

Jeffrey Zheng *Editor*

Variant Construction from Theoretical Foundation to Applications

OPEN

 Springer

Variant Construction from Theoretical Foundation to Applications

Jeffrey Zheng
Editor

Variant Construction from Theoretical Foundation to Applications

OPEN

 Springer

Editor
Jeffrey Zheng
School of Software
Yunnan University
Kunming, Yunnan, China



ISBN 978-981-13-2281-5 ISBN 978-981-13-2282-2 (eBook)
<https://doi.org/10.1007/978-981-13-2282-2>

Library of Congress Control Number: 2018958351

© The Editor(s) (if applicable) and The Author(s) 2019. This book is an open access publication.

Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Dedicated to
I-Ching—First Variant Construction
Lan Z. Yin & Su M. Zheng—Mother & Father
Qing S. Gao—Mentor on Parallel Sorting
Algorithm & Computer Architecture
Tasiyasu L. Kunii—Master on Meta
Knowledge
Bob Beaumont—Adviser on Optimization
Ping Zhang—Wife
Graduate School of USTC & UCAS—40-th
Anniversary (1978–2018)

Foreword

Dr. Jeffrey Zheng was one of the first postgraduate students supervised by Prof. Qingshi Gao (Member, Chinese Academy of Sciences) at the Institute of Computing Technology, Chinese Academy of Sciences. I have known Dr. Zheng for 40 years since then. Building upon his postgraduate work (Parallel Sorting Algorithm and 0-1 Transformation), Dr. Zheng has made significant contribution to the field of Variant Construction, ranging from theoretical foundations to various applications. His research has been published at many academic journals and conferences. For the convenience of readers, Dr. Zheng compiled his representative works of 40 years into two monographs with complementary contents. I believe that professionals in related fields will find this book both an excellent reference and a source of inspiration. Other readers will enjoy this book as an introduction to topics of Variant Construction. I am very happy to recommend this book in the form of a foreword.

Beijing, China
April 2018

Yunmei Dong
Professor, The Institute of Software
Chinese Academy of Sciences
Member, Chinese Academy of Sciences

As head of the R&D team for Lenovo Chinese Systems, I am very pleased to see the research work of former colleague Dr. Jeffrey Zheng, which began 30 years ago with the “Smoothly Enlarging Chinese Font Algorithm of 0-1 logic operations” at the Institute of Computing Technology of the Chinese Science Academy. His most recent work “Variant Construction” is summarized as a professional monograph. I expect this new measurement system to be used efficiently for advanced cryptographic tests in modern cyberspace security. I am pleased to give this foreword.

Beijing, China
April 2018

Guangnan Ni
Professor, The Institute of Computing Technology
Chinese Academy of Sciences
Member, Chinese Academy of Engineering

Dr. Jeffrey Zheng and I were in the first group of postgraduates major in Computer Architecture at the Graduate School of the Chinese Academy of Sciences 40 years ago. Professor Qingshi Gao (Member, Chinese Academy of Sciences) supervised him in particular in the areas of parallel algorithm and computer architecture.

Dr. Jeffrey Zheng is one of the few classmates who continue to works in basic research and advanced applications. It is great for Dr. Jeffrey Zheng to collect his research work in a monograph. Variant Measurement Technology could be used in the next generation of Quantum Cryptographic Communication Services.

On the occasion of the 40th anniversary of the Graduate School of Chinese Academy of Sciences, I would like to express my good wishes as a classmate for this monograph in the foreword.

Beijing, China
April 2018

Guojie Li
Professor, The Institute of Computing Technology
Chinese Academy of Sciences
Member, Chinese Academy of Engineering

Preface

Associated with the fast development of science and technology in the twenty-first century, the modern computer and communication system in optical fiber communication supporting the global Internet shows profound influence on society and economy. As a result, globalization has become an extremely important issue in social and economic systems. The Internet and optical fiber communication systems have revolutionized the geographic and communication patterns of the world, by creating an open era of integrated global Internet connectivity. Quantum key communication technology and quantum entanglement experiments on a quantum satellite represent typical examples of China's world-leading science and technology from the perspective of frontier application research. The latest achievements of artificial intelligence, which is the lead of Alpha-Go, show the potential intelligence prospect of advanced technology based on deep learning, artificial neural networks, and knowledge-based support vector machine systems. Related achievements are very attractive, such as poetry robots, service robots, industrial robots, face recognition, gesture recognition, unmanned aerial vehicles, self-driving cars, and unmanned underwater vehicles. A list of military and civilian high-tech achievements supports daily life with rich and colorful intelligent products.

From the viewpoint of mathematics and logics, the foundation framework to design and simulate both modern computer systems and optical fiber communication networks is dependent on the 0-1 logical system and representations of multiple bit states. For integrated circuits, the theoretical basis can be traced back to the 1930s. Shannon developed the Boolean algebra to design circuits establishing switch circuit theory, Turing proposed the Turing machine, and von Neumann established a modern computer architecture. After more than 50 years of development follows Moore's Law: the observation that the number of transistors in a dense integrated circuit doubles approximately every 2 years. Optimization of very large-scale integrated circuit technology appears everywhere with evolution of magical functions.

Looking ahead, the development of advanced science and technology is subject to the limitations of basic theory and applications on foundational supports. From the perspective of basic research, how we can extend this classical level is a very interesting issue and an extremely difficult research topic.

Purpose of This Book

After four decades of deep exploration on 0-1 logical systems, the authors expended vector 0-1 logical systems to establish a variant logic framework in 2010. After further research and development for one decade, three theoretical components were established: variant logic, variant measurement, and variant map. At the same time, various sample applications were investigated and developed. However, because most published papers are scattered in professional journals, conference proceedings, and academic books, it is difficult for other people to obtain comprehensive information on the topic.

In addition, each article may be focused on a specific issue, and it is difficult for readers to understand the whole structure from a few papers. We are going to organize relevant papers in this book, which will be the first book on variant construction with intrinsic logical connections on the selected papers. Selected papers are composed of different parts. Based on this architecture, different readers can easily access suitable content from specific chapters.

The Need for a New Logic System

In modern computer and communication systems, the theory of switch circuits uses multiple bits, states, and logic operations for state automata and combinatorial logic units to design and implement complex computing and communication systems. For solving linear equations with n variables as algebraic equation, Boolean equation or differential equation, it is useful to apply a matrix associated with a set of eigenvectors. Matrices and eigenvalues are valid to provide solutions on periodic problems of special basis in periodic functions or periodic boundary conditions. However, it is difficult for periodic models to resolve exhaustive cases on the conditions of quasi-periodic, nonperiodic random, and chaotic forms. For example, modern cryptographic generation/analysis systems such as block ciphers are dependent on a Substitution–Permutation Net (SPN). This type of network connection on n bit vectors of input/output transformation includes permutation operations, where the total number of configuration functions is proportional to $2^n!$. From a measuring viewpoint, cryptographic sequences need to have relevant measurements, analysis models, and methods with huge complexity far beyond based on state automata and combinatorial logic circuits.

Modern digital computing and communication technologies are based on classical logic systems, the global Internet network with huge amounts of data models, deep learning, artificial neural networks, and knowledge-based vector support machines cannot meet internal states of exponentially increased models. Although Fourier transform and wavelet transform are the most important tools for modern spectrum analysis, there are significant limitations for this type of periodic schemes to process arbitrary random state and aperiodic types of complex functions in big data environments. It is difficult for random applications to obtain the convergence results. Quantum mechanics and modern photonic-electronic applications are confirmed the effectiveness of this frontier science.

Nobel Prize Winner G. t'Hooft proposed a cellular automaton interpretation of quantum mechanics. The research results show that there is a commonplace overlapped between classical logic and quantum mechanics, at the Planck scale in 10^{-43} range. It is necessary to use 0-1 vectors in permutation condition to represent quantum states. From a counting viewpoint, the complexity of such structures is related to 2^n !

In classical statistics, the Ising model provides an analysis mechanism on 0-1 states. Based on the assumption of exhaustive states, an exact solution can be compared with the average field on one- and two-dimensional lattices. In general, whether there is an exact solution under the condition of random permutation distribution is an interesting topic worth further exploration. Modern experiments made good progress in advanced nanotechnology, fiber optics, laser photonics, and ultrafast laser pulse in quantum optics technology. Advanced experiments in nanotechnologies can be used to distinguish a series of the quantum block/surface/line and dot macro- to nanostructures, and relevant emission and absorption spectrum can be observed. Both wider continuous spectrum of thermal noises and narrower discrete spectrum of coherent laser beams are observed. In current research problems, the measurement models and methods discussed are far different from the quantum scale, and all results can be described in modern probability statistics. However, the complex operation associated with the shift operations on the phase space of permutations, modern statistical probability methods, and tools have difficulties to handle symmetric groups directly with arbitrary random permutation requirements.

The advanced Quantum Key Distribution (QKD), from a stochastic analysis viewpoint, needs to have effective measurement model and quantitative method to identify the source of a random sequence. Is it generated from a quantum random resource as a truly random sequence or a stream cipher as a pseudo-random sequence? It is impossible to make a classification use the NIST random testing package. This type of targets is also impossible to apply spectrum analysis and linear equation tools. More advanced models and methods are required.

For a 0-1 vector with multiple bits, analysis tools use classical probabilistic statistical models and methods. Since the specific problem of randomness testing is far beyond the combinatorial analysis and state automata, it is difficult to handle the demand of actual measurement and quantitative analysis due to ultra-complexity of the substitution and permutation on complicated modes. Similar to modern

physics applying classical statistics, it is necessary to establish a solid logic foundation to support permutation and substitution operations in logic mechanism to make extension of analytical frontier to support both theoretical foundation and practical applications.

From mathematical logic, automatic control, quantum mechanics, artificial intelligence, etc., using probability and statistics, the demand for random sequence analysis and measurement uses the n variable 0-1 vectors and their linear combination cannot meet measurement requirements on various applications. Modern measuring methodology and technology need to use permutation and substitution operations on different levels of logic foundation to satisfy the frontier measurements on quantum physics, cryptographies, and artificial intelligence. From a measuring viewpoint, the emergence of a new measuring system is urgently required to deal with advanced applications.

Overview of Modern Group Theory

From a discrete representative viewpoint, every abstract group is isomorphic to a subgroup of the symmetric group of some set (Cayley's theorem) and permutations are the core basis in modern group theory.

The beginning of modern group theory can be traced back to Galois' contribution in the 1830s; Klein studied transformation group in the 1870s to propose Erlangen program to show the group theory as an invariant structure for symmetrical patterns and transformations. Inspired by Klein, Lie used infinitesimal symmetry transformations to establish a Lie algebra system.

Using the multiple tuples of variable structures, Hamilton proposed complex and quaternion expressions. Influenced by Gordon on invariant formula, Hilbert using finite basis constructed a complete system of an algebraic structure on n variables. In 1906, an infinite-dimensional Hilbert space of complex variables was developed. Based on the series of automorphic functions, Poincaré was the first person to discover a chaotic deterministic system which laid the foundations of modern complex dynamic system, fractal and chaos theory.

Through Noether's investigations on Einstein general relativity to determine the conserved quantities for every physical laws that possess some continuous symmetry as Noether theorem. A series of studies on invariants and symmetries were promoted the development of abstract algebra in the 1930s by refining algebraic structures as groups, rings, algebras, fields, and lattices.

In the 1930s, Weyl established the group theory of quantum mechanics; the theoretical basis of quantum mechanics was established based on the symmetry operator. Since the 1940s, Hua developed a complex matrix representation under symplectic group using the unit circle as the core. In the 1950s, Yang proposed the gauge invariance that plays a foundation role in modern field theory. Chern established the fiber bundle structure for the differential geometry of the complex function.

From 1980s, the gauge field theory became the basic mathematical tool of modern physics. The eightfold/tenfold way of quark model plays a key role in the standard model of particle physics and the exploration of grand unified theory; the corresponding group structures are $SU(3)/SU(5)$.

Brief History on 0-1 Logic Systems

From the perspective development of mathematical logic, the origin of the modern 0-1 logic system can be traced back to Leibniz's invention on binary counting and combinatorial analysis in the 1670s. In the 1850s, Boole proposed Boolean algebra; in the 1900s, Logic school made logic as the foundation of modern mathematics.

In the 1930s, Gödel proposed incompleteness theorem to be unprovable in a given formal system for Hilbert's decision problem. In 1936, Turing used infinite length of 0-1 sequence with read/write operation to be the Turing machine. Under Church's Lambda calculus, the Church-Turing thesis lays the theoretical foundation of computable and recursive theory.

Using 0-1 variables and logic operators, Shannon in 1937 proposed switch theory to provide module design, simulation, and implementation bases for modern computers and communication systems of technical supports. After more than half a century revolutionary development of semiconductor chips, electronic circuits from discrete separated components to integrated circuits, and then very large-scale integrated circuits, switch theory provides solid foundation on the basic theory, application analysis, and design tools.

Although the modern logic system was original developed from Leibniz, use of permutation modes in state transformations can be traced back ancient time for several thousand years ago in oriental history. In the *I-Ching* system developed from the early days, Yin and Yang's representations are identified as the roots. Five thousand years ago, Fu-hsi proposed eight trigrams as an initial set that can be represented as eight states of three 0-1 variables. Using modern mathematics, one can see that the representations of the three layers of trigrams of Yin/Yang are equivalent to the eight diagrams and eight states of three 0-1 variables. Three thousand years ago, King Wen of Zhou dynasty proposed another order of eight trigrams to be different from Fu-hsi, that is, a permutation of the Fu-hsi group. In the 1050s, Shao Yung proposed a balanced binary tree as a natural order of a binary system same as the Leibniz binary counting.

Ancient Oriental philosophers have developed the logical foundation of Chinese traditional culture using this Yin/Yang symbol system. However, it must be pointed out that subsets of states are contained in this system with various logic paradoxes at different levels. This dialectical logic system based on the *I-Ching* is difficult to meet a list of important characteristics in formal logic: consistency, completeness, noncontradiction, soundness, etc.

Modern 0-1 Vector Algebra

For using 0-1 vectors and logic operators in vector operation mode, it is a natural way to extend parallel bit operations from a single bit to multiple bits. In addition, in order that bit operations can be effectively performed on multiple bits, it is necessary to implement permutation operations among bits. It is convenient to define a pair of bits with a fixed distance and cyclic shift operations on a given vector.

In the 1970s, Lee described cyclic shift operations in *Modern Switch Circuit Theory and Digital Design*. From the formula of vector switching functions, the canonical forms of vector switching functions are extremely complex and very powerful transformations.

Associated with the advanced development on block ciphers in cryptography, a new vector extension has been developed as Advanced Vector Extensions (AVS). Specific development of the new instruction for AES cipher algorithm is AES-NI package, which shows the latest achievements for block ciphers.

Under this type of vector permutation–substitution components, complex cryptographic algorithms can efficiently perform encryption and decryption requirements under permutation and substitution commands.

Introduction to Variant Construction

In the 1980s, the author studied the sorting problem on a vector of N integer elements using the symmetric group under 0-1 vector control, and constructed high-performance parallel sorting algorithms. Then, smoothly enlarging algorithms for Chinese fonts were proposed using logic operations on 2D bitmaps. In the 1990s, multiple levels of invariants were used to organize a state set as a phase space, and the conjugate classification and transformation of binary images was established.

In 2010, a new vector logic system was proposed using two composite operations: permutation and complement, to form a new vector logic system: Variant Logic. After 8 years of in-depth exploration, the variant construction is composed of three core components: variant logic, variant measurement, and variant map.

Using four meta states, multiple probability and statistical measurements can be constructed. By associating these measurements with quantitative expressions and combinatorial projections, more than 60 research papers and book chapters were published. Relevant contents are covered from theoretical foundation to sample applications. Since all these papers are published in various places all over the world, it is difficult for readers to systematically collect them for further reading. This book is the first one to collect the most relevant papers from theoretical foundation to sample applications to organize the variant construction as variant

logic, variant measurement, variant map, meta model, and sample application systematically.

The Organization of This Book

This book is composed of nine subparts in two main parts: theoretical foundation and sample application. The theoretical foundation is composed of four subparts: Variant Logic, Variant Measurement, Variant Map, and Meta Model.

Variant Logic describes n variable 0-1 vectors with 2^n states which form a variant configuration space with $2^n!2^n$ members.

Variant Measurement defines on n tuple 0-1 vectors, four meta measures, and ten expansion operators established.

Variant Map illustrates 2^n states and 2^{2^n} transforming states, and multiple statistical probability distributions are investigated using four meta measures and their combinations in higher dimensional distributions.

Meta Model describes a concept cell model of knowledge representation and a multiple probability model on voting.

The part of ample application is composed of five subparts: Global Visualization, Quantum Interaction, Random Sequence, DNA Sequence, and Multi-valued Pulse Sequence. In Global Visualization, a list of function maps is used on medical image analysis, cellular automata rule space on exhaustive arrangement. In Quantum Interaction, conditional and relative probability distributions simulate two paths of quantum interactive effects. Random Sequence provides variant random number generators, a unified measurement model to handle both pseudo and truly random sequences in modern cryptographic applications on variant maps. In DNA Sequence, whole gene sequences are mapped on variant maps. In Multiple-valued Pulse Sequence, bat echo/ECG sequences are mapped on variant maps.

Suitable Readers of This Book

This book includes a wide range of topics from theoretical foundation to sample applications. Different parts may be suitable for specific groups. Variant Logic, Meta Model, and Variant Measurement are useful for basic researchers on logic, probability, statistics, analysis, and measures on mathematical foundation, combinatorial mathematics, metamathematics, quantum logic, and combinatorial group theory on levels of researchers and graduate students; Variant Measurement and Variant Map are suitable for application researchers and engineers in big data, complicated system analysis, feature extraction, artificial intelligence, applied mathematics, software engineers, senior college students, and postgraduate

students; Variant Map and sample applications are suitable for requirements of complex system analysis/design, data engineer, big data engineer, artificial intelligence engineer, application development engineer, postgraduate, and senior undergraduate students.

Kunming, Yunnan, China
April 2018

Jeffrey Zheng

Acknowledgements

The author would like to thank colleagues: Chris Zheng, Jianzhong Liu, Tao Chen, Yuzhong Luo, Tong Li, Yixian Yang, Lizhen Li, Zhengfu Han, Dawu Gu, Weizhong Yang, Jing Luo, Wei Zhou, Shaowen Yao, Lian Lu, Yinfu Xie, Chu Zhang, Xiazhou Yang, Xiaoyun Pu, Weilian Wang, Lu Shan, Ying Lin, Yunchun Zhang, Dennis Heim, Olga Heim, and Colin Campbell for their criticism, encouragement, suggestions, discussions, corrections, and help of various kind on this book.

I am particularly grateful to my students for the past 10 years: Bingjing Cai, Wenjia Zhao, Qin Kang, Qiping Li, Zhiqiang Yu, Yao Zhou, Jie Wan, Huan Wang, Jie-ao Zhu, Qinxian Bu, Weiqiong Zhang, Zu Wan, An Wang, Yuqian Liu, Lei Du, Ruoyu Shen, Heyuan Chen, Yan Ji, Guoxiu Zhai, Pingan Zeng, Wenjia Liu, Ruoxue Wu, Lixin Wu, Zhonghao Yang, Lihua Leng, Zhihui Hou, Yuyuan Mao, Yamin Luo, Zhefei Li, Yifeng Zheng, and many other students in a series of research courses and projects to explore extensive topics from data streams of binary/DNA/multiple-valued sequences to wider applications under variant construction.

I specially thank Tosiyasu Kunii and Bob Beaumont for lifetime friendship in encouragement and information guided us to explore meta models, various applications on Binary/DNA/ECG sequences, and other complicated signals in variant construction.

I sincerely thank four main funding resources to support us to complete this book.

- The Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002).
- NSF of China (61362014).
- Yunnan Advanced Overseas Scholar Project.
- Australian Commercialising Emerging Technologies (COMET) program.
- Finally, I thank the following publishers for permission to include seven papers in previous OA publications:

- Scienpress Ltd for one paper, Chapter “[Synchronous Property—Key Fact on Quantum Interferences](#)”.
- Research Online of Edith Cowan University for three papers, Chapters “[Novel Pseudorandom Number Generation Using Variant Logic Framework](#)”, “[2D Spatial Distributions for Measures of Random Sequences Using Conjugate Maps](#)”, and “[3D Visual Method of Variant Logic Construction for Random Sequence](#)”.
- Scientific Research for two paper, Chapters “[Permutation and Complementary Algorithm to Generate Random Sequences for Binary Logic](#)” and “[Variant Map System to Simulate Complex Properties of DNA Interactions Using Binary Sequences](#)”.
- OMICS International for one papers, Chapter “[Successful Creation of Regular Patterns in Variant Maps from Bat Echolocation Calls](#)”.

Contents

Part I Theoretical Foundation—Variant Logic	
Variant Logic Construction Under Permutation and Complementary Operations on Binary Logic	3
Jeffrey Zheng	
Hierarchical Organization of Variant Logic	23
Jeffrey Zheng	
Part II Theoretical Foundation—Variant Measurement	
Elementary Equations of Variant Measurement	39
Jeffrey Zheng	
Triangular Numbers and Their Inherent Properties	51
Chris Zheng and Jeffrey Zheng	
Symmetric Clusters in Hierarchy with Cryptographic Properties	67
Jeffrey Zheng	
Part III Theoretical Foundation—Variant Map	
Variant Maps of Elementary Equations	97
Jeffrey Zheng	
Variant Map System of Random Sequences	105
Jeffrey Zheng	
Stationary Randomness of Three Types of Six Random Sequences on Variant Maps	133
Jeffrey Zheng, Yamin Luo, Zhefei Li and Chris Zheng	

Part IV Theoretical Foundation—Meta Model

Meta Model on Concept Cell 159
Jeffrey Zheng and Chris Zheng

Voting Theory for Two Parties Under Approval Rule 169
Jeffrey Zheng

Part V Applications—Global Variant Functions

Biometrics and Knowledge Management Information Systems 193
Jeffrey Zheng and Chris Zheng

Recursive Measures of Edge Accuracy on Digital Images 203
Jeffrey Zheng and Chris Zheng

**2D Spatial Distributions for Measures of Random Sequences
Using Conjugate Maps** 217
Qingping Li and Jeffrey Zheng

**Permutation and Complementary Algorithm to Generate
Random Sequences for Binary Logic** 237
Jie Wan and Jeffrey Zheng

**3D Visual Method of Variant Logic Construction
for Random Sequence** 247
Huan Wang and Jeffrey Zheng

Part VI Applications—Quantum Simulations

Synchronous Property—Key Fact on Quantum Interferences 265
Jeffrey Zheng

The n th Root of NOT Operators of Quantum Computers 279
Jeffrey Zheng

Part VII Applications—Binary Sequences

**Novel Pseudorandom Number Generation Using Variant
Logic Framework** 289
Jeffrey Zheng

RC4 Cryptographic Sequence on Variant Maps 297
Zhonghao Yang and Jeffrey Zheng

**Refined Stationary Randomness of Quantum Random
Sequences on Variant Maps** 307
Jeffrey Zheng, Yamin Luo and Zhefei Li

Using Information Entropy to Measure Stationary Randomness of Quantum Random Sequences 321

Weizhong Yang, Yamin Luo, Zhefei Li and Jeffrey Zheng

Visual Maps of Variant Combinations on Random Sequences 333

Jeffrey Zheng and Jie Wan

Part VIII Applications—DNA Sequences

Variant Map System to Simulate Complex Properties of DNA Interactions Using Binary Sequences 353

Jeffrey Zheng, Weiqiong Zhang, Jin Luo, Wei Zhou and Ruoyu Shen

Whole DNA Sequences of *Cebus capucinus* on Variant Maps 379

Yuyuan Mao, Jeffrey Zheng and Wenjia Liu

Part IX Applications—Multiple Valued Sequences

Successful Creation of Regular Patterns in Variant Maps from Bat Echolocation Calls 391

D. M. Heim, O. Heim, P. A. Zeng and Jeffrey Zheng

Visual Analysis of ECG Sequences on Variant Maps 401

Zhihui Hou and Jeffery Zheng

Contributors

D. M. Heim Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China

O. Heim Leibniz Institute for Zoo and Wildlife Research, Berlin, Germany; Animal Ecology, Institute of Biochemistry and Biology, University of Potsdam, Potsdam, Germany

Zhihui Hou Yunnan University, Kunming, China

Qingping Li School of Software, Yunnan University, Kunming, China

Zhefei Li Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China

Wenjia Liu Yunnan University, Kunming, China

Jin Luo School of Life Sciences, Yunnan University, Kunming, China

Yamin Luo Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China

Yuyuan Mao School of Software, Yunnan University, Kunming, China

Ruoyu Shen School of Software, Yunnan University, Kunming, China

Jie Wan Yunnan University, Kunming, China; The People's Bank of China, Kunming, China

Huan Wang Yunnan University, Kunming, China

Weizhong Yang Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China; Key Laboratory of Quantum Information of Yunnan, School of Software, Yunnan University, Kunming, China

Zhonghao Yang Yunnan University, Kunming, China

P. A. Zeng Yunnan University, Kunming, China

Weiqiong Zhang School of Software and Microelectronics, Peking University, Beijing, China

Chris Zheng Tahto, Sydney, Australia; Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China

Jeffrey Zheng Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China; Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China; Key Laboratory of Yunnan Software Engineering, Yunnan University, Kunming, Yunnan, China

Wei Zhou School of Software, Yunnan University, Kunming, China

Part I

Theoretical Foundation—Variant Logic

I-Ching has three key properties: 1. Simple, 2. Variant, 3. Invariant.
—Zheng Xuan

The Monad, of which we shall here speak, is nothing but a simple substance, which enters into compounds. By simple is meant without parts.

—Gottfried W. Leibniz

Quaternions came from Hamilton after his really good work had been done, and though beautifully ingenious, have been an unmixed evil to those who have touched them in any way.

—Lord Kelvin

From a historical viewpoint, the first paper of variant logic foundation (A framework to express variant and invariant functional spaces for binary logic) was published in *Frontiers of Electrical and Electronic Engineering in China*, Higher Education Press and Springer 5(2):163–167 (2010). An extensive book chapter (Chapter “A framework of variant-logic construction for cellular automata”) was published in the OA book of *Cellular Automata—Innovative Modelling for Science and Engineering*:325–352 (2011) by InTech Press to describe a variant logic framework systematically.

The Part I is composed of two chapters (1–2).

Chapter “[Variant Logic Construction Under Permutation and Complementary Operations on Binary Logic](#)” is shown the core construction of variant logic under two vector operations (Permutation, Complement) on 0-1 logic.

Chapter “[Hierarchical Organization of Variant Logic](#)” describes complex hierarchical organization under variant logic construction to compare with other logic systems.

Variant Logic Construction Under Permutation and Complementary Operations on Binary Logic



Jeffrey Zheng

Abstract This chapter presents a binary logic framework whose function elements are invariant under permutation and complementary operations. The entire framework is described using 4 levels of hierarchy: n variables, 2^n states, 2^{2^n} functions, and $2^n!2^{2^n}$ logic functionals. Under the proposed framework, it is possible to determine higher level function complexity by analysing lower levels of organisation characteristics. These characteristics can be determined quite accurately because the symmetry conditions of variable and state organisations have invariant logic functions and a corresponding logic functional organisation. More symmetrical arrangement at state level creates more symmetrical permutations within the function space. Lower level properties are highly influential on the higher level properties of function components within a logic functional space. The proposed framework provides a logic foundation to describe complex binary systems using lower level properties, making analysis of systems more efficient and less calculation intensive. Different global coding schemes are discussed and typical two-variable cases of logic functionals are illustrated.

Keywords Vector permutation · Complement · Variant logic · Functional space
Binary logic framework

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_1

1 Introduction

Mathematical invariance [1, 2] is key in the understanding and development of new scientific theories and technologies [3]. Most scientific theories rely on invariant properties of group behaviour and transformations [4] to describe the rules of the world we live in. Theories such as relativity and quantum mechanics all rely on invariance properties for their constructs [5]. In the field of mathematical logic, construction of theoretical frameworks [6, 7] focus upon three hierarchical levels: variables, states and function spaces. Boolean algebra and switching theory [8, 9] exploit combinatorial invariant properties, and use these foundational properties for implementing new theories and applications.

For reasons of consistency and symmetry of structure, logical operations are restricted to two types of canonical forms namely, the product-of-sums and the sum-of-products approach. Any complex logic function can be rewritten as these two canonical forms. The use of a truth table enables analysis and the transformation into the canonical representations [6].

Following the introduction of Conway's Game of Life [10], Stephan Wolfram from the 1980s [11, 12] started to apply Boolean algebra to describe the behaviour of Cellular Automata. His approach used a binary counting sequence to naming different rules of behaviour based upon the functions generating the next iteration in the game. Wolfram identified four classes of transformations within the rules of Cellular Automata (CA). Results of findings are published in his book [13]—"A New Kind of Science". The main method of analysis in this area of research chooses a CA operation, recursively applying the operation to different initial conditions to find emergent patterns from the process. This approach creates many interesting results that can be visually identified [14, 15].

In the analysis of dynamic systems, it is essential to identify transformation spaces with functional invariance [16, 17]. An example in physics is phase space [2]. The phase space plays an essential role to describe key properties of a given dynamic system. Phase characteristics are more difficult to construct under a logic framework. A mechanism for linking lower level characteristics with higher levels properties such as symmetry currently does not exist. Under combinatorial logic, different permutations add no additional information to access information in phase space [14].

1.1 *Western and Eastern Logic Traditions*

Beginning with Aristotle (384–322 B.C.), the foundations of Western logic have played a key role in the development of today's global society [18]. The modern theory of logic systems comprise of a series of outstanding individuals and their contributions to the theory of logic: G. Leibniz and the introduction of the Binary Number System (1646–1716) [19, 20]; G. Boole and the development of Boolean

Logic (1854) [21]; G. Cantor and Set Theory (1879); G. Frege and Conceptual Logic (1879) [22, 23]; B. Russell and Russell's Paradox (1910) [24]; J. Lukasiewicz and Multiple-Valued Logic (1920); D. Hilbert and Foundations of Geometric Logic (1923) [25], K. Gödel and his Incomplete Theorem (1931) [22], A. Turing and the Turing Machine (1936) [26]; C. Shannon and Switching Theory (1937) [27]; H. Reichenbach and Probability Logic (1949) [28]; as well as L. Zadeh and Fuzzy Logic (1965) [29]. Development of such theorems and mathematical frameworks have enabled Western culture to understand the operation of our world as a set of implementable rules. Logic and the development of rules for the expression of logic have provided a language that enabled the construction of today's scientific societies.

In contrast to the binary on-off nature of Western logic, Oriental culture have been influenced by spiritual traditions of balance and harmony. The theme of balance can be summarised in the I-Ching or 'The Book of Changes', one of the most influential books of classic Oriental literature [30–37]. The concept of Yin and Yang forces and the subtle interplay of the two opposing forces yield combinations and permutations of change. Orient philosophy believed that 'the only constant phenomena is change' and such a worldview emphasised the dynamic nature of a system; rather than focusing on the individual states of a system (on, off), prominence was instead placed on operations that yield change (on to off, off to on). The structure of thought introduced by the I-Ching allowed change to be systematically documented and analysed. Complex interactions, cyclic behaviour and the interplay of nature at all levels of oriental culture—sociology, literature, medicine, astrology and religion—were able to be described using the tools of dynamic logic provided by the I-Ching; the framework remains a complete philosophy as well as a universal language and has remained unchanged over the past two thousand years [38].

Leibniz in as early as 1690 realised that the balanced yin-yang structure proposed by Shao Yong (1050) was equivalent to the binary number system [33, 38]. However the Western scientific community have mostly disregarded the I-Ching; due mainly to cultural and language barriers as well as local superstitions that cloud the essence of the framework. In its ancient form of allegories and metaphors, the I-Ching is unable to satisfy the logician's requirement for completeness, consistence and other such properties. The challenge then is to be able present this philosophy for modern times, in the language of mathematics. Stripped of its colourful language, what insights does this ancient system contain? What are the essential differences between modern binary logic and the I-Ching's dynamic binary structures? The unification of these two schools of thought would bring greater understanding of the world we live in [35]. As the modern formulation of Cellular Automata generates complexity through binary logic whilst the I-Ching analyses complexity through binary logic, the modern language of the I-Ching can be found in the creation of a structural definition of CA.

1.2 Logic and Dynamic Systems

In the field of mathematical logic, construction of theoretical frameworks focus upon three spatial hierarchies: variables, states and function spaces [6, 7]. Boolean algebra and switching theory exploit such properties, using the combinatorial invariance of the framework for implementing new theories and applications [8, 9]. Logical operations are restricted to two types of canonical forms, namely the product-of-sums and the sum-of-products approaches. Any complex logic function can be rewritten as these two canonical forms. This is done for reasons of consistency, simplicity and symmetry of structure; as such the use of a truth table enables analysis and the transformation into the canonical representations [6].

In the analysis of dynamic systems, it is essential to identify transformation spaces with functional invariance [16, 17]. The Ising model is arguably the simplest binary system that undergoes a nontrivial phase transition [14]. In modern physics, this type of model uses a structure linked to phase space representation of a dynamic systems [2]. The phase space plays an essential role to describe key properties of any dynamic system, however under classical logic, phase characteristics are difficult to construct. A mechanism for linking low-level representations such as variables and states with higher level group properties such as symmetric conditions currently does not exist. This is more a limitation of the language and the operations allowed by the language. Classical logic is based on static combinatorial structures. Permutations, which are intrinsic to phase space, cannot be expressed under such a framework of classical combinatorial logic [14]. Cellular Automata frameworks [39], however, are fully dynamic and have been used to describe phase space [2]. Inspired by the traditional I-Ching hierarchical structures, new conditions, operations and relationships have been proposed on top of the Classical Logic framework to incorporate the dynamic nature of CA. The additional constructs provide support for CA using framework that is logically consistent and complete [40].

The [40] proposal builds upon earlier studies of logic systems from a structural viewpoint. Kunii and Takai [41] applied a n -cell structure for analysis, classification and generation of visual objects using topology and homotopy tools in computer graphics [42–46]. Zheng and Maeder [47] proposed a balanced classification on binary images for conjugate classification and transformation of binary images on regular plan lattices in 1990s to visualise different configurations [15, 48–50]. All such work used partial constructs of the [40] framework. The proposed framework supports classical logic, vector permutation and complementary operations. The new construction requires five spatial hierarchies containing $2^{2^n} \times 2^n!$ functional configurations for any n variables. This structure is much larger than classical logic having three spatial hierarchies supporting 2^{2^n} functions for n variables. Newly defined symmetric properties play an important role in predictions and classifications of possible recursive results. Using such properties, global behaviour can be identified and classified. A disadvantages of the new framework lies in its extreme complexity. It is possible to use parallel computers to do analysis of the configurations contained by $n = 3$ (the space already includes more than 10^7 configurations). It is impossible

using today's technology to process the $n = 5$ space due to the extreme growth of structural complexity ($2^{32} \times 32!$ configurations).

This chapter describes a logic framework, using invariant characteristics of permutations and complementary operations to identify an invariant structure under such mixed operations. This allows the definition of a phase space to be introduced into logic. The transformation does not change the relevant function space. A proposed 2D representation provides additional properties to predict different behaviours from permutations that influence higher level structures in a logic functional space.

2 Truth Table Representation for a Logic Function Space

The proposed framework describes three levels of a logic function space and the truth table representation of the space.

2.1 Basic Definitions

$$\begin{aligned} f : X \rightarrow Y; \quad Y = f(X); \quad X, Y \in B_2^N \\ X = X_{N-1}X_{N-2} \dots X_j \dots X_1X_0, \quad Y = Y_{N-1}Y_{N-2} \dots Y_j \dots Y_1Y_0 \quad (1) \\ X_j, Y_j \in B_2, 0 \leq j < N \end{aligned}$$

An example of a transform: the sequence $X = 0001110100$, $N = 10$ is an input for a function operation f , the output is a sequence of the same length $Y = 1101011001$; $X, Y \in B_2^{10}$.

Definition 1 Let $\dots X_j \dots$ be a n bit structure:

$$\begin{aligned} \dots X_j \dots = x_{n-1}x_{n-2} \dots x_i \dots x_1x_0 = \mathbf{x} \\ 0 \leq i < n, 0 \leq j < N, \mathbf{x} \in B_2^n \end{aligned} \quad (2)$$

where $X_j = x_i$ is a corresponding position.

$$Y_j = f(\dots X_j \dots) = f(x_{n-1}x_{n-2} \dots x_i \dots x_1x_0) = f(\mathbf{x}) \quad (3)$$

In Boolean logic, n variables correspond to a full truth table with $2^n \times 2^{2^n}$ entries. The I th meta-state $0 \leq I < 2^n$ has n -bit number to occupy the I th column position, the J th function $T(J)$ has the J th row with 2^n bits $0 \leq J < 2^{2^n}$, the function value of the I th entry is determined by $T(J)_I$. The full table can be represented as follows (Table 1):

Table 1 Truth Tables of n -variables

$0 \leq I < 2^n$	S_{2^n-1}	...	S_I	...	S_1	S_0
$I_{n-1} \dots I_i \dots I_0$	1...1...1	...	$I_{n-1} \dots I_i \dots I_0$...	0...0...1	0...0...0
$0 \leq J < 2^{2^n}$	J_{2^n-1}	...	J_I	...	J_1	J_0
$T(0)$	0	...	0	...	0	0
$T(1)$	0	...	0	...	0	1
$T(2)$	0	...	0	...	1	0
...			...			
$T(J)$	J_{2^n-1}	...	J_I	...	J_1	J_0
...			...			
$T(2^{2^n} - 2)$	1	...	1	...	1	0
$T(2^{2^n} - 1)$	1	...	1	...	1	1

Method 1: Process Method of Truth Table

Input: \mathbf{x} : n variables in a $\{0, 1\}$ sequence, J : selected function number

Process: Using the input sequence \mathbf{x} , the meta-state number I is to select the I -th column of function $T(J)$

Output: Return $T(J)_I$'s value (1 for true and 0 for false) as output.

2.2 Permutation Invariants

Proposition 1 *Sequential Mapping Under sequential order, $T(J) = J$.*

Proof The relevant output entries of $T(J)$ are mapped to the binary number J having 2^n bits:

$$\begin{aligned}
 T(J) &= T(S_{2^n-1}(J_{2^n-1})) \dots T(S_I(J_I)) \dots T(S_0(J_0)) \\
 &= T(J)_{2^n-1} \dots T(J)_I \dots T(J)_0 = J \in B_2^{2^n} \\
 T(J)_I &= T(S_I(J_I)) = J_I \in B_2; 0 \leq I < 2^n, 0 \leq J < 2^{2^n}
 \end{aligned} \tag{4}$$

■

Definition 2 For any n binary logic variables, let $\Omega(N)$ be a symmetric group with N elements and P be a permutation operator, $P \in \Omega(2^n)$, then for any $J, \exists K, J, K \in B_2^{2^n}$, $P(T(J)) = K, 0 \leq J, K < 2^{2^n}$, the following permutation can be represented in Truth Table form:

$$\begin{aligned}
 P : J &\rightarrow K \\
 P(T(J)) &= P(T(S_{2^n-1}(J_{2^n-1}))) \dots P(T(S_I(J_I))) \dots P(T(S_0(J_0))) \\
 &= P(T(J)_{2^n-1}) \dots P(T(J)_I) \dots P(T(J)_0) \\
 &= K_{2^n-1} \dots K_I \dots K_0 = K \in B_2^{2^n} \\
 P(T(J)_I) &= P(T(S_I(J_I))) = T(S_{P(I)}(J_{P(I)})) \\
 &= T(J)_{P(I)} = J_{P(I)} = K_I \in B_2 \\
 &0 \leq I < 2^n, 0 \leq J, K < 2^{2^n}, P \in \Omega(2^n)
 \end{aligned} \tag{5}$$

Proposition 2 *The Truth Table under permutation operation on 2^n meta-states can generate $2^n!$ sequences for 2^{2^n} length of integers.*

Proof For any $P \in \Omega(2^n)$, 2^n are independent, it is composed of $\Omega(2^n)$ elements. ■

For the one-variable condition (i.e. $n = 1$), there are only two possible arrangements. The initial sequence is represented as $\mathbf{S} = S_1S_0 = 10$, and a permutation operation generates the output $P(\mathbf{S}) = S_0S_1 = 01$. The following shows two groups of results:

Mate-state	\mathbf{S}	1	0	$P(\mathbf{S})$	0	1
Function	J			$P(J)$		
0	0	0	0	0	0	0
\bar{x}	1	0	1	2	1	0
x	2	1	0	1	0	1
1	3	1	1	3	1	1

For any permutation operation, the function $T(J) = P(T(J))$ is always invariant. The inequality $J \neq K = P(J)$ holds in general.

3 Fourth Level of Organisation

Building upon the three levels (variables, states and functions), a fourth level of organisation is introduced.

3.1 Complementary Operation

Definition 3 Complementary Operator, for any binary (0–1) variable $y \in B_2$, let the relevant index $\delta \in B_2$ be a complementary operator:

$$y^\delta = \begin{cases} \bar{y} & \delta = 0 \\ y & \delta = 1 \end{cases} \tag{6}$$

Definition 4 Complementary Function Operation, for any n variable function of 2^n meta function vectors $\mathbf{S} = S_{2^n-1} \dots S_I \dots S_0$ Let $\Delta = \delta_{2^n-1} \dots \delta_I \dots \delta_0, 0 \leq I < 2^n, \delta_I \in B_2, \Delta \in B_2^{2^n}$.

For this type of complementary operations on function, Δ is

$$\begin{aligned}
\Delta &: T(J) \rightarrow K; J, K \in B_2^{2^n}, 0 \leq J, K < 2^{2^n} \\
\mathbf{S}^\Delta &= S_{2^{2^n-1}}^{\delta_{2^n-1}} \dots S_I^{\delta_I} \dots S_0^{\delta_0}, S_I \in B_2^{2^n} \\
T(J)^\Delta &= T(S_{2^{2^n-1}}^{\delta_{2^n-1}}(J_{2^n-1})) \dots T(S_I^{\delta_I}(J_I)) \dots T(S_0^{\delta_0}(J_0)) \\
&= T(J)_{2^{2^n-1}}^{\delta_{2^n-1}} \dots T(J)_I^{\delta_I} \dots T(J)_0^{\delta_0} \\
&= K_{2^{2^n-1}} \dots K_I \dots K_0 = K \in B_2^{2^n} \\
T(J)_I^{\delta_I} &= T(S_I^{\delta_I}(J_I)) = J_I^{\delta_I} = K_I \in B_2 \\
&0 \leq I < 2^n, 0 \leq J, K < 2^{2^n}, \delta_I \in \Delta
\end{aligned} \tag{7}$$

3.2 Invariant Logic Functions Under Permutation and Complementary

Definition 5 Permutation and Complementary Operations. For any of the n variables expressed as 2^n meta vectors, Complementary Operations $\Delta \in B_2^{2^n}$ and Permutation Operations $P \in \Omega(2^n)$ are expressed as

$$\begin{aligned}
(P, \Delta) &: T(J) \rightarrow K; J, K \in B_2^{2^n}, P \in \Omega(2^n), \Delta \in B_2^{2^n} \\
P(T(J)^\Delta) &= P(T(S_{2^{2^n-1}}^{\delta_{2^n-1}}(J_{2^n-1}))) \dots P(T(S_I^{\delta_I}(J_I))) \dots P(T(S_0^{\delta_0}(J_0))) \\
&= P(T(J)_{2^{2^n-1}}^{\delta_{2^n-1}}) \dots P(T(J)_I^{\delta_I}) \dots P(T(J)_0^{\delta_0}) \\
&= K_{2^{2^n-1}} \dots K_I \dots K_0 = K \in B_2^{2^n} \\
P(T(J)_I^{\delta_I}) &= P(T(S_I^{\delta_I}(J_I))) = J_{P(I)}^{\delta_{P(I)}} = K_I \in B_2 \\
&0 \leq I < 2^n, 0 \leq J, K < 2^{2^n}, P \in \Omega(2^n), \delta_I \in \Delta
\end{aligned} \tag{8}$$

3.3 Logic Functional Spaces

Theorem 1 (Logic Function Invariants under Permutation & Complementary Operations) *For any logic function, the output of Method 2 provides an equivalent output as the original Truth Table under all conditions.*

Proof A J th row on the permutation and complementary table of $P(T^\Delta)$ for any $I \in B_2^n, J \in B_2^{2^n}$ is constructed by

$$P(T(J)_I^\Delta) = T(J)_{P(I)}^{\delta_{P(I)}} = \begin{cases} \neg T(J)_I & \delta_{P(I)} = 0 \\ T(J)_I & \delta_{P(I)} = 1 \end{cases} \tag{9}$$

Counting Order	7	6	5	4	3	2	1	0	
S	111	110	101	100	011	010	001	000	Binary counting
0	0	0	0	0	0	0	0	0	a full 0 vector
Δ	1	1	0	0	1	1	0	0	a Δ - vector
$\neg\Delta$	0	0	1	1	0	0	1	1	a not Δ - vector
1	1	1	1	1	1	1	1	1	a full 1 vector
$T(178)$	1	0	1	1	0	0	1	0	initial value
$T(178)^1$	1	0	1	1	0	0	1	0	$T(178)$ Truth
$T(178)^{\neg\Delta}$	0	1	1	1	1	1	1	0	$T(178)$ Δ -Variant
$T(178)^0$	0	1	0	0	1	1	0	1	$T(178)$ False
$T(178)^\Delta$	1	0	0	0	0	0	0	1	$T(178)$ Δ -Invariant

Method 2: Permutation and Complementary Methods Table $P(T^\Delta)$

Input: \mathbf{x} : n variables in a binary $\{0, 1\}$ sequence, J : is the selected function number, $P \in \Omega(2^n)$ and $\Delta \in B_2^{2^n}$ are Permutation and Complementary operators

Process: Input sequence \mathbf{x} is established, the $P(I)$ -th column is selected using the meta-state number I . This represents the I -th column of the function $P(T(J)^\Delta)$

Output: If $\delta_{P(I)} = 1$, return the value of $T(J)_{P(I)}^{\delta_{P(I)}}$ (1 for true and 0 for false);
 if $\delta_{P(I)} = 0$, return $\neg T(J)_{P(I)}^{\delta_{P(I)}}$.

After using Method 2, the results are shown:

$$P(T(J)^\Delta)_I = \begin{cases} \neg\neg T(J)_I = T(J)_I & \delta_{P(I)} = 0 \\ T(J)_I & \delta_{P(I)} = 1 \end{cases} \tag{10}$$

■

Theorem 2 (Permutation Group for Meta Function Vector) *For 2^n meta function vectors, a total of permutation numbers is $2^n!$.*

Theorem 3 (Permutation & Complementary Structure) *Under permutation and complementary operations, a total of $2^n!2^{2^n}$ permutations can be generated to form a logic functional space for the n variables.*

4 Different Coding Schemes: One- and Two-Dimensional Representations

The initial step to construct a series of logic functionals. Permutation and complementary differences can be shown in the proposed invariant function structures. Different coding schemes under different symmetric restrictions are established. Four schemes are described, in which one of them is in one-dimensional representation and other three schemes are two-dimensional representations. For binary sequences in sequential counting order, the scheme is known as the SL (Shao Yong & Leibniz) coding scheme.

4.1 G Coding

The General Code (G) is used to map permutation & complementary operations. For any state in the G coding scheme having 2^n bits,

$$G : (J, \Delta, P) \rightarrow K; J, K \in B_2^{2^n}; \Delta \in B_2^{2^n}, P \in \Omega. \quad (11)$$

4.2 W Coding

From the G coding scheme, their bit numbers are separated into two equal parts in the same bits to form a 2D representation. This mapping mechanism can represent a function space as a W coding scheme.

$$W : (J, \Delta, P) \rightarrow K = \langle J^1 | J^0 \rangle \quad (12)$$

$$J, K \in B_2^{2^n}; J^1, J^0 \in B_2^{2^{n-1}}; S^1, S^0 \in \mathbf{S}, \Delta \in B_2^{2^n}, P \in \Omega$$

Under this representation, a given logic functional for the function space is illustrated as a fixed matrix.

$$\{W(J)\}_{J=0}^{2^{2^n}} = \begin{matrix} & \langle 0|0 \rangle & \dots & \langle 0|J^0 \rangle & \dots & \langle 0|2^{2^{n-1}} - 1 \rangle \\ \dots & & & \dots & & \dots \\ \langle J^1|0 \rangle & \dots & \langle J^1|J^0 \rangle & \dots & \langle J^1|2^{2^{n-1}} - 1 \rangle & \\ \dots & & \dots & & \dots & \\ \langle 2^{2^{n-1}} - 1|0 \rangle & \dots & \langle 2^{2^{n-1}} - 1|J^0 \rangle & \dots & \langle 2^{2^{n-1}} - 1|2^{2^{n-1}} - 1 \rangle & \end{matrix} \quad (13)$$

$$0 \leq J^0, J^1 < 2^{2^{n-1}}; 0 \leq J < 2^{2^n}$$

In the one-variable condition, there are eight cases in their logic functional spaces as follows:

f	J^{11}, T	W	$J^{01}, \Delta-V$	W	$J^{10}, \Delta-IV$	W	J^{00}, F	W
0	0	$\langle 0 0 \rangle$	2	$\langle 1 0 \rangle$	1	$\langle 0 1 \rangle$	3	$\langle 1 1 \rangle$
\bar{x}	1	$\langle 0 1 \rangle$	3	$\langle 1 1 \rangle$	0	$\langle 0 0 \rangle$	2	$\langle 1 0 \rangle$
x	2	$\langle 1 0 \rangle$	0	$\langle 0 0 \rangle$	3	$\langle 1 1 \rangle$	1	$\langle 0 1 \rangle$
1	3	$\langle 1 1 \rangle$	1	$\langle 0 1 \rangle$	2	$\langle 1 0 \rangle$	0	$\langle 0 0 \rangle$
f	$P(J^{11}), T$	W	$P(J^{01}), \Delta-V$	W	$P(J^{10}), \Delta-IV$	W	$P(J^{00}), F$	W
0	0	$\langle 0 0 \rangle$	1	$\langle 0 1 \rangle$	2	$\langle 1 0 \rangle$	3	$\langle 1 1 \rangle$
\bar{x}	2	$\langle 1 0 \rangle$	3	$\langle 1 1 \rangle$	0	$\langle 0 0 \rangle$	1	$\langle 0 1 \rangle$
x	1	$\langle 0 1 \rangle$	0	$\langle 0 0 \rangle$	3	$\langle 1 1 \rangle$	2	$\langle 1 0 \rangle$
1	3	$\langle 1 1 \rangle$	2	$\langle 1 0 \rangle$	1	$\langle 0 1 \rangle$	0	$\langle 0 0 \rangle$

For better visualisation and expression, the one-dimensional G coding scheme is converted into a two-dimensional W coding scheme.

$$W = \begin{array}{|c|c|} \hline \text{Truth} & \Delta\text{-Variant} \\ \hline 0 \bar{x} & x 1 \\ x 1 & 0 \bar{x} \\ \hline \bar{x} 0 & 1 x \\ 1 x & \bar{x} 0 \\ \hline \Delta\text{-Invariant} & \text{False} \\ \hline \end{array} \quad PW = \begin{array}{|c|c|} \hline \text{Truth} & \Delta\text{-Variant} \\ \hline 0 x & x 0 \\ \bar{x} 1 & 1 \bar{x} \\ \hline \bar{x} 1 & 1 \bar{x} \\ 0 x & x 0 \\ \hline \Delta\text{-Invariant} & \text{False} \\ \hline \end{array}$$

4.3 F Coding

Using 2D representation, symmetric condition can be added to arrange meta-states into specific order. For each pair of states in W, if they satisfy following condition, then a refined code: F coding scheme is determined.

$$\begin{array}{ccc} J^1 \text{ the } I\text{th meta-state} & \Leftrightarrow & J^0 \text{ the } I\text{th meta-state} \\ \Downarrow & \text{F coding scheme} & \Downarrow \\ X \in S^1 & \Leftrightarrow & \bar{X} \in S^0 \end{array}$$

4.4 C Coding

In addition to a pair of states in complementary relationship, further structure is introduced onto F code. When the pair of states in F have the same values in their *i*th position, they form a C coding scheme.

$$\begin{array}{ccc} S^1 \text{ the } I\text{th} & \Leftrightarrow & S^0 \text{ the } I\text{th} \quad \text{F coding scheme} \\ \Downarrow & \text{C coding scheme} & \Downarrow \quad + \\ \forall x_i \in S^1, x_i = 1(0) & \Leftrightarrow & \forall x_i \in S^0, x_i = 0(1) \text{ general conjugate} \end{array}$$

The C coding scheme, have the strongest symmetric conditions available. Only a relatively small number among the three invariant groups can be identified within this scheme.

5 Two-Variable Cases

Four groups of the proposed schemes are selected as examples. Each group of a logic functional represents 16 logic functions as 4×4 images. 4 groups are arranged as 2×2 blocks to arrange as Truth/False, Δ -Variant/ Δ -Invariant properties. The 2×2 blocks correspond to:

Truth Block	Δ -Variant
Δ -Invariant	False Block

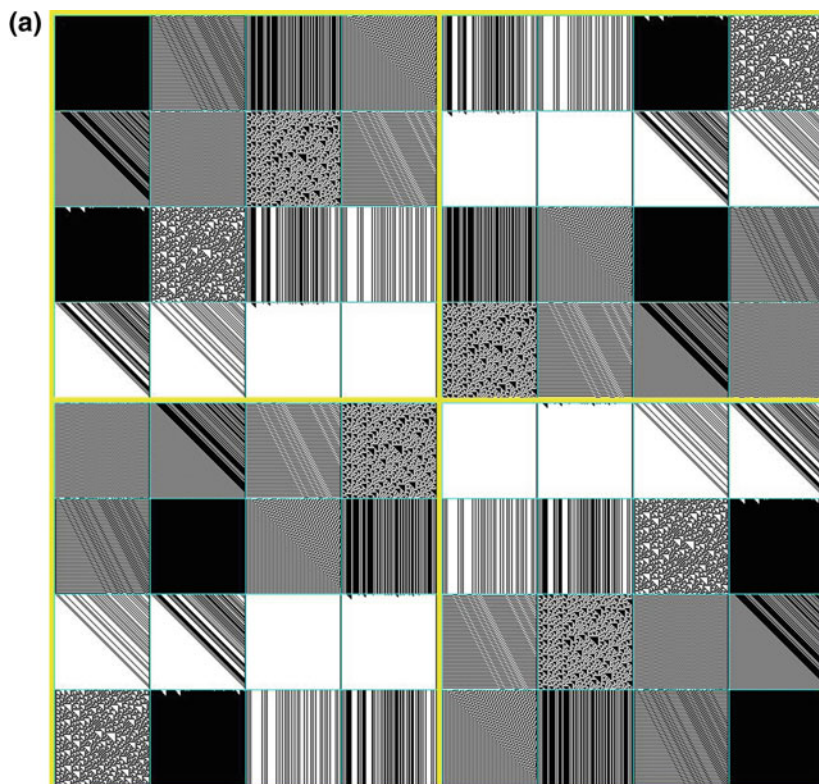
Each block contains 16 entries of function images as a 4×4 ($2^2 \times 2^2$) configuration. Each image entry denotes a transformed number and its function number in the form: $\begin{matrix} \langle J^1 | J^0 \rangle \\ J \end{matrix}$ where $K = \langle J^1 | J^0 \rangle$ is a transformed number and J is the function number. In all four figures, (a) 2×2 base blocks to represent function images and (b) 2×2 vector blocks to represent relevant coding schemes respectively.

In Fig. 1, the counting order of meta-states has been arranged as W coding (SL code): $P = (3210)$, $P(\Delta) = 1010$. In this group, only Functions 6 and 9 can be observed in complementary symmetric condition in main diagonal direction.

In Fig. 2, variation the configurations among W coding: $P = (2301)$, $P(\Delta) = 0101$ creates similar effects seen in Fig. 1.

In Fig. 3, the F coding scheme is shown: under this configuration, $P = (2310)$, $P(\Delta) = 0110$. Six pairs (0:15, 1:7, 2:11, 4:13, 6:9, 8:14) of complementary functions can be identified. The group has four blocks containing the same pairs of configurations.

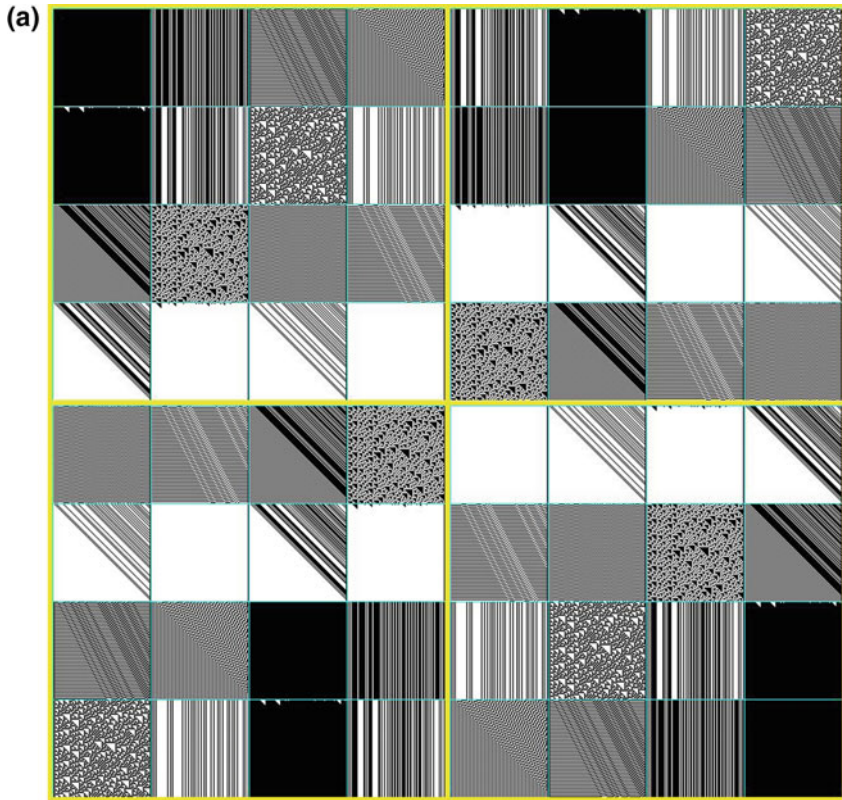
In Fig. 4, C coding has represented: $P = (3102)$, $P(\Delta) = 1100$. In addition to six pairs as same as F coding, four corners are 4 functions (0, 5, 10, 15) in all blocks. This makes most regular structures compared to all other coding schemes.



(b)

$\langle 0,0 \rangle$ 0	$\langle 0,1 \rangle$ 1	$\langle 0,2 \rangle$ 2	$\langle 0,3 \rangle$ 3	$\langle 0,0 \rangle$ 10	$\langle 0,1 \rangle$ 11	$\langle 0,2 \rangle$ 8	$\langle 0,3 \rangle$ 9
$\langle 1,0 \rangle$ 4	$\langle 1,1 \rangle$ 5	$\langle 1,2 \rangle$ 6	$\langle 1,3 \rangle$ 7	$\langle 1,0 \rangle$ 14	$\langle 1,1 \rangle$ 15	$\langle 1,2 \rangle$ 12	$\langle 1,3 \rangle$ 13
$\langle 2,0 \rangle$ 8	$\langle 2,1 \rangle$ 9	$\langle 2,2 \rangle$ 10	$\langle 2,3 \rangle$ 11	$\langle 2,0 \rangle$ 2	$\langle 2,1 \rangle$ 3	$\langle 2,2 \rangle$ 0	$\langle 2,3 \rangle$ 1
$\langle 3,0 \rangle$ 12	$\langle 3,1 \rangle$ 13	$\langle 3,2 \rangle$ 14	$\langle 3,3 \rangle$ 15	$\langle 3,0 \rangle$ 6	$\langle 3,1 \rangle$ 7	$\langle 3,2 \rangle$ 4	$\langle 3,3 \rangle$ 5
$\langle 0,0 \rangle$ 5	$\langle 0,1 \rangle$ 4	$\langle 0,2 \rangle$ 7	$\langle 0,3 \rangle$ 6	$\langle 0,0 \rangle$ 15	$\langle 0,1 \rangle$ 14	$\langle 0,2 \rangle$ 13	$\langle 0,3 \rangle$ 12
$\langle 1,0 \rangle$ 1	$\langle 1,1 \rangle$ 0	$\langle 1,2 \rangle$ 3	$\langle 1,3 \rangle$ 2	$\langle 1,0 \rangle$ 11	$\langle 1,1 \rangle$ 10	$\langle 1,2 \rangle$ 9	$\langle 1,3 \rangle$ 8
$\langle 2,0 \rangle$ 13	$\langle 2,1 \rangle$ 12	$\langle 2,2 \rangle$ 15	$\langle 2,3 \rangle$ 14	$\langle 2,0 \rangle$ 7	$\langle 2,1 \rangle$ 6	$\langle 2,2 \rangle$ 5	$\langle 2,3 \rangle$ 4
$\langle 3,0 \rangle$ 9	$\langle 3,1 \rangle$ 8	$\langle 3,2 \rangle$ 11	$\langle 3,3 \rangle$ 10	$\langle 3,0 \rangle$ 3	$\langle 3,1 \rangle$ 2	$\langle 3,2 \rangle$ 1	$\langle 3,3 \rangle$ 0

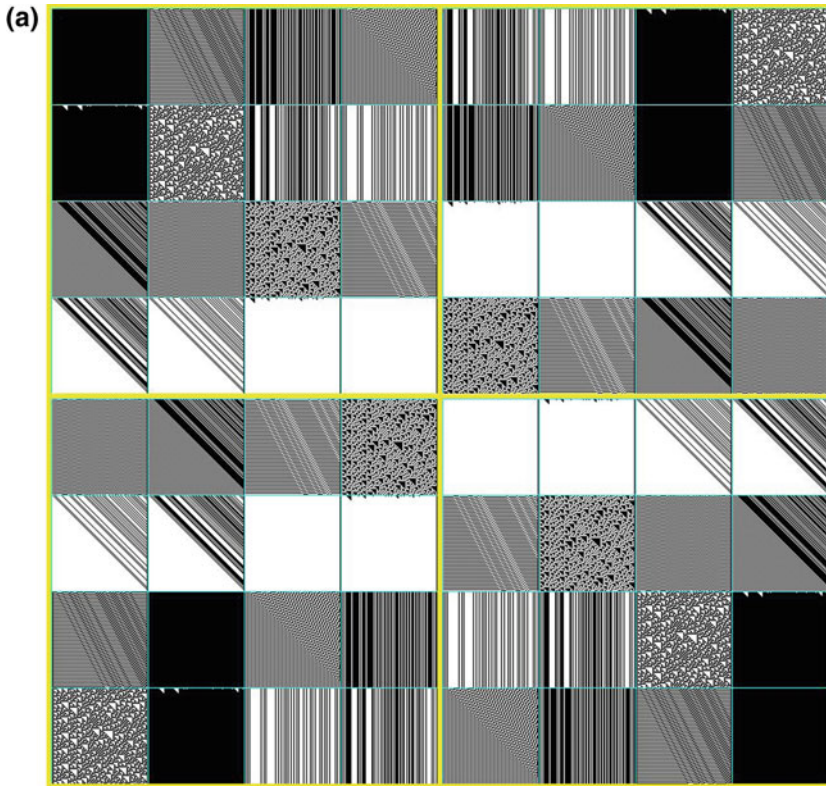
Fig. 1 W coding (SL code): $P = (3210)$, $P(\Delta) = 1010$; **a** 2×2 base blocks **b** 2×2 vector blocks



(b)

$\langle 0,0 \rangle$ 0	$\langle 0,1 \rangle$ 2	$\langle 0,2 \rangle$ 1	$\langle 0,3 \rangle$ 3	$\langle 0,0 \rangle$ 10	$\langle 0,1 \rangle$ 8	$\langle 0,2 \rangle$ 11	$\langle 0,3 \rangle$ 9
$\langle 1,0 \rangle$ 8	$\langle 1,1 \rangle$ 10	$\langle 1,2 \rangle$ 9	$\langle 1,3 \rangle$ 11	$\langle 1,0 \rangle$ 2	$\langle 1,1 \rangle$ 0	$\langle 1,2 \rangle$ 3	$\langle 1,3 \rangle$ 1
$\langle 2,0 \rangle$ 4	$\langle 2,1 \rangle$ 6	$\langle 2,2 \rangle$ 5	$\langle 2,3 \rangle$ 7	$\langle 2,0 \rangle$ 14	$\langle 2,1 \rangle$ 12	$\langle 2,2 \rangle$ 15	$\langle 2,3 \rangle$ 13
$\langle 3,0 \rangle$ 12	$\langle 3,1 \rangle$ 14	$\langle 3,2 \rangle$ 13	$\langle 3,3 \rangle$ 15	$\langle 3,0 \rangle$ 6	$\langle 3,1 \rangle$ 4	$\langle 3,2 \rangle$ 7	$\langle 3,3 \rangle$ 5
$\langle 0,0 \rangle$ 5	$\langle 0,1 \rangle$ 7	$\langle 0,2 \rangle$ 4	$\langle 0,3 \rangle$ 6	$\langle 0,0 \rangle$ 15	$\langle 0,1 \rangle$ 13	$\langle 0,2 \rangle$ 14	$\langle 0,3 \rangle$ 12
$\langle 1,0 \rangle$ 13	$\langle 1,1 \rangle$ 15	$\langle 1,2 \rangle$ 12	$\langle 1,3 \rangle$ 14	$\langle 1,0 \rangle$ 7	$\langle 1,1 \rangle$ 5	$\langle 1,2 \rangle$ 6	$\langle 1,3 \rangle$ 4
$\langle 2,0 \rangle$ 1	$\langle 2,1 \rangle$ 3	$\langle 2,2 \rangle$ 0	$\langle 2,3 \rangle$ 2	$\langle 2,0 \rangle$ 11	$\langle 2,1 \rangle$ 9	$\langle 2,2 \rangle$ 10	$\langle 2,3 \rangle$ 8
$\langle 3,0 \rangle$ 9	$\langle 3,1 \rangle$ 11	$\langle 3,2 \rangle$ 8	$\langle 3,3 \rangle$ 10	$\langle 3,0 \rangle$ 3	$\langle 3,1 \rangle$ 1	$\langle 3,2 \rangle$ 2	$\langle 3,3 \rangle$ 0

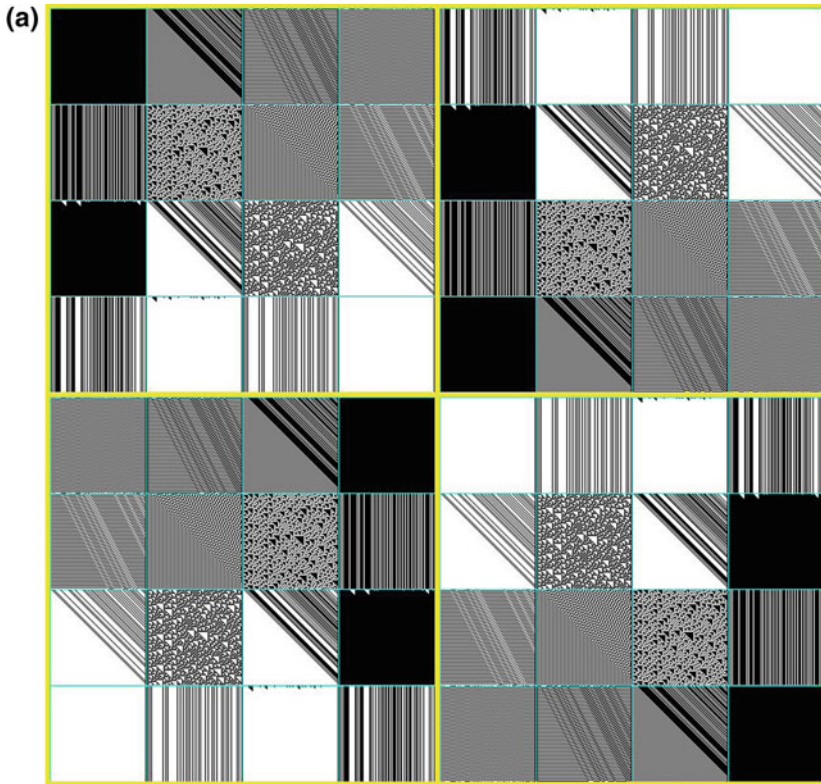
Fig. 2 W coding: $P = (2301)$, $P(\Delta) = 0101$; **a** 2×2 base blocks **b** 2×2 vector blocks



(b)

<0,0> 0	<0,1> 1	<0,2> 2	<0,3> 3	<0,0> 10	<0,1> 11	<0,2> 8	<0,3> 9
<1,0> 8	<1,1> 9	<1,2> 10	<1,3> 11	<1,0> 2	<1,1> 3	<1,2> 0	<1,3> 1
<2,0> 4	<2,1> 5	<2,2> 6	<2,3> 7	<2,0> 14	<2,1> 15	<2,2> 12	<2,3> 13
<3,0> 12	<3,1> 13	<3,2> 14	<3,3> 15	<3,0> 6	<3,1> 7	<3,2> 4	<3,3> 5
<0,0> 5	<0,1> 4	<0,2> 7	<0,3> 6	<0,0> 15	<0,1> 14	<0,2> 13	<0,3> 12
<1,0> 13	<1,1> 12	<1,2> 15	<1,3> 14	<1,0> 7	<1,1> 6	<1,2> 5	<1,3> 4
<2,0> 1	<2,1> 0	<2,2> 3	<2,3> 2	<2,0> 11	<2,1> 10	<2,2> 9	<2,3> 8
<3,0> 9	<3,1> 8	<3,2> 11	<3,3> 10	<3,0> 3	<3,1> 2	<3,2> 1	<3,3> 0

Fig. 3 F coding: $P = (2310)$, $P(\Delta) = 0110$; **a** 2×2 base blocks **b** 2×2 vector blocks



(b)

$\langle 0,0 \rangle$ 0	$\langle 0,1 \rangle$ 4	$\langle 0,2 \rangle$ 1	$\langle 0,3 \rangle$ 5	$\langle 0,0 \rangle$ 10	$\langle 0,1 \rangle$ 14	$\langle 0,2 \rangle$ 11	$\langle 0,3 \rangle$ 15
$\langle 1,0 \rangle$ 2	$\langle 1,1 \rangle$ 6	$\langle 1,2 \rangle$ 3	$\langle 1,3 \rangle$ 7	$\langle 1,0 \rangle$ 8	$\langle 1,1 \rangle$ 12	$\langle 1,2 \rangle$ 9	$\langle 1,3 \rangle$ 13
$\langle 2,0 \rangle$ 8	$\langle 2,1 \rangle$ 12	$\langle 2,2 \rangle$ 9	$\langle 2,3 \rangle$ 13	$\langle 2,0 \rangle$ 2	$\langle 2,1 \rangle$ 6	$\langle 2,2 \rangle$ 3	$\langle 2,3 \rangle$ 7
$\langle 3,0 \rangle$ 10	$\langle 3,1 \rangle$ 14	$\langle 3,2 \rangle$ 11	$\langle 3,3 \rangle$ 15	$\langle 3,0 \rangle$ 0	$\langle 3,1 \rangle$ 4	$\langle 3,2 \rangle$ 1	$\langle 3,3 \rangle$ 5
$\langle 0,0 \rangle$ 5	$\langle 0,1 \rangle$ 1	$\langle 0,2 \rangle$ 4	$\langle 0,3 \rangle$ 0	$\langle 0,0 \rangle$ 15	$\langle 0,1 \rangle$ 11	$\langle 0,2 \rangle$ 14	$\langle 0,3 \rangle$ 10
$\langle 1,0 \rangle$ 7	$\langle 1,1 \rangle$ 3	$\langle 1,2 \rangle$ 6	$\langle 1,3 \rangle$ 2	$\langle 1,0 \rangle$ 13	$\langle 1,1 \rangle$ 9	$\langle 1,2 \rangle$ 12	$\langle 1,3 \rangle$ 8
$\langle 2,0 \rangle$ 13	$\langle 2,1 \rangle$ 9	$\langle 2,2 \rangle$ 12	$\langle 2,3 \rangle$ 8	$\langle 2,0 \rangle$ 7	$\langle 2,1 \rangle$ 3	$\langle 2,2 \rangle$ 6	$\langle 2,3 \rangle$ 2
$\langle 3,0 \rangle$ 15	$\langle 3,1 \rangle$ 11	$\langle 3,2 \rangle$ 14	$\langle 3,3 \rangle$ 10	$\langle 3,0 \rangle$ 5	$\langle 3,1 \rangle$ 1	$\langle 3,2 \rangle$ 4	$\langle 3,3 \rangle$ 0

Fig. 4 C coding: $P = (3102)$, $P(\Delta) = 1100$; **a** 2×2 base blocks **b** 2×2 vector blocks

6 Conclusion

It is shown in this chapter that the arrangement of binary function space using four levels of classification can be used to add symmetry and regular structure onto the entire space of binary functions. For ease of visualisation, it is convenient to apply 2D representation mechanism that enables symmetric configurations of the system to be analysed via different coding schemes. Binary functional spaces provide additional optimal information to generate large numbers of potential configurations in order to arrange and organise logic phase spaces.

The mechanism can be developed further to establish a solid logic foundation on logic functional levels for theoretical explorations and practical applications. We aim to make refined investigation on different coding schemes within the highest levels of organisation in our future work.

Acknowledgements Thanks Mr. J. Wan for generation all sample images and configurations and Dr. D. Heim for editing the chapter. Financial support was given by School of Software, Yunnan University.

References

1. T.A. Springer, *Invariant Theory* (Springer, Berlin, 1977)
2. A. Holden, *Shapes, Spaces and Symmetry* (Columbia University Press, New York, 1971)
3. G. Birkhoff, *Lattice Theory*, vol. 25, 3rd edn. (American Mathematical Society Colloquium Publications, 1984)
4. R.P. Burn, *Groups: A Path to Geometry* (Cambridge University Press, Cambridge, 1985)
5. H. Weyl, *Symmetry* (Princeton, 1952)
6. M. Bonnet, *Handbook of Boolean Algebras* (North Holland, 1989)
7. R. Sikorski, *Boolean Algebra* (Springer, Berlin, 1960)
8. S. Lee, *Modern Switching Theory and Digital Design* (Prentice-Hall Inc., Englewood Cliffs, 1978)
9. S. Vingron, *Switching Theory: Insight Through Predicate Logic* (Springer, Berlin, 2004)
10. H. Umeo, S. Morishita, K. Nishinari, *Cellular Automata* (Springer, Berlin, 2008)
11. S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific, Singapore, 1986)
12. S. Wolfram, *Cellular Automata and Complexity* (Addison-Wesley, New York, 1994)
13. S. Wolfram, *A New Kind of Science* (Wolfram Media Inc., Champaign, 2002). <http://www.wolframscience.com/>
14. A. Ilachinski, *Cellular Automata—A Discrete Universe* (World Scientific, Singapore, 2001)
15. Z. Zheng, C. Leung, Visualising global behaviour of 1d cellular automata image sequences in 2d maps. *Phys. A* **233**, 785–800 (1996)
16. P. Dunn, *The Complexity of Boolean Networks* (Academic Press, New York, 1988)
17. M. Paterson, *Boolean Function Complexity* (Cambridge University Press, Cambridge, 1992)
18. M. Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, Oxford, 1972)
19. G. Leibniz, *Philosophical Papers and Letters* (Springer, Berlin, 1976)
20. G. Leibniz, R. Ariew, D. Garber, *Philosophical Essays* (Hackett Publishing, 1989)
21. G. Boole, *An Investigation of the Laws of Thought* (Dover, 1850/1940/1958)
22. J. Dawson, *Logical Dilemmas—The Life and Work of Kurt gödel* (A.K. Peters, Ltd., Wellesley, 2005)

23. W. Demopoulos, *Frege's Philosophy of Mathematics* (Harvard University Press, Cambridge, 1995)
24. B. Russell, *Principles of Mathematics* (Forgotten Books, 1942)
25. D. Hilbert, *Grundlagen der Geometrie* (Göttingen, 1899)
26. A. Turing, On computable numbers, with an application to the entscheidungs problem. Proc. Lond. Math. Soc. Ser. 2 **42**, 230–265 (1936)
27. C. Shannon, N. Sloane, A. Wyner, *Claude Elwood Shannon: Collected Papers* (IEEE Press, IEEE Information Theory Society, 1993)
28. H. Reichenbach, *The Theory of Probability* (The University of California Press, Berkeley, 1949)
29. L. Zadeh, Fuzzy sets. Information and Control **8**, 338–357 (1965)
30. W. Chu, W. Sherrill, *An Anthology of I Ching* (Routledge and Kegan Paul Ltd., London, 1977)
31. J. Cooper, Yin and Yang, The Taoist Harmony of Opposites (The Theftford Press, 1981)
32. L. Govinda, *The Inner Structure of I Ching: The Book of Transformation* (Wheelwright Press, 1981)
33. D. Hook, *The I Ching and Mankind* (Routledge and Kegan Paul Ltd., London, 1975)
34. I. Shchutshii, *Researches on the I Ching* (Princeton University Press, Princeton, 1979)
35. G. Whincup, *Rediscovering of I Ching* (GreyWhincup, 1986)
36. H. Wilhelmi, *Change: Eight Lectures on I Ching* (Princeton University Press, Princeton, 1979)
37. R. Wilhelmi, *Lectures on the I Ching: Constancy and Change* (Princeton University Press, Princeton, 1979)
38. J. Needham, L. Wang, *Science and Civilisation in China: History of Scientific Thought* (Cambridge University Press, Cambridge, 1954–1988), p. 2
39. D. Griffefeath, C. Moor, *New constructions in cellular automata*, Santa Fe Institute Studies in the Sciences of Complexity (2003)
40. J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic. Front. Electr. Electron. Eng. China **5**(2), 163–172 (2010) (Higher Education Press and Springer). <http://www.springerlink.com/content/91474403127n446u/>
41. T. Kunii, Y. Takai, Cellular self-reproducing automata as a parallel processing model for botanical colony growth pattern simulation, in *New Advances in Computer Graphics* (Springer, Berlin, 1989), pp. 7–22
42. T. Kunii, H. Hioki, Y. Shinagawa, Visualizing Highly Abstract Mathematical Concepts: A Case Study in Animation of Homology Groups, in *Multimedia Modeling* (World Scientific, Singapore, 1993), pp. 3–30
43. T. Kunii, H. Kunii, A cellular model for information systems on the web—integrating local and global information, in *Proceedings of 1999 International Symposium on Database Applications in Non-Traditional Environments* (IEEE CS Press, 1999)
44. T. Kunii, Y. Shinagawa, *Modern Geometric Computing for Visualization* (Springer, Berlin, 1992)
45. T. Kunii, S. Takahashi, Area guide map modeling by manifolds and CW-complexes, in *Modeling in Computer Graphics* (Springer, Berlin, 1993), pp. 5–20
46. T. Kunii, Y. Tsuchida, Y. Arai, H. Matsuda, M. Shirahama, S. Miura, A model of hands and arms based on manifold mappings, in *Communicating with Virtual Worlds* (Springer, Berlin, 1993), pp. 381–398
47. Z. Zheng, A. Maeder, The conjugate classification of the kernel form of the hexagonal grid, in *Modern Geometric Computing for Visualization* (Springer, Berlin, 1992), pp. 73–89
48. Z. Zheng, Conjugate transformation of regular plan lattices for binary images, Ph.D. Thesis, Monash University, 1994
49. Z. Zheng, Conjugate visualisation of global complex behaviour. Complex. Int. **3** (1996). <http://www.complexity.org.au/ci/vol03/zheng/>
50. Z. Zheng, A. Maeder, The elementary equation of the conjugate transformation for hexagonal grid, in *Modeling in Computer Graphics* (Springer, Berlin, 1993), pp. 21–42

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Hierarchical Organization of Variant Logic



Jeffrey Zheng

Abstract In modern logic, various systems have been proposed extending classical Boolean logic & switching theory. Such logic frameworks include multiple-valued logic, probability logic, fuzzy logic, module logic, quantum logic and various other frameworks. Although these extensions have been applied to many applications in mathematics, in science and in engineering, all extensions to Boolean logic invalidates at least one of the six fundamental rules of Boolean logic shown in L1 to L6. We propose a new framework of logic, variant logic, extending Boolean logic whilst satisfying the six fundamental rules (L1–L6). By defining the Variant–Invariant behaviour of logical operations, this framework can be constructed using four types of general operators. Main results of the chapter are summarized in **Theorems 8–10**, respectively. To show significant differences between classical logic and new variant logic, invariant properties of this hierarchical organization are discussed. Simplest cases of one-variable conditions are illustrated. Variant logic can provide the necessary framework to support analysis and description of Cellular Automata, Fractal Theory, Chaos Theory and other systems dealing with complexity. Such applications of this framework will be explored in future papers.

Keywords Switching theory · Boolean/multiple valued/probability/fuzzy logic
Variant/invariant property · Hierarchical organization · Variant logic

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_2

1 Laws of Logic Systems

1.1 Laws in Classical Logic Systems

Classical logic identifies a class of formal logic that are characterized by a number of properties [1–17].

Definition 1 For any logic system if all CL1–CL5 are satisfied, then it is a classical logic system. The five properties of classical logic (CL1–CL5) are listed as follows:

CL1: Law of the excluded middle and double negative elimination

CL2: Law of non-contradiction

CL3: Monotonicity and idempotency of entailment

CL4: Commutativity of conjunction

CL5: De Morgan duality

Examples of such classical logic systems include works of philosophy and religion (Aristotle’s Organon; Nagarjuna’s tetralemma; and Avicenna’s temporal modal logic) as well as foundational logic systems such as reformulations by George Bool and Gottlob Frege [4–17]. These properties can be rewritten as simplified equations describing basic properties of a logic system using characteristics of the five classical properties. The following equations (L1–L6) describe such a system.

L1: $P \cup P = P$ Idempotency

L2: $P \cap P = P \dots$

L3: $\neg P \cup P = P$ Excluded Middle

L4: $\neg P \cap P = P \dots$

L5: $\neg\neg P = P$ Double Negative Elimination

L6: $P, P \rightarrow Q$

The set of equations can be applied in the analysis of modern logic systems to determine if they are all satisfied. The equations will be defined as canonical properties and a logic system satisfying all six properties will be defined as a canonical system. If any logic system does not, they are categorized as non-canonical.

1.2 Current Logic Systems

Many modern logic systems cannot satisfy the six canonical properties. Three-valued logic proposed by Lukasiewicz 1920 can satisfy L3–L6, cannot satisfy L1–L4. Probability logic proposed by Reichenbach 1949 can satisfy L5–L6, cannot satisfy L1–L4. Fuzzy logic proposed by Zadeh 1965 satisfy L1, L2, L5, L6, cannot satisfy L3–L4. Since they cannot satisfy canonical properties, they are all non-canonical logic systems [1–22].

2 Truth Valued Representation in Boolean Logic Systems

For any n -variable Boolean logic system, it is natural to establish 2^n states. Under either selected or not selected operation, it can be building up a truth table for a given Boolean function. Collecting all possible selections, a full truth table is constructed in 2^n columns and 2^{2^n} rows in presentation. We can list this table as follows:

$0 \leq I < 2^n$	$2^n - 1$...	I	...	1	0
$0 \leq i < n$	1...1	...1	$I_{n-1} \dots I_i \dots I_0$...	0...0	...1 0...0...0
$0 \leq J < 2^{2^n}$						
0	0	...	0	...	0	0
1	0	...	0	...	0	1
2	0	...	0	...	1	0
...						
J	J_{2^n-1}	...	J_I	...	J_1	J_0
...						
$2^{2^n} - 2$	1	...	1	...	1	0
$2^{2^n} - 1$	1	...	1	...	1	1

where there are three parameters: $i, I, J : 0 \leq i < n, 0 \leq I < 2^n, 0 \leq J < 2^{2^n}$ corresponding to variable, state and function numbers, respectively. Under such conditions, for any J , it is convenient to use Karnaugh map or relevant logic tools to construct the given Boolean function in combination [6–17].

3 Cellular Automata Representations

Cellular Automata—CA uses a different mechanism [23–35] to represent a given function. In a one-dimensional form of CA, a N -length binary sequence is

$$X = X_{N-1}X_{N-2} \dots X_j \dots X_1X_0, 0 \leq j < N, X_j \in \{0, 1\} = B_2$$

For a given function f , the output sequence is defined as follows: $f : X \rightarrow Y, Y = f(X)$,

$$Y = Y_{N-1}Y_{N-2} \dots Y_j \dots Y_1Y_0, 0 \leq j < N, Y_j \in B_2$$

It is feasible to use a moving window with a fixed length n to separate X into a local kernel in length n . The kernel can be presented as

$$[\dots X_j \dots] = x_{n-1} \dots x_i \dots x_0, x_i \in B_2.$$

For a given function f

$$y = f(x_{n-1} \dots x_i \dots x_0)$$

It is necessary to assign a certain position i in the kernel for special care to associated with j position of both sequences. We have

$$y = f(x_{n-1} \dots x_i \dots x_0) = f(\dots X_j \dots) == Y_j$$

or $X_j = X_j^{t-1}, Y_j = X_j^t$ i.e.

$$f : X_j^{t-1} \rightarrow X_j^t, X_j^{t-1}, X_j^t \in B_2$$

4 Variant Construction

4.1 Four Variation Forms

Considering $f : X_j^{t-1} \rightarrow X_j^t$ for any function of Boolean logic system to analyse their variation properties [36–40], it is normal to have following proposition.

Proposition 1 For any $f : X_j^{t-1} \rightarrow X_j^t$ transformation, four forms of transforming classes are identified: $TA : 0 \rightarrow 0, TB : 0 \rightarrow 1, TC : 1 \rightarrow 0, TD : 1 \rightarrow 1$.

Proof X_j, Y_j are 0-1 variables, only four classes listed are possible. ■

Definition 2 Four transforming forms are corresponding to following sets: TA: Invariant class for 0 value, TB: Variant class for 0 value, TC:Variant class for 1 value, TD: Invariant class for 1 value.

Under such definition, the following proposition can be established.

Proposition 2 Using four classes of transformation, four variant operations are defined.

Type	X_j	\rightarrow	Y_j	Truth	Variant	Invariant	False
TA	0	0	0	0	0	1	1
TB	0	1	1	1	1	0	0
TC	1	0	0	1	1	0	1
TD	1	1	1	1	0	1	0

Proof Truth (False) values are determined by $Y_j(\bar{Y}_j)$ and Variant(Invariant) values are determined by {TB, TC} for 1(0) and {TA, TD} for 0(1) respectively. ■

Theorem 1 In { Truth, Variant, Invariant, False} groups, only two pairs of groups: {Truth, False} and {Variant, Invariant} satisfy L1–L6 to form a canonic logic system.

Proof Both groups are composed of 0-1 variables, in addition, Truth/False, Variant/Invariant are formed complement relationships. Other combinations contain common parts, it is not possible for them to satisfy logic canonic conditions L1–L6. ■

Definition 3 Sequential number of binary is defined as SL coding to remember Y. Shao and Leibniz contribution [41–49] on binary logic.

Definition 4 The operator $BN : J \rightarrow B$ converts an integer to its binary representation. The operator $DC : B \rightarrow J$ converts a binary number to its decimal representation.

Definition 5 The SL coding scheme is an ordering of binary table outputs $T : B_2^{2^n} \rightarrow J$. An element $J_I \in SL$ at position I , where $0 \leq I < 2^n$ represents function T_I such that the binary representation of T_I is defined as

$$BN(J) = T_{2^n-1}[J_{2^n-1}] \dots T_I[J_I] \dots T_0[J_0]$$

For any n variable structure, J is composed of 2^n bits to represent $0 \leq J < 2^{2^n}$ numbers.

Definition 6 A G coding scheme is defined as an ordering of binary table outputs $T : B_2^{2^n} \rightarrow J$. An element $J_I \in SL$ at position I where $0 \leq I < 2^n$ represents function T_I such that the binary representation of T_I is defined as

$$G = \{\forall J | T(J), 0 \leq J < 2^{2^n}\};$$

$$T(J) = T_{2^n-1}[Y(J_{2^n-1})] \dots T_I[Y(J_I)] \dots T_0[Y(J_0)], 0 \leq I < 2^n$$

Where $\{Y(J_I), 0 \leq I < 2^n\}$ are 2^{2^n} length 0-1 vectors, $Y(J_{2^n-1}) \neq \dots \neq Y(J_I) \neq \dots \neq Y(J_0)$, respectively.

Under G coding scheme, ordering number is an integer sequence with 2^{2^n} positions. Different transformations will make this sequence extremely complex. In convenient to do representation, a two-dimensional W coding scheme is proposed.

Definition 7 A W coding scheme is defined as an ordering pair of binary table outputs $T : B_2^{2^n} \rightarrow \langle J^1 | J^0 \rangle$. Each component is composed of 2^{n-1} bits in representation:

$$\langle J^1 | J^0 \rangle = T_{2^n-1}[Y(J_{2^n-1})] \dots T_I[Y(J_I)] \dots T_0[Y(J_0)], 0 \leq I < 2^n$$

$$J^0 = \{\forall I | BN(J_I \bmod 2^{n-1}), 0 \leq I < 2^{n-1}\}$$

$$J^1 = \{\forall I | BN(J_I \bmod 2^{n-1}), 2^{n-1} \leq I < 2^n\}$$

Under this construction, a G coding scheme is transformed into a W coding scheme to represent two-dimensional structure for different permutation results. In general, J^0 represents lower 2^{n-1} bits and J^1 represents higher 2^{n-1} bits, respectively. A general structure of W coding is a $2^{2^{n-1}} \times 2^{2^{n-1}}$ matrix shown in the following figure.

$\langle 0 0\rangle$...	$\langle 0 J^0\rangle$...	$\langle 0 2^{2^n-1} - 1\rangle$
...	
$\langle J^1 0\rangle$...	$\langle J^1 J^0\rangle$...	$\langle J^1 2^{2^n-1} - 1\rangle$
...	
$\langle 2^{2^n-1} - 1 0\rangle$...	$\langle 2^{2^n-1} - 1 J^0\rangle$...	$\langle 2^{2^n-1} - 1 2^{2^n-1} - 1\rangle$

$0 \leq J^0, J^1 < 2^{2^n-1} \quad \{\langle J^1|J^0\rangle\}$: 2D Space for 2^{2^n} Functions

4.2 Complement and Variant Operators

Definition 8 In B_2^n , the generalized complement Y^Q , $Q \in B_2^{2^n}$ of a variable Y is defined to be the element obtained from complementing the components of Y according to the value of corresponding component of Q ; Y_I is complemented or un-complemented if Q_I is 0 or 1, respectively, where Y_I and Q_I designate the I th component of Y and Q .

For example, given B_2^4 for $Q = \{0101, 0110\}$ are as follows:

Y	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Y^{0101}	1010	1011	1000	1001	1110	1111	1100	1101	0010	0011	0000	0001	0110	0111	0100	0101
Y^{0110}	1001	1000	1011	1010	1101	1100	1111	1110	0001	0000	0011	0010	0101	0100	0111	0110

To apply Q operator on 2^n meta vectors, a vector family can be generated.

Proposition 3 In $B_2^{2^n}$, generalized complement operator $Q \in B_2^{2^n}$ has 2^{2^n} different cases.

Proof Q is a 2^n bits vector, each position can be selected as 0 or 1, so a total of selections is equal to 2^{2^n} . ■

Definition 9 For 2^n meta states composed of vector Ψ , the i th vector $\Psi(i)$, $0 \leq i < n$ has 2^n bits. Four vectors: $\{\mathbf{0}, \Psi(i), \neg\Psi(i), \mathbf{1}\}$ in 2^n bits can be selected as Q operators. This special form of Q type operations is defined as QV operation.

Proposition 4 For a QV operator, $QV \in \{\mathbf{0}, \Psi(i), \neg\Psi(i), \mathbf{1}\}$, four QV vectors provide following complement results respectively in transformation:

- $\mathbf{0}$: False Operator
- $\mathbf{1}$: Truth Operator
- $\Psi(i)$: Invariant Operator
- $\neg\Psi(i)$: Variant Operator

Proof $\mathbf{1}$ operator keeps original truth table values; $\mathbf{0}$ operator reverses all values; $\Psi(i)$ operator makes invariant condition and $\neg\Psi(i)$ operator generates variant property. ■

Proposition 5 *Undertaken QV operations, 2^{n+1} cases are generated as a complement variant group.*

Proof Only $0 \leq i < n$ selected, each position have two selections associated with i plus two constant vectors. So a total of $2 \times 2^n = 2^{n+1}$ cases can be generated. ■

Definition 10 For 2^n meta vectors Y , its I th component $Y(I) \in B_2^{2^n}$, $Y(I)$ has 2^n bits. A permutation operator P makes the I th component into $P(I)$ th component for $\forall I, 0 \leq I < 2^n$, respectively.

Proposition 6 *Undertaken P operation to 2^n meta vectors in Y , a total of $2^n!$ permutations can be generated.*

Proof P operator is equal to permutation on 2^n integers. This generates a symmetric group contained $2^n!$ members. ■

Proposition 7 *Undertaken Q and P operators in Y , a total of $2^{2^n} \cdot 2^n!$ cases can be created. This creates a Complement Permutation Structure—CPS.*

Proof Q and P operators are independent of each other. Their results can be multiplied together. ■

Proposition 8 *Undertaken QV and P operators in Y , a total of $2^{n+1} \cdot 2^n!$ cases can be created. This creates a Complement Variant Structure—CVS.*

Proof QV and P operators are independent each other. Their results can be multiplied together. ■

4.3 Other Global Coding Schemes

Under $QV + P$ and $Q + P$ operations, more coding schemes can be defined.

Definition 11 The F coding scheme is defined as a subset W . For any W code, if any two meta state can be paired, such that $\forall j_1, j_1 - 2^{n-1} = j_0, 0 \leq j_0 < 2^{n-1} \leq j_1 < 2^n, I_{j_1} = \bar{I}_{j_0}$ indicate state I_{j_1} be I_{j_0} 's complement.

F coding provides restricted pair conditions to the structure. Its corresponding forms are as follows:

$$\begin{array}{ccc}
 J^1 \text{ } j\text{-th meta state} & \rightleftharpoons & J^0 \text{ } j\text{-th mate state} \\
 \Downarrow & \text{F coding base} & \Downarrow \\
 X & \rightleftharpoons & \bar{X}
 \end{array}$$

Definition 12 A coding scheme satisfies general conjugate condition if $\forall I_{j_0} \in I_{J^0}$, for the selected position $i, \forall a_i \in I_{j_0}, a_i = 0, 0 \leq i < n$.

In other words, the general conjugate condition makes selected position on lower part in 0 valued and higher part in 1-valued, respectively.

Definition 13 The C coding scheme is defined as a set of the F coding whereby $\forall I_{j_0} \in I_{J^0}$, for the selected position i , $\forall a_i \in I_{j_0}, a_i = 0, 0 \leq i < n$.

C coding provides more strong restrictions to separate all 0-valued meta states in lower part and all 1-valued meta states in higher part.

$$\begin{array}{ccc}
 J^1 \text{ } j\text{-th mate state} & \Leftrightarrow & J^0 \text{ } j\text{-th F coding} \\
 \downarrow & \text{C coding base} & \downarrow \\
 \forall x_i \in J^1, x_i = 1 & \Leftrightarrow & \forall x_i \in J^0, x_j = 0 \text{ General Conjugate}
 \end{array}$$

Some coding samples are listed in following table:

No.	7	6	5	4	3	2	1	0	Normal sequential number
SL	111	110	101	100	011	010	001	000	Ordering sequence
Truth	0	0	0	1	1	1	1	0	G: $J = 30$; W: $\langle 1 12 \rangle$
Variant	1	1	0	1	0	0	1	0	G: $J = 210$; W: $\langle 13 2 \rangle$
W	111	110	010	011	001	000	100	101	General Conjugate, without pairs
Truth	0	0	1	1	1	0	1	0	G: $J = 58$; W: $\langle 3 10 \rangle$
Variant	1	1	0	0	1	0	1	0	G: $J = 202$; W: $\langle 12 10 \rangle$
F	111	110	101	100	000	001	010	011	Meta states in pairs
Truth	0	0	0	1	0	1	1	1	G: $J = 23$; F: $\langle 1 7 \rangle$
Variant	1	1	0	1	0	1	0	0	G: $J = 212$; F: $\langle 13 4 \rangle$
C	111	110	010	011	000	001	101	100	General Conjugate + pairs
Truth	0	0	1	1	0	1	0	1	G: $J = 54$; C: $\langle 3 5 \rangle$
Variant	1	1	0	0	0	1	0	1	G: $J = 197$; C: $\langle 12 5 \rangle$

4.4 Sizes of Variant Spaces

Definition 14 Under $QV + P$ operations, W, F and C coding schemes are defined as WV, FV and CV coding schemes, respectively.

Theorem 2 For a W coding scheme of n variables, it has a total of $2^{2^n} \cdot 2^n!$ cases distinguished.

Theorem 3 For a WV coding scheme of n variables, it has a total of $2^{n+1} \cdot 2^n!$ cases distinguished.

Theorem 4 For a F coding scheme of n variables, it has a total of $2^{2^n} \cdot 2^{2^{n-1}} \cdot 2^{n-1}! = 2^{2^n(1+1/2)} \cdot 2^{n-1}!$ cases distinguished.

Theorem 5 For a FV coding scheme of n variables, it has a total of $2^{n+1} \cdot 2^{2^{n-1}} \cdot 2^{n-1}! = 2^{2^n+n+1} \cdot 2^{n-1}!$ cases distinguished.

Theorem 6 For a C coding scheme of n variables, it has a total of $2^{2^n} \cdot 2^{n-1}!$ cases distinguished.

Theorem 7 For a CV coding scheme of n variables, it has a total of $2^{n+1} \cdot 2^{n-1}!$ cases distinguished.

Using definitions of different coding schemes, shown in various sequences of one variable cases in the following table:

Function	Truth	W coding	Variant	W coding	Invariant	WV coding	False	WV coding
0	0	$\langle 0 0 \rangle$	2	$\langle 1 0 \rangle$	1	$\langle 0 1 \rangle$	3	$\langle 1 1 \rangle$
\bar{x}	1	$\langle 0 1 \rangle$	3	$\langle 1 1 \rangle$	0	$\langle 0 0 \rangle$	2	$\langle 1 0 \rangle$
x	2	$\langle 1 0 \rangle$	0	$\langle 0 0 \rangle$	3	$\langle 1 1 \rangle$	1	$\langle 0 1 \rangle$
1	3	$\langle 1 1 \rangle$	1	$\langle 0 1 \rangle$	2	$\langle 1 0 \rangle$	0	$\langle 0 0 \rangle$
0	0	$\langle 0 0 \rangle$	1	$\langle 0 1 \rangle$	2	$\langle 1 0 \rangle$	3	$\langle 1 1 \rangle$
\bar{x}	2	$\langle 1 0 \rangle$	3	$\langle 1 1 \rangle$	0	$\langle 0 0 \rangle$	1	$\langle 0 1 \rangle$
x	1	$\langle 0 1 \rangle$	0	$\langle 0 0 \rangle$	3	$\langle 1 1 \rangle$	2	$\langle 1 0 \rangle$
1	3	$\langle 1 1 \rangle$	2	$\langle 1 0 \rangle$	1	$\langle 0 1 \rangle$	0	$\langle 0 0 \rangle$

using 2D W coding to arrange 1D sequences into 2D matrices:

Original:	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Truth</td> <td style="padding: 5px;">Variant</td> </tr> <tr> <td style="padding: 5px; text-align: center;">0 \bar{x}</td> <td style="padding: 5px; text-align: center;">x 1</td> </tr> <tr> <td style="padding: 5px; text-align: center;">x 1</td> <td style="padding: 5px; text-align: center;">0 \bar{x}</td> </tr> <tr> <td style="padding: 5px; text-align: center;">\bar{x} 0</td> <td style="padding: 5px; text-align: center;">1 x</td> </tr> <tr> <td style="padding: 5px; text-align: center;">1 x</td> <td style="padding: 5px; text-align: center;">\bar{x} 0</td> </tr> <tr> <td style="padding: 5px;">Invariant</td> <td style="padding: 5px;">False</td> </tr> </table>	Truth	Variant	0 \bar{x}	x 1	x 1	0 \bar{x}	\bar{x} 0	1 x	1 x	\bar{x} 0	Invariant	False	Permutation:	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Truth</td> <td style="padding: 5px;">Variant</td> </tr> <tr> <td style="padding: 5px; text-align: center;">0 x</td> <td style="padding: 5px; text-align: center;">x 0</td> </tr> <tr> <td style="padding: 5px; text-align: center;">\bar{x} 1</td> <td style="padding: 5px; text-align: center;">1 \bar{x}</td> </tr> <tr> <td style="padding: 5px; text-align: center;">\bar{x} 1</td> <td style="padding: 5px; text-align: center;">1 \bar{x}</td> </tr> <tr> <td style="padding: 5px; text-align: center;">0 x</td> <td style="padding: 5px; text-align: center;">x 0</td> </tr> <tr> <td style="padding: 5px;">Invariant</td> <td style="padding: 5px;">False</td> </tr> </table>	Truth	Variant	0 x	x 0	\bar{x} 1	1 \bar{x}	\bar{x} 1	1 \bar{x}	0 x	x 0	Invariant	False
Truth	Variant																										
0 \bar{x}	x 1																										
x 1	0 \bar{x}																										
\bar{x} 0	1 x																										
1 x	\bar{x} 0																										
Invariant	False																										
Truth	Variant																										
0 x	x 0																										
\bar{x} 1	1 \bar{x}																										
\bar{x} 1	1 \bar{x}																										
0 x	x 0																										
Invariant	False																										

5 Invariant Properties of Variant Constructions

It is interesting to notice that under QV operations, there are $2n + 2$ vectors available to generate QVS. This makes significant differences among classical logic and Variant logic construction [50–56]. The main results of this chapter are summarized in the following theorems.

Theorem 8 (Four Invariant Points for One Variable Condition) For a W coding scheme under one variable condition, four points of the structure correspond to four functions: $\{0, x, \bar{x}, 1\}$, respectively.

Proof When $n = 1$, four vectors are available for any Q or QV operations. ■

Theorem 9 (Two Invariant Points for Truth and False Schemes) *For any $n > 1$, $W(WV)$ coding schemes, for any truth or false representation, only full 0 or full 1 valued vectors can be invariant undertaken P operations.*

Proof Undertaken P operation, if there is any not full 0 or 1 vectors, its binary number sequences will be changed. ■

Theorem 10 (Four Invariant Points for C Coding Scheme) *For any C (CV) coding scheme in variant construction, four corner positions of 2D function matrix have extreme invariant properties.*

Proof Under C(CV) coding scheme, four functions: $\{0, x, \bar{x}, 1\}$ correspond as follows: $x = \langle 0|0 \rangle$; $0 = \langle 2^{2^{n-1}} - 1|0 \rangle$; $1 = \langle 0|2^{2^{n-1}} - 1 \rangle$; $\bar{x} = \langle 2^{2^{n-1}} - 1|2^{2^{n-1}} - 1 \rangle$. Four positions are all corner points of the variant matrix. ■

6 Comparison

It is convenient to list numeric parameters to compare the different coding schemes in the following table.

Var	State	Function	ExPower	SL	W coding	WV coding	C coding	CV coding
n	2^n	2^{2^n}	$2^n!$	1	$2^{2^n} 2^n!$	$2^{n+1} 2^n!$	$2^{2^n} 2^{n-1}!$	$2^{n+1} 2^{n-1}!$
1	2	4	2	1	8	8	4	4
2	4	16	24	1	384	192	32	32
3	8	256	40320	1	10321920	645120	6144	384
4	16	2^{16}	$16!$	1	$2^{16} 16!$	$32 \cdot 16!$	$2^{16} \cdot 8!$	$32 \cdot 8!$
5	32	2^{32}	$32!$	1	$2^{32} 32!$	$64 \cdot 32!$	$2^{32} \cdot 16!$	$64 \cdot 16!$

where we use Var: variable number; State: state number; Function: function number; ExPower: exponent power products; SL: SL coding number; W coding: W coding number under $Q + P$ operations; WV coding: WV coding number under $QV + P$ operations; C coding: C coding number under $Q + P$ operations; CV coding: CV coding number under $QV + P$ operations in the table, respectively.

7 Conclusion

In this chapter, variant logic has been proposed to extend truth table representation that describes variant properties of binary sequences. This extension is required to ex-

pand traditional Boolean logic framework to a new variation space. Under two types of vector operations, the new space has $2^{2^n} 2^n!$ times more complexity than traditional Boolean function space with 2^{2^n} members. In order to manage this complexity, the framework has proposed a series of global coding schemes encoded through symmetric properties representing the elements in a matrix as a 2D map. Under this two-dimensional model, coding mechanism can be constructed and their invariant properties can be discussed.

Boolean function space represents a core invariant functional space and the newly expanded space broadens the descriptions and coding schemes used. Thus, a wide area of variation coding can be developed. In essence, the space of binary sequence functions can be thought of as a keyboard with 2^{2^n} notes. Each note contains a complete Boolean function set and its own representation. The set of notes can be represented using a coding scheme that orders the notes in a particular sequence (SL and G codes) or their 2D maps (W, F and C codes).

Under W coding representation mechanism, 2D matrix is suitable to visualize permutation sequences of n variable logic structures. Using invariant properties, classical logic and variant logic can be clearly identified. Further work on dynamic behaviours of complex dynamic systems can be explored. This chapter outlines the construction and notation of variant logic only. Future papers will show that the proposed scheme, with its foundation in symmetry, will have definite uses for predicting convergent and chaotic behaviour in dynamic binary systems such as the analysis of cellular automata rules using various visual methodologies.

References

1. G. Birkhoff, *Lattice Theory*, vol. 25, 3rd edn. (American Mathematical Society Colloquium Publications, 1984)
2. V.N. Salii, *Lattices with Unique Complements* (Amer. Math. Soc. 1988)
3. F. Maeda, S. Maeda, *Theory of Symmetric Lattices* (Springer, Berlin, 1970)
4. A.B. Rosser, A.R. Turquette, *Many-Valued Logics* (North-Holland Publishing Company, Amsterdam, 1952)
5. S. Vickers, *Topology via Logic* (Cambridge University Press, Cambridge, MA, 1989)
6. R. Sikorski, *Boolean Algebra* (Springer, Berlin, 1960)
7. F.H. Edwards, *The Principles of Switching Circuits* (MIT Press, Cambridge, MA, 1973)
8. S.C. Lee, Vector Boolean algebra and calculus. *IEEE Trans. Comput.* **C-25**(9), 865–874 (1976)
9. S.C. Lee, *Modern Switching Theory and Digital Design* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1978)
10. S. Muroga, *Logic Design and Switching Theory* (Wiley-Interscience Publication, 1979)
11. A. Thayse, *Boolean Calculus of Differences* (Springer, Berlin, 1981)
12. K.H. Kim, *Boolean Matrix Theory and Applications* (Marcel Dekker Inc., New York, 1982)
13. W. Markek, J. Onyszkiewicz, *Elements of Logic and Foundations of Mathematics in Problems* (Kluwer Academic Publishers Group, Dordrecht, 1982)
14. P.E. Dunn, *The Complexity of Boolean Networks* (Academic Press, London, 1988)
15. M. Bonnet, *Handbook of Boolean Algebras* (North Holland, New York, 1989)
16. M.S. Paterson (ed.), *Boolean Function Complexity* (Cambridge University Press, Cambridge, MA, 1992)
17. S.P. Vingron, *Switching Theory: Insight Through Predicate Logic* (Springer, Berlin, 2004)

18. R.P. Burn, *Groups: A Path to Geometry* (Cambridge University Press, Cambridge, MA, 1985)
19. M. Greutz, *Quarks, Gluons and Lattices* (Cambridge University Press, Cambridge, MA, 1983)
20. F.P. Greenleaf, *Invariant Means on Topological Groups* (Van Nostrand Reinhold Company, 1969)
21. J. Fogarty, *Invariant Theory* (W.A. Benjamin Inc., 1969)
22. T.A. Springer, *Invariant Theory* (Springer, Berlin, 1977)
23. J. Von Neumann, The general and logical theory of automata, in *Collected Works* vol. 5, ed. by J. von Neumann (1963)
24. S. Noguchi, Oizumi, A survey of cellular logic. *J. Inst. Electron. Commun. Engrg.* **54**(2), 206–220 (1971)
25. A.W. Burks, Cellular automata and natural systems, in *Proceedings of the 5th Congress of Deutsche Gessellschaft für Kybernetik* (1973)
26. T. Toffoli, Cellular Automata Mechanics. Ph.D. Thesis (University of Michigan, 1977)
27. A. Kandel, S.C. Lee, *Fuzzy Switching and Automata: Theory and Applications* (Crane Russak & Company Inc., New York, 1979)
28. K. Preston Jr., M.J.B. Duff, *Modern Cellular Automata: Theory and Application* (Plenum Press, New York, 1984)
29. S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific, Singapore, 1986)
30. C.G. Langton, Life at the edge of chaos, in *Artificial Life II* (Addison-Wesley, Reading, MA, 1992)
31. S. Wolfram, *Cellular Automata and Complexity* (Addison-Wesley, Reading, MA, 1994)
32. S. Wolfram, *A New Kind of Science* (Wolfram Media Inc., 2002). <http://www.wolframscience.com/>
33. A. Iachinski, *Cellular Automata-A Discrete Universe* (World Scientific, Singapore, 2001)
34. D. Griffieath, C. Moor, *New Constructions in Cellular Automata* (Santa Fe Institute Studies in the Sciences of Complexity, 2003)
35. H. Umeo, S. Morishita, K. Nishinari, *Cellular Automata* (Springer, Berlin, 2008)
36. Z.J. Zheng, A.J. Maeder, *The Conjugate Classification of the Kernel Form of the Hexagonal Grid, Modern Geometric Computing for Visualization* (Springer, Berlin, 1992), pp. 3–89
37. Z.J. Zheng, A.J. Maeder, *The Elementary Equation of the Conjugate Transformation for Hexagonal Grid, Modeling in Computer Graphics* (Springer, Berlin, 1993), pp. 21–42
38. Z.J. Zheng, Conjugate Transformation of Regular Plan Lattices for Binary Images. Ph.D. Thesis (Monash University, 1994)
39. Z.J. Zheng, *Conjugate Visualisation of Global Complex Behaviour, Complexity International*, vol. 3 (1996). <http://www.complexity.org.au/ci/vol03/zheng/>
40. Z.J. Zheng, C.H.C. Leung, Visualising global behaviour of 1D cellular automata image sequences in 2D maps. *Phys. A* **233**, 785–800 (1996)
41. J. Needham, *Science and Civilisation of China vol. 2 History of Scientific Thought* (Cambridge Press, Cambridge, 1954–1988)
42. D.F. Hook, *The I Ching and Mankind* (Routledge and Kegan Paul Ltd, London, 1975)
43. W.K. Chu, W.A. Sherrill, *An Anthology of I Ching* (Routledge and Kegan Paul Ltd, London, 1977)
44. I.K. Shchutshii, *Researches on the I Ching* (Princeton University Press, Princeton, 1979)
45. H. Wilhelmi, *Chang: Eight Lectures on I Ching* (Princeton Press, Princeton, 1979)
46. R. Wilhemi, *Lectures on the I Ching: Constancy and Change* (Princeton University Press, Princeton, 1979)
47. J.C. Cooper, *Yin and Yang, The Taoist Harmony of Opposites* (The Thetford Press, 1981)
48. L. A. Govinda, *The Inner Structure of I Ching: The Book of Transformation* (Wheelwright Press, 1981)
49. G. Whincup, *Rediscovering of I Ching* (GreyWhincup, 1986)
50. E.S. Fedorov, *Symmetry of Crystals* (1890) (American Crystallographic Association, 1971)
51. H. Weyl, *Symmetry* (Princeton, 1952)
52. P. Benacerraf, H. Putnam, *Philosophy of Mathematics* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1964)

53. C.H. McGillavry, *Symmetry Aspects of M.C. Escher's Periodic Drawings* (A. Oosthoek's Uitgeversmaatschappij N.V., Utrecht, 1965)
54. P.B. Yale, *Geometry and Symmetry* (Holden Day, 1968)
55. A. Holden, *Shapes, Spaces and Symmetry* (Columbia Uni. Press, New York, NY, 1971)
56. F.H. Bool, *Escher: with A Complete Catalogue of Graphic Works* (Thames Hudson, 1982)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part II

Theoretical Foundation—Variant Measurement

All of mathematics is a tale about groups.

—*Henri Poincaré*

In geometric and physical applications, it always turns out that a quantity is characterized not only by its tensor order, but also by symmetry.

—*Hermann Weyl*

Nothing exists until it is measured.

—*Niels Bohr*

A list of research papers were published on variant measurements during 2011–2012. Two OA book chapters that are important to express core results of variant measurements (Chapter “From Local Interactive Measurements to Global Matrix Representations on Variant Construction, From Conditional Probability Measurements to Global Matrix Representations on Variant Construction”) are published in *Advanced Topics in Measurements:339–400* (2012) by InTech Press.

Part II is composed of three chapters (3–5).

Chapter “[Elementary Equations of Variant Measurement](#)” provides the elementary equation of variant measurement to discuss four meta measures under permutative and associative properties. Two sets of sample partitions are expressed as sum of product of binomial coefficients in the elementary equation. This is a systematic approach to handle configuration space under four meta measures.

Chapter “[Triangular Numbers and Their Inherent Properties](#)” uses triangular numbers to express inherent properties of 1D binary sequences under three parameters as an elementary equation. A set of interesting properties were explored.

This scheme provides efficient partitions to handle rotational invariant properties on binary sequences.

Chapter “[Symmetric Clusters in Hierarchy with Cryptographic Properties](#)” describes symmetric clusters in hierarchy under multiple symmetric operations: combination, crossing, variant, and rotation conditions. Rich clusters were observed under various conditions.

Elementary Equations of Variant Measurement



Jeffrey Zheng

Abstract Four variant measures are used to represent combinatorial functions including binomial coefficients. These variant measures are based on two types of m -bit vectors. Type A corresponds to non-periodic boundary conditions, while Type B corresponds to periodic boundary conditions. For each type, groups containing the four variant measures are formed, which are invariant against permutative and associative operations. By mapping two group elements of Type B on coefficients of binomial decompositions, patterns similar to Pascal's triangle are observed.

Keywords Variant measurement · m variable vector · Multinomial coefficient
Permutative and associative operations · Global invariant

1 Introduction

For any n 0–1 variables, variant logic provides a $2^n! \times 2^{2^n}$ -dimensional configuration space [16, 17] to support measurement and analysis [14, 15], which is a real difficulty for any practical activities [1, 9–11]. From a measuring analysis viewpoint [6–8, 13], it is essential to manipulate static states and their measuring clustering as effective measures to be a core content of any 0–1 measuring framework. In this chapter, starting from m variables of a 0–1 vector, binomial expressions are applied to support the four meta measures of variant partitions and associated multinomial expressions.

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_3

Using permutative and associative operations, various variation and invariant properties are investigated. From a global invariant viewpoint, various combinatorial clustering properties are systematically explored.

2 Elementary Equation

Let x be an m -bit vector, $x = x_0x_1 \dots x_i \dots x_{m-1}$, $x_i \in \{0, 1\}$, $0 \leq i < m$, $x \in B_2^m$. Each x is an m bit state. From a variation viewpoint, there are two types $\{A, B\}$ distinguished. Let $\{m_\perp, m_+, m_-, m_\top\}$ be four measuring operators.

2.1 Type A Measures

For a pair of $(i, i + 1)$ elements, (x_i, x_{i+1}) , $0 \leq i < (m - 1)$ form partitions. (Non-periodic boundary conditions)

Four measures can be calculated from the following equations.

$$m_\perp(x) = \sum_{i=0}^{m-2} [(x_i, x_{i+1}) == (0, 0)] \quad (1)$$

$$m_+(x) = \sum_{i=0}^{m-2} [(x_i, x_{i+1}) == (0, 1)] \quad (2)$$

$$m_-(x) = \sum_{i=0}^{m-2} [(x_i, x_{i+1}) == (1, 0)] \quad (3)$$

$$m_\top(x) = \sum_{i=0}^{m-2} [(x_i, x_{i+1}) == (1, 1)] \quad (4)$$

$$m = m_\perp(x) + m_+(x) + m_-(x) + m_\top(x) + 1 \quad (5)$$

From a clustering viewpoint, the last bit of x , x_{m-1} can be used to distinguish relevant combinatorial numbers. While $x_{m-1} == 1$, there are $\binom{m-1}{m_+ + m_\top + 1}$ and for $x_{m-1} == 0$, there are $\binom{m-1}{m_+ + m_\top}$, possible x vectors, where $m_+ + m_\top$ is the number of 1 elements in a vector. By adding both binomial coefficients, Pascal's rule [4] is obtained.

$$\binom{m}{p} = \binom{m-1}{p} + \binom{m-1}{p-1}, \quad (6)$$

$$p(x) = m_+(x) + m_\top(x) + 1, 0 \leq p \leq m, x \in B_2^m$$

2.2 Type B Measures

A pair of $(i, i + 1)$ elements is linked as a ring, $(x_i, x_{i+1(mod\ m)})$, $0 \leq i < m$ (Periodic boundary conditions).

$$m_{\perp}(x) = \sum_{i=0}^{m-1} [(x_i, x_{i+1(mod\ m)}) == (0, 0)] \quad (7)$$

$$m_{+}(x) = \sum_{i=0}^{m-1} [(x_i, x_{i+1(mod\ m)}) == (0, 1)] \quad (8)$$

$$m_{-}(x) = \sum_{i=0}^{m-1} [(x_i, x_{i+1(mod\ m)}) == (1, 0)] \quad (9)$$

$$m_{\top}(x) = \sum_{i=0}^{m-1} [(x_i, x_{i+1(mod\ m)}) == (1, 1)] \quad (10)$$

$$m = m_{\perp}(x) + m_{+}(x) + m_{-}(x) + m_{\top}(x) \quad (11)$$

Let p be the number of 1 elements, $p(x) = m_{+}(x) + m_{\top}(x)$, then the number of possible x vectors is

$$\binom{m}{p}, \quad 0 \leq p \leq m. \quad (12)$$

3 Partition

Either Type A or B, internal parameters are associated with the four meta measures. For a brief analysis, Type B will be selected as initial part, multinomial coefficients are applied to partition relevant binomial coefficients. Using m variable, p number and q branches, the following equations are formulated. Under the partition condition, vector x can be ignored.

$$m = m_{\perp} + m_{+} + m_{-} + m_{\top} \quad (13)$$

$$p = m_{+} + m_{\top} \quad (14)$$

$$m - p - q = m_{\perp} \quad (15)$$

$$q = m_{+} = m_{-} \quad (16)$$

$$p - q = m_{\top} \quad (17)$$

Based on equivalent quantitative numbers, there are one-to-one corresponding on the four meta measures and relevant quantitative measures:

$$\{m_{\perp}, m_{+}, m_{-}, m_{\top}\} \leftrightarrow \{m - p - q, q, q, p - q\}$$

from a global restriction to establish an equivalent expressional framework.

From an expressional viewpoint, different partitions are investigated from a single binomial coefficient to a set of multinomial coefficients with equivalent properties among different expressions. Their partitions undertaken on various levels are illustrated in the following sections. From a binomial coefficient, there are multiple levels of representations involved, the first level and the n th level can be connected as

$$\binom{m_{\perp} + m_{+} + m_{-} + m_{\top}}{p} \rightarrow \sum_{k=0}^p \prod_{l=1}^n \binom{f_l(m_{\perp}, m_{+}, m_{-}, m_{\top})}{g_l(p, k)} \quad (18)$$

$$0 \leq p \leq m \quad 0 \leq k \leq m.$$

The core content of this chapter is to establish a global invariant framework using n levels of representations by deriving the functions f_l and g_l .

4 Variation Space

Let $\{a, b, c, d\}$ be a set of four distinct measures. Two operations, permutative and associative, can be determined. For an ordered tuple with four measures (a, b, c, d) , Permutative operator $\pi: (a, b, c, d) \rightarrow (\pi(a), \pi(b), \pi(c), \pi(d))$ to map one measure to another measure.

Associative operator $\alpha: \{a, b, c, d\} \rightarrow \alpha\{a, b, c, d\}$ to group one to multiple measures keeping the initial ordering.

e.g. $(a, b, c, d) \rightarrow (b, d, a, c)$ is a permutative operation and $\{a, b, c, d\} \rightarrow \{a, b\}\{c\}\{d\}$ is an associative operation.

A permutative operation changes the order of four tuple variables and an associative operation changes sequential relationship on its neighbourhood elements. In a normal arithmetical condition, two operations have conservative under add operations with global invariant properties. From an algebraic viewpoint, two operations are independent.

Lemma 1 *For an ordering structure with four measures under two operations: permutative and associative, there are 192 configurations identified.*

Proof For a vector with 4 members, there are a total of 24 distinct permutations $4! = 24$. For an ordered set of 4 elements, 8 associated patterns are identified as follows: $\{\{a, b, c, d\}; \{a\}\{b, c, d\}; \{a, b\}\{c, d\}; \{a, b, c\}\{d\}; \{a\}\{b\}\{c, d\}; \{a\}\{b, c\}\{d\}; \{a, b\}\{c\}\{d\}; \{a\}\{b\}\{c\}\{d\}\}$. Two operations are independent, so the whole system contains $24 \times 8 = 192$ configurations.

5 Invariant Combination

Using both permutative and associative operations, various combinatorial invariants can be identified.

5.1 Type A Invariants

Five invariant groups can be distinguished.

Item	Set	Cluster
0	{ }	1
1	{a,b,c,d}	1
2a	{a}{b,c,d}; {b}{a,c,d}; {c}{a,b,d}; {d}{a,b,c}	4
2b	{a,b}{c,d}; {a,c}{b,d}; {a,d}{b,c}	3
3	{a,b}{c}{d}; {a,c}{b}{d}; {a,d}{b}{c}; {b,c}{a}{d}; {b,d}{a}{c}; {c,d}{b}{a}	6
4	{a}{b}{c}{d}	1

Proposition 1 For a measuring structure with four members, Type A has 16 combinatorial invariants distinguished (0 item: 1 cluster; 1 item: 1 cluster; 2a item: 4 clusters; 2b item: 3 clusters; 3 item: 6 clusters; 4 item: 1 cluster).

Proof Checking Type A conditions listed, all combinatorial conditions are exhaustive included.

5.2 Type B Invariants

For Type B, let $b = c$, following simplification can be performed.

Item	Set	Cluster
0	{ }	1
1	{a,b,b,d}	1
2a	{a}{b,b,d}; {b}{a,b,d}; {b}{a,b,d}; {d}{a,b,b}	
→	{a}{b,b,d}; {b}{a,b,d}; {d}{a,b,b}	3
2b	{a,b}{b,d}; {a,b}{b,d}; {a,d}{b,b}	
→	{a,b}{b,d}; {a,d}{b,b}	2
3	{a,b}{b}{d}; {a,b}{b}{d}; {a,d}{b}{b}; {b,b}{a}{d}; {b,d}{a}{b}; {b,d}{b}{a}	
→	{a,b}{b}{d}; {a,d}{b}{b}; {b,b}{a}{d}; {b,d}{a}{b}	4
4	{a}{b}{b}{d}	1

Proposition 2 For a measuring structure with four members, Type B has 12 combinatorial invariants distinguished (0 item: 1 cluster; 1 item: 1 cluster; 2a item: 3 clusters; 2b item: 2 clusters; 3 item: 4 clusters; 4 item: 1 cluster).

Proof Checking Type B conditions listed, all combinatorial conditions are exhaustive included.

6 Combinatorial Expressions of Type B Invariants

Applying $m_{\perp} = m - p - q$, $m_{+} = m_{-}$, $m_{\top} = p - q$ to replace $\{a, b, c, d\}$, there are 11 effective formula:

Item	Set of measures	Cluster
1	$\{m\}$	1
2a	$\{m - p - q\}\{p + q\}; \{q\}\{m - q\}; \{p - q\}\{m - p + q\}$	3
2b	$\{m - p\}\{p\}; \{m - 2q\}\{2q\}$	2
3	$\{m - p\}\{q\}\{p - q\}; \{m - 2q\}\{q\}\{q\}; \{2q\}\{m - p - q\}\{p - q\}; \{p\}\{m - p - q\}\{q\}$	4
4	$\{m - p - q\}\{q\}\{q\}\{p - q\}$	1

Corollary 1 Type B invariants include 11 nontrivial expressions.

Proof Only 0 item is a trivial one.

7 Two Combinatorial Formula and Quantitative Distributions

From a combinatorial viewpoint, 1. item formula is a binomial coefficient $\binom{m}{p}$, $0 \leq p \leq m$, to show various partition properties with relevant parameters. For convenient illustration, two expressions are selected: $\{m - p\}\{p\}$ and $\{2q\}\{m - 2q\}$ from 2 clusters of 2b item of Type B.

7.1 Case I. $\{m - p\}\{p\}$

In combinatorics, the following identity for binomial coefficients:

$$\binom{m+n}{r} = \sum_{k=0}^r \binom{m}{k} \binom{n}{r-k}$$

is Vandermonde's identity (or Vandermonde's convolution), for any nonnegative integers r, m, n . The identity is named after Alexandre-Théophile Vandermonde (1772), although it was already known in 1303 by the Chinese mathematician Zhu Shijie (Chu Shi-Chieh) [2, 3, 5, 12].

Applying Chu-Vandermonde's identity to identify $\{m - p\}\{p\}$ as f_1 and f_2 in Eq. (18), the binomial coefficient in level $n = 2$ can be written as

$$\begin{aligned} \binom{m}{p} &= \sum_{k=0}^p \binom{m-p}{k} \binom{p}{p-k} \\ &= \sum_{k=0}^p \binom{m-p}{k} \binom{p}{k}, 0 \leq p \leq m. \end{aligned} \quad (19)$$

In this way, each binomial coefficient $\binom{m}{p}$ is composed of $p + 1$ pairs of binomial coefficient multiplications and a total of sums on relevant groups.

Theorem 1 For all coefficients of Type B, sum of all coefficients in $\{m - p\}\{p\}$, $0 \leq p \leq m$ is equal to 2^m .

Proof Since

$$\forall m > 0, \sum_{p=0}^m \binom{m}{p} = 2^m, \sum_{k=0}^p \binom{m-p}{k} \binom{p}{k} = \binom{m}{p},$$

so

$$\sum_{p=0}^m \sum_{k=0}^p \binom{m-p}{k} \binom{p}{k} = 2^m.$$

According to Theorem 1, all parameters of $\{\binom{m-p}{k} \binom{p}{k}\}$ are distributed in $(m + 1)^2$ 2D array.

For e.g., while $m = 10$, all coefficients are in 11×11 region and nontrivial values are composed of a triangle shape with reflect symmetric properties on p values.

$$m > 0, 0 \leq k, p \leq m, \{f(m, p, k) = \binom{m-p}{k} \binom{p}{k}\} :$$

$f(10, p, k)$	0	1	2	3	4	5	6	7	8	9	10	p
10												
9												
8												
7												
6												
5						1						
4					15	25	15					
3				35	80	100	80	35				
2			28	63	90	100	90	63	28			
1		9	16	21	24	25	24	21	16	9		
0	1	1	1	1	1	1	1	1	1	1	1	
k												

7.2 Case II. $\{2q\}\{m - 2q\}$

Applying Chu-Vandermonde’s identity to identify $\{2q\}\{m - 2q\}$ as f_1 and f_2 in Eq. (18), the binomial coefficient in level $n = 2$ can be written as

$$\binom{m}{p} = \sum_{k=0}^p \binom{2q}{k} \binom{m - 2q}{p - k} \tag{20}$$

$$0 \leq p \leq m, 0 \leq q \leq \lfloor m/2 \rfloor$$

By using this formula, it is possible to select a special q value in $\{\binom{2q}{k} \binom{m-2q}{p-k}\}$ to form $\lfloor m/2 \rfloor + 1$ 2D coefficient distributions.

Theorem 2 For Type B $\{2q\}\{m - 2q\}$, $0 \leq p \leq m, 0 \leq q \leq \lfloor m/2 \rfloor$ equation, selecting a proper value of q , all coefficients are distributed in $\lfloor m/2 \rfloor + 1$ 2D arrays and the sum of total coefficients in a 2D array is equal to 2^m .

Proof Since

$$\forall m > 0, 0 \leq q \leq \lfloor m/2 \rfloor, \binom{m}{p} = \sum_{k=0}^p \binom{2q}{k} \binom{m - 2q}{p - k} \& \sum_{i=0}^m \binom{m}{p} = 2^m,$$

so

$$\sum_{p=0}^m \sum_{k=0}^p \binom{2q}{k} \binom{m - 2q}{p - k} = 2^m$$

According to Theorem 2, $\{\binom{2q}{k} \binom{m-2q}{p-k}\}$ coefficients are distributed in $\lfloor m/2 \rfloor + 1$ levels of $(m + 1) \times (m + 1)$ 2D planes.

$m = 10, q = 5:$

$f(10, 5, p, k)$	0	1	2	3	4	5	6	7	8	9	10	p
10												
9												
8												
7												
6												
5												
4												
3												
2												
1												
0	1	10	45	120	210	252	210	120	45	10	1	
k												

7.3 Result Analysis

Two formulas selected from 2b item of Type B show completely different properties. In Case I, for a given m , all coefficients are distributed in one triangle area with reflection properties on p direction.

However, Case II provides multiple levels of 2D distributions and each one is corresponding to a selected q value. From three listed conditions, $q = 0$ and $q = 5$ are linear structures, the first one is located on diagonal positions of the plane and the second one is located on $k = 0, p = \{0, 1, \dots, 10\}$ a horizontal region. While $0 < q < 5$, all distributions are shown in as parallelograms. Each line is shown in special symmetries. We can observe associated with variations of q values, horizontal projection keeps the same, however, the vertical projection will be changed from $q = 0$ binomial distribution, to be a pulse on $q = \lfloor m/2 \rfloor$ condition. This type of controllable properties could be useful to explore future advanced applications.

8 Conclusion

A new approach to decompose binomial coefficients under permutative and associative operations is proposed. Using this approach, it is feasible to investigate four meta measures in global invariant spaces. The resulting set of 192 configurations is categorized into standard group theory mechanism. From a statistic viewpoint, Type A (Five levels in 16 clusters) and Type B (Five levels in 12 clusters) provide global identifications on complicated partitions on wider restrictions, further theoretical explorations and practical applications are deeply expected in the coming period.

Acknowledgements The author would like to thank Chris Zheng for refined clustering analysis on random sequences to open a new way in binomial expressions, Yifeng Zheng and Kaiyu Yang for generating binomial coefficients in different conditions and Dr. Dennis Heim for correction of the chapter.

References

1. J.R. Chen, *Combinatorial Mathematics* (Harbin Institute of Technology Press, Harbin, 2012). (in Chinese)
2. H.W. Gould, Some generalizations of Vandermonde's convolution. *Am. Math. Mon.* **63**(2), 84–91 (1956)
3. H.W. Gould, *Combinatorial Identities* (Morgantown Printing and Binding Company, Morgantown, 1972)
4. M. Hall, *Combinatorial Theory*, 2nd edn. (Blaisdell, New York, 1986)
5. L.K. Hua, *Loo-Keng Hua Selected Papers* (Springer, New York, 1982)
6. D.E. Knuth, *The Art of Computer Programming*, vol. 1, 3rd edn. (Addison-Wesley, Reading, Massachusetts, 1998)
7. D.E. Knuth, *The Art of Computer Programming, Volume 4: Combinatorial Algorithms, Part 1*, (Addison-Wesley, Boston, 2011)
8. F. Morgan, *Geometric Measure Theory*, 4th edn. (Elsevier, Amsterdam, 2009)
9. R.P. Stanley, *Enumerative Combinatorics*, vol. 1, 2nd edn. (Cambridge University Press, Boston, 1997)
10. D. Stanton, R. Stanton, D. White, *Constructive Combinatorics* (Springer, New York, 1986)
11. G.Z. Tu, *Combinatorial Enumeration Methods and Applications* (Science Press, 1981) (in Chinese)
12. Vandermonde's identity. https://en.wikipedia.org/wiki/Vandermonde's_identity
13. L.Z. Xu, M.S. Jiang, Z.Q. Zhu, *Combinatorial Mathematics of Computation* (Shanghai Science and Technology Press, 1983) (in Chinese)
14. Z.J. Zheng, A. Maeder, The conjugate classification of the kernel form of the hexagonal grid, in *Modern Geometric Computing for Visualization* (Springer, 1992), pp. 73–89
15. Z.J. Zheng, *Conjugate transformation of regular plan lattices for binary images*, Ph.D. thesis, Monash University, 1994
16. J.Z.J. Zheng, C.H.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* **5**(2), 163–172 (2010). Higher Educational Press and Springer
17. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, A framework of variant logic construction for cellular automata, in *Cellular Automata—Innovative Modeling for Science and Engineering*, ed. by A. Salcido (InTech Press, 2011)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Triangular Numbers and Their Inherent Properties



Chris Zheng and Jeffrey Zheng

Abstract A method to classify one-dimensional binary sequences using three parameters intrinsic to the sequence itself is introduced. The classification scheme creates combinatorial patterns that can be arranged in a two-dimensional triangular structure. Projections of this structure contain interesting properties related to the Pascal triangle numbers. The arrangement of numbers within the triangular structure has been named “triangular numbers”, and the essential parameters, elementary equation, and sequencing schemes are discussed as well as visualizations of sample distributions, special cases, and search results. We believe this to be a novel finding as sequences generated using this method are not contained in the On-Line Encyclopedia of Integer Sequences or OEIS.

Keywords Binary sequence · Classification · Combinatorial patterns · Triangular number · Elementary equation · Variant triangle

1 Introduction

Additive number theory [7], the study of integer subsets and their behavior under addition, is a branch of mathematics related to combinatorics. The simplest constructs within this field are binomial coefficients [6]. The properties of binomial

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

C. Zheng
Tahto, Sydney, Australia
e-mail: z@caudate.me

J. Zheng (✉)
Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

coefficients have been explored by many over the history of mathematics [8, 9]. One generalization of the binomial coefficient is the multinomial coefficient [5, 8, 9]. Any multinomial coefficient can be expressed as the products of multiple binomial coefficients:

$$\binom{k_1 + k_2 + \dots + k_m}{k_1, k_2, \dots, k_m} = \binom{k_1 + k_2}{k_1} \binom{k_1 + k_2 + k_3}{k_1 + k_2} \dots \binom{k_1 + k_2 + \dots + k_m}{k_1 + k_2 + \dots + k_{m-1}}. \quad (1)$$

For this type of expansions, the simplest is the trinomial coefficient [10–13]:

$$\binom{r}{k, m-k, r-m} = \binom{r}{m} \binom{m}{k}. \quad (2)$$

1.1 Geometric Arrangement of Combinatorial Data

In discrete geometry [2], as the most basic 2D shape, triangular patterns are found in such series as combinatorial triangle [A102639](#), differential triangle [A194005](#) [1], additive triangle [A035312](#), and Pascal triangle [A007318](#) [8, 9, 11, 13].

This chapter proposes a novel method of classification of binary sequences that is shown to be combinatorial properties in nature. By using a simple basis of binary (0–1) sequences and applying simple classification rules, a triangular structure can be generated. The set of results has been named “Generative Triangular Numbers”. The term generative [3] is used to describe the technique of using a simple input and a repeatedly applied process, creating emergent properties through repetition. Generative science [4] is a multidisciplinary science that explores the natural world and its complex behaviors as a generative process. Generative approaches can be used to simulate describe behaviors in fractals, cellular automata, and various nonlinear systems.

The generated patterns are not currently found in the On-Line Encyclopedia of Integer Sequences (OEIS) potentially making them an interesting area for further research.

1.2 Previous Work

The current scheme is a derivative of the work of Zheng et al. [16, 17] to organize 1D 0–1 sequences as certain $N > 1$ length vectors using three parameters in variant measurement construction and classifications on hierarchical discrete phase spaces in general.

A trinomial equation is proposed as an elementary equation using three control parameters $\{q, p, N\}$ [14, 15] to describe 0–1 vectors of N length as a subgroup, where N is the length of a vector, p indicates the number of elements with 1 values,

and q records the number of changes from either 0–1 or 1–0 as the vector in a circular form to form a 2D array with nontrivial triangular numbers. This type of elementary equation can be generatively applied to make relevant triangular numbers as a geometric distribution to form a hierarchical 3D array generatively. Based on this hierarchical 3D array, different integer sequences can be observed from this type of generative triangular numbers, and one projection on p direction is collected by Vandermonde’s identities to show their correspondences to standard binomial coefficients. Main results are provided by algorithms, theorems, and corollaries. Sample cases are illustrated and possible meanings are discussed.

2 Definitions and Sample Cases

2.1 Definitions

Definition 1 Let X be a 0–1 vector, $X = x_{N-1} \dots x_i \dots x_0$ with N elements as a state, $x_i \in \{0, 1\}$, $0 \leq i < N$.

Definition 2 Let $\Omega(N)$ denote a vector space contained all 0–1 vectors of N length $\Omega(N) = \{\forall X | 0 \leq X < 2^N\}$ as an initial data set.

Definition 3 Let $\binom{n}{k}$ be a binomial coefficient, it satisfies

$$\binom{n}{k} = \begin{cases} 1, & \text{if } n = k; \\ 0, & \text{if } n \neq k, k > n \text{ or } k < 0; \\ \frac{n!}{k! \times (n-k)!}, & \text{otherwise.} \end{cases} \quad (3)$$

Under this condition, $|\Omega(N)| = 2^N$ forms a vector space with N length, respectively.

Definition 4 For any selected vector $X \in \Omega(N)$, $p(X)$ can be determined by

$$p(X) = \sum_{i=0}^{N-1} x_i, x_i \in \{0, 1\}. \quad (4)$$

Lemma 1 For a vector space $\Omega(N)$, p provides a complete partition on a subgroup and the number of vectors in the subgroup is a binomial coefficient.

Proof For a given p , $0 \leq p \leq N$, its combinatorial property makes a total number of $\binom{N}{p} = \frac{N!}{p! \times (N-p)!}$ vectors identified to partition the vector space $\Omega(N)$.

Definition 5 For a circular vector $X \in \Omega(N)$, $q(X)$ can be determined by

$$q(X) = \sum_{0 \leq i < N} (x_i \equiv 0) \& (x_{i+1} \equiv 1); x_i, x_{i+1} \in 0, 1, (i + 1) \bmod(N), \quad (5)$$

e.g., $N = 10, X = 1110011001, p(X) = 6(i = \{0, 3, 4, 7, 8, 9\}); q(X) = 2(i = \{2, 6\})$.

2.2 Sample Cases

Under this construction, any selected vector can be evaluated by the three parameters. Applying this set of parameters to create subgroups, interesting inner structures can be identified. That is, $N = 4$, all 16 vectors in the vector space, can be distinguished as six subgroups as a pair of (q, p) values shown in Table 1.

Each subgroup is linked to their corresponding vectors in Table 2

Enumeration numbers of relevant subgroup numbers are shown in Table 3.

Table 1 Six subgroups for $N = 4$ vector space in (q, p) partitions

$q \backslash p$	0	1	2	3	4
0	(0, 0)				(0, 4)
1		(1, 1)	(1, 2)	(1, 3)	
2			(2, 2)		

Table 2 Six subgroups, vectors, and enumerating numbers

(q, p)	$\{X\}, N = 4$	No.
(0, 0)	{0000}	1
(0, 4)	{1111}	1
(1, 1)	{0001, 0010, 0100, 1000}	4
(1, 2)	{0011, 0110, 1100, 1001}	4
(1, 3)	{0111, 1110, 1101, 1011}	4
(2, 2)	{0101, 1010}	2

Table 3 $N = 4, (q, p)$ subgroup numbers and a projection

$q \backslash p$	0	1	2	3	4
0	1				1
1		4	4	4	
2			2		
$\sum_{\forall q}$	1	4	6	4	1

Table 4 Six levels of binomial coefficients and generative triangular numbers

N	$\{p, N\}$ Binomial Numbers	$\{q, p, N\}$ Generative Triangular Numbers
1	1 1	1 1
2	1 2 1	1 1 2
3	1 3 3 1	1 1 3 3
4	1 4 6 4 1	1 1 4 4 4 2
5	1 5 10 10 5 1	1 1 5 5 5 5 5 5
6	1 6 15 20 15 6 1	1 1 6 6 6 6 6 9 12 9 2

From Table 3, it is easy to verify that 16 vectors are sum of all possible numbers from six subgroups. Subgroup sequences of all numbers are as the same as $N = 4$ binomial coefficients. Applying this corresponding from $N = 1-6$, six rows of original binomial coefficients can be created generatively as three-dimensional organization and each row $\{p, N\}$ sequence corresponds a (q, p) triangular shape, respectively, shown in Table 4.

This type of relationship can be expanded on generative mechanism from special cases of $N = 1-6$ to general conditions for any given N value. The detailed generative triangular mechanism is described in the next section.

3 Elementary Equations

Definition 6 Let $f(q, p, N)$ denote a function for generative triangular numbers $0 \leq p \leq N, 0 \leq q \leq \lfloor N/2 \rfloor$, for two initial and end subgroups $p = \{0, N\}, q = 0$, let two functions of subgroups be $f(0, 0, N) = f(0, N, N) = 1$.

For other subgroups, each case $0 < p < N, 0 < q \leq \lfloor N/2 \rfloor$ is a subgroup under a given condition. Elementary equation of generative triangular numbers is proposed to use binomial coefficient expression in Eq. 6.

$$f(q, p, N) = \frac{N}{N-p} \binom{N-p}{q} \binom{p-1}{q-1}. \tag{6}$$

Table 5 $N = 5, f(q, p, 5)$ subgroup numbers

$q \backslash p$	0	1	2	3	4	5
0		1				1
1			5	5	5	5
2				5	5	

Using this elementary equation, the list of values can be verified. For example, $f(1, 1, 5) = \frac{5}{4} \binom{4}{1} \binom{0}{0} = 5$; $f(2, 3, 5) = \frac{5}{2} \binom{2}{2} \binom{2}{1} = 5$; ... $f(2, 4, 5) = \frac{5}{1} \binom{1}{2} \binom{3}{1} = 0$. All $\{f(q, p, 5)\}$ calculations are listed in Table 5.

Corollary 1 *The elementary equation has equivalent identities on a pair of $\{p, N - p\}$.*

$$\begin{aligned}
 f(q, p, N) &= \frac{N}{N-p} \binom{N-p}{q} \binom{p-1}{q-1} \\
 &= \frac{N}{N-(N-p)} \binom{N-(N-p)}{q} \binom{(N-p)-1}{q-1} \tag{7} \\
 &= f(q, N-p, N).
 \end{aligned}$$

Proof Using the elementary equation, we have

$$\begin{aligned}
 f(q, p, N) &= \frac{N}{N-p} \binom{N-p}{q} \binom{p-1}{q-1}: \text{(equation 6)} \\
 &= \frac{N}{(N-p)} \frac{(N-p)!}{(N-p-q)!q!} \binom{p-1}{q-1} \\
 &= \frac{N}{q} \frac{(N-p-1)!}{(N-p-q)!(q-1)!} \binom{p-1}{q-1} \\
 &= \frac{N}{q} \binom{N-p-1}{q-1} \binom{p-1}{q-1} \\
 &= \frac{N}{q} \binom{p-1}{q-1} \binom{N-p-1}{q-1} \\
 &= \frac{N}{p} \binom{p}{q} \binom{N-p-1}{q-1} \\
 &= \frac{N}{N-(N-p)} \binom{N-(N-p)}{q} \binom{(N-p)-1}{q-1}: \text{(equation 7)} \\
 &= f(q, N-p, N).
 \end{aligned}$$

p parameters are in the vertical direction. In general condition for any given N , triangular numbers can be arranged in Table 6 (Fig. 1).

Table 6 $N = 5, f(q, p, 5)$ subgroup numbers in vertical direction

$p \backslash q$	0	1	2
0	1		
1		5	
2		5	5
3		5	5
4		5	
5	1		

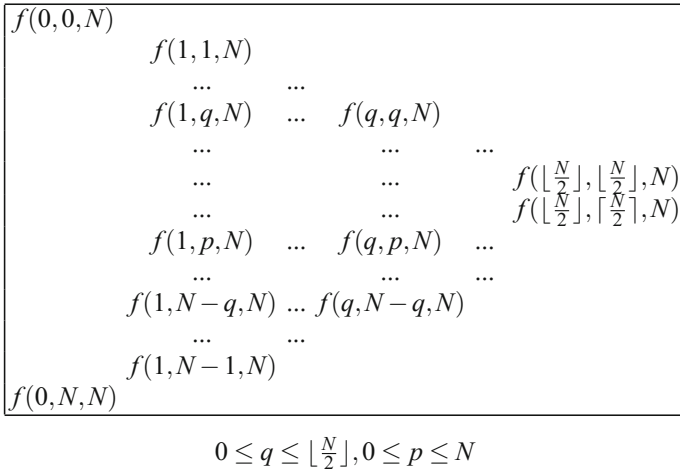


Fig. 1 Triangular numbers for a given $N > 1$

4 Local Propensities

It is necessary to investigate different relationships for symmetry properties from the elementary equations to distinguish functions for generative triangular numbers.

4.1 Nontrivial Areas

Corollary 2 (A pair of symmetric properties) *In either $0 < q \leq p \leq N - q$ or $q = 0, p = \{0, N\}$, a pair of nontrivial trinomial coefficients on triangular numbers satisfies*

$$f(q, p, N) = f(q, N - p, N). \tag{8}$$

Proof Using the elementary equation, two cases are required.

Case 1: If $q > 0$, Eqs. 6 and 7 provide relevant combinatorial identities. Case 2: If $q = 0$, we have $f(0, 0, N) = f(0, N, N) = 1$ by Definition 6.

4.2 Trivial Areas

Corollary 3 (Five areas for trivial values) *If case 1— $q > 0, 0 < p < q$; case 2— $N - q < p < N$; case 3— $q = 0, 0 < p < N$; case 4— $q > 0, p = 0$; case 5— $q > 0, p = N$, then*

$$f(q, p, N) = 0. \quad (9)$$

Proof For cases 1, 2 and 3, we have

$$\begin{aligned} f(q, p, N) &= \frac{N}{N-p} \binom{N-p}{q} \binom{p-1}{q-1} \\ &= \frac{N}{N-p} \binom{N-p}{q} \left[\binom{p-1}{q-1} = 0 \right], 0 < p < q : \text{Case 1} \\ &= \frac{N}{N-p} \left[\binom{N-p}{q} = 0 \right] \binom{p-1}{q-1}, N - q < p < N : \text{Case 2} \\ &= \frac{N}{N-p} \binom{N-p}{0} \left[\binom{p-1}{-1} = 0 \right], q = 0, 0 < p < N : \text{Case 3} \\ &= 0. \end{aligned}$$

For cases 4 and 5, we have

$$\begin{aligned} f(q, p, N) &= \frac{N}{q} \binom{N-p-1}{q-1} \binom{p-1}{q-1} \\ &= \frac{N}{q} \binom{N-1}{q-1} \left[\binom{-1}{q-1} = 0 \right], q > 0, p = 0 : \text{Case 4} \\ &= \frac{N}{q} \left[\binom{-1}{q-1} = 0 \right] \binom{N-1}{q-1}, q > 0, p = N : \text{Case 5} \\ &= 0. \end{aligned}$$

5 Projection Properties

5.1 Linear Projection

In this section, the algebraic properties of linear projection are investigated.

Definition 7 Let $L(p, N)$ denote a function as a linear projection to collect all possible values for a given $p, 0 \leq p \leq N$.

Table 7 $N = 5, f(q, p, 5)$ subgroup numbers and two projections

$p \backslash q$	0 1 2	$L(p, 5) = \sum_{\forall q} f(q, p, 5)$
0	1	1
1	5	5
2	5 5	10
3	5 5	10
4	5	5
5	1	1
		$ \Omega(5) = \sum_{\forall q} \sum_{\forall p} f(q, p, 5) = 32$

For the case of $N = 5$, two projections and their generative triangular numbers are shown in Table 7, respectively.

Following theorems and corollaries are claimed.

Theorem 4 *If $L(p, N) = \sum_{q=1}^p f(q, p, N), 0 < p < N$, then the projection function $L(p, N)$ is a binomial coefficient and*

$$L(p, N) = \binom{N}{p}. \tag{10}$$

Proof For a fixed $p, 0 < p < N$, all possible $\{f(q, p, N)\}$ are collected to form the following equation:

$$\begin{aligned} L(p, N) &= \sum_{q=1}^p f(q, p, N) \\ &= \sum_{q=1}^p \frac{N}{N-p} \binom{N-p}{q} \binom{p-1}{q-1} \\ &= \frac{N}{N-p} \sum_{q=1}^p \binom{N-p}{q} \binom{p-1}{q-1} \\ &= \frac{N}{N-p} \sum_{q=1}^p \binom{N-p}{q} \binom{p-1}{p-q}; \quad \binom{n}{k} = \binom{n}{n-k} \\ &= \frac{N}{N-p} \binom{N-1}{p}; \quad \binom{x+y}{n} = \sum_{k=0}^n \binom{x}{k} \binom{y}{n-k} \\ &= \frac{N}{(N-p)} \frac{(N-1)!}{(N-p-1)!p!} \end{aligned}$$

$$\begin{aligned}
&= \frac{N!}{(N-p)!p!} \\
&= \binom{N}{p}.
\end{aligned}$$

For a complete sequence of binomial coefficients, it is necessary to include both initial and end subgroups. Further corollaries can be established.

Corollary 5 For any given $N > 0$ under the listed condition, a set of projection function $\{L(p, N)\}$, $0 \leq p \leq N$ is composed of the same sequence of binomial coefficients

$$L(p, N) = \binom{N}{p}. \quad (11)$$

Proof For $0 < p < N$ condition, they are well determined by Theorem 5.1 and two end subgroups $p = \{0, N\}$, $\binom{N}{0} = \binom{N}{N} = 1$ by defined initial conditions.

Corollary 6 The sum of all possible $\{L(p, N)\}_{p=0}^N$ is

$$\sum_{p=0}^N L(p, N) = 2^N. \quad (12)$$

Proof Collecting all possible numbers by Corollary 2, we have

$$\begin{aligned}
\sum_{p=0}^N L(p, N) &= \sum_{p=0}^N \binom{N}{p} \\
&= (1+1)^N \\
&= 2^N.
\end{aligned}$$

Corollary 7 For $0 \leq p \leq N$, a pair of functions has an equivalent formula

$$L(p, N) = L(N-p, N). \quad (13)$$

Proof By Corollary 2.1, both equations are equal.

Theorem 8 For any $N > 0$, the sum of all possible functions on $\{f(q, p, N)\}_{\forall p, \forall q}$ or $\{L(p, N)\}_{p=0}^N$ is equal to 2^N

$$\sum_{\forall p} \sum_{\forall q} f(q, p, N) = \sum_{p=0}^N L(p, N) = 2^N. \quad (14)$$

Proof By Corollary 6, two equations are equal.

5.2 Triangular Sequence

Definition 8 For a given $N \geq 1$, let $T(N)$ denote a 2D structure with all nontrivial triangular numbers.

$$T(N) = \{f(q, p, N) | f(q, p, N) > 0, 0 \leq q \leq \lfloor N/2 \rfloor, 0 \leq p \leq N\} \quad (15)$$

Corollary 9 For a given N , if $|T(N)|$ be a total number of distinguishable elements for nontrivial triangular numbers, then $|T(N)|$ has the following equation:

$$|T(N)| = \begin{cases} N^2/4 + 2; & N \equiv 0, \pmod{2} \\ (N^2 - 1)/4 + 2; & N \equiv 1, \pmod{2}. \end{cases} \quad (16)$$

Proof By Corollary 2 for a given N , a triangular shape for nontrivial members is composed of two parts: a triangular area and two $q = 0$ points. The triangular area has $(N - 1)$ length and $\lfloor N/2 \rfloor$ high. If $N \equiv 0, \pmod{2}$, the triangular area is a regular triangle contained $N^2/4$ elements, so the total number of the generative triangular shape is $N^2/4 + 2$. For an odd valued N , a triangular area has additional $\lfloor N/2 \rfloor$ members side on a regular triangle with $\lfloor N/2 \rfloor^2$ elements, so the total number of elements is $\lfloor N/2 \rfloor^2 + \lfloor N/2 \rfloor + 2 = (N^2 - 1)/4 + 2$.

Definition 9 For a given $N \geq 1$, let $TS(N)$ denote an integer sequence with $|T(N)|$ elements for all nontrivial triangular numbers in $T(N)$

$$\begin{aligned} TS(N) := & [f(0, 0, N), f(0, N, N), \dots, \\ & \dots, f(q, q, N), \dots, f(q, p, N), \dots, f(q, N - q, N), \dots, \\ & \dots, f(\lfloor N/2 \rfloor, \lfloor N/2 \rfloor, N), f(\lfloor N/2 \rfloor, \lceil N/2 \rceil, N)], \\ & 1 \leq q \leq \lfloor N/2 \rfloor, q \leq p \leq N - q. \end{aligned} \quad (17)$$

5.3 Linear Sequence

Definition 10 For a given $N \geq 1$, let $L(N)$ denote a 1D structure with relevant linear numbers.

$$L(N) = \{L(p, N) | 0 \leq p \leq N\} \quad (18)$$

Corollary 10 For a given N , if $|L(N)|$ be a total number of distinguishable elements for linear numbers, then $|L(N)|$ satisfies Eq. 19.

$$|L(N)| = N + 1 \quad (19)$$

Table 8 $\{T(4), T(5), T(6)\}, \{L(4), L(5), L(6)\}$ subgroup numbers in three levels

N	$T(N)$	$q := 0$	1	2	3	p	$L(N)$	$L(N)$
4	$T(4) :=$		1			0		1
			4			1		4
			4	2		2	$L(4) :=$	6
			4			3		4
			1			4		1
5	$T(5) :=$		1			0		1
			5			1		5
			5	5		2	$L(5) :=$	10
			5	5		3		10
			5			4		5
	1			5		1		
6	$T(6) :=$		1			0		1
			6			1		6
			6	9		2		15
			6	12	2	3	$L(6) :=$	20
			6	9		4		15
			6			5		6
	1			6		1		

Definition 11 For a given $N \geq 1$, let $LS(N)$ denote an integer sequence with $|T(N)|$ elements for all linear numbers in $L(N)$ (Table 8)

$$LS(N) := [L(0, N), \dots, L(p, N), \dots, L(N, N)], \quad 0 \leq p \leq N. \quad (20)$$

From the listed six groups of $\{T(4), T(5), T(6)\}$ and $\{L(4), L(5), L(6)\}$ structures, two integer sequences are arranged as follows:

$$TS(4), TS(5), TS(6) := [1, 1, 4, 4, 4, 2, 1, 1, 5, 5, 5, 5, 5, 5, 1, 1, 6, 6, 6, 6, 6, 9, 12, 9, 2];$$

$$LS(4), LS(5), LS(6) := [1, 4, 6, 4, 1, 1, 5, 10, 10, 5, 1, 1, 6, 15, 20, 15, 6, 1].$$

6 Sample Cases

Two sample cases are selected for $N = \{17, 18\}$ to show their triangular numbers and generative structures in Table 9. In relation to relevant integer sequences, both $\{L(16), L(17)\}$ and $\{T(16), T(17)\}$ are shown in Table 9. Two integer sequences are significantly different. The triangular number sequence in this case with a total length of 140 integers is three times longer than the linear number sequence with a total

Table 9 Triangular number arrays for $N = \{16, 17\}$ cases

N	$L(N)$	$T(N)$
16	$L(16) :=$	1
		16
		120
		560
		1820
		4368
		8008
		11440
		12870
		11440
		8008
		4368
		1820
		560
		120
		16
		1
$ L(16) = 17$	$ T(16) = 66, \sum_{\forall q, \forall p} f(q, p, 16) = 65536 = 2^{16}$	
17	$L(17) :=$	1
		17
		136
		680
		2380
		6188
		12376
		19448
		24310
		24310
		19448
		12376
		6188
		2380
		680
		136
		17
1		
$ L(17) = 18$	$ TN(17) = 74, \sum_{\forall q, \forall p} f(q, p, 17) = 131072 = 2^{17}$	
\sum	$ LS(16), LS(17) = 35 = T(16) + T(17) $	$ T(16), T(17) = T(16) + T(17) = 140, \sum_{\forall q, \forall p} \sum_{n=16}^{17} f(q, p, n) = 196608 = 2^{16} + 2^{17}$

length of 35 integers. Two integer sequences represent different partition results on the same number $196608 = 2^{16} + 2^{17}$ for generative binomial and trinomial coefficients, respectively.

7 Conclusion

Due to the proposed elementary equation of trinomial coefficients with excellent symmetric properties on a 2D grid similar to binomial coefficients on a 1D line, projecting operation makes 2D $T(N)$ array be 1D linear $L(N)$ array, respectively. Two types of $TS(N)$ and $LS(N)$ integer sequences can be generated. As the simplest expansion of multinomial coefficients, discrete 2D geometry could provide solid combinatorial foundation to support multinomial explorations.

From a combinatorial geometry viewpoint, triangular numbers provide a key construction to link between trinomial and binomial representation in mathematical foundation. Trinomial integer sequences, as representatives, need to be deeply explored by modern combinatorial & discrete mathematical societies. Further explorations are expected on detailed analysis and systematic construction on both and practical applications.

Acknowledgements Both authors would like to thank Mr. Zhonghao Yang for his contribution to work on sample sequences, sincerely to gratitude @Qiaochu Yuan for the suggestion of combinatorial description and @Zander to provide a set of combinatorial equations to answer @zcaudate's question [18] in 2012.

References

1. J.R. Chen, *Combinatorial Mathematics* (Harbin Institute of Technology Press, Harbin, 2012). (in Chinese)
2. Discrete geometry. http://en.wikipedia.org/wiki/Discrete_geometry
3. Generative. <http://en.wikipedia.org/wiki/Generative>
4. Generative science. http://en.wikipedia.org/wiki/Generative_science
5. H.W. Gould, Some generalizations of Vandermonde's convolution. *Am. Math. Mon.* **63**(2), 84–91 (1956)
6. H.W. Gould, *Combinatorial Identities* (Morganton, Morgantown, 1972)
7. L.K. Hua, *Loo-Keng Hua Selected Papers* (Springer, 1982)
8. L.K. Hua, *Selected Work of Hua Loo-Keng on Popular Sciences* (Shanghai Education Press, 1984) (in Chinese)
9. D.E. Knuth, *The Art of Computer Programming*, vol. 1, 3rd edn. (Addison-Wesley, Upper Saddle River, 1998)
10. G. Polya, R. Tarjan, D. Woods, *Notes on Introductory Combinatorics* (Birkhauser, Boston, 1983)
11. G.Z. Tu, *Combinatorial Enumeration Methods and Applications* (Science Press, Beijing, 1981). (in Chinese)
12. J.H. van Lint, R.M. Wilson, *A Course in Combinatorics*, 2nd edn. (Cambridge University Press, New York, 2001)

13. L.X. Wang, *An Elementary Treatise on Combinations* (Harbin Institute of Technology Press, Harbin, 2012) (in Chinese)
14. Z.J. Zheng, A. Maeder, The conjugate classification of the kernel form of the hexagonal grid, in *Modern Geometric Computing for Visualization* (Springer, 1992), pp. 73–89. <https://doi.org/10.1007/978-4-431-68207-3>
15. Z.J. Zheng, *Conjugate transformation of regular plan lattices for binary images*, Ph.D. thesis, Monash University, 1994
16. J.Z.J. Zheng, C.H.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Fron. Electr. Electron. Eng. China* **5**(2), 163–172 (2010). Higher Educational Press and Springer. <https://doi.org/10.1007/s11460-010-0011-4>
17. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, A framework of variant logic construction for cellular automata, In A. Salcido (Ed.), *Cellular Automata—Innovative Modeling for Science and Engineering* (InTech Press, 2011). <https://doi.org/10.5772/15400>
18. @zcaudate’s question. <http://math.stackexchange.com/questions/155289/>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Symmetric Clusters in Hierarchy with Cryptographic Properties



Jeffrey Zheng

Abstract Symmetric Boolean functions play a key role in stream ciphers. Symmetric constructions provide core components in cryptographic applications. In this chapter, four meta symmetric clustering schemes (combination, crossing, variant and rotation) are organized in a hierarchy for n variables of 0–1 vectors in measuring phase spaces. Local counting properties in a cluster and global counting properties in a given level are formulated. From selected symmetric clusters, a number of various symmetric Boolean functions are formulated. Counting properties on symmetric clusters, vectors in selected clusters and special symmetric Boolean functions are listed. Four sets of symmetric Boolean functions are compared. Properties of symmetric clusters and Boolean functions are discussed. Main results are expressed in theorems and tables. Among four meta schemes, the variant scheme presents novel properties approximately with $O(n^2/4)$ clusters on a 2D phase space different from other schemes: combinatorial $O(n)$, crossing $O(n/2)$ and rotation $O(2^n/n)$ on 1D measuring phase spaces, respectively. The variant pseudorandom number generator is a similar approach on RC4 and HC128 stream ciphers using word-oriented 0–1 vectors. Further advanced researches and explorations on relevant optimal configurations are required.

Keywords Symmetric construction · Meta symmetric Cluster · hierarchy Boolean function · Four meta schemes · Phase space

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Kunming, China
e-mail: conjugatelagic@yahoo.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_5

1 Introduction

Symmetric Boolean functions [5] have been widely used as components of different cryptosystems [25] (e.g. in stream ciphers, block ciphers or hash functions). In combinatorial mathematics [10], a symmetric Boolean function is a Boolean function whose value does not depend on the permutation of its input bits [4], i.e. it depends only on the number of ones in the input on n variables of 0–1 vectors [21]. A total of 2^n vectors are composed of a vector space or a phase space for the construction [19]. For a specific symmetric Boolean function, it is necessary to have invariant properties undertaken a special group of permutations [18]. For example, rotation symmetric Boolean functions are invariant under the circular translation of indices. In addition to rotation symmetric properties, multiple invariants (combination, crossing, reflection, translation) may be composed of various symmetric subgroups of permutations [10, 22]. Various combinatorial counting schemes are explored [34–36].

1.1 Symmetric Functions—Combinatorial Invariant

From a combinatorial viewpoint, symmetric Boolean functions are a combinatorial invariant that links to the number of one elements p , $0 \leq p \leq n$ in a vector [35]. In combinatorics, this type of function has been linked to binomial coefficients, and normally, there are $n + 1$ partitions to distinct the parameter of a measuring phase space into various clusters [30]. Symmetric Boolean functions are characterized [36] by the fact that their outputs only depend on the p numbers of their inputs. The usefulness of symmetric functions in a cryptographic context has been widely explored which possess good cryptographic properties [6, 7].

1.2 Crossing Number - Topological Invariant

A zero-crossing [23] describes a point where the sign of a mathematical function changes (e.g. from positive to negative), represented by a crossing of the axis (zero value) in the graph of the function. It is a commonly used term in electronics, mathematics, sound and image processing.

From a measuring viewpoint, a 0–1 vector with n bits can be expressed as a circular ring that has a fixed crossing number q , $0 \leq q \leq \lfloor \frac{n}{2} \rfloor$ distinguished a number of derivative changes on either 0–1 or 1–0, respectively. This type of derivative invariant is widely used in crypto-analysis for many years. In NIST random data testing packages [1], binary derivative [3] and Runs tests [2] play an important role to measure the randomness of a binary sequence formed by a pseudorandom number generator for use in cipher systems. From an analytic viewpoint, this parameter is a

topological invariant and different from a combinatorial invariant to provide another type of partition capacities to organize a set of clusters in a measuring phase space.

1.3 Rotation Symmetric Functions - Geometric Invariant

In combinatorial mathematics, rotation symmetric properties are widely explored from early stage of abstract group theories and symmetric group constructions [10, 22] as a geometric invariant. Filiol and Fontaine [12] were initially explored on balanced Boolean functions with a good correlation immunity. Pieprzyk and Qu [26] were applied in crypto-applications to use Rotation Symmetric Boolean Functions (RSBF) as components in the rounds of a hashing algorithm.

Extensive R&D activities on RSBF are continuous for last decades, a list of advanced works explored, such as degree and non-linearity [6], optimal algebraic immunity [7], bent and semi-bent functions [8, 33], non-linearity of resilient, non-linear Boolean functions [20, 28], balanced Boolean functions [12, 16], non-linear balanced Boolean functions [31], weights and non-linearity [11], immune combining functions [32], count and cryptographic properties [13, 29], etc.

1.4 Trinomial Coefficients

It is a natural approach [10, 18, 19] to apply binomial coefficients to partition a measuring phase space on 0–1 vector sets. However, when parameters increase more than three, a generalization [34–36] using multinomial coefficients may not provide a general solution on further refined partitions, if the processed phase space is composed of 0–1 vectors. It is convenient for us to use a trinomial expression to show this fact.

Let $n = n_1 + n_2 + n_3, 0 < n,$

$$\binom{n}{n_1, n_2, n_3} = \frac{n!}{n_1!n_2!n_3!},$$

collecting all possible trinomial coefficients, we have

$$\sum_{\forall n_1, n_2, n_3} \binom{n}{n_1, n_2, n_3} = 3^n \neq 2^n. \quad (1)$$

From Eq. 1, it is interesting to notice that trinomial coefficients provide further segments to partition three-valued 0–2 vectors. Due to this reason, extensions using multinomial coefficients may not be directly relevant to binary-valued 0–1 vector sets. Refined identity equations of combinatorics are required [14, 15].

1.5 Variant Symmetric Schemes - Variant Invariants

Various schemes to use multiple invariants to partition special phase spaces have been explored in binary image analysis and processing for many years. In 1990s, Zheng [39, 40] proposed conjugate classifications to apply seven invariants in a hierarchy to partition the kernels of four regular plane lattices on $n = \{4, 5, 7, 9\}$ cases for 2D binary images. For n -tuple 0–1 vectors, variant logic frameworks [41, 42] are proposed in 2010s, various applications are explored, such as 3D visual method [37], variant Pseudorandom Number Generator (PRNG) [38, 43], computational simulation on quantum interactions [44–47] and non-coding DNA analysis [48–50].

1.6 Organization of the Chapter

In this chapter, an algebraic equation of variant trinomial will be proposed as a kernel structure to arrange a hierarchical phase space. This extension provides a general framework of multiple symmetric operations to support three numeric numbers: combinatorial, crossing and variant in a hierarchy. Three meta clusters of measuring phase spaces are identified by the three invariants: $\{n, p, q\}$ and their combinations. Refined levels can be compared with the rotation symmetric scheme under $n = \{1, 2, 3, 4, 5\}$ conditions. Similarities and differences among the four schemes are explored.

In Sect. 2, symbols and local counting properties of symmetric clusters in measuring spaces are defined, algebraic equations are formulated and two important projections are discussed. In Sect. 3, variant symmetric clusters and their elementary equation are proposed. In Sect. 4, four number sets of symmetric clusters are explored from a global viewpoint. In Sect. 5, symmetric Boolean functions of selected clusters are constructed and both algebraic and approximate numeric properties are discussed. In Sect. 6, cryptographic properties of symmetric Boolean functions in a hierarchy are discussed and special properties on the variant scheme are stressed. Section 7 is the conclusion of the chapter. Main results of the chapter are expressed in a list of theorems and corollaries in Sects. 2–5, respectively.

2 Symmetric Clusters in Measuring Phase Spaces

In this section, basic symbols, primary definitions and algebraic formulas are defined for different clusters in their measuring phase spaces.

2.1 Basic Symbols

Main symbols in this chapter are listed in Table 1.

2.2 Primary Definitions

Definition 1 (*x* an n -tuple vector on 0–1 variables) Let x be a 0–1 vector with n length.

$$x = (x_{n-1}, \dots, x_i, \dots, x_0), 0 \leq i < n, x_i \in \{0, 1\} = B_2, x \in B_2^n, \quad (2)$$

e.g. $x = 110010, n = 6$.

Table 1 Basic symbols

Symbol	Notes
n	Number of 0–1 variables, $1 \leq n$
x	0–1 vector $x = (x_{n-1}, \dots, x_i, \dots, x_0), x_i \in \{0, 1\} = B_2, 0 \leq i < n$
I	$I(x)$ index for a vector x
$\Omega(n)$	Phase space of vector set $\{x\}, \Omega(n) = \{\forall x 0 \leq I < 2^n\}$
$f_\Omega(n)$	Number of vectors in $\Omega(n)$
R	$R(x, r)$ rotation operator
F	$F(x)$ reflection operator
p	$p(x)$ number of 1's elements in $x, 0 \leq p \leq n$
q	$q(x)$ number of cyclic crossings either 0–1 or 1–0 in x
$L(p, n)$	Combinatorial cluster of vectors in $\Omega(n), L(p, n) \subset \Omega(n)$
$E(q, n)$	Crossing cluster of vectors in $\Omega(n), E(q, n) \subset \Omega(n)$
$V(q, p, n)$	Variant cluster of vectors in $\Omega(n), V(q, p, n) \subset \Omega(n)$
$G(m, n)$	m -th rotation symmetric cluster of vectors in $\Omega(n), G(m, n) \subset \Omega(n)$
$f_E(q, n)$	Crossing number of vectors in a cluster $E(q, n)$
$f_L(p, n)$	Combinatorial number of vectors in a cluster $L(p, n)$
$f_V(q, p, n)$	$f_V(q, p, n)$ variant number of vectors in a cluster $V(q, p, n)$
$f_G(m, n)$	Rotation number of vectors in the m -th cluster $G(m, n)$
$O(N)$	Approximate number of N
$C_X(n)$	Approximate number of clusters in a set of $\{X(\cdot)\}, X \in \{E, L, V, G\}$
$f_X(n)$	Approximate number of clusters in a set of $\{X(\cdot)\}, X \in \{E, L, V, G\}$
$SF_X(n)$	Number of Symmetric Boolean Functions (SBF) in $\{X(\cdot)\}, X \in \{E, L, V, G\}$
$SF_{Xb}(n)$	Number of balanced SBF_X in $\{X(\cdot)\}, X \in \{L, V, G\}, n = 0 \pmod 2$
$SF_{Eb}(n)$	Number of balanced SBF_E in $\exists q, \{E(q, n)\}, n = 0 \pmod 4$

Definition 2 (*I index for a vector x*) For a vector x , let I or $I(x)$ be an index:

$$I = I(x) = \sum_{i=0}^{n-1} x_i * 2^i, \quad (3)$$

e.g. $x = 110010$, $I(x) = 2^5 + 2^4 + 2 = 32 + 16 + 2 = 50$.

Definition 3 ($\Omega(n)$ a full set of n -tuple 0–1 vectors) Let $\Omega(n)$ be a vector space or a phase space of all n -tuple 0–1 vectors,

$$\Omega(n) = \{\forall x | 0 \leq I < 2^n, x \in B_2^n\} \text{ and } \Omega(n) = B_2^n. \quad (4)$$

Definition 4 Let $f_\Omega(n)$ denote a number of vectors in $\Omega(n)$.

Lemma 1 $f_\Omega(n)$ is equal to 2^n .

Proof For a vector $x \in B_2^n$ from $0 \dots 0$ to $1 \dots 1$, its index I can cover a full region of $0 \leq I < 2^n$, so $\Omega(n)$ contains 2^n distinct vectors and $f_\Omega(n) = 2^n$.

Definition 5 (*Measuring Phase Space*) If a phase space can be organized by various invariants, then it is a measuring phase space and its dimension is determined by a number of active invariants.

Corollary 1 For any $n > 0$, $\Omega(n)$ is a measuring phase space in zero dimension.

Proof For any $n > 0$, $\Omega(n)$ is composed of one cluster of vectors as a single point.

Definition 6 (*R rotation operator*) Let $R(x; r)$ be a rotation operator on a vector x rotation $-n < r < n$ positions:

$$\begin{aligned} R(x; r) &= R(x_{n-1}, \dots, x_i, \dots, x_0; r) \\ &= (x_{n-1+r \bmod n}, \dots, x_{i+r \bmod n}, \dots, x_{0+r \bmod n}), \end{aligned} \quad (5)$$

e.g. $x = 110010$, $\{R(x; r)\}_{r=0}^5 = \{110010, 100101, 001011, 010110, 101100, 011001\}$ with six distinct vectors.

Lemma 2 (*Maximal cyclic structure*) Initially from any vector x under a rotation operator, at most n distinct vectors will be distinguished under the rotation operator.

Proof From any x , a set of $\{R(x; r)\}_{r=0}^{n-1}$ with n vectors can be generated. If the listed set of n vector sequences contains more than one cycle, then the number of distinct vectors will be less than n .

For example, $x = 110110$, $\{R(x; r)\}_{r=0}^5 = \{110110, 101101, 011011, 110110, 101101, 011011\}$ with only a set of three distinct vectors: $\{110110, 101101, 011011\}$.

Definition 7 (*F reflection operator*) Let $F(x)$ be a reflect operator,

$$F(x) = F(x_{n-1}, \dots, x_i, \dots, x_0) = (x_0, \dots, x_i, \dots, x_{n-1}), 0 \leq i < n. \quad (6)$$

Lemma 3 (*A pair of reflections*) For any vector x , only two results are distinguished under $F(x)$ operation: (1) $F(x) = x$; (2) $F(x) \neq x$.

Proof (1) If $F(x) = x$, then the values of the vector x are distributed as a central symmetric form; (2) if $F(x) \neq x$, then the vector x does not have a symmetric distribution.

For example, $x = 110010$, $F(x) = 010011$; $y = 110011$, $F(y) = 110011$.

Definition 8 (*p number of one elements*) Let p or $p(x)$ be a number of one elements in x ,

$$p = p(x) = \sum_{i=0}^{n-1} x_i, 0 \leq p \leq n. \quad (7)$$

For example, $x = 110010$, $p(x) = 3$; $y = 110011$, $p(y) = 4$.

Definition 9 (*q number of cyclic crossings*) Let q or $q(x)$ be a number of cyclic crossings either 0-1 or 1-0 in a vector x ,

$$\begin{aligned} q = q(x) &= \sum_{0 \leq i < n} (x_i \equiv 0) \& (x_{i+1} \equiv 1); x_i, x_{i+1} \in B_2, (i+1) \pmod n; \\ &= \sum_{0 \leq i < n} (x_i \equiv 0) \& (x_{i-1} \equiv 1); x_i, x_{i-1} \in B_2, (i-1) \pmod n; \\ &0 \leq q \leq \lfloor \frac{n}{2} \rfloor. \end{aligned} \quad (8)$$

For example, $x = 110010$, $q(x) = 2$; $y = 110011$, $q(y) = 1$.

2.3 Counting Properties on Rotation Clusters

Definition 10 (*G(m, n) m-th rotation symmetric cluster*) Let $G(m, n)$ be an m -th rotation symmetric cluster of vectors, $G(m, n) = \Omega(n|m) \subset \Omega(n)$ in $\Omega(n)$, and let a total number of rotation symmetric clusters be $C_G(n)$, $1 \leq m \leq C_G(n)$,

$$\Omega(n) = \bigcup_{m=1}^{C_G(n)} \Omega(n|m) = \bigcup_{m=1}^{C_G(n)} G(m, n). \quad (9)$$

Corollary 2 A set of $\{G(m, n)\}_{m=1}^{C_G(n)}$ is composed of a measuring phase space in one dimension.

Proof Using the parameter m , $\{G(m, n)\}_{m=1}^{C_G(n)}$ can be listed in a linear order.

Lemma 4 By Burnside's lemma, ϕ being Euler's phi-function,

$$C_G(n) = \frac{1}{n} \sum_{k|n} \phi(k) 2^{\frac{n}{k}}. \quad (10)$$

Proof A brief proof of this lemma can be found in [29].

Definition 11 Let $f_G(m, n)$ denote a number of vectors in the m -th cluster $G(m, n)$.

Corollary 3 For any $f_G(m, n)$, $1 \leq f_G(m, n) \leq n$.

Proof Due to Lemma 2, each $f_G(m, n) \leq n$ in general; for two special vectors in $\{0 \dots 0, 1 \dots 1\}$, we have $f_G(m, n) = 1$.

Corollary 4 Collecting all possible rotation clusters, the total number of vectors is equal to $f_\Omega(n)$

$$\begin{aligned} \sum_{m=1}^{C_G(n)} f_G(m, n) &= 2^n \\ &= f_\Omega(n). \end{aligned} \quad (11)$$

Proof From Lemma 4 and Corollary 3, it contains a full set of 2^n vectors in $\Omega(n)$.

Lemma 5 For a given n , $C_G(n)$ has an approximate number,

$$C_G(n) \approx O\left(\frac{2^n}{n}\right). \quad (12)$$

Proof Using Corollaries 3 and 4, each distinct cluster contains at most n vectors; it is a natural to have such an approximate number in enumeration.

It is convenient to list defined rotation parameters in Table 2 for $n = 4$ condition.

2.4 Counting Properties on Measuring Phase Spaces

For any vector $x \in \Omega(n)$, three measuring parameters $\{n, p, q\}$ are represented as three invariants. Three measurements transfer a phase space into a set of measuring phase spaces in a hierarchy.

Table 2 Six rotation clusters, various vectors in $\{G(m, 4)\}$

(m, n)	$G(m, n)$	$f_G(m, n)$
(1, 4)	{0000}	1
(2, 4)	{0001, 0010, 0100, 1000}	4
(3, 4)	{0011, 0110, 1100, 1001}	4
(4, 4)	{0101, 1010}	2
(5, 4)	{0111, 1110, 1101, 1011}	4
(6, 4)	{1111}	1
	$C_G(4) = 6$	$f_\Omega(n) = 16$

Definition 12 ($L(p, n)$ combinatorial cluster) Let $L(p, n)$ be a combinatorial cluster of vectors in $\Omega(n)$, $L(p, n) = \Omega(n|p) \subset \Omega(n)$. Two parameters $\{n, p\}$ partition the phase space $\Omega(n)$ to form a set of clusters $\{L(p, n)\}$ in a measuring phase space.

$$\Omega(n|p) = L(p, n) = \{\forall x | 0 \leq p \leq n, x \in \Omega(n)\}. \quad (13)$$

Corollary 5 A set of $\{L(p, n)\}_{p=0}^n$ is composed of a measuring phase space in one dimension.

Proof The parameter p is the active invariant to arrange the phase space in a linear order.

Definition 13 Let $C_L(n)$ be a number of clusters in $\forall p, \{L(p, n)\}$.

Lemma 6 For a given n ,

$$C_L(n) = n + 1. \quad (14)$$

Proof Using Definition 12, $0 \leq p \leq n$ and for any p , $L(p, n) \neq \emptyset$, the parameter p partitions the whole set $\Omega(n)$ into $n + 1$ distinct subsets as clusters.

Definition 14 ($f_L(p, n)$ combinatorial number) Let $f_L(p, n)$ be a combinatorial number of vectors in a cluster $L(p, n)$.

Lemma 7 For a pair of $\{n, p\}$ parameters,

$$f_L(p, n) = \binom{n}{p} \quad (15)$$

Proof Using Definition 12, this number is equal to a binomial coefficient selected p elements from n positions.

It is convenient to list defined measuring parameters in Table 3 for $n = 4$ condition.

Table 3 Five clusters, various vectors in $\{L(p, 4)\}$

(p, n)	$L(p, n)$	$f_L(p, n)$
(0, 4)	{0000}	1
(1, 4)	{0001, 0010, 0100, 1000}	4
(2, 4)	{0011, 0110, 1100, 1001, 0101, 1010}	6
(3, 4)	{0111, 1110, 1101, 1011}	4
(4, 4)	{1111}	1
	$C_L(4) = 5$	$f_\Omega(4) = 16$

Definition 15 ($E(q, n)$ crossing cluster of vectors) Let $E(q, n)$ be a crossing cluster of vectors in $\Omega(n)$, $E(q, n) = \Omega(n|q) \subset \Omega(n)$. Two parameters $\{n, q\}$ partition the phase space $\Omega(n)$ to form a set of clusters $\{E(q, n)\}$ in a measuring phase space.

$$\Omega(n|q) = E(q, n) = \{\forall x | 0 \leq q \leq \lfloor \frac{n}{2} \rfloor, x \in \Omega(n)\} \tag{16}$$

Corollary 6 A set of $\{E(q, n)\}_{q=0}^{\lfloor n/2 \rfloor}$ is composed of a measuring phase space in one dimension.

Proof The parameter q is the active invariant to arrange the phase space in a linear order.

Definition 16 Let $C_E(n)$ be a number of crossing clusters in $\forall q, \{E(q, n)\}$.

Lemma 8 For a given $n > 0$,

$$C_E(n) = \lfloor \frac{n}{2} \rfloor + 1. \tag{17}$$

Proof According to Definition 15 and each $E(q, n) \neq \emptyset, 0 \leq q \leq \lfloor \frac{n}{2} \rfloor$, the parameter q partitions the whole set $\Omega(n)$ into $\lfloor \frac{n}{2} \rfloor + 1$ distinct subsets as clusters.

Definition 17 ($f_E(q, n)$ number of vectors) Let $f_E(q, n)$ be a number of vectors in a cluster $E(q, n)$.

Lemma 9 For a pair of $\{n, q\}$ parameters,

$$f_E(q, n) = 2 * \binom{n}{2q}, 0 \leq q \leq \lfloor \frac{n}{2} \rfloor. \tag{18}$$

Proof Two cases can be distinguished: Case 1: $q = 0$; Case 2: $1 \leq q \leq \lfloor \frac{n}{2} \rfloor$.

Case 1: All n values are either 1 or 0, $2 * \binom{n}{0} = 2$.

Case 2: For a given $q, 2q$ crossing positions are composed of a pair of a 0–1 crossing then a 1–0 crossing repeatedly for q times in a vector and this configuration has a total of $\binom{n}{2q}$ vectors included, and the same pair of positions can be exchanged as a

Table 4 Three clusters, vectors in $\{E(q, 4)\}$ cases

(q, n)	$E(q, n)$	$f_E(q, n)$
(0, 4)	{0000, 1111}	2
(1, 4)	{0001, 0010, 0100, 1000, 0011, 0110, 1100, 1001, 0111, 1110, 1101, 1011}	12
(2, 4)	{0101, 1010}	2
	$C_E(4) = 3$	$f_\Omega(4) = 16$

pair of 1–0 and 0–1 crossings with the same number of different vectors, so a total of $2 * \binom{n}{2q}$ vectors are involved in each q selection.

It is convenient to list above defined measuring parameters in Table 4 for $n = 4$ condition.

3 Variant Symmetric Clusters

Definition 18 ($V(q, p, n)$ variant cluster) Let $V(q, p, n)$ be a variant cluster of vectors in $\Omega(n)$, $V(q, p, n) = \Omega(n|p, q) \subset \Omega(n)$. Three parameters $\{n, p, q\}$ partition the phase space $\Omega(n)$ to form a set of clusters $\{V(q, p, n)\}$ in a measuring phase space.

$$\Omega(n|p, q) = V(q, p, n) = \{\forall x | 0 \leq p \leq n, 0 \leq q \leq \lfloor \frac{n}{2} \rfloor, x \in \Omega(n)\} \quad (19)$$

Corollary 7 A set of $\{V(q, p, n)\}_{\forall q, p}$ is composed of a measuring phase space on two dimensions.

Proof Both invariants q and p are two active invariants to arrange the phase space on a 2D plane lattice.

Lemma 10 Both $\{L(p, n)\}$ combinatorial clusters and $\{E(q, n)\}$ crossing clusters can be generated from special subsets of $\{V(q, p, n)\}$ variant clusters.

Proof For a given p , $L(p, n)$ can be determined by

$$L(p, n) = \bigcup_{q=0}^{\lfloor \frac{n}{2} \rfloor} V(q, p, n).$$

For a given q , $E(q, n)$ can be determined by

$$E(q, n) = \bigcup_{p=0}^n V(q, p, n).$$

Table 5 Three sets of variant clusters for $n = 4$ in $\{V(q, p, n)\}$ condition

$q \backslash p$	0	1	2	3	4	$E(q, n)$
0	$V(0, 0, 4)$				$V(0, 4, 4)$	$E(0, 4)$
1		$V(1, 1, 4)$	$V(1, 2, 4)$	$V(1, 3, 4)$		$E(1, 4)$
2			$V(2, 2, 4)$			$E(2, 4)$
$L(p, n)$	$L(0, 4)$	$L(1, 4)$	$L(2, 4)$	$L(3, 4)$	$L(4, 4)$	$\Omega(4)$

Applying this set of partitions, three sets of relevant clusters can be identified.

For example, $n = 4$, all 16 vectors in the vector space, three sets of clusters can be distinguished as six clusters $\{V(q, p, n)\}$, five clusters for $\{L(p, n)\}$ and three clusters for $\{E(q, n)\}$ shown in Table 5, respectively.

Definition 19 Let $C_V(n)$ be a number of non-trivial variant clusters in $\forall q, p, \{V(q, p, n)\}$.

In general condition for any given $n > 1$, three sets of variant clusters could be shown in Fig. 1.

Theorem 1 For a given n , $C_V(n)$ satisfies Eq. 20

$$C_V(n) = \begin{cases} n^2/4 + 2; & n \equiv 0 \pmod{2} \\ (n^2 - 1)/4 + 2; & n \equiv 1 \pmod{2}. \end{cases} \tag{20}$$

Proof From Fig. 1 for a given n , a triangular shape for non-trivial variant clusters is composed of two parts: a triangular area and two $q = 0$ points. The triangular

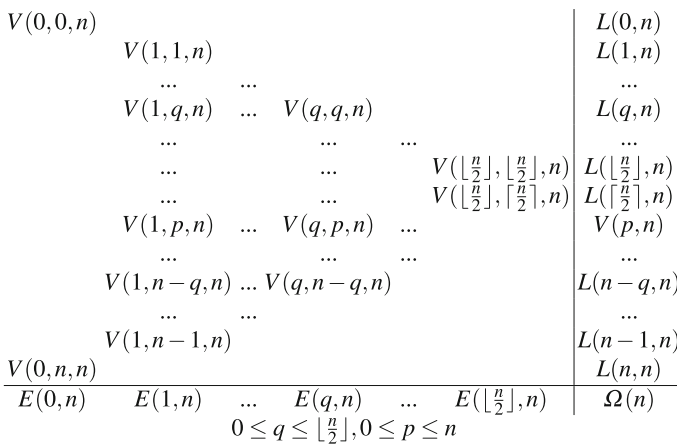


Fig. 1 Three sets of variant clusters $\{V(q, p, n)\}$, $\{E(q, n)\}$, $\{L(p, n)\}$ for $n > 1$

area has $(n - 1)$ length and $\lfloor n/2 \rfloor$ high. If $n \equiv 0 \pmod 2$, the triangular area is a regular triangle contained $n^2/4$ clusters, so the total number of this triangular shape contains $n^2/4 + 2$ clusters. For an odd valued n , a triangular area has additional $\lfloor n/2 \rfloor$ clusters side on a regular triangle with $\lfloor n/2 \rfloor^2$ clusters, so the total number of clusters is $\lfloor n/2 \rfloor^2 + \lfloor n/2 \rfloor + 2 = (n^2 - 1)/4 + 2$.

3.1 Variant Trinomial Coefficients – Elementary Equation

Definition 20 Let $f_V(q, p, n)$ or $f(q, p, n)$ $0 \leq p \leq n, 0 \leq q \leq \lfloor \frac{n}{2} \rfloor$ denote an enumeration function for a number of 0–1 vectors in a variant cluster.

It is convenient to list relevant measuring parameters in Table 6 for $n = 4$ conditions.

Definition 21 For two initial and end clusters $p = \{0, n\}, q = 0$, let two cases be $f(0, 0, n) = f(0, n, n) = 1$. For other cases, each cluster $0 < p < n, 0 < q \leq \lfloor \frac{n}{2} \rfloor$ contains a subgroup of vectors under a given condition. A variant trinomial coefficient for a number of vectors in a cluster is defined as an elementary equation in Equation 21,

$$f(q, p, n) = \frac{n}{n - p} \binom{n - p}{q} \binom{p - 1}{q - 1}. \tag{21}$$

Applying variant trinomial coefficients in Eq. 21, there is no difficult to process more complicated cases in enumeration. Global arrangements on their triangular shapes are convenient to be arranged by p measures in vertical direction. Two cases $n = \{4, 5\}$ are shown in Table 7.

In a general condition for any given $n > 1$, three sets of various numbers can be shown in Fig. 2.

Table 6 Six clusters, vectors in $\{V(q, p, 4)\}$

(q, p, n)	$V(q, p, 4)$	$f(q, p, 4)$
$(0, 0, 4)$	{0000}	1
$(0, 4, 4)$	{1111}	1
$(1, 1, 4)$	{0001, 0010, 0100, 1000}	4
$(1, 2, 4)$	{0011, 0110, 1100, 1001}	4
$(1, 3, 4)$	{0111, 1110, 1101, 1011}	4
$(2, 2, 4)$	{0101, 1010}	2
	$C_V(4) = 6$	$f_\Omega(4) = 16$

Table 7 Three sets of vector numbers $\{f(q, p, n)\}, \{f_E(q, n)\}, \{f_L(p, n)\};(a) n = 4;(b) n = 5$

$p \backslash q$	0	1	2	$f_L(p, 4)$
0	1			1
1		4		4
2		4	2	6
3		4		4
4	1			1
$f_E(q, 4)$	2	12	2	$f_\Omega(4) = 16$

(a) $n = 4$

$p \backslash q$	0	1	2	$f_L(p, 5)$
0	1			1
1		5		5
2		5	5	10
3		5	5	10
4		5		5
5	1			1
$f_E(q, 5)$	2	20	10	$f_\Omega(5) = 32$

(b) $n = 5$

$f(0, 0, n)$ $f(1, 1, n)$ \dots $f(1, q, n) \dots f(q, q, n)$ \dots \dots $f(1, p, n) \dots f(q, p, n) \dots$ \dots $f(1, n - q, n) \dots f(q, n - q, n)$ \dots $f(1, n - 1, n)$ $f(0, n, n)$		$f_L(0, n)$ $f_L(1, n)$ \dots $f_L(q, n)$ \dots $f_L(\lfloor \frac{n}{2} \rfloor, n)$ $f_L(\lceil \frac{n}{2} \rceil, n)$ $f(p, n)$ \dots $f_L(n - q, n)$ \dots $f_L(n - 1, n)$ $f_L(n, n)$
$f_E(0, n)$		$f_\Omega(n)$
$f_E(1, n) \dots f_E(q, n) \dots f_E(\lfloor \frac{n}{2} \rfloor, n)$		$0 \leq q \leq \lfloor \frac{n}{2} \rfloor, 0 \leq p \leq n$

Fig. 2 Three sets of $\{f(q, p, n)\}, \{f_E(q, n)\}, \{f(p, n)\}$ variant numbers for $n > 1$

3.2 Combinatorial Projection on Variant Clusters

From an algebraic viewpoint, the following theorems and corollaries are established for a general condition to meet any $n \geq 1$ cases.

Lemma 11 *If $f_L(p, n) = \sum_{q=1}^p f(q, p, n), 0 < p < n$, then the projection function $f_L(p, n)$ is a binomial coefficient and*

$$f_L(p, n) = \binom{n}{p}. \tag{22}$$

Proof For a fixed $p, 0 < p < n$, all possible $\{f(q, p, n)\}$ are collected to form the following combinatorial identities: [14, 15, 21],

$$\begin{aligned}
f_L(p, n) &= \sum_{q=1}^p f(q, p, n) \\
&= \sum_{q=1}^p \frac{n}{n-p} \binom{n-p}{q} \binom{p-1}{q-1} \\
&= \frac{n}{n-p} \sum_{q=1}^p \binom{n-p}{q} \binom{p-1}{q-1} \\
&= \frac{n}{n-p} \sum_{q=1}^p \binom{n-p}{q} \binom{p-1}{p-q}; \quad \binom{N}{k} = \binom{N}{N-k} \\
&= \frac{n}{n-p} \binom{n-1}{p}; \quad \binom{x+y}{N} = \sum_{k=0}^N \binom{x}{k} \binom{y}{N-k} \\
&= \frac{n}{(n-p)} \frac{(n-1)!}{(n-p-1)!p!} \\
&= \frac{n!}{(n-p)!p!} \\
&= \binom{n}{p}.
\end{aligned}$$

For a complete sequence of binomial coefficients, it is necessary to include both initial and end clusters. Further Theorem 2 can be established.

Theorem 2 For any given $n > 0$, a set of projection function $\{f_L(p, n)\}_{p=0}^n$ is composed of the same sequence of binomial coefficients

$$f_L(p, n) = \binom{n}{p}. \quad (23)$$

Proof For $0 < p < n$ condition, the equation has been determined by Lemma 11 and two end clusters $p = \{0, n\}$, $\binom{n}{0} = \binom{n}{n} = 1$ are determined by Definition 21.

Corollary 8 The sum of all possible $\{f_L(p, n)\}_{p=0}^n$ is equal to $f_\Omega(n)$,

$$\sum_{p=0}^n f_L(p, n) = f_\Omega(n) = 2^n. \quad (24)$$

Proof Collecting all possible numbers in Theorem 2, we have

$$\begin{aligned}
\sum_{p=0}^n f_L(p, n) &= \sum_{p=0}^n \binom{n}{p} \\
&= (1 + 1)^n \\
&= 2^n \\
&= f_\Omega(n).
\end{aligned}$$

3.3 Crossing Projection on Variant Clusters

Lemma 12 *If $f_E(q, N) = \sum_{p=q}^{n-q} f(q, p, n)$, $1 \leq q \leq \lfloor \frac{n}{2} \rfloor$, then the enumeration function $f_E(q, n)$ is a double of a binomial coefficient*

$$f_E(q, n) = 2 \binom{n}{2q}. \quad (25)$$

Proof For a fixed q , collecting all possible $\{f(q, p, n)\}_{p=q}^{n-q}$, the following combinatorial identities [14, 15, 21] are deduced:

$$\begin{aligned}
f_E(q, n) &= \sum_{p=q}^{n-q} f(q, p, n) \\
&= \sum_{p=q}^{n-p} \frac{n}{n-p} \binom{n-p}{q} \binom{p-1}{q-1} \\
&= \sum_{p=q}^{n-p} \frac{n}{q} \binom{n-p-1}{q-1} \binom{p-1}{q-1}; \quad \frac{N}{q} \binom{N-p-1}{q-1} = \frac{N}{N-p} \binom{N-p}{q} \\
&= \frac{n}{q} \sum_{p=q}^{n-p} \binom{n-p-1}{q-1} \binom{p-1}{q-1}
\end{aligned}$$

$$\begin{aligned}
&= \frac{n}{q} \binom{n-1}{2q-1}; \quad \binom{N+1}{r+s+1} = \sum_{k=r}^{N-s} \binom{k}{r} \binom{N-k}{s} \\
&= 2 \frac{n}{2q} \frac{(n-1)!}{(n-2q)!(2q-1)!} \\
&= 2 \frac{n!}{(2q)!(n-2q)!} \\
&= 2 \binom{n}{2q}.
\end{aligned}$$

Theorem 3 For any given $n > 0$ under the listed condition, a set of projection function $\{f_E(q, n)\}_{0 \leq q \leq \lfloor \frac{n}{2} \rfloor}$ are composed of the subsequence of binomial coefficients,

$$f_E(q, n) = 2 \binom{n}{2q}. \quad (26)$$

Proof For $1 \leq q \leq \lfloor n/2 \rfloor$ condition, equations are determined by Lemma 12 and for the initial subgroup, we have $q = 0$, $f_E(0, n) = \binom{n}{0} + \binom{n}{n} = 2 \binom{n}{0}$.

Corollary 9 For $n \equiv 0 \pmod{2}$, $0 \leq q \leq n/2$, there are a pair of symmetric functions

$$f_E(q, n) = f_E(n/2 - q, n). \quad (27)$$

Proof Under $n \equiv 0 \pmod{2}$ condition,

$$\begin{aligned}
f_E(q, n) &= 2 \binom{n}{2q} \\
&= 2 \binom{n}{n-2q} = 2 \binom{n}{2(n/2 - q)} \\
&= f_E(n/2 - q, n).
\end{aligned}$$

Corollary 10 For $n \equiv 0 \pmod{4}$, $q = n/4$, $f_E(n/4, n)$ has the maximal value

$$f_E(n/4, n) > f_E(q, n), \quad q \neq n/4. \quad (28)$$

Proof Under $n \equiv 0 \pmod{4}$ condition,

$$f_E(q, n) = 2 \binom{n}{2q} < 2 \binom{n}{n/2} = 2 \binom{n}{2n/4} = f_E(n/4, n).$$

Corollary 11 *The sum of all possible $\{f_E(q, n)\}_{0 \leq q \leq \lfloor \frac{n}{2} \rfloor}$ is equal to $f_\Omega(n)$,*

$$\sum_{q=0}^{\lfloor \frac{n}{2} \rfloor} f_E(q, n) = f_\Omega(n) = 2^n. \quad (29)$$

Proof Collecting all possible numbers, we have the following equations:

$$\begin{aligned} \sum_{q=0}^{\lfloor \frac{n}{2} \rfloor} f_E(q, n) &= \sum_{q=0}^{\lfloor \frac{n}{2} \rfloor} 2 \binom{n}{2q} \\ &= 2 \sum_{q=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2q}, \quad \sum_{k \geq 0} \binom{n}{2k} = \sum_{k \geq 0} \binom{n}{2k+1} = 2^{n-1} \\ &= 2 \times 2^{n-1} \\ &= 2^n \\ &= f_\Omega(n). \end{aligned}$$

3.4 Relationships of Four Symmetric Clusters

Theorem 4 *For any $n > 0$, the sum of all possible functions on $\{f(q, p, n)\}_{\forall p, \forall q}$ or $\{f_E(q, n)\}_{0 \leq q \leq \lfloor \frac{n}{2} \rfloor}$ or $\{f_L(p, n)\}_{p=0}^n$ or $\{f_G(m, n)\}$, $1 \leq m \leq C_G(n)$ is equal to $f_\Omega(n)$*

$$\begin{aligned} f_\Omega(n) &= \sum_{\forall p} \sum_{\forall q} f(q, p, n) = \sum_{q=0}^{\lfloor \frac{n}{2} \rfloor} f_E(q, n) = \sum_{p=0}^n f_L(p, n) \\ &= \sum_{m=1}^{C_G(n)} f_G(m, n) \\ &= 2^n. \end{aligned} \quad (30)$$

Proof From the results of Corollaries 4, 8 and 11, four schemes provide various partitions to the same set of vectors on $\Omega(n)$ completely.

Corollary 12 *Numbers of four symmetric clusters can be expressed by*

Table 8 Numbers of four symmetric clusters in $1 \leq n \leq 16$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$C_E(n)$	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9
$C_L(n)$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$C_V(n)$	2	3	4	6	8	11	14	18	22	27	32	38	44	51	58	66
$C_G(n)$	2	3	4	6	8	14	20	36	60	108	188	352	632	1182	2192	4116

$$C_E(n) = \lfloor \frac{n}{2} \rfloor + 1;$$

$$C_L(n) = n + 1;$$

$$C_V(n) = \begin{cases} n^2/4 + 2, & n \equiv 0 \pmod{2} \\ (n^2 - 1)/4 + 2, & n \equiv 1 \pmod{2} \end{cases};$$

$$C_G(n) = \frac{1}{n} \sum_{k|n} \phi(k) 2^{\frac{n}{k}}.$$

Proof Due to Lemmas 4, 6, 8 and Theorem 1, four equations for numbers of various symmetric clusters are listed.

In convenient for comparison, their values on $1 \leq n \leq 16$ are listed in Table 8, respectively.

Checking real clusters in four schemes, the following corollaries can be provided.

Corollary 13 When $n = \{1, 2, 3\}$, three cluster schemes $C_L(n)$, $C_V(n)$, $C_G(n)$ provide the same partitions of clusters.

Proof Checking the three schemes, we have $C_L(1) = C_V(1) = C_G(1) = 2$, $C_L(2) = C_V(2) = C_G(2) = 3$, $C_L(3) = C_V(3) = C_G(3) = 4$. Relevant cluster contains the same set of vectors.

Corollary 14 When $n = \{1, 2, 3, 4, 5\}$, two cluster schemes $C_V(n)$, $C_G(n)$ provide the same partitions of clusters.

Proof Due to Corollary 13, we need to check $n = \{4, 5\}$ cases. For the two schemes, we have $(C_L(4) = 5) \neq (C_V(4) = C_G(4) = 6)$, $(C_L(5) = 6) \neq (C_V(5) = C_G(5) = 8)$. Relevant cluster contains the same set of vectors.

Corollary 15 When $n \geq 6$, four cluster schemes $C_E(n)$, $C_L(n)$, $C_V(n)$, $C_G(n)$ provide different partitions on their clusters.

Proof Due to Corollaries 13 and 14, we need to check $n = \{6, \dots\}$ cases. For the four schemes, $C_E(6) = 4$, $C_L(6) = 7$, $C_V(6) = 11$, $C_G(6) = 14$. Only a few clusters can contain the same set of vectors.

Corollary 16 *When $n \geq 6$, three cluster schemes: combinatorial, crossing and variant $\{C_E(n), C_L(n), C_V(n)\}$ may contain more symmetric properties than rotation clusters on $C_G(n)$.*

Proof Considering a special case on $\{n = 6, p = 3, q = 2\}$, $V(2, 3, 6) = \{001101, 011010, 110100, 101001, 010011, 100110, 011001, 110010, 100101, 001011, 010110, 101100\}$; this cluster contains two cycles: $\{001101, 011010, 110100, 101001, 010011, 100110\}$ and $\{011001, 110010, 100101, 001011, 010110, 101100\}$ with six vectors, respectively. Both cycles have rotation symmetries only without reflection symmetries. It is possible to use reflection symmetric operators to distinct two relative cycles to form a pure rotation symmetric structure. However, other clusters may contain more cycles such as $L(3, 6)$ with four cycles and $E(2, 6)$ with six cycles, respectively. It is necessary to apply other symmetric operators different from rotation for further separations.

4 Four Number Sets of Symmetric Clusters

4.1 Four Approximates on Numbers of Clusters

Using the four numeric equations, relevant approximates can be expressed as follows.

Lemma 13 *Four approximates can be expressed as*

$$C_E(n) \approx O\left(\frac{n}{2}\right); \quad (31)$$

$$C_L(n) \approx O(n); \quad (32)$$

$$C_V(n) \approx O\left(\frac{n^2}{4}\right); \quad (33)$$

$$C_G(n) \approx O\left(\frac{2^n}{n}\right). \quad (34)$$

Proof Using the four equations, the following approximates can be expressed:

$$C_E(n) = \lfloor \frac{n}{2} \rfloor + 1 \approx O\left(\frac{n}{2}\right);$$

$$C_L(n) = n + 1 \approx O(n);$$

$$C_V(n) = \begin{cases} n^2/4 + 2, & n \equiv 0 \pmod{2} \\ (n^2 - 1)/4 + 2, & n \equiv 1 \pmod{2} \end{cases} \approx O\left(\frac{n^2}{4}\right);$$

$$C_G(n) = \frac{1}{n} \sum_{k|n} \phi(k) 2^{\frac{n}{k}} \approx O\left(\frac{2^n}{n}\right).$$

4.2 Four Approximates on Numbers of Vectors

Definition 22 Let $f_X(n)$, $X \in \{L, E, V, G\}$ denote an approximate number of vectors in X cluster.

Lemma 14 Four approximates can be expressed as

$$f_E(n) \approx O\left(\frac{2^{n+1}}{n}\right); \quad (35)$$

$$f_L(n) \approx O\left(\frac{2^n}{n}\right); \quad (36)$$

$$f_V(n) \approx O\left(\frac{2^{n+2}}{n^2}\right); \quad (37)$$

$$f_G(n) \approx O(n). \quad (38)$$

Proof Since all clusters partition the same phase space $\Omega(n)$ with 2^n vectors, their approximates for vectors in a cluster can be evaluated,

$$f_E(n) = \frac{2^n}{O\left(\frac{n}{2}\right)} \approx O\left(\frac{2^{n+1}}{n}\right);$$

$$f_L(n) = \frac{2^n}{O(n)} \approx O\left(\frac{2^n}{n}\right);$$

$$f_V(n) = \frac{2^n}{O\left(\frac{n^2}{4}\right)} \approx O\left(\frac{2^{n+2}}{n^2}\right);$$

$$f_G(n) = \frac{2^n}{O\left(\frac{2^n}{n}\right)} \approx O(n).$$

It is convenient to list approximate numbers on clusters, vectors and dimension of measuring phase spaces in Table 9.

Table 9 Four approximate numbers on both clusters and vectors

X	$C_X(n)$	$f_X(n)$	Measuring phase space
E	$O\left(\frac{n}{2}\right)$	$O\left(\frac{2^{n+1}}{n}\right)$	1D
L	$O(n)$	$O\left(\frac{2^n}{n}\right)$	1D
V	$O\left(\frac{n^2}{4}\right)$	$O\left(\frac{2^{n+2}}{n^2}\right)$	2D
G	$O\left(\frac{2^n}{n}\right)$	$O(n)$	1D

5 Symmetric Boolean Functions for Selected Clusters

5.1 Four Numbers on Symmetric Boolean Functions

Definition 23 Let $SF_X(n)$ denote a number of Symmetric Boolean Functions (SBF) in $\{X(\cdot)\}$, $X \in \{E, L, V, G\}$.

Theorem 5 (Four types of symmetric Boolean functions) *Total numbers of four types of symmetric Boolean functions $SF_X(n)$, $X \in \{E, L, V, G\}$ are*

$$SF_E(n) = 2^{C_E(n)} = 2^{\lfloor \frac{n}{2} \rfloor + 1}; \quad (39)$$

$$SF_L(n) = 2^{C_L(n)} = 2^{n+1}; \quad (40)$$

$$SF_V(n) = 2^{C_V(n)} = \begin{cases} 2^{n^2/4+2}, & n \equiv 0 \pmod{2} \\ 2^{(n^2-1)/4+2}, & n \equiv 1 \pmod{2} \end{cases}; \quad (41)$$

$$SF_G(n) = 2^{C_G(n)} = O\left(2^{\frac{n}{n}}\right). \quad (42)$$

Proof For any selected cluster, there are two selections for its symmetric Boolean functions.

5.2 Four Numbers of Balanced Symmetric Clusters

Definition 24 Let $SF_{Xb}(n)$ be a maximal number of balanced SBF_X in $\{X(\cdot)\}$, $X \in \{L, V, G\}$, $n = 0 \pmod{2}$.

Definition 25 Let $SF_{Eb}(n)$ be a maximal number of balanced SBF_E in $\exists q$, $\{E(q, n)\}$, $n = 0 \pmod{4}$.

Lemma 15 *Four selected numbers $\{C_{Xb}(n)\}$, $X \in \{E, L, V, G\}$ for balanced symmetric clusters are*

$$C_{Eb}(n) = \begin{cases} 1, & n \equiv 0 \pmod{4} \\ 0, & n \not\equiv 0 \pmod{4} \end{cases}; \quad (43)$$

$$C_{Lb}(n) = 1; \quad (44)$$

$$C_{Vb}(n) = \frac{n}{2}; \quad (45)$$

$$C_{Gb}(n) = O\left(\frac{1}{n} \binom{n}{n/2}\right). \quad (46)$$

Proof From Corollary 10 for Eb groups $n \equiv 0 \pmod{4}$ cases, $q = n/4$ provides a cluster with a maximal number of vectors in a balanced condition and other cases cannot satisfy balanced conditions; for Lb groups $n \equiv 0 \pmod{2}$ cases, $p = n/2$

Table 10 Numbers of four balanced symmetric functions in $2 \leq n \leq 20$

n	2	4	6	8	10	12	14	16	18	20
$2^{C_{Eb}(n)}$	1	2	1	2	1	2	1	2	1	2
$2^{C_{Lb}(n)}$	2	2	2	2	2	2	2	2	2	2
$2^{C_{Vb}(n)}$	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
$2^{C_{Gb}(n)}$	2^1	2^2	2^4	2^{10}	$O(2^{25.2})$	$O(2^{77})$	$O(2^{245.1})$	$O(2^{804.3})$	$O(2^{2701.1})$	$O(2^{9237.8})$

provides a cluster with a maximal number of vectors in a balanced condition; for Vb groups $n \equiv 0 \pmod{2}$ cases, $p = n/2$, $1 \leq q \leq n/2$, there are $n/2$ clusters involved in a balanced condition; for Gb groups $n \equiv 0 \pmod{2}$ cases, $p = n/2$, a total of rotation symmetric clusters $O\left(\frac{1}{n} \binom{n}{n/2}\right)$ could be involved in a balanced condition.

5.3 Four Numbers of Balanced Symmetric Boolean Functions

Theorem 6 (Four balanced SYMMETRIC Boolean functions) *Total numbers of four balanced symmetric Boolean functions $\{SF_Xb(n)\}$, $X \in \{E, L, V, G\}$ are*

$$SF_{Eb}(n) = 2^{C_{Eb}(n)} = \begin{cases} 2, & n \equiv 0 \pmod{4}; \\ 1, & n \not\equiv 0 \pmod{4}; \end{cases} \quad (47)$$

$$SF_{Lb}(n) = 2^{C_{Lb}(n)} = 2; \quad (48)$$

$$SF_{Vb}(n) = 2^{C_{Vb}(n)} = 2^{\frac{n}{2}}; \quad (49)$$

$$SF_{Gb}(n) = 2^{C_{Gb}(n)} = O\left(2^{\frac{1}{n} \binom{n}{n/2}}\right). \quad (50)$$

Proof Each number of clusters in a selected scheme has been determined in Lemma 15. For any selected cluster in the scheme, there are two selections to form relevant symmetric Boolean functions.

In convenient for comparison, four types of SBF_{Xb} numbers on $2 \leq n \leq 20$ are listed in Table 10, respectively.

6 Cryptographic Properties of Symmetric Boolean Functions in Hierarchy

Boolean functions are of great importance in the design of random number generators for stream ciphers [25] that are widely used in modern network environment.

Due to cryptographically secure consideration, the sequence produced by the random number generator must satisfy the various properties [6, 8]: the longer period, the period complexity and good statistical distributions. There exists a huge theoretical knowledge of such combining generators [25].

A symmetric Boolean function must fulfil different necessary criteria to yield a cryptographically secure scheme, at least to resist known attacks [11]. In this direction, various measuring parameters play an important role such as balanced, support set, hamming weight, hamming distance, balanced function, non-linearity, correlation immunity, etc. [6, 8].

In relation to balanced properties, when n is even, the functions of highest non-linearity are the bent functions, and it is well known that the bent functions cannot be the balanced functions [28, 33]. From a structural viewpoint, the balanced functions having the highest possible non-linearity need to be considered. However, finding such functions is a very difficult problem [29, 31, 33]. When n is odd, exhibiting functions of the highest non-linearity is a hard problem in itself. Among the available candidates, balanced ones exist [16, 33].

To explore optimal functions in rotation symmetric Boolean function sets, many researchers are faced extremely difficulties on computational complexity even for $n > 10$ symmetric Boolean functions [29]. Exponentially increasing complexity makes a complex exhaustive search be quickly impossible. Compared with both variant and rotation schemes listed in Table 10, it is interesting to notice that the variant scheme takes a numeric complexity on $n = 20$ as same as the rotation symmetric scheme on $n = 10$. Much faster computation on optimal functions could be feasibly explored.

From a meta analytic viewpoint, measuring phase spaces provide multiple levels of construction in a hierarchy linked to various symmetric Boolean functions. They support an n tuple 0–1 vector construction as a word-based 0–1 vector to satisfy various design and analysis purposes. The variant PRNG construction [38, 43] is a similar approach to RC4 and HC128 stream ciphers [25] in their meta phase spaces using the word-oriented vector structure with the higher speed and efficiency. Measuring phase spaces could support advanced cryptographic applications on the direction.

Due to significant differences between measuring phase spaces proposed and algebraic normal forms classically formulated, in addition to initial balanced symmetric properties discussed in the chapter, other advanced comparison mechanisms need to be established for all interesting cryptographic properties to satisfy practical and optimal requirements for stream ciphers. Further detailed researches and explorations are required.

7 Conclusion

Symmetric clusters in a hierarchy provide the additional information to organize various symmetric Boolean functions into hierarchical constructions as multiple meta

levels of structures efficiently. The variant symmetric functions proposed in this chapter provide a meta construction on a 2D measuring phase space to contribute richer capacities compared with the three classical schemes (combinatorial, crossing and rotation) on 1D measuring phase spaces.

From a measuring viewpoint, three schemes (combinatorial, variant and rotation) in Tables 8, 9 and 10 have similar values in $n = \{1, 2, 3\}$ and $\{4, 5\}$ or different values in $n \geq 6$ conditions. The variant scheme provides a 2D intermediate structure different from other two schemes in 1D structure. From an approximate viewpoint, both combinatorial and rotation schemes are shown in stronger similar properties. Their approximate number of clusters and number of vectors in a cluster can be exchanged in Table 9. From an abstract system viewpoint, this pair of exchangeable measurements may provide approximate symmetric properties for both combinatorial and rotation schemes.

From a clustering viewpoint, the most important results are summarized in Theorem 4 to show that the four symmetric cluster schemes are different partition schemes on the same 0–1 vector set.

From a balanced analysis viewpoint, the key results of balanced symmetric Boolean functions are summarized in Theorem 6 and Table 10. This set of results provides a basic measurement to illustrate relevant computational difficulties to explore further optimal properties in balanced symmetric conditions. Different from other three schemes (combinatorial, crossing and rotation) in either very simpler or extremely complex associated with n increasing, balanced variant symmetric Boolean functions present very interesting patterns to support even $n \geq 20$ cases for future explorations.

Many advanced properties are existed to use a meta hierarchical construction to manage relevant measuring phase spaces into multilevels of a hierarchical structure. Various measuring parameters can be used as control parameters in detailed cases. Refined design and analysis can be performed under this meta hierarchy to provide powerful models and tools on design and optimization for future generations of stream ciphers.

References

1. E.B. Barker, A Statistical test suite for random and pseudorandom number generators for cryptographic applications, ITLB NIST (2000)
2. J.V. Bradley, *Distribution-free statistical tests* (Prentice-Hall 1968)
3. J. Carroll, The binary derivative test: noise filter, crypto aid, and random-number seed selector. *Simulation* **53**(3), 129–135 (1989)
4. P.J. Cameron, *Combinatorics: Topics, Techniques, Algorithms* (Cambridge University Press, Cambridge, 1994)
5. A. Canteaut, M. Videau, Symmetric boolean functions. *IEEE Trans. Inf. Theory* **51**(8), 2791–2811 (2005)
6. C. Carlet, On the degree, nonlinearity, algebraic thickness and nonnormality of boolean function, with developments on symmetric functions. *IEEE Trans. Inf. Theory* **50**(9), 2178–2185 (2004)

7. C. Carlet, K. Feng, An infinite class of balanced functions with optimal algebraic immunity, good immunity for fast algebraic attacks and good nonlinearity, in *ASIACRYPT* ed. by J. Pieprzyk, LNCS, vol. 5350 (Springer 2008), pp. 425–440
8. C. Carlet, G. Gao, W. Liu, A secondary construction and a transformation on rotation symmetric functions, and their action on bent and semi-bent functions. *J. Comb. Theory, Ser. A*, **127**, 161–175 (2014)
9. F.N. Castro, L.A. Medina, Linear recurrences and asymptotic behavior of exponential sums of symmetric boolean functions. *Electron. J. Combin.* **18**(2), P8 (2011)
10. J.R. Chen. *Combinatorial Mathematics* (Harbin Institute of Technology Press, 2012) (in Chinese)
11. T.W. Cusick, P. Štáňičá. Fast Evaluation, weights and nonlinearity of rotation-symmetric functions. *Discrete Mathe.* **258**(1-3), 289–301 (2002)
12. E. Filiol, C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation immunity, in *Eurocrypt 1998*, number 1403 in Lecture Notes in Computer Science, vol. 475488 (Springer-Verlag, 1998)
13. S.J. Fu, C. Li, L.J. Qu, On the number of rotation symmetric boolean functions. *Sci. China Inf. Sci.* **53**(3), 537–545 (2010)
14. H.W. Gould, Some generalizations of vandermonde’s convolution. *Am. Math. Mon.* **63**(2), 84–91 (1956)
15. H.W. Gould. *Combinatorial Identities* (Morganton, 1972)
16. Y.M. Guo, G.P. Gao, Y.Q. Zhao. Recent results on balanced symmetric boolean functions, available: <http://eprint.iacr.org/2012/093> (2012)
17. G. Gao, X. Zhang, W. Liu, C. Carlet, Constructions of quadratic and cubic rotation symmetric bent functions. *IEEE Trans. Inf. Theory* **58**(7), 4908–4913 (2012)
18. M. Hall, *Combinatorial Theory*, 2nd edn. (Blaisdell, 1986)
19. L.K. Hua, *Loo-Keng Hua Selected Papers* (Springer, 1982)
20. S. Kavut, S. Maitra, M.D. Ycel, Search for boolean functions with excellent profiles in the rotation symmetric class. *IEEE Trans. Inf. Theory* **53**(5), 1743–1751 (2007)
21. D.E. Knuth. *The Art of Computer Programming*, vol. 1, 3rd edn. (Addison-Wesley, 1998)
22. D.E. Knuth, *The Art of Computer Programming, A: Combinatorial Algorithms*, Part 1, vol. 4 (Addison-Wesley, 2011)
23. B. Logan Jr., Information in the zero crossings of bandpass signals. *Bell Syst. Tech. J.* **56**, 487–510 (1977)
24. Q. Meng, L. Chen, F. Fu, On homogeneous rotation symmetric bent functions. *Discr. Appl. Math.* **158**(10), 1111–1117 (2010)
25. G. Paul, S. Maitra. *RC4 Stream Cipher and Its Variants* (CRC Press, 2012)
26. J. Pieprzyk, C.X. Qu, Fast hashing and rotation-symmetric functions. *J. Universal Comput. Sci.* **5**(1), 20–31 (1999)
27. L. Qu, C. Li, K. Feng, A note on symmetric boolean functions with maximum algebraic immunity in odd number of variables. *IEEE Trans. IT-53*, 2908–2910 (2007)
28. Sarkar, P., Maitra, S, Construction of nonlinear Boolean functions with important cryptographic properties, in *Advances in Cryptology EUROCRYPT 2000*, vol. 1807 in LNCS (Springer Verlag, 2000), pp. 485–506
29. P. Štáňičá, S. Maitra, Rotation symmetric boolean functions - count and cryptographic properties, *Discr. Appl. Math.* **156**, 1567–1580 (2008)
30. R.P. Stanley, *Enumerative Combinatorics*, Vol. 1, 2nd edn. (Cambridge University Press, 1997)
31. W. Su, X.H. Tang, A. Pott, A note on a conjecture for balanced elementary symmetric boolean functions. *IEEE Trans. Inf. Theory* **59**(1), 665–671 (2013)
32. S.H. Su, X.H. Tang. Construction of rotation symmetric boolean functions with optimal algebraic immunity and high nonlinearity. *Des. Codes Cryptography* **71**(2), 183–199 (2014)
33. S.H. Su, X.H. Tang, On the systematic constructions of rotation symmetric bent functions with any possible algebraic degrees. *IACR Cryptology ePrint Archive* **2015**, 451 (2015)
34. G.Z. Tu, *Combinatorial Enumeration Methods & Applications* (Science Press, 1981) (in Chinese)

35. A. Tucker, *Applied Combinatorics* (Wiley, 2007)
36. J.H. van Lint, R.M. Wilson, *A Course in Combinatorics*, 2nd edn. (Cambridge University Press, 2001)
37. H. Wang, J. Zheng, 3D Visual Method of Variant Logic Construction for Random Sequence. Australian Information Warfare and Security, pp. 16–27 (2013)
38. W.Z. Yang, J. Zheng, Variant pseudo-random number generator, Hakin9 extra. Timing Attack **06**(13), 28–31 (2012)
39. Z.J. Zheng, A. Maeder, The conjugate classification of the kernel form of the hexagonal grid, in *Modern Geometric Computing for Visualization* (Springer-Verlag, 1992) pp. 73–89. http://link.springer.com/chapter/10.1007/978-4-431-68207-3_5 e-version
40. Z.J. Zheng. *Conjugate Transformation of Regular Plan Lattices for Binary Images*, Ph.D. Thesis, Monash University, 1994
41. J.Z.J. Zheng, C.H.H. Zheng, A framework to express variant and invariant functional spaces for binary logic, *Frontiers of Electrical and Electronic Engineering in China*, **5**(2), 163–172, Higher Educational Press and Springer-Verlag, 2010. <http://link.springer.com/article/10.1007%2Fs11460-010-0011-4>, <https://doi.org/10.1007/s11460-010-0011-4>
42. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, A framework of variant logic construction for cellular automata, *Cellular Automata - Innovative Modeling for Science and Engineering*, ed by A. Salcido (InTech Press, 2011). <http://www.intechopen.com/books/cellular-automata-innovative-modelling-for-science-and-engineering/a-framework-of-variant-logic-construction-for-cellular-automata>, <https://doi.org/10.5772/15400>
43. J. Zheng, Novel pseudo-random number generation using variant logic framework, in *2nd International Cyber Resilience Conference*, pp. 100–104, 2011. <http://igneous.scis.ecu.edu.au/proceedings/2011/icr/zheng.pdf>
44. J. Zheng, C. Zheng, Variant simulation system using quaternion structure. *J. Modern Opt.* Taylor & Francis Press **59**(5), 484–492 (2012)
45. J. Zheng, C. Zheng, T.L. Kunii, From conditional probability measurements to global matrix representations on variant construction, in *Advanced Topics in Measurements* (InTech Press, 2012), pp. 339–370
46. J. Zheng, C. Zheng, T.L. Kunii. From Local Interactive Measurements to Global Matrix Representations on Variant Construction, in *Advanced Topics in Measurements* (InTech Press, 2012), pp. 371–400
47. J. Zheng, C. Zheng, T.L. Kunii, Interactive maps on variant phase space, in *Emerging Application of Cellular Automata* (InTech Press, 2013), pp. 113–196
48. J. Zheng, W. Zhang, J. Luo, W. Zhou, R. Shen, Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Math.* **3**(7A), 5–24 (2013)
49. J. Zheng, J. Luo, W. Zhou, Pseudo DNA sequence generation of non-coding distributions using variant maps on cellular automata. *Appl. Math.* **5**(1), 153–174 (2014)
50. J. Zheng, W. Zhang, J. Luo, W. Zhou, V. Liesaputra, Variant map construction to detect symmetric properties of genomes on 2D distributions. *J. Data Mining Genomics Proteomics* **5**, 150 (2014). <https://doi.org/10.4172/2153-0602.1000150>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part III

Theoretical Foundation—Variant Map

Arc, amplitude, and curvature sustain a similar relation to each other as time, motion, and velocity, or as volume, mass, and density.

—*Carl Friedrich Gauss*

As long as algebra and geometry have been separated, their progress have been slow and their uses limited; but when these two sciences have been united, they have lent each mutual forces, and have marched together towards perfection.

—*Joseph-Louis Lagrange*

The arithmetical symbols are written diagrams and the geometrical figures are graphic formulas.

—*David Hilbert*

In relation to variant map, a longer book chapter (Chapter “Interactive Maps on Variant Phase Spaces”) was published in the OA book of Emerging Application of Cellular Automata: 113–196 (2013) by InTech Press. This provides systematic approaches under statistical mechanics in comparison. Possible projections and their mapping mechanisms are explored.

Part III is composed of three chapters (6–8).

Chapter “[Variant Maps of Elementary Equations](#)” provides variant maps of elementary equation to generate visual distributions using two cases of combinatorial expressions. From two cases, it is interesting to see symmetric distributions under various parameters and complex distributions are created by control parameters shown in 2D and 3D distributions and their projections.

Chapter “[Variant Map System of Random Sequences](#)” describes variant map system of random sequences; five types of maps are defined and proposed on two types of 1D maps and three types of 2D maps. A sample sequences from the AES cipher is selected and multiple maps are illustrated.

Chapter “[Stationary Randomness of Three Types of Six Random Sequences on Variant Maps](#)” proposes a testing system for stationary randomness of random

sequences on variant maps. Three types of six random sequences are selected. Six samples are composed of three random resources: two block ciphers, two stream ciphers, and two quantum ciphers. Three variation categories are observed.

Variant Maps of Elementary Equations



Jeffrey Zheng

Abstract Using four measures in Type B, there are 11 invariant expressions to form elementary equations of variant measurement. In this chapter, two invariant expressions are selected to illustrate sample procedures from elementary equations to relevant variant maps. Using various projections and multiple levels of representations, complicated binomial coefficients and their variations are illustrated under various conditions. Using multinomial coefficients, multiple viewpoints are used for references. Due to this type of variation framework contains rich structures, further explorations are required from multiple levels on both theoretical foundation and practical applications.

Keywords Variant measurement · Elementary equation · Variant map
Multinomial coefficient · Coefficient array

1 Introduction

Variant construction starts from n 0–1 variables to form 2^n states and 2^{2^n} functions, via vector permutation and complement operations on state space to establish a variant logic framework to contain $2^n! \times 2^{2^n}$ configurations as a variation space. Variant measurement acts as a core of quantitative measurement, starting from m 0–1 variables to explore relevant clustering conditions on 2^m states. Since this type of variations has a close relationship to partition and recombination using binomial and multinomial coefficients under identically combinatorial expressions. Apply-

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_6

ing the results in Chapter “Elementary Equations of Variant Measurement”, Type B measures are composed of 11 nontrivial invariants. Two invariants are selected in this chapter, their different partition properties are illustrated to use coefficients on 2D and 3D distributions. Variant maps are generated from coefficient arrays as samples.

2 Measures and Maps

Two combinatorial invariants are selected: $\{m - p\}\{p\}$ and $\{2q\}\{m - 2q\}$. Different distributions on their coefficients are explored.

2.1 Case 1. $\{m - p\}\{p\}$

For $\{m - p\}\{p\}$ formula, relevant equation is

$$\binom{m}{p} = \sum_{k=0}^p \binom{m-p}{k} \binom{p}{k} \tag{1}$$

A binomial coefficient is separated by sum of $(p + 1)$ pairs of binomial coefficient products. For a selected value p , coefficients $\{\binom{m-p}{k} \binom{p}{k}\}, 0 \leq k \leq p$ are arranged in a linear order.

This property is true for all p values. A special three tuple structure (m, p, k) has 1–1 correspondence with a coefficient $f(m, p, k) = \binom{m-p}{k} \binom{p}{k}$. While m value increased, coefficient array will be increased as a 3D rectangular steps, each m value has a $(m + 1)^2$ region.

The nontrivial coefficients are distributed as a triangle. Let $F(m, p) = \sum_{\forall k} f(m, p, k), 0 \leq p \leq m$ and $G(m, k) = \sum_{\forall p} f(m, p, k), 0 \leq k \leq m$, two projections $\{F(m, p), G(m, k)\}$ can be projected. Coefficients and relevant four maps are shown in Fig. 1.

Lemma 1 For $\{m - p\}\{p\}$ equation, coefficients are distributed in $(m + 1)^2$ and all nontrivial coefficients are clustered in 1/4 region and 3/4 regions has coefficient 0.

2.2 Case 2. $\{2q\}\{m - 2q\}$

Briefly $\{m - p\}\{p\}$ and $\{2q\}\{m - 2q\}$ are simple invariants. For $\{2q\}\{m - 2q\}$ invariant, it has the following equation.

$f(10, p, k) = (.)$	0	1	2	3	4	5	6	7	8	9	10	p	$G(10, k) = \sum_{\forall p} (.)$
0	1	1	1	1	1	1	1	1	1	1	1	1	11
1		9	16	21	24	25	24	21	16	9			165
2			28	63	90	100	90	63	28				462
3				35	80	100	80	35					330
4					15	25	15						55
5						1							1
6													
7													
8													
9													
10													
k													
$F(10, p) = \sum_{\forall k} (.)$	1	10	45	120	210	252	210	120	45	10	1		$\sum_{\forall p, k} (.) = 1024 = 2^{10}$

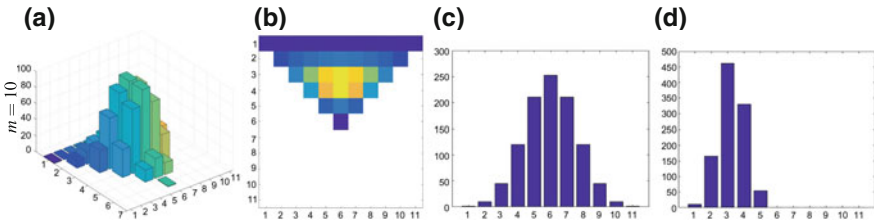


Fig. 1 One set of coefficients and its two projections in four maps (a)–(d); **a** 3D $f(10, p, k)$; **b** 2D $f(10, p, k)$; **c** 1D $F(10, p)$; **d** 1D $G(10, k)$

$$\binom{m}{p} = \sum_{k=0}^p \binom{2q}{k} \binom{m-2q}{p-k} \tag{2}$$

where q is a free variable, $0 \leq q \leq \lfloor m/2 \rfloor$. Different from Case 1, this equation can determine $\lfloor \text{floorm}/2 \rfloor + 1$ levels of coefficients according to different q values selected to form a 3D coefficient structure.

Let $f(m, q, p, k) = \binom{2q}{k} \binom{m-2q}{p-k}$ under $0 \leq q \leq \lfloor m/2 \rfloor, 0 \leq k, p \leq m$ conditions, nontrivial coefficients are distributed in special shapes on multiple 2D regions.

Using color coding scheme, it is feasible to map coefficients into greyscale or color pixels as variant maps.

A binomial coefficient can be separated as sum of $(p + 1)$ pairs of coefficient products $\{\binom{2q}{k} \binom{m-2q}{p-k}\}, 0 \leq k \leq p$ to be a linear order.

This type of property is true for all p values, a special tuple of four parameters (m, q, p, k) has 1–1 correspondence with coefficient $\binom{2q}{k} \binom{m-2q}{p-k}$. Each selected m value is corresponding to $(m + 1)^2 \times (\lfloor m/2 \rfloor + 1)$ region to locate all coefficients.

Lemma 2 For $\{2q\}\{m - 2q\}$ combinatorial invariant, all coefficients are restricted in $(m + 1)^2 \times (\lfloor m/2 \rfloor + 1)$ region.

3 Visual Results

It is convenient to use color coding to transfer each coefficient as a pixel in a variant map. Invariant coefficients provide ideal conditions for a practical measurement, it is feasible to check physical differences between an idea distribution and a practical measurement.

From a quantitative viewpoint, multinomial expressions provide proper basis on corresponding partitions to be a relative measurement in representation.

3.1 Case 1. Maps

Using $\binom{m}{p} \rightarrow \{\binom{m-p}{k}\binom{p}{k}\}$, three maps are shown in Fig. 1 as 2D coefficients, 3D histograms, and 2D projections on four parameters $m = \{10, 11, 15, 16\}$, respectively.

3.2 Case 2. Maps

Different from Case 1, each m is associated with one 2D coefficient. In $\binom{m}{p} \rightarrow \{\binom{2q}{k}\binom{m-2q}{p-k}\}$ conditions, each q selection determines a 2D array of coefficients. Under $0 \leq q \leq \lfloor m/2 \rfloor$ conditions, $\lfloor m/2 \rfloor + 1$ levels are required. For $m = 10$, it is necessary to have 6 levels.

To observe global properties, a 3D color map is shown in Fig. 3 to illustrate 3D coefficients under color coding.

4 Result Analysis

In maps of Figs. 1, 2, and 3, it is convenient to see variant maps transformed from elementary equations. From a certain viewpoint, $\{m - p\}\{p\}$ coefficients have symmetric properties on horizontal direction on $p : m - p$ with reflective properties. Nontrivial coefficients are located in 1/4 region of $(m + 1)^2$ square. An isosceles triangle is composed of all nontrivial coefficients. Selecting any m , there is only one 2D coefficient associated with to be a unified distribution.

$\{2q\}\{m - 2q\}$ coefficients are corresponding to multiple 2D distributions under various q values. While $q = 0$, each nontrivial coefficient is located on diagonal position of $p = k$ and each coefficient is a $\binom{2q}{k}\binom{m-2q}{p-k}$ equation. In $0 \leq q \leq 5$ conditions, 2D coefficient matrices are shown in six groups of $\{0 : 10, 2 : 8, 4 : 6, 6 :$

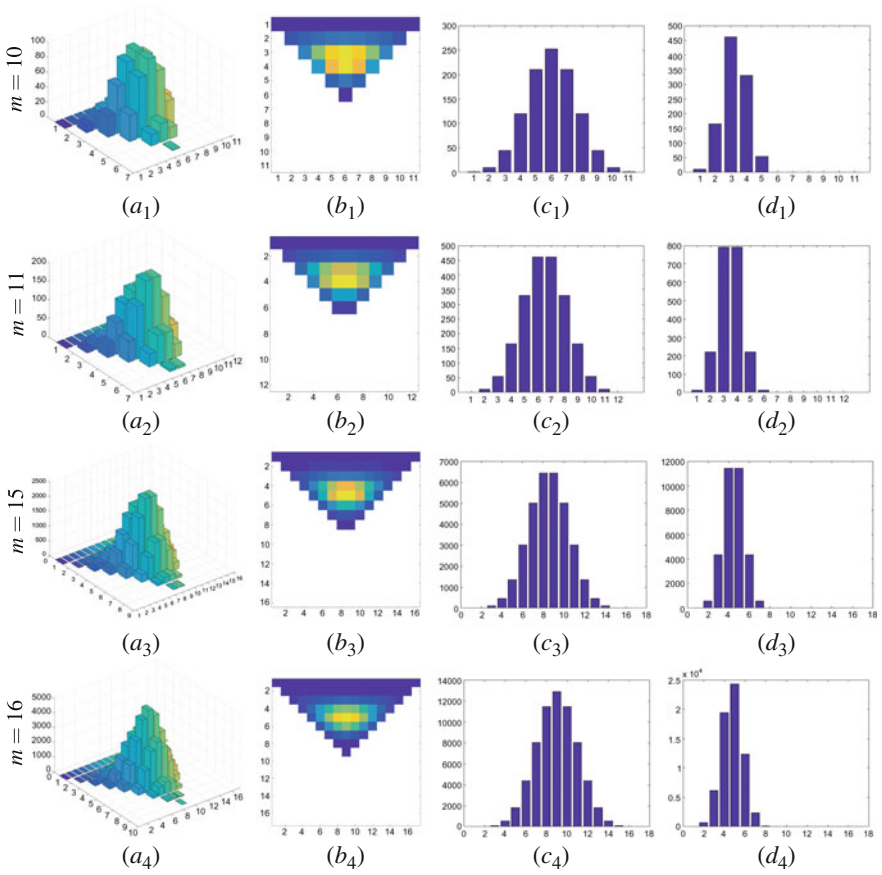


Fig. 2 $\{m - p\}\{p\}$ maps: $m = \{10, 11, 15, 16\}$; $(a_1) - (d_1)$ $m = 10$; $(a_2) - (d_2)$ $m = 11$; $(a_3) - (d_3)$ $m = 15$; $(a_4) - (d_4)$ $m = 16$

4, 8 : 2, 10 : 0}, this can be described as $(x + y)^{n+l} = (x + y)^n(x + y)^l$ coefficient distributions that can be illustrated in Fig. 2 $\{(a_0) - (c_0)\} - \{(a_5) - (c_5)\}$ maps.

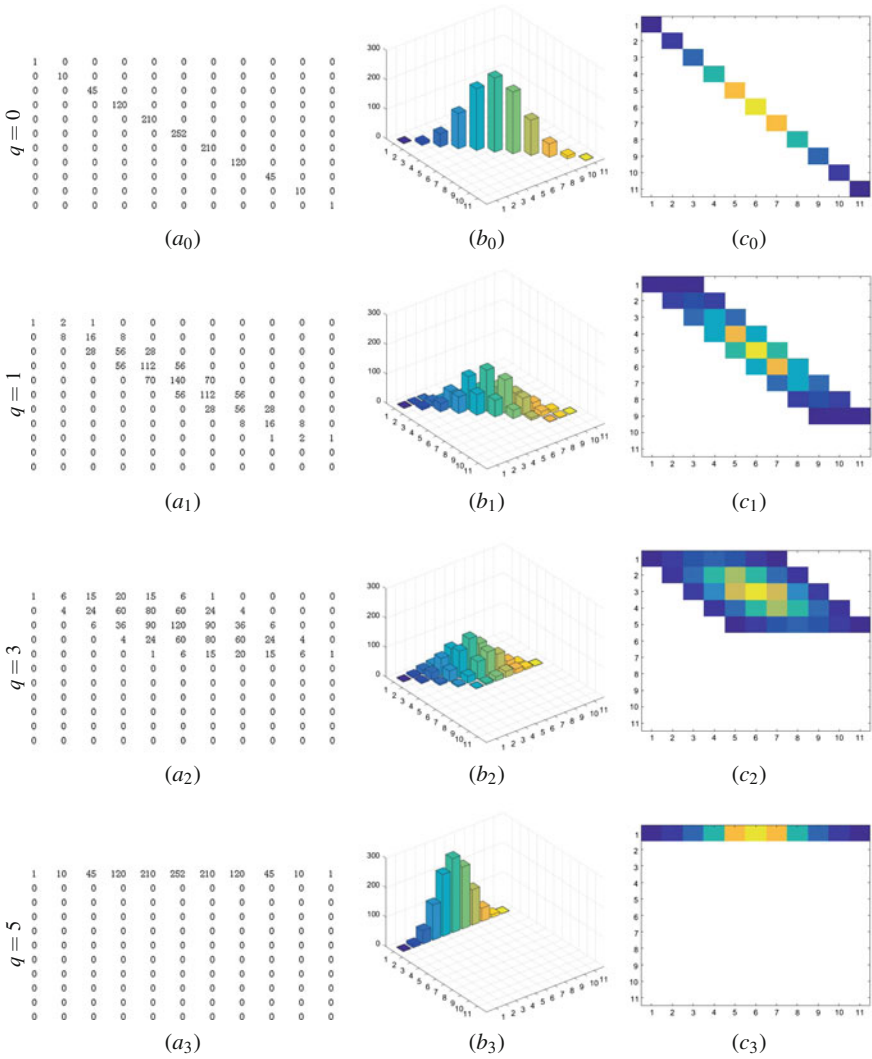
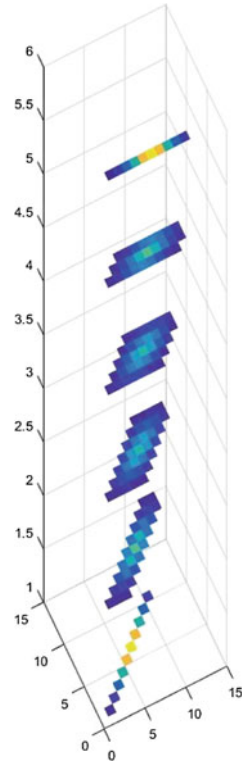


Fig. 3 $\{2q\}\{m-2q\}$ maps: $m = 10$; $(a_0)-(c_0)q = 0$; $(a_1)-(c_1)q = 1$; $(a_2)-(c_2)q = 2$; $(a_3)-(c_3)q = 3$; $(a_4)-(c_4)q = 4$; $(a_5)-(c_5)q = 5$

Fig. 4 $\{2q\}\{m - 2q\}$
 map: $m = 10$; 3D color map



5 Conclusion

It is a new exploration to use elementary equation to illustrate relevant variant maps. Based on the described model and calculation, it is convenient to do various analysis and visualization. It is an initial step to check two invariants from Type B for four variant measures. Further explorations are required on five levels of 11 nontrivial invariants in Type B. From results in this chapter, distinct distributions are observed on the two selected invariants. Other nine invariants in Type B will be discussed in future papers (Fig. 4).

Acknowledgements The author would like to thank Yifeng Zheng and Kaiyu Yang for generating binomial coefficients in different conditions and Dr. Dennis Heim for correction of the chapter.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Variant Map System of Random Sequences



Jeffrey Zheng

Abstract Sequences of random variables play a key role in probability theory, stochastic processes, and statistics to analyze dynamic behavior. Speckle patterns have emerged as useful tools to explore space–time variations of random sequences in various measurement applications of comprehensive properties in complex space–time variation events. In this chapter, a variant map system is proposed to analyze statistical properties of random sequences in visual representations. An input 0–1 sequence will be divided into multiple segments and each segment of a fixed length will be transformed into a 2-tuple pair of measures. Five measuring sets are identified and rearranged in a 1D or 2D numerical array as a histogram representing a visual map. These five types of maps consist of two types in 1D format as classical maps and three types in 2D format as variant maps. Properties are analyzed on all five types of maps. A cryptographic sequence of the AES cipher is selected as a sample stream. The five types of visual maps are generated and refined clustering characteristics are organized into four groups on changes of segmented and shifted lengths for visual comparisons on enlarged 2DP maps. Speckle patterns of various distributions are observed. Three variant maps with distinct statistic distributions could be useful to provide new visual tools to explore comprehensive cryptographic sequences on complex nonlinear dynamic behavior in global network environments.

Keywords Variant map · Visual representation · Multiple segment · Statistical probability distribution · Clustering characteristics

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_7

1 Introduction

Associated with network communication and internet technology [1] in global applications, web communication, internet of things, cloud computing, big data, mobile phone, and smart wireless technologies [2] are significantly developed in the last decade and widely adapted over the world market. In the current situation, it is a key issue for cryptographic researchers and applications [3] to use advanced technologies of stream ciphers to protect data security of ultrafast and extra-big data streams in global network environments.

1.1 Pseudo-Random Sequences

1.1.1 From Linear Stream Ciphers

Traditional stream ciphers [4] on LFSR Linear Feedback Shift Register structure (in military cryptography) are used as pseudo-random number generators, due to the ease of implementation from simple hardware, long periods, and uniformly distributed streams. The LFSR stream ciphers are the core in classical stream ciphers through the mathematical theory of algebraic functions for system simulation and analysis.

However, an LFSR is a linear system leading to fairly easy cryptanalysis using the Berlekamp–Massey algorithm. Important LFSR-based stream ciphers use A5/1 & A5/2 in GSM cell phones and E0 in Bluetooth. But the A5/2 cipher has been broken and both A5/1 and E0 have serious weaknesses [5, 6].

1.1.2 From Nonlinear Stream Ciphers

The new generation of stream ciphers [7, 8] are widely used in advanced web communications. Three general methods are applied to improve security weaknesses in LFSR-based stream ciphers:

1. **Nonlinear Functions:** Nonlinear combination of several bits from the LFSR state [9].
2. **Nonlinear Parts:** Nonlinear combination of the output bits of two or more LFSRs or using Evolutionary algorithm for nonlinearity [10].
3. **Clock Control:** Irregular clocking of the LFSR, as in the alternating step generator [11].

With batch, a series of nonlinear algorithms have emerged [12]: nonlinear equivalence [13], evolutionary methods [10], AES cipher [14], RC4 [15], ZUC [9], cellular automata [16], and nonlinear dynamic system [17].

The new generation of stream ciphers are being shifted from the traditional mode: LFSR [4] to various nonlinear modes: NLFSR [18, 19], clock control [11], nonlinear

functions [9] etc., it is essential for ciphers to be integrated and implemented [20] to satisfy security models. However, different from LFSR with well-established linear mathematical theories and simulation tools, it is extremely difficult to use advanced nonlinear mathematical theories, recursive models, descriptive tools, and implementing schemes [17] in nonlinear dynamic environments.

How to evaluate cryptographic sequences generated from the nonlinear stream ciphers is an urgent problem for modern stream ciphers.

1.2 Truly Random Sequences from Hardware Devices and Speckle Patterns

In addition to pseudo-random sequences generated by stream ciphers, high-quality stochastic oscillators of truly random sequences are generated from special hardware devices such as laser photonics [21], nonlinear optics [22], quantum optics [23], quantum noises [24], thermal noise [25], chaos, and fractal nonlinear dynamics [26].

A list of truly random number generators are developed to extract stochastic information from speckle patterns [27], i.e., random bits from turbulence [28] to get random numbers from the speckle positions, generation of random arrays using laser speckle [29], 2D generation of random numbers by multimode fiber speckle [30], Markov speckle for efficient random bit generation [31] and dynamic laser speckle and applications [11].

Since various truly random sequences are created from specific physical models with special principles and uncertain methodologies, it is extremely difficult for cryptographic researchers to make proper measurements explore nonlinear dynamic properties.

1.3 Statistic Testing Packages on Cryptographic Sequences

Randomness has been explored for many years [32] on a series of statistic testing theories and methods. The NIST 800-22 testing package [33] is an effective statistic package on random sequences collecting a set of 16 statistic testing schemes in evaluations of statistic properties on cryptographic sequences. Statistic testing packages are very useful to catch a list of quantitative measurements evaluating randomness properties of cryptographic sequences in wider applications. However, testing schemes in various packages are mainly focused on P-value or a list of static properties of a testing sequence.

Since comprehensive behaviors in nonlinear dynamics may increase computational complexities tragically to involve complicated dynamic properties in the multivariate environment, those dynamic behaviors are completely ignored.

1.4 Gaussian Distribution and Speckle Pattern

Multivariate normal probability distribution models are the most important and powerful tools that are used to test stochastic characteristics of a random data sequence [34] under the framework of probability, stochastic process, and statistics [35] for nonlinear problems. In this kind of measuring models, when the data sequence is sufficiently long, the high-dimensional probability distribution of the sequence [36] is similar to the continuous Gaussian distribution.

A typical projection model is shown in Fig. 1a; the central part shows a Gaussian surface with an unbalanced distribution in a 2D plane distributed as $P(X, Y)$ measures with pseudo-colors and its two 1D projections shown in both horizontal $P(X)$ and vertical $P(Y)$ planes, respectively. In Fig. 1b, a standard Gaussian surface with symmetric shapes is illustrated and the 2D projection of its pseudo-color map is shown in Fig. 1c with an ideal continuous distribution of color on the map. Different from ideally continuous distributions, in Fig. 1d, a real image generated from the Laser speckle phenomena [37] is illustrated as an objective speckle pattern [38] scattered by a laser beam from a plastic surface onto a wall. It is convenient for us to compare different color maps in Fig. 1c, d, respectively.

From these set of figures, the relationship between the projection curve and two 1D Gaussian distributions can be observed in the multivariate normal probability environment. Multivariate Gaussian probability distributions may support classical schemes to analyze complex stochastic data sets of measuring sequences in many applications in continuous conditions. But speckle patterns in Fig. 1d provide intrinsically discrete random patterns that may not be easily simulated by smoothed Gaussian map in Fig. 1c, further exploration on proper simulation and control mechanisms are required.

1.5 Controlling Deterministic Chaos

Controlling deterministic chaos has been an active R&D field in nonlinear dynamics over the past decades. From the pioneering work, significant progress has been achieved in control spatiotemporal chaos [39], plasma device, laser systems [40], chemical reactions, and biological systems both spatial and temporal dependence considered. The complex Ginzburg–Landau equation (CGLE) system [41] describes universal dynamics features near a supercritical Hopf bifurcation. It exhibits defected mediate turbulence or spiral turbulence in a wide parameter region. The control by generating a spiral wave seed has been described [42, 43] to grow into a stable spiral in the CGLE system.

Systematic approaches on simulation of nonlinear behaviors, speckle phenomena in optics [37] and pattern dynamics [44] have been actively explored.

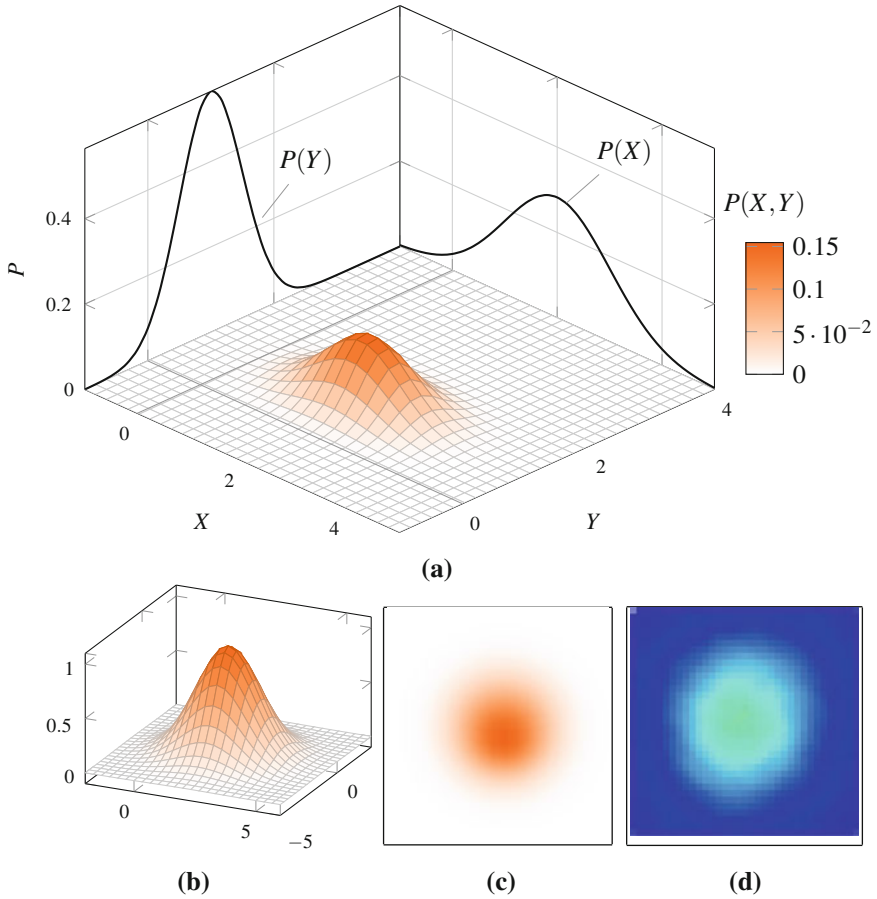


Fig. 1 Multivariate Gaussian Probability Distributions and an objective speckle pattern; **a** Bivariate normal distribution with two probability projections; **b** A symmetric bivariate normal surface with pseudo-colors; **c** A 2D pseudo-color map of the symmetric bivariate normal surface; **d** An objective speckle pattern scattered by a laser beam from a plastic surface onto a wall. [38]

1.6 Poincaré Map

From a measuring viewpoint, spatial variations of a stochastic sequence will be changed by overall macro characteristics showing statistic measurements of distributed patterns [45] in a vector space, so that a random sequence is measured by an analytic space. From an analysis viewpoint, the Poincaré section [46] corresponds to a discrete map proposed by the eminent French scientist Henri Poincaré 100 years ago.

The Poincaré map handles additional information from sequential changes of ordered measurements in the phase space of classical dynamics, nonlinear dynamic systems [47] and chaos.

The mapping mechanism of the Poincaré map may be useful to handle dynamic patterns on cryptographic sequences of stream ciphers. This mapping scheme has been applied to observe the global randomness of cellular automata sequences on 2D maps [48] 20 years ago.

1.7 Variant Framework

Various schemes following the top-down strategy are explored to use multiple measures to partition special phase spaces from a top state set to multiple bottom states via multi-levels of a hierarchy in combinatorial algorithms [49], image analysis and processing for many years.

The conjugate classification [50] is proposed to apply seven measures in a hierarchy to partition the kernels of four regular plane lattices on $n = \{4, 5, 7, 9\}$ cases for 2D binary images. For 1D cellular automata sequences, global random behaviors [48] are visualized in 2D maps.

For n -tuple bit vectors, the variant logic framework [51] was proposed and various applications were explored: 3D visual method on random number sequences [52], variant Pseudo-Random Number Generator PRNG [53, 54], computational simulation on quantum interactions [55, 56], noncoding DNA analysis [57] and bat echolocation [58].

1.8 Proposed Scheme

For the purpose of system characterization based on comprehensive measurements of cryptographic sequences, we propose a variant map system for a 0–1 stochastic sequence with length N . Multiple segments M are divided from the sequence by a given length m . A 2-tuple pair of measures can be extracted from a 0–1 segment that is the number of a single element and the number of 01 patterns in the segment. All paired measures are composed of a sequence of M pairs of measures as an ordered measuring set with M elements.

The pairs of the measuring sequence are directly separated into two independent measuring sequences to keep each parameter in the same order. Applying the pairing scheme of the Poincaré section, one single measuring sequence can be reorganized by two consequent measures as a 2-tuple pair of measures. Two measuring sequences in the Poincaré section and the original pairs of measuring sequence are arranged as the three sequences of 2-tuple measures. So a total of five sequences of distinct measures are constructed including two sequences on single measures and three sequences on 2-tuple measures.

Following this approach, two sets of single measuring sequences are sorted as two 1D numerical arrays as statistical histograms being classic 1D maps and three sets of 2-tuple measuring sequences are sorted as three 2D integer arrays as statistic histograms being three variant maps. Under the controlling operations on the changes of the segment lengths and shift displacements, multiple results of the five measuring sequences are transformed into 1D statistic histograms and 2D pseudo-color maps to show effective speckle patterns from the selected cryptographic sequence under various conditions of the combination on the two controlling parameters.

1.9 Organization of the Chapter

This chapter describes the variant map system in diagrams of the system architecture and the core modules with input/output and processing functions in Sect. 2. In Sect. 3, the relationships among measuring sequences and the five statistical distribution maps are analyzed. In Sect. 4, an AES cipher sequence is selected to form a series of statistical maps based on changes of the two control parameters. From the results of the visual maps in Sect. 4, intuitive analysis and brief comparisons are carried out in Sect. 5. Finally, in Sect. 6, the main results are summarized.

2 Framework of Variant Map System

2.1 Framework

For the variant map system, the block diagrams of the system framework and the core modules of the system are shown in Fig. 2. The framework of the system architecture in Fig. 2a is composed of three core modules: the Shift Segment Measurement SSM, the Measuring Sequence Combination MSC, and the Projective Color Map PCM. The three modules are shown in Fig. 2b–d in more detail, respectively.

2.2 Shift Segment Measurement SSM

The SSM module is shown in Fig. 2b.

Let X be a 0–1 vector with N elements as an input sequence,

$$X = X[0]X[1] \cdots X[I] \cdots X[N - 1], 0 \leq I < N; X[I] \in \{0, 1\} \quad (1)$$

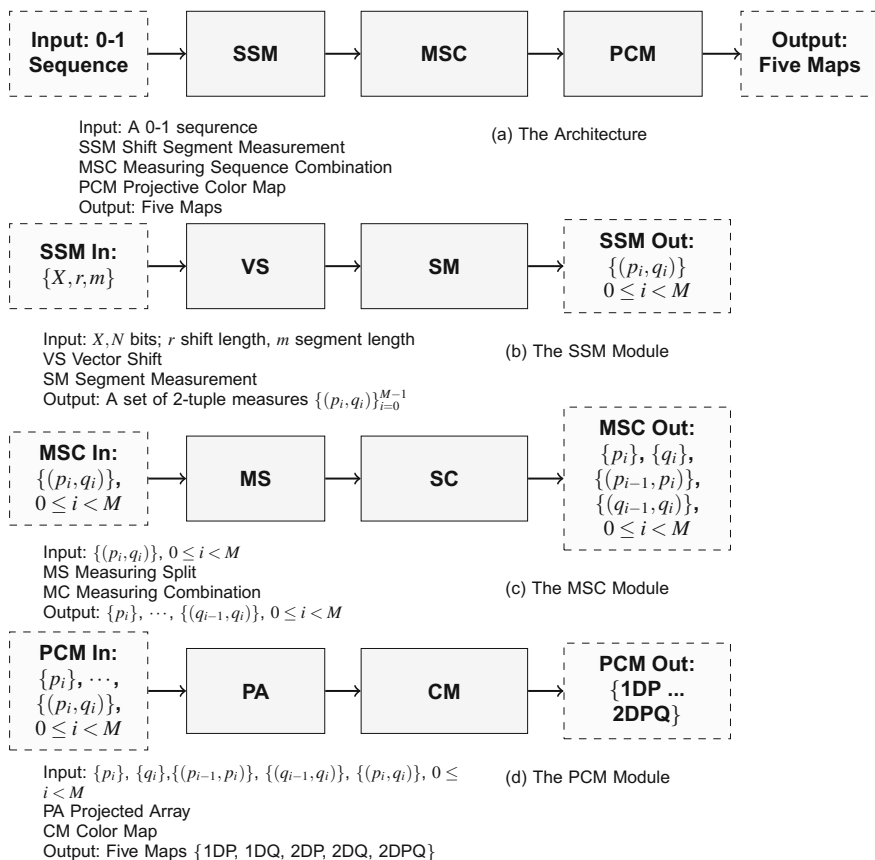


Fig. 2 The framework of the variant map system for cryptographic sequences; **a** The system architecture; **b** The SSM module; **c** The MSC module; **d** The PCM module

The SSM module consists of two processing units: the Vector Shift VS and the Segment Measurement SM, respectively. The two input control parameters: $\{r, m\}$ are defined as shift length r and segment length m .

Let Y be a 0–1 vector with N elements, this vector is generated by the shift operation under the loop displacement condition from the input sequence (i.e., a cyclic shift right + or shift left –)

$$Y = X(r), Y[I] = X[I \pm r], I \pm r \pmod{N}, 0 \leq I < N; X[I], Y[I] \in \{0, 1\} \quad (2)$$

The shifted vector is inputted into the SM unit for a segmentation process. The input sequence will be divided from a long sequence with N elements into $M = \lfloor N/m \rfloor$ segments as a set of sub-vectors with m elements and each segment

contains m bits. The i -th sub-vector $0 \leq i < M$ on the j -th position $0 \leq j < m$ is denoted as $Y_{i,j}$.

This sequence of sub-vectors after the segmenting operation forms the following $m \times M$ matrix, m positions for the i -th complete row vector in the sequence correspond to a pair of 2-tuple measures: (p_i, q_i) , and incomplete parts of the last sub-vector are ignored.

$$Y = \begin{bmatrix} Y_{0,0} & Y_{0,1} & \cdots & Y_{0,j} & \cdots & Y_{0,m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{i,0} & Y_{i,1} & \cdots & Y_{i,j} & \cdots & Y_{i,m-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{M-1,0} & Y_{M-1,1} & \cdots & Y_{M-1,j} & \cdots & Y_{M-1,m-1} \\ \cdots & & & & & \end{bmatrix} \rightarrow \begin{bmatrix} (p_0, q_0) \\ \vdots \\ (p_i, q_i) \\ \vdots \\ (p_{M-1}, q_{M-1}) \\ \cdots \end{bmatrix} \quad (3)$$

$$= \{(p_i, q_i)\}_{i=0}^{M-1}$$

The pair of 2-tuple measures (p_i, q_i) is determined by the following formula:

$$Y_{i,j} = Y[J] \in \{0, 1\}; J = i \times m + j, \quad (4)$$

$$0 \leq i < M, 0 \leq j < m, 0 \leq J < m \times M \leq N$$

$$p_i = \sum_{j=0}^{m-1} Y_{i,j}, Y_{i,j} \in \{0, 1\}, 0 \leq p_i \leq m; \quad (5)$$

$$q_i = \sum_{j=0}^{m-1} [(Y_{i,j-1}, Y_{i,j}) == (0, 1)], j - 1(\text{mod } m), 0 \leq q_i \leq \lfloor m/2 \rfloor; \quad (6)$$

i.e., $X = 0011010010, N = 10, M = 2, m = 5; (p_0 = 2, q_0 = 1); (p_1 = 2, q_1 = 2)$.

The parameter p_i is the number of single elements in the i -th sub-vector, the parameter q_i is the number of 01 pattern overlapped in the i -th sub-vector in a cyclic condition. For any segment $m > 0, 0 \leq p_i \leq m, 0 \leq q_i \leq \lfloor m/2 \rfloor$, all segments are transformed from a random sequence with N elements into a measuring sequence with M elements.

The SSM module outputs the ordered pairs of 2-tuple measures $\{p_i, q_i\}_{i=0}^{M-1}$.

2.3 Measuring Sequence Combination MSC

The MSC module is described in Fig. 2c, the module is composed of two units: the Measuring Split MS and the Measuring Combination MC. The MS unit processes the SSM module's output, and splits the measuring sequence with 2-tuple measures

into two independent measuring sequences: $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$ to keep the original measuring number invariant.

Recombining each single measuring sequence by overlapping consequent elements as a pair, the MC unit will form two independent measuring sequences organized in 2-tuple measures: $\{p_i\}_{i=0}^{M-1} \rightarrow \{(p_{i-1}, p_i)\}_{i=0}^{M-1}$ and $\{q_i\}_{i=0}^{M-1} \rightarrow \{(q_{i-1}, q_i)\}_{i=0}^{M-1}$, $i - 1 \pmod{M}$ to provide appropriate sequences for subsequent processing modules.

The MSC module produces the following four measure sequences: $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$, $\{(p_{i-1}, p_i)\}_{i=0}^{M-1}$, $\{(q_{i-1}, q_i)\}_{i=0}^{M-1}$, respectively.

2.4 Projective Color Map PCM

The PCM module consists of two units: PA, CM. For five measuring sequences, 1D and 2D measures will be processed separately.

The PA unit processes relevant measuring sequences to transform them into integer arrays and the CM unit will visualize these on either normalized histograms (1D measures) or color maps (2D measures), respectively.

2.4.1 1D Measures

The 1D measures involve two measuring sequences: $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$. Let $P[m + 1]$, $Q[\lfloor m/2 \rfloor + 1]$ and $NP[m + 1]$, $NQ[\lfloor m/2 \rfloor + 1]$ be two 1D (integer, float) arrays to represent the corresponding elements, which are defined in the following.

2.4.2 1DP Map

The 1DP statistic histogram: for a sequence $\{p_i\}_{i=0}^{M-1}$, NP , P are two arrays (float, integer) with $(m + 1)$ elements. The j -th elements $NP[j]$, $P[j]$, $0 \leq j \leq m$, can be obtained from the following procedure:

Initialization: $\forall NP[j] = 0.0$, $P[j] = 0$, $0 \leq j \leq m$;
 Calculation: *for* ($i = 0$; $i < M$; $i++$) { $P[p_i]++$; }
 Normalization: *for* ($j = 0$; $j \leq m$; $j++$) { $NP[j] = P[j]/M$; }

In the 1DP map, the PA unit corresponds to Initialization and Calculation; the CM unit handles Normalization.

2.4.3 1DQ Map

The 1DQ statistic histogram: for a sequence $\{q_i\}_{i=0}^{M-1}$, NQ , Q are two arrays (float, integer) with $(\lfloor m/2 \rfloor + 1)$ elements. The j -th elements $NQ[j]$, $Q[j]$, $0 \leq j \leq \lfloor m/2 \rfloor$, can be obtained from the following procedure:

Initialization: $\forall NQ[j] = 0.0, Q[j] = 0, 0 \leq j \leq \lfloor m/2 \rfloor$;
 Calculation: *for* ($i = 0; i < M; i++$) $\{Q[q_i]++\}$;
 Normalization: *for* ($j = 0; j \leq \lfloor m/2 \rfloor; j++$) $\{NQ[j] = Q[j]/M\}$;

Using P, NP, Q, NQ arrays, it is possible to generate the corresponding 1D statistical histograms as 1D maps.

In the 1DQ map, the PA unit corresponds to Initialization and Calculation; the CM unit handles Normalization.

2.4.4 2D Measures

The 2D measures specially process three measuring sequences: $\{(p_{i-1}, p_i)\}_{i=0}^{M-1}$, $\{(q_{i-1}, q_i)\}_{i=0}^{M-1}$, $\{(p_i, q_i)\}_{i=0}^{M-1}$. Let $P[m+1 : m+1]$, $Q[\lfloor m/2 \rfloor + 1 : \lfloor m/2 \rfloor + 1]$, $PQ[m+1 : \lfloor m/2 \rfloor + 1]$ be three 2D integer arrays to represent the corresponding elements, which are defined in the following.

2.4.5 2DP Map

2DP statistic histogram: for a sequence $\{(p_{i-1}, p_i)\}_{i=0}^{M-1}$, P is a 2D integer array with $(m+1)^2$ elements. The i, j -th elements $P[i, j]$, $0 \leq i, j \leq m$, can be obtained from the following procedure:

Initialization: $\forall P[i, j] = 0, 0 \leq i, j \leq m$;
 Calculation: $P[p_{M-1}, p_0]++$;
 for ($i = 1; i < M; i++$) $\{P[p_{i-1}, p_i]++\}$;
 Pseudo-color: Matching proper color $\forall P[i, j], 0 \leq i, j \leq m$

In the 2DP map, the PA unit corresponds to Initialization and Calculation; the CM unit handles pseudo-color.

2.4.6 2DQ Map

2DQ statistic histogram: for a sequence $\{(q_{i-1}, q_i)\}_{i=0}^{M-1}$, Q is a 2D integer array with $(\lfloor m/2 \rfloor + 1)^2$ elements. The i, j -th element $Q[i, j]$, $0 \leq i, j \leq \lfloor m/2 \rfloor$, can be obtained from the following procedure:

Initialization: $\forall Q[i, j] = 0, 0 \leq i, j \leq \lfloor m/2 \rfloor$;
 Calculation: $Q[q_{M-1}, q_0] ++$;
 $\text{for}(i = 1; i < M; i ++)\{Q[q_{i-1}, q_i] ++;\}$
 Pseudo-color: Matching proper color $\forall Q[i, j], 0 \leq i, j \leq \lfloor m/2 \rfloor$

In the 2DQ map, the PA unit corresponds to Initialization and Calculation; the CM unit handles Pseudo-color.

2.4.7 2DPQ Map

2DPQ statistic histogram: for a sequence $\{(p_i, q_i)\}_{i=0}^{M-1}$, PQ is a 2D integer array with $(m+1) \times (\lfloor m/2 \rfloor + 1)$ elements. The i, j -th elements $PQ[i, j], 0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$, can be obtained from the following procedure:

Initialization: $\forall PQ[i, j] = 0, 0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$;
 Calculation: $\text{for}(i = 0; i < M; i ++)\{PQ[p_i, q_i] ++;\}$
 Pseudo-color: Matching proper color $\forall PQ[i, j], 0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$

In the 2DPQ map, the PA unit corresponds to Initialization and Calculation; the CM unit handles Pseudo-color.

Through the PCM module, five measuring sequences are transformed into two 1D arrays and three 2D arrays with $(m+1), (\lfloor m/2 \rfloor + 1), (m+1)^2, (\lfloor m/2 \rfloor + 1)^2$ and $(m+1) \times (\lfloor m/2 \rfloor + 1)$ clusters, respectively.

The final results of the variant map system are five maps: 1DP, 1DQ, 2DP, 2DQ, and 2DPQ as expected statistic distributions of the input 0–1 sequence.

3 Sequence Analysis

3.1 Ideal Condition

From a viewpoint of sequence analysis, it is a classical technology to sort the $\{p_i\}_{i=0}^{M-1}$ measuring sequence as a 1D statistic histogram. When the measuring sequence meets ideal conditions, the 1D statistical distribution is a binomial distribution.

Lemma 1 *For an input 0–1 sequence, if the total number of segments is equal to $M = 2^m$, and each segment of m bits appears only once in the sequence, then the 1DP array satisfies the binomial distribution:*

$$P[i] = \binom{m}{i}, 0 \leq i \leq m \quad (7)$$

Corollary 1 *If the input sequence meets the conditions of Lemma 1, then the total number of items in the 1DP array is equal to*

$$\sum_{i=0}^m P[i] = 2^m = M \quad (8)$$

Lemma 2 *If the input sequence meets the conditions of Lemma 1, then the 1DQ array satisfies the following relation:*

$$Q[i] = 2 \binom{m}{2i}, 0 \leq i \leq \lfloor m/2 \rfloor \quad (9)$$

Corollary 2 *If the input sequence meets the conditions of Lemma 1, then the total number of items in the 1DQ array is equal to*

$$\sum_{i=0}^{m/2} Q[i] = 2^m = M \quad (10)$$

3.2 General Condition

Theorem 1 *For any 0–1 sequence with N elements, a 2DP array has two projections in both vertical and horizontal directions and they are corresponding to the 1DP array.*

Proof A 2DP array is generated from a measuring sequence $\{(p_{i-1}, p_i)\}_{i=0}^{M-1}$ and the 2DP array is $\{P[i, j]\}_{i=0}^m \{j=0\}^m$, from both directions $P[i] = \sum_{j=0}^m P[i, j]$, $0 \leq i \leq m$; $P[j] = \sum_{i=0}^m P[i, j]$, $0 \leq j \leq m$; so $\{P[i]\}_{i=0}^m = \{P[j]\}_{j=0}^m$. Both projections are the same 1DP array.

Corollary 3 *For an arbitrary input sequence, the total number of items in the 2DP array is equal to*

$$\sum_{i=0}^m \sum_{j=0}^m P[i, j] = \sum_{i=0}^m P[i] = M \quad (11)$$

Theorem 2 *For any 0–1 sequence with N elements, a 2DQ projection in both directions is the 1DQ array.*

Proof A 2DQ array is generated from a measuring sequence $\{q_{i-1}, q_i\}_{i=0}^{M-1}$ and the 2DQ array is $\{Q[i, j]\}_{i=0}^{\lfloor m/2 \rfloor} \{j=0\}^{\lfloor m/2 \rfloor}$, from both directions $Q[i] = \sum_{j=0}^{\lfloor m/2 \rfloor} Q[i, j]$, $0 \leq i \leq \lfloor m/2 \rfloor$; $Q[j] = \sum_{i=0}^{\lfloor m/2 \rfloor} Q[i, j]$, $0 \leq j \leq \lfloor m/2 \rfloor$; so $\{Q[i]\}_{i=0}^{\lfloor m/2 \rfloor} = \{Q[j]\}_{j=0}^{\lfloor m/2 \rfloor}$. Both projections are the same 1DQ array.

Corollary 4 For an arbitrary input sequence, the total number of items in the 2DQ array is equal to

$$\sum_{i=0}^{\lfloor m/2 \rfloor} \sum_{j=0}^{\lfloor m/2 \rfloor} Q[i, j] = \sum_{i=0}^{\lfloor m/2 \rfloor} Q[i] = M \quad (12)$$

Theorem 3 For any 0–1 sequence with N elements, a 2DPQ projection in two directions is corresponding to either a 1DP array or a 1DQ array, respectively.

Proof A 2DPQ array is generated from a measuring sequence $\{p_i, q_i\}_{i=0}^{M-1}$ and the 2DPQ array is $\{PQ[i, j]\}_{i=0}^m \sum_{j=0}^{\lfloor m/2 \rfloor}$, from two directions $P[i] = \sum_{j=0}^{\lfloor m/2 \rfloor} PQ[i, j]$, $0 \leq i \leq m$; $Q[j] = \sum_{i=0}^m PQ[i, j]$, $0 \leq j \leq \lfloor m/2 \rfloor$. So the two projections are corresponding to either a 1DP or a 1DQ array.

Corollary 5 For an arbitrary 0–1 input sequence, the total number of items in the 2DPQ array is equal to

$$\sum_{i=0}^m \sum_{j=0}^{\lfloor m/2 \rfloor} PQ[i, j] = M = \sum_{i=0}^m P[i] = \sum_{j=0}^{\lfloor m/2 \rfloor} Q[j] \quad (13)$$

Corollary 6 For an arbitrary input sequence, five measuring sequences are corresponding to two 1D and three 2D arrays. Let $|G|$ denote the number of associated possible clusters in G . If $m > 3$, then $|2DP| > |2DPQ| > |2DQ| > |1DP| > |1DQ|$ is satisfied.

Proof Five arrays: (2DP, 2DPQ, 2DQ, 1DP, 1DQ) contain $\{(m+1)^2, (m+1) \times (\lfloor m/2 \rfloor + 1), (\lfloor m/2 \rfloor + 1)^2, (m+1), (\lfloor m/2 \rfloor + 1)\}$ items, respectively. If $m > 3$, then the inequalities are true.

3.3 Brief Discussion

From the listed statement in lemmas, theorems, and corollaries, Lemmas 1 and 2 described an ideal input sequence where each segment is a uniform distribution which appears only once. Under this ideal condition, both 1DP and 1DQ arrays are corresponding to a binomial distribution. Corollaries 1 and 2 have shown that both 1DP and 1DQ arrays meet the number of quantitative characteristics for the ideal input sequence.

Theorems 1 and 2 establish projective conditions on any input sequence. A 2DP or 2DQ array has its 1D projection of two directions on the same array. Theorem 3 claims that for any 2DPQ array, two projections are corresponding to both 1DP and 1DQ arrays, respectively.

Corollaries 3 and 4 treat 2DP and 2DQ arrays, respectively, in the total number of summing conditions on their quantitative characteristics. Corollary 5 is associated

with Theorem 3 on a 2DPQ array to share with other four projections the same quantitative characteristics. In Corollary 5, the total number of each component on five statistic arrays is equal to the total number of segments M , a 2DPQ array occupies a central position in the projection to other arrays. Corollary 6 uses inequalities to show five scales of numbers of items in five arrays to provide the maximal number of items involved in the structure.

From a viewpoint of complex stochastic sequence analysis, this partition mode corresponds to the maximum number of clusters distinguished in the condition of multiple segments. Different from surface analysis based on the multivariate Gaussian probability distribution, variant maps provide only a limited finite number of lattice points that form space-related clusters on the projection position. Under the condition of segments in larger length, the 2DP array has the maximum number of distinct items and can be clearly distinguished among the five arrays to make the most visible map showing the largest refined distribution in details.

4 Sample Maps

Since the ideal distribution may appear merely on specific conditions, it is very difficult to use algebraic formulas to describe measuring sequences on statistical maps of an arbitrary cryptographic sequence. For complicated data sequences, the most effective scheme is using the computational approach directly to generate relevant maps and then to make feasible comparisons. Among the five maps generated from an input 0–1 sequence, more 2DP maps are selected in this section to illustrate a series of changes among segment lengths and shifting lengths for refined details.

In this section, one cryptographic sequence generated from an AES cipher is selected as a sample sequence, and various control parameters will be changed. This sample sequence has a fixed length $N = 10^6$ in one million stochastic bits. Various changes are made on the length m of segment and shift displacement r . Five maps will be applied to show their special statistical distributions.

4.1 Dramatically Changing the Segment Lengths: 1DP, 1DQ, 2DP, 2DQ, and 2DPQ Maps $m = \{8, 16, 128\}$, $r = 0$

Three groups of Figs. 3, 4, and 5 are involved in comparison based on the five maps.

In Fig. 3, nine maps from both 1DQ and 2DQ forms are selected in $m = \{8, 16, 128\}$, $r = 0$ condition; (a)–(c) showing three 1DQ maps with different segments; (d)–(f) showing 2DQ maps in normal sizes and (g)–(i) being the same 2DQ maps with enlarged sizes.

In Fig. 4, 12 maps from 1DP, 2DPQ, and 1DQ forms are selected in $m = \{8, 16, 128\}$, $r = 0$ condition; (a)–(c) showing three 1DQ maps with differ-

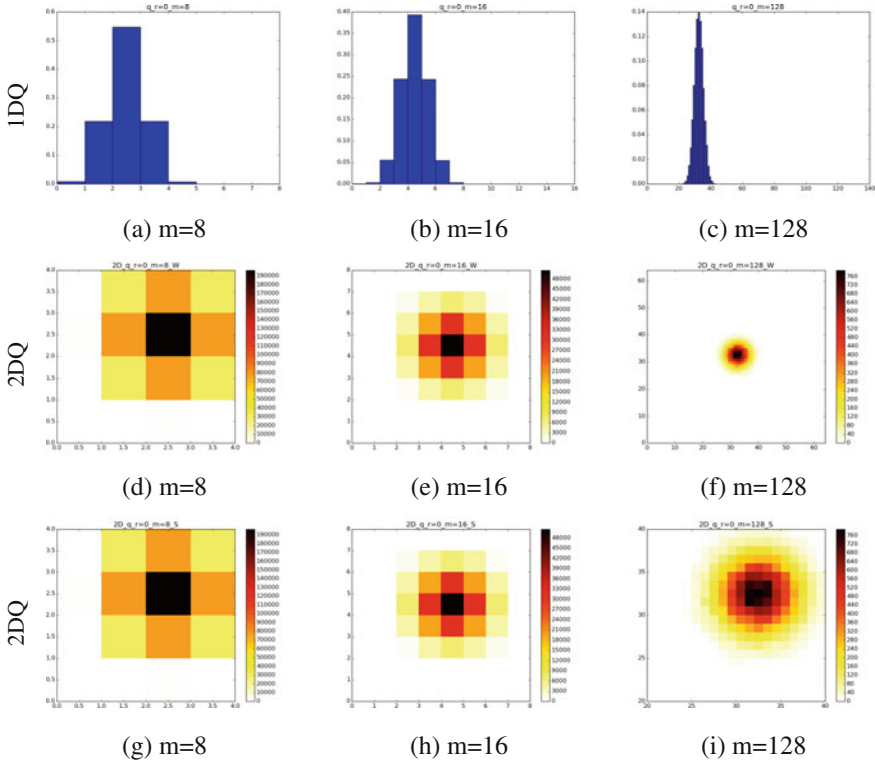


Fig. 3 1DQ and 2DQ maps on $m = \{8, 16, 128\}$, $r = 0$; **a-c** 1DQ maps; **d-f** 2DQ Regular maps; **g-i** 2DQ Enlarged maps

ent segments; (d)–(f) showing 2DPQ maps in normal sizes; (g)–(i) being the same 2DPQ maps with enlarged sizes and (j)–(l) illustrating 1DQ maps for convenient comparison.

In Fig. 5, nine maps from both 1DP and 2DP forms are selected in $m = \{8, 16, 128\}$, $r = 0$ condition; (a)–(c) showing three 1DP maps with different segments; (d)–(f) showing 2DP maps in normal sizes and (g)–(i) being the same 2DP maps with enlarged sizes.

4.2 Small Changes in Segment Lengths: 2DP Maps; Variation Series in Lengths of Segments $m = \{125, 126, 127\}$, $r = 0$

Two groups of maps are compared in Fig. 6 based on slightly changing segment lengths.

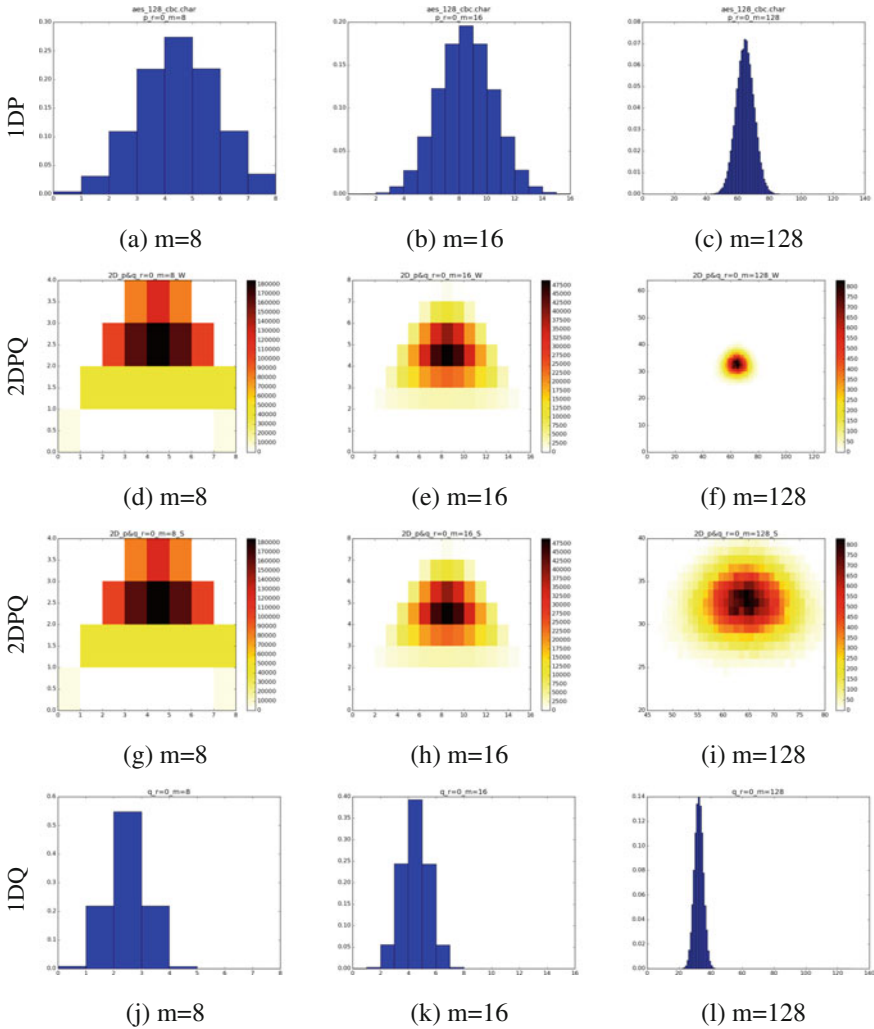


Fig. 4 1DP, 2DPQ, and 1DQ maps on $m = \{8, 16, 128\}$, $r = 0$; **a–c** 1DP maps; **d–f** 2DPQ Regular maps; **g–i** 2DPQ Enlarged maps; **j–l** 1DQ maps

In Fig. 6, nine maps from both 1DP and 2DP forms are selected in $m = \{125, 126, 127\}$, $r = 0$ condition; (a)–(c) showing three 1DP maps with different segments; (d)–(f) being 2DP maps in normal sizes and (g)–(i) showing the same 2DP maps with enlarged sizes.

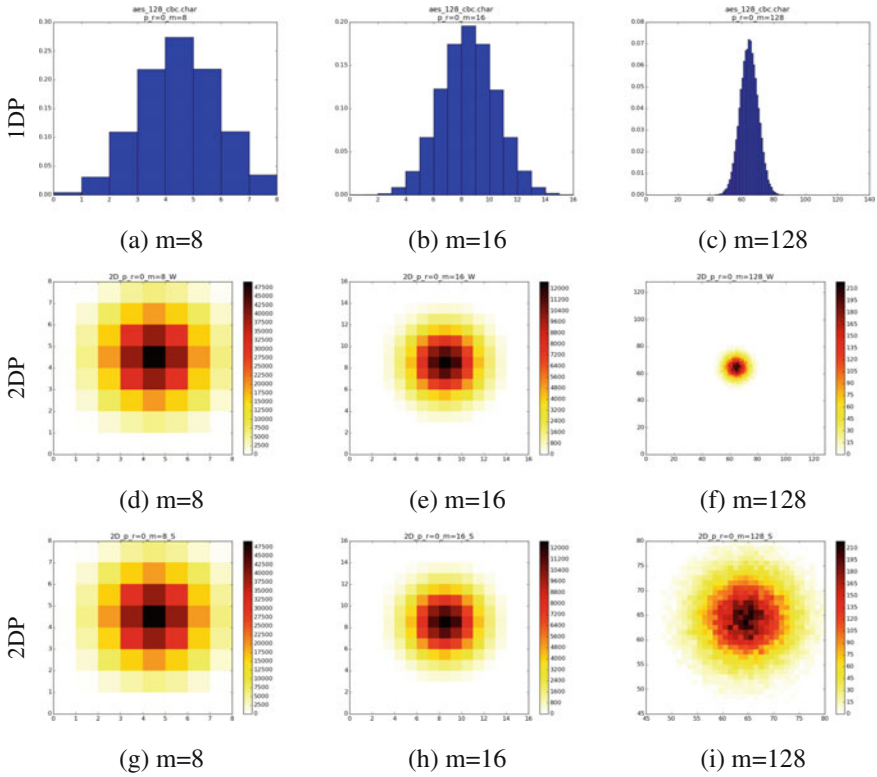


Fig. 5 1DP and 2DP maps on $m = \{8, 16, 128\}$, $r = 0$; **a-c** 1DP maps; **d-f** 2DP Regular maps; **g-i** 2DP Enlarged maps

4.3 Changing the Lengths of Shift Displacement: 2DP Maps Change on Displacement Series $m = 128$, $r = \{1, 2, 8\}$

Two groups of maps are compared in Fig. 7 under changing shift lengths.

In Fig. 7, nine maps from both 1DP and 2DP forms are selected in $m = 128$, $r = \{1, 2, 8\}$ condition; (a)–(c) showing three 1DP maps with different segments; (d)–(f) being 2DP maps in normal sizes and (g)–(i) showing the same 2DP maps with enlarged sizes.

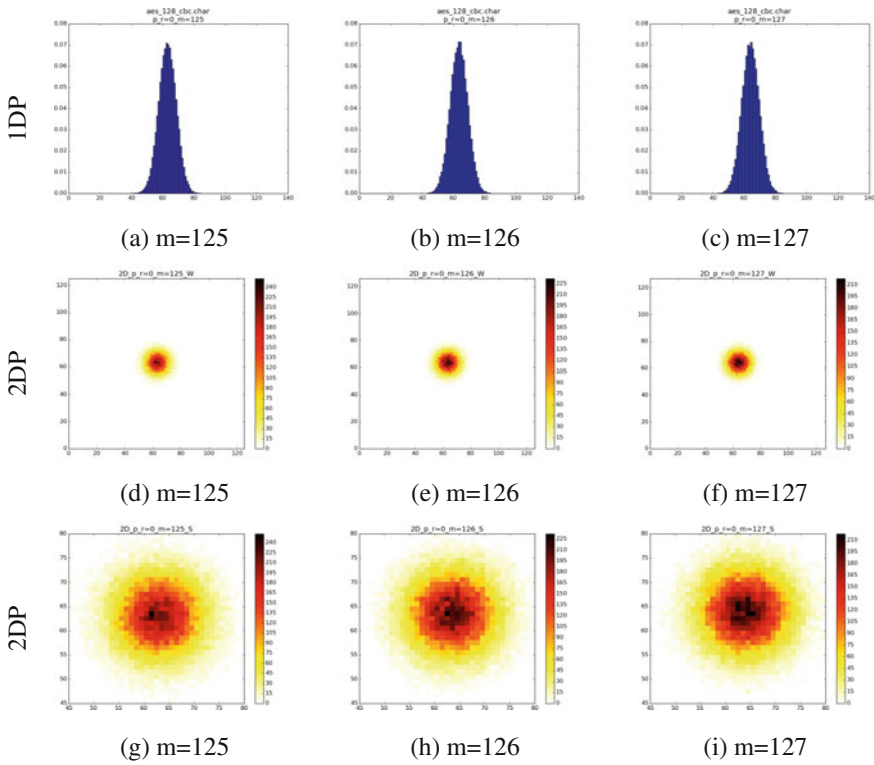


Fig. 6 1DP and 2DP maps on $m = \{125, 126, 127\}$, $r = 0$; **a-c** 1DP maps; **d-f** 2DP Regular maps; **g-i** 2DP Enlarged maps

4.4 Enlarged Maps: 2DP Maps on $m = \{125, 127, 128\}$, $r = \{0, 8\}$

1DP maps are selected in both Figs. 8 and 9 on enlarged forms.

In Fig. 8, four maps from the 2DP form are selected in $m = \{125, 127, 128\}$, $r = \{0, 8\}$ condition; (a) $r = 0$, $m = 125$; (b) $r = 0$, $m = 127$; (c) $r = 0$, $m = 128$, and (d) $r = 8$, $m = 128$. Four maps are showing the same 2DP maps on enlarged sizes.

In Fig. 9a and b, two maps of speckle patterns are selected from two distinct patterns for comparison. (a) a larger map from the 2DP form is generated in $m = 128$, $r = 0$ condition; (b) a larger map of Fig. 1d is illustrated for a laser beam reflected from a plastic surface onto a wall. It is convenient for readers to observe the two speckle pattern maps in refined details.

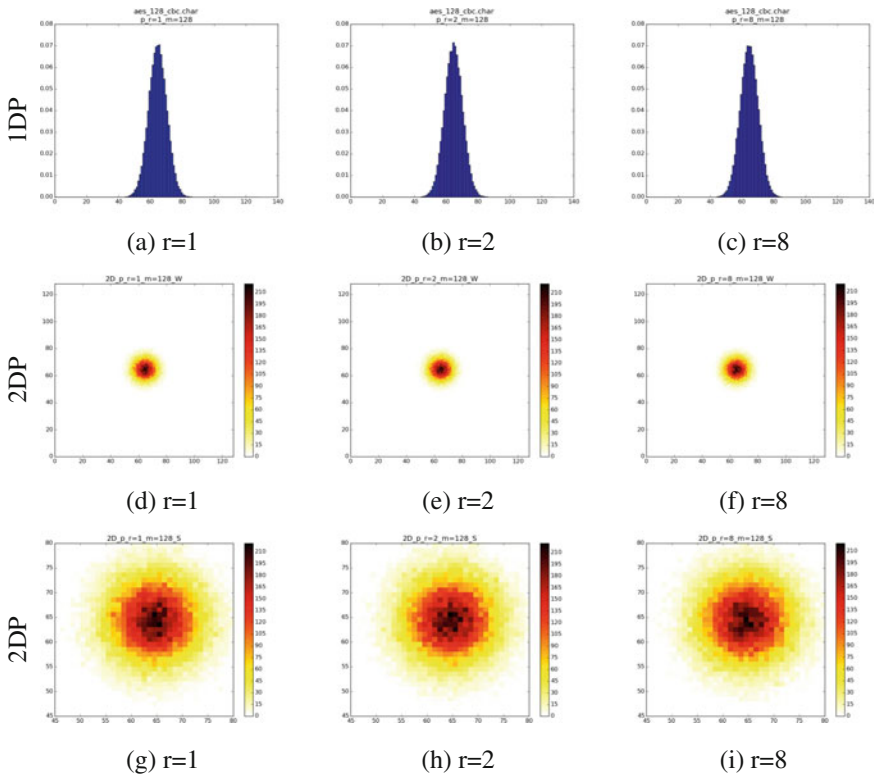


Fig. 7 1DP and 2DP maps on $m = 128, r = \{1, 2, 8\}$; **a-c** 1DP maps; **d-f** 2DP Regular maps; **g-i** 2DP Enlarged maps

5 Result Analysis

5.1 Figures 3, 4 and 5

In Figs. 3, 4, and 5, six maps are listed on both 1DP (Figs. 4 and 5a-c) and 1DQ (Figs. 3a-c and 4j-l) forms, their distributions are generally corresponding to binomial coefficients. Under the changes of different lengths on segments, 1D maps are showing distributions of binomial patterns in the symmetric bell curves with the maximal value on the middle area.

From Figs. 3 and 5, six 2DQ maps (Fig. 3d-i) and six 2DP maps (Fig. 5d-i) are listed, when $m = \{8, 16\}$, significant regular distributions along both horizontal and vertical directions (Figs. 3d-h and 5d-h) appear as symmetric patterns. The central cluster is collected the largest number of measures located on the center point of relevant maps. But checking maps in Figs. 3f-i and 5f-i, regular patterns with the central symmetry are severely destroyed when the length of segments is increased to

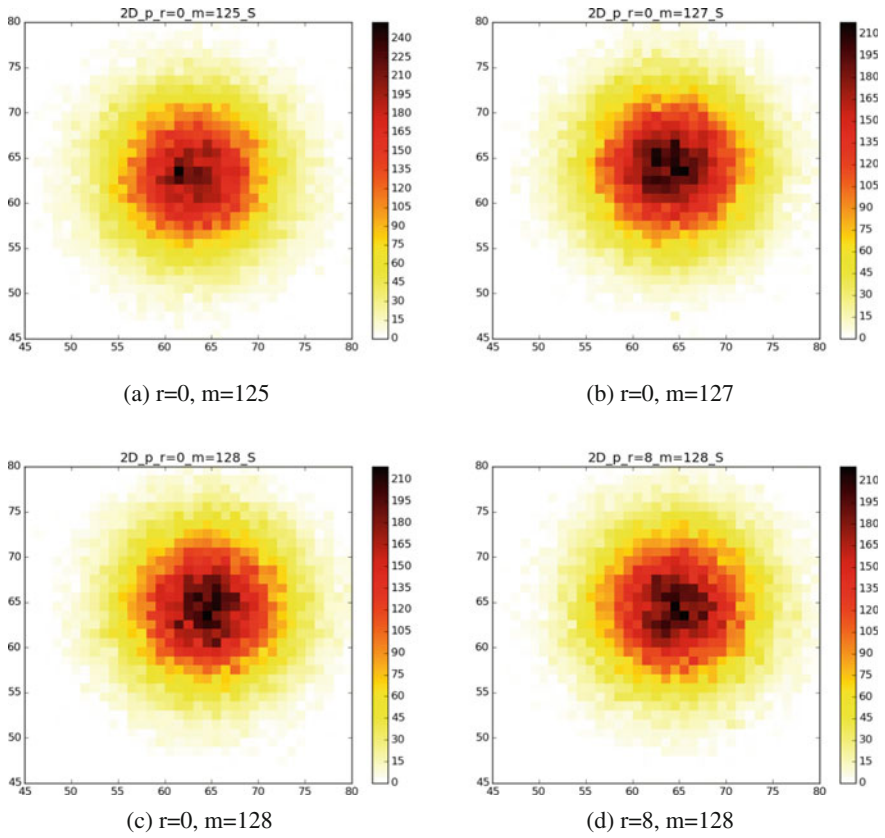


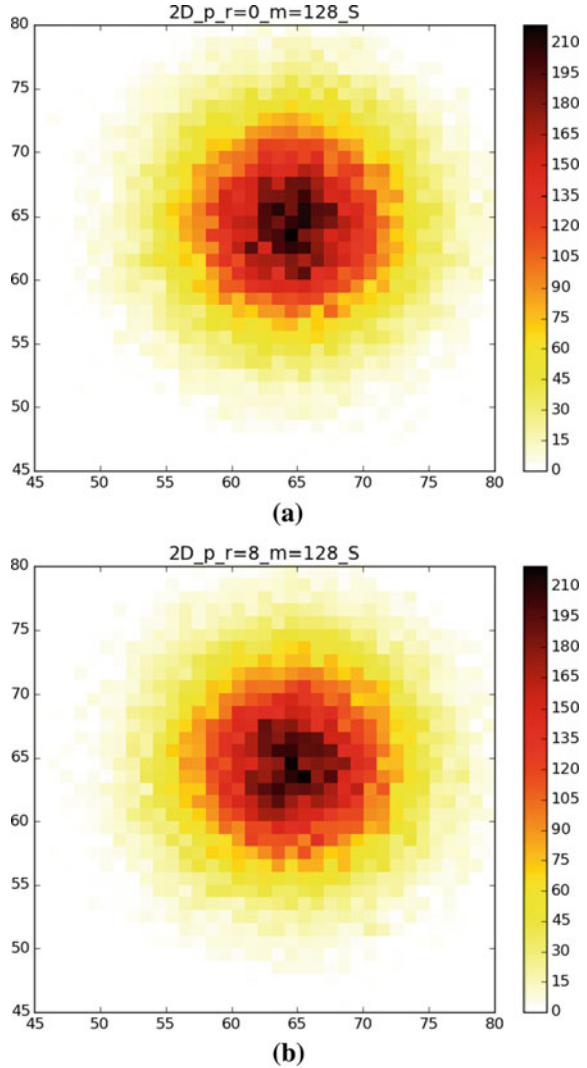
Fig. 8 2DP larger maps on $m = \{125, 127, 128\}, r = \{0, 8\}$; **a** $r = 0, m = 125$ map; **b** $r = 0, m = 127$ map; **c** $r = 0, m = 128$ map; **d** $r = 8, m = 128$ map

$m = 128$. Regarding the two maps in Figs. 3f and 5f, both maps show circular disks with the central position at the highest number of collected measures. However, the two enlarged maps in Figs. 3i and 5i clearly show that significant speckle patterns are visualized around the central areas with stochastic higher numbers of measures. By comparing the two maps in Figs. 3i and 5i, Figure 5i provides much more visible asymmetry than Fig. 3i.

Because a 2DQ map covers only a quarter of a 2DP map, the damaging ratio of its symmetric properties appears much weaker than on the 2DP map. Applying a sufficiently larger segment length, central areas are observed with random speckle patterns and visible symmetric properties significantly damaged.

In general, it is feasible for a 2DP map to observe its middle areas in an approximately rotational symmetry in small sizes. But when the segment length is big enough, significant speckle patterns emerge in the central area with stronger stochastic properties.

Fig. 9 Speckle patterns in enlarged maps of the 2DP form; **a** $m = 128, r = 0$; **b** $m = 128, r = 8$



In the 2DPQ maps of Fig. 4d–i, when $m = \{8, 16\}$, there appears a single central point as a key cluster to collect the maximal number with visible symmetrical patterns on the horizontal direction, but without symmetrical pattern on the vertical direction in Fig. 4d–h. However, when $m = 128$, the 2DPQ map of Fig. 4f appears as an irregular disk with higher values in the central area.

From the 2DPQ map of Fig. 4i, the enlarged map shows that stochastic speckle patterns appear in the central area with better horizontal symmetry than vertical direction with significantly damaged details.

5.2 Figure 6

In Fig. 6a–i, the nine maps are listed to show small changes on lengths of segments $m = \{126, 127, 128\}$. By checking the three 1DP maps in Fig. 6a–c, three middle areas appear slightly different from the bell shape: (a) left is higher than right; (b) right is higher than left; (c) right is higher than left and the middle one is lower than its nearest neighbors.

The three 2DP maps in (d)–(f) appear significantly as circular disks with an approximate symmetry and higher clusters around central areas. In the three enlarged 2DP maps in (g)–(i), there appear various speckle patterns in central areas.

Comparing the six maps of (a)–(c) and (g)–(i), speckle patterns in the three 2DP maps (g)–(i) are much easier identified than broken curving patterns in the three 1DP maps (a)–(c).

5.3 Figure 7

In Fig. 7a–i, the nine maps are listed to analyze changes of the parameters $m = 128, r = \{1, 2, 8\}$. By checking the three 1DP maps in Fig. 7a–c, middle areas of three maps appear slightly different from the regular bell shape: (a) left is lower than middle and middle is equal to right; (b) left and right are lower than middle, and right is higher than left; (c) left-middle-right are equal.

The three 2DP maps in (d)–(f) appear as similar circular disks with an approximate symmetry and higher clusters around central areas. In the three enlarged 2DP maps (g)–(i), there are various speckle patterns distinguishably placed in central areas.

Comparing the six maps of (a)–(c) and (g)–(i), distinguishable speckle patterns in the three 2DP maps (g)–(i) are much easier identified than broken curving patterns in the three 1DP maps (a)–(c).

5.4 Figures 8–9

In Fig. 8a–d, four enlarged 2DP maps are listed by using the parameters $m = \{125, 127, 128\}, r = \{0, 8\}$. Three maps (a)–(c) are created with $m = \{125, 127, 128\}, r = 0$ and two maps (c)–(d) with $m = 128, r = \{0, 8\}$. Four larger 2DP maps in (a)–(d) show stronger speckle patterns distinguishable in their central areas with significant distributions identified differently from mixed reflection and rotational effects.

In Fig. 9a–b, two enlarged maps of speckle patterns are selected. The map (a) with $m = 128, r = 0$ provides refined details to illustrate stochastic speckle patterns in the central area and the map (b) with $m = 128, r = 8$ has the same segment length, but a different shift length. The highest color clusters of the map (b) appear more

compact and simpler than the highest color clusters of the map (a). The two maps are showing different speckle patterns as a result of simple geometric transformations.

By comparing the two enlarged speckle pattern maps, significant similarities and differences in details could be recognized.

6 Conclusion

For any 0–1 sequence with N elements, the variant map system processes multiple segments to transform each segment in a pair of measures. Using the cryptographic sequence generated from the AES cipher, five statistic maps were created. Two 1D maps show binomial distributions to which we refer as classical maps. Three 2D maps are constructed as variant maps. Selecting smaller segmented lengths, both classical and variant maps were illustrated in four groups. With larger segmented lengths increased, there are significant speckle patterns observed. From a brief comparison of the two larger maps, the enlarged 2DP maps in Fig. 9a, b show better refined visual details than other smaller maps.

For the 2DPQ map, there are significant horizontal symmetries observed, however, there is no reflection effect in the vertical direction.

From different 2DP maps with parameters $m = \{125, \dots, 128\}$, significant changes are observed: various speckle patterns are developed by both changes between lengths of segments and shift displacements. Enlarged maps are convenient to illustrate stochastic speckle patterns visibly. Some significant clusters are collected with speckle patterns associated to different control parameters in relevant maps.

From a viewpoint of system operation, two types of control parameters: length of segments and shift length of the sequence, provide an effective control mechanism to form clear speckle patterns on 2D distributions. It is necessary for us to put more attention on systematically exploring this type of issues, for refined researches on further directions.

The variant map system is different from both technologies: extracting information of speckle patterns to form random sequences and NIST 800-22 statistic testing package to use a single measurement of a P-value or a list of static parameters for evaluation. The variant framework provides five maps to identify complicated measurements through speckle patterns in details for any cryptographic sequence. Three refined 2D maps have more accurate properties than two 1D maps to describe nonlinear dynamic behavior as possible quantitative measurements.

In relation to the variant map system, future explorations on both theoretical foundation and key applications on cryptographic sequences are urgently required.

References

1. S. Pyne, B. Rao, S. Rao Edited, *Big Data Analytics - Methods and Applications* (Springer India, 2016)
2. L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing. *Inf. Sci.* 371–386 (2014)
3. D. Puthal et al., A dynamic prime number based efficient security mechanism for big sensing data streams. *J. Comput. Syst. Sci.* **83**(1), 22–42 (2017)
4. S. Golomb, *Shift-Register Sequences*, Revised edn. (Aegean Park Press, Laguna Hills, California, 1982)
5. E. Barkam, E. Biham, N. Keller, Instant ciphertext-only cryptanalysis of GSM encrypted communication. *J. Cryptology* **21**(3), 392–429 (2008)
6. Y. Lu, W. Meier, S. Vaudenay, The conditional correlation attack: a practical attack on bluetooth encryption. *Crypto* **2005**(3621), 97–117 (2005)
7. <https://en.wikipedia.org/wiki/ESTREAM>
8. P. Junod, A. Canteaut, *Advanced Linear Cryptanalysis of Block and Stream Ciphers* (IOS Press, 2011), p. 2. ISBN 9781607508441
9. ZUC. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 and 128-EIA3: Document 2: ZUC Specification
10. A. Poorghanad, A. Sadr, A. Kashanipour, Generating high quality pseudo random number using evolutionary methods. *IEEE Congr. Comput. Intell. Secur.* **9**, 331–335 (2008)
11. A. de Queiroz, J. Schechtman, Elimination of nonlinear clock feed through in component-simulation switched-current circuits, in *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, pp. II378–II381 (1998)
12. A. Fuster-Sabater, F. Vitini, Classes of nonlinear filters for stream ciphers, chapter *Geometry, Algebra and Applications: From Mechanics to Cryptography*, Volume 161 of the series Springer Proceedings in Mathematics and Statistics, 107–119 (2016)
13. S. Ronjom, C. Cid. Nonlinear Equivalence of Stream Ciphers, in *Proceedings of Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, Lecture Notes in Computer Science*, vol. 6147 (Springer, 2010), pp. 40–54.
14. J. Nechvatal, E. Barker, L. Bassham, et al. (2000), Report on the development of the advanced encryption standard (AES), *National Institute of Standards and Technology (NIST)*, <http://csrc.nist.gov/archive/aes/round2/r2report.pdf>
15. G. Paul, S. Maitra. *RC4 Stream Cipher and Its Variants* (CRC Press, 2012)
16. S.D. Cardell, A. Fuster-Sabater, linear models for the self-shrinking generator based on CA. *J. Cell. Automata* **11**(23), 195211 (2016)
17. N. Nagaraj, One-time pad as a nonlinear dynamical system. *Commun. Nonlinear Sci. Numer. Simul.* **17**, 4029–4036 (2012)
18. E. Dubrova, M. Teslenko, H. Tenhunen. On analysis and synthesis of (n,k)-non-linear feedback shift registers, in *Proceedings of the Conference on Design, Automation and Test in Europe*, 1286–1291 (2008)
19. E. Dubrova, A list of maximum period NLFSRs, *Cryptology ePrint Archive*, Report 2012/166 (2012)
20. Y. Zhao, Y. Hu, S. Li, A new analysis method for nonlinear component of stream ciphers. *J. Inf. Comput. Sci.* **10**(16), 5313–5321 (2013)
21. D. Meschede. *Optics, Light and Lasers*, 2 ed. (Wiley-VCH, 2007)
22. R. Boyd. *Nonlinear Optics*, 3rd ed. (Academic Press, 2008)
23. M. Nakazawa et al., QAM quantum stream cipher using digital coherent optical transmission. *Opt. Express* **22**(4), 4098–4107 (2014)
24. M. Yoshida et al., Single-channel 40 Gbit/s digital coherent QAM quantum noise stream cipher transmission over 480 km. *Opt. Express* **24**1, 652–661 (2016)
25. J. Barry, E. Lee, D.G. Messerschmitt, *Digital Communications* (Springer, 2004)
26. S. Lian, et al., A chaotic stream cipher and the usage in video protection. *Chaos Solitons and Fractals* **34**(3), 851–859 (2007)

27. J.W. Goodman, Some fundamental properties of speckle. *J. Opt. Soc. Am.* **66**, 1145 (1976)
28. D.G. Marangon, G. Vallone, P. Villorosi, Random bits, true and unbiased, from atmospheric turbulence. *Sci. Rep.* **4**, 5490 (2014). <https://doi.org/10.1038/srep05490>
29. J. Marron, A.J. Martino, G.M. Morris, Generation of random arrays using clipped laser speckle. *Appl. Opt.* **25**, 26 (1986)
30. P. Lalanne et al., 2-D generation of random numbers by multimode fiber speckle for silicon arrays of processing elements. *Opt. Commun.* **76**, 387–394 (1990)
31. R. Horstmeyer, R.Y. Chen, B. Judkewitz, C. Yang, Markov speckle for efficient random bit generation. *Opt. Express* **20**, 26394–26410 (2012)
32. D.E. Knuth, *The Art of Computer Programming*, vol. 2: *Seminumerical Algorithms* (Addison-Wesley, 1969)
33. NIST. A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST, Special Publication* (2010)
34. D. Makovoz, Noise variance estimation in signal processing, in *International Symposium on Signal Processing and Information Technology* (2006), pp. 364–369
35. K. Ito, Gaussian filter for nonlinear filtering problems, in *Conference on Decision and Control*, pp. 1218–1223 (2000)
36. F. Orieux, O. Feron, J. Giovannelli, Sampling high-dimensional gaussian distributions for general Linear inverse problems. *IEEE Signal Process. Lett.* **19**(5), 251–254 (2012)
37. J.W. Goodman. *Speckle Phenomena in Optics Theory and Applications*, (Ben Roberts and Cmpnany, 2007)
38. Speckle pattern, https://en.wikipedia.org/wiki/Speckle_pattern
39. M. Cross, P. Hohenberg, *Science* **263**, 1569 (1994)
40. P. Colet, R. Roy, K. Wiesenfeld, *Phys. Rev. E* **50**, 3453 (1994)
41. I.S. Aranson, L. Kramer, *Rev. Mod. Phys.* **74**, 99 (2002)
42. M. Jiang, X. Wang, Q. Ouyang, H. Zhang, *Phys. Rev. E* **69**, 056202 (2004)
43. H. Zhang, B. Hu, G. Hu, Q. Ouyang, J. Kurths, *Phys. Rev. E* **66**, 046303 (2002)
44. Q. Ouyang, *Introduction on Nonlinear Sciences and Pattern Dynamics* (Peking University Press, 2010) (in Chinese)
45. P.J.A. Holmes, Nonlinear oscillator with a strange attractor. *Philos. Trans. Royal Soc. A* **292**(1394), 419–448 (1979)
46. F. Haake. *Quantum Signatures of Chaos* (Springer-Verlag, 2010)
47. G. Teschl, *Ordinary Differential Equations and Dynamical Systems, Graduate Studies in Mathematics*, vol. 140 (Amer. Math. Soc, Providence, 2012)
48. Z.J. Zheng, C.H.C. Leung, Visualising global behaviour of 1D cellular automata image sequences in 2D Map. *Phys. A* **3–4**, 785–800 (1996)
49. D. E. Knuth. *The Art of Computer Programming*, vol. 4A: *Combinatorial Algorithms Part 1* (Addison-Wesley, 2011)
50. Z.J. Zheng. *Conjugate transformation of regular plan lattices for binary images*, Ph.D. Thesis, Monash University (1994)
51. J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic, *Frontiers of Electrical and Electronic Engineering in China*, 5(2), 163–172. Higher Educational Press and Springer-Verlag (2010). <https://doi.org/10.1007/s11460-010-0011-4>
52. H. Wang, J. Zheng, 3D Visual Method of Variant Logic Construction for Random Sequence, in *Australian Information Warfare and Security*, pp. 16–27 (2013)
53. W.Z. Yang, J. Zheng, Variant pseudo-random Number generator, *Hakin9 Extra. Timing Attack* **06**(13), 28–31 (2012)
54. J. Zheng, Novel Pseudo-Random Number Generation Using Variant Logic Framework, in *2nd International Cyber Resilience Conference*, 10bit04 (2011). <http://igneous.scis.ecu.edu.au/proceedings/2011/icr/zheng.pdf>
55. J. Zheng, C. Zheng, Variant simulation system using quaternion structure. *J. Mod. Opti.* **59**(5), 484–492 (2012)
56. J. Zheng, C. Zheng, T.L. Kunii, Interactive Maps on Variant Phase Space, in *Emerging Application of Cellular Automata*, pp. 113–196 (InTech Press, 2013)

57. J. Zheng, W. Zhang, J. Luo, W. Zhou, R. Shen, Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Math.* **3**(7A), 5–24 (2013)
58. D.M. Heim, O. Heim, P.A. Zeng, J. Zheng, Successful creation of regular patterns in variant maps from bat echolocation calls. *Biol. Syst.: Open Access* **5**, 2 (2016). <https://doi.org/10.4172/2329-6577.1000166>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Stationary Randomness of Three Types of Six Random Sequences on Variant Maps



Jeffrey Zheng, Yamin Luo, Zhefei Li and Chris Zheng

Abstract Various random streams have different stationary properties. It is necessary to use statistical probability and time series to evaluate quality of stationary randomness. In this chapter, a testing model is used on three maps for a random sequence. Multiple segments are divided on the shifted sequence as three measuring sets. For a map, the maxima are extracted and three maximal values are identified. 2D maps represent stationary randomness. Conditions of station random/stationary sequences are investigated. Testing sets are collected from three types of six random resources: AES, DES, A5, RC4, Australian National University (ANU), and University of Science and Technology of China (USTC) (two block ciphers, two stream ciphers, and two quantum ciphers). Six random sequences are selected. Measurements of stationary randomness are compared. There are only 0.0034–4.27% differences that are recognized. Using variation ratios, six samples are composed of three variation categories on {AES, DES}, {A5, RC4}, and {ANU, USTC}, respectively. From a measuring viewpoint, all six samples are showing distinguished stationary randomness properties.

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014) and Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉) · Y. Luo · Z. Li · C. Zheng
Key Laboratory of Quantum Information of Yunnan,
Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Software Engineering of Yunnan,
Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

Y. Luo
e-mail: 1047668416@qq.com

Z. Li
e-mail: 576167164@qq.com

C. Zheng
e-mail: z@caudate.me

Keywords Stationary randomness · Segment · Shifted sequence · Maxima
Quantum sequence · Variation ratio

1 Introduction

In modern cyberspace environment [1], network communication technologies play the essential role to support advanced developments of science, technology, and social daily life in every aspect. From a security viewpoint of network communication, Communication Security (COMSEC) systems [2] are the most important part. Every COMSEC system depends on block cipher/stream cipher/hash technologies, and its core component is linked to a random number generator for any cryptographic applications.

Quantum satellite [3] using Quantum Key Distribution (QKD) systems [4] in cryptographic applications is the most advanced ICT development to establish ultra-secure quantum communications. For a QKD system, a truly random number generator [5], quantum random number generator, plays a key role.

From a reliable viewpoint, it is necessary to test stationary randomness degrees on shift operations in evaluations. In this section, a list of relevant schemes, pseudo-random/truly random sequences, P_value, statistical probability distribution, optical statistics, stationary/nonstationary properties, and variant maps, are discussed.

1.1 Pseudorandom Sequences from Linear Stream Ciphers

Traditional stream ciphers [6] on Linear Feedback Shift Register (LFSR) structure (in military cryptography) are used as pseudorandom number generators, due to the ease of implementation from simple hardware, long periods, and uniformly distributed streams. The LFSR stream ciphers are the core in classical stream ciphers through the mathematical theory of algebraic functions for system simulation and analysis.

However, an LFSR is a linear system leading to fairly easy cryptanalysis using the Berlekamp–Massey algorithm. Important LFSR-based stream ciphers A5/1 & A5/2 are used in GSM cell phones and E0 is used in Bluetooth protocol. But from cryptanalysis viewpoint, the A5/2 cipher has been broken and both A5/1 and E0 have serious weaknesses [7, 8].

1.2 Pseudorandom Sequences from Nonlinear Stream Ciphers

The new generation of stream ciphers [9, 10] is widely used in advanced cyber communications. Three general methods are applied to improve security weaknesses in LFSR-based stream ciphers:

1. **Nonlinear Functions:** Nonlinear combination of several bits from the LFSR state [11];

2. **Nonlinear Parts:** Nonlinear combination of the output bits of two or more LFSRs or using evolutionary algorithm for nonlinearity [12]; and
3. **Clock Control:** Irregular clocking of the LFSR, as in the alternating step generator [13].

With batch a series of nonlinear algorithms are emerged [14]: nonlinear equivalence [15], evolutionary methods [12], AES cipher [16], RC4 [17], ZUC [11], cellular automata [18], and nonlinear dynamic system [19].

The new generation of stream ciphers has being shifted from the traditional mode: LFSR [6] to various nonlinear modes: NLFSR [20, 21], clock control [13], nonlinear functions [11], etc.; it is essential for ciphers to be integrated and implemented [22] to satisfy security models. However, different from LFSR with well-established linear mathematical theories and simulation tools, it is extremely difficult to use advanced nonlinear mathematical theories, recursive models, descriptive tools, and implementing schemes [19] in nonlinear dynamic environments. How to evaluate cryptographic sequences generated from the nonlinear stream ciphers is an urgent problem for modern stream/block ciphers.

1.3 Truly Random Sequences from Hardware Devices

In addition to pseudorandom sequences generated by stream ciphers, high-quality stochastic oscillators of truly random sequences are generated from special hardware devices such as laser photonics [23], nonlinear optics [24], quantum optics [25], quantum noises [26], thermal noise [27], and chaos and fractal nonlinear dynamics [28].

Since various truly random sequences are created from specific physical models with special principles and uncertain methodologies, it is extremely difficult for cryptographic researchers to make proper measurements explore nonlinear dynamic properties.

1.4 P_value Schemes—Statistical Tests on Cryptographic Sequences

Randomness has being explored for many years [29] on a series of statistic testing theories and methods. From a testing viewpoint, it is feasible to apply statistic testing packages to measure randomness properties on a given cryptographic sequence. NIST 800-22 package is a typical representative to provide more than 15 testing schemes for evaluation. Using the testing package, it is essential to check whether $P_value > 0.01$ for the sequence. Since such measuring scheme provides static property, it is difficult to use only P_value parameter to express complex dynamic behaviors intrinsically involved in cryptographic sequences.

Since comprehensive behaviors in nonlinear dynamics may increase computational complexities tragically to involve complicated dynamic properties in the multivariate environment, those dynamic behaviors are completely ignored in P -value schemes.

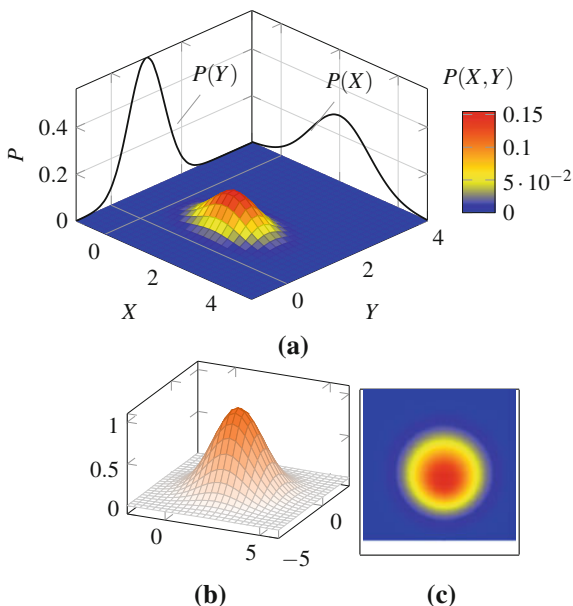
1.5 Multiple Statistical Probability Distributions

Measuring cryptographic sequences under segment conditions, multiple statistical probability schemes are useful to create various distributions to illustrate complex spatial relationships.

Multivariate normal probability distributions are the most important and powerful tool to test stochastic characteristics of a random data sequence [30] under the framework of probability, stochastic process, and statistics [31] for nonlinear problems. In this kind of measuring models, when a data sequence is sufficiently long, the high-dimensional probability distribution of the sequence [32] is converted into a continuous Gaussian distribution.

A typical projection model is shown in Fig. 1a; the central part shows a Gaussian surface with an unbalanced distribution in a 2D plane distributed as $P(X, Y)$ measures with pseudo-colors and two 1D projections shown in horizontal $P(X)$ and vertical $P(Y)$ planes, respectively. In Fig. 1b, a standard Gaussian surface with

Fig. 1 Multivariate Gaussian Probability Distributions (a)–(c); **a** Bivariate normal distribution with two probability projections; **b** A symmetric bivariate normal surface with pseudo-colors; **c** A 2D pseudo-color map of the symmetric bivariate normal surface



symmetric shapes is illustrated and the 2D projection of its pseudo-color map is shown in Fig. 1c with continuous distribution of color on the map.

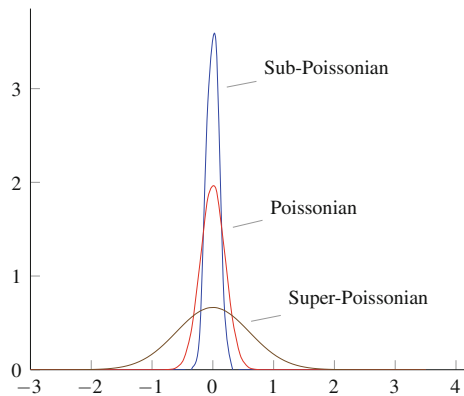
From sample figures, the relationship between the projection curve and two 1D Gaussian distributions are observed in the multivariate normal probability environment. Multivariate Gaussian probability distributions support various schemes to analyze complex stochastic data set of measuring sequences in many applications in continuous conditions.

1.6 Photon Statistic in Quantum Optics

Photon statistics is the theoretical and experimental approach on the statistical distributions in photon counting experiments to analyze the statistical nature of photons in a light source.

Three types of statistical distributions shown in Fig. 2 can be obtained by the light source [33]: Poissonian, super-Poissonian, and sub-Poissonian. The variance and average number of photon counts are identified for the corresponding distribution. Both Poissonian and super-Poissonian light are described by a semi-classical theory in which the light source is modeled as an electromagnetic wave and the atom is modeled by quantum mechanics. In contrast, sub-Poissonian light requires the quantization of the electromagnetic field for a proper description and is a direct measure of the particle nature of light.

Fig. 2 Three-photon statistical distributions



1.7 Stationary and Non-stationary Properties

In mathematics and statistics, a stationary process is a stochastic process [34] whose joint probability distribution does not change when shift operations performed. Consequently, parameters such as mean and variance, if they are present, also do not change over time. Stationarity is an interesting property for many statistical procedures in time series analysis.

In 1938, Kolmogorov established the basic theorems for smoothing and predicting stationary stochastic processes [35, 36] that had major military applications during the Cold War.

In applied mathematics, the Wiener–Khinchin theorem [37–39] states that the Autocorrelation Function (ACF) of a wide-sense-stationary process has a spectral decomposition given by the power spectrum of the process. One of the effective ways identifying stationary times series is the ACF plot [40]. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of nonstationary data decreases slowly [41].

1.8 Datastreams

1.8.1 Pseudorandom Number Resources

Four cryptographic sequences are selected: {AES, DES, A5, RC4}. For each cipher, a cryptographic sequence of 100MB data streams is collected.

{AES, DES} are block ciphers [16] on OFB mode to transfer block cipher output as a stream cipher stream.

A5/1 is a stream cipher [42] based around a combination of three LFSRs with irregular clocking.

RC4 is a stream cipher [43] designed by Ron Rivest in 1987. The design of RC4 avoids the use of LFSRs, its structure is ideal for software implementation, and it requires only byte manipulations.

1.8.2 Two Quantum Random Number Resources

Reliable and unbiased random numbers are important in cryptographic applications. Many algorithms can be used to generate pseudorandom numbers, but they can never be perfectly random or indeterministic.

Quantum random numbers can be generated from a physical quantum source of a coherent laser light to be splitting a beam of light into two beams and then measuring the power in each beam. Due to the light intensity in each beam, it fluctuates about the mean. Those fluctuations can be converted into a source of random numbers [44–46] being a stationary Poisson distribution.

Two quantum cryptographic resources are selected: {ANU, USTC}. For each quantum cipher, a truly random sequence of 1GB data streams is collected.

USTC resource: In the Key Laboratory of Quantum Information, USTC, CAS, true random number sequences are generated [45]. This type of true random sequences supports advanced quantum communication devices of QKD systems [47, 48].

More than 20GB quantum random number sequences are provided by USTC for randomness testing.

ANU resource: The ANU Quantum Random Numbers Server is an open website [49] to offer true random numbers to anyone on the Internet. Such random numbers are generated in real time by measuring the quantum fluctuations of the vacuum. The electromagnetic field of the vacuum exhibits random fluctuations in phase and amplitude at all frequencies. By carefully measuring these fluctuations, ultra-high bandwidth random numbers can be generated. Relevant data streams are downloaded.

1.9 Variant Framework

The conjugate classification [50] is proposed to apply seven measures in a hierarchy to partition the kernels of four regular plane lattices on $n = \{4, 5, 7, 9\}$ cases for 2D binary images. For 1D cellular automata sequences, global random behaviors [51] are visualized in 2D maps.

Various schemes following the top-down strategy are explored to use multiple measures to partition special phase spaces from a top state set to multiple bottom states via multilevels of a hierarchy in combinatorial algorithms [52], image analysis, and processing for many years.

For n -tuple bit vectors, the variant logic framework [53] is proposed, and various applications are explored: 3D visual method on random number sequences [54], variant Pseudorandom Number Generator (PRNG) [55, 56], computational simulation on quantum interactions [57, 58], noncoding DNA analysis [59], and bat echolocation [60].

1.10 Proposed Scheme

For the convenience of testing stationary randomness on six cryptographic sequences, we propose a testing system for a stationary random sequence with length N ; multiple segments M are divided from the sequence by a given length m ; a 2-tuple pair of measures can be extracted from a 0–1 segment that is the number of 1 element and the number of 01 pattern in the segment. All paired measures are composed of a sequence of M pairs of measures as an ordered measuring set with M elements.

The pairs of the measuring sequence are directly separated as two independent measuring sequences to keep each parameter in the same order. A total of three

sequences of distinct measures are constructed including two sequences on single measures and one sequence on 2-tuple measures.

Following this approach, two sets of single measuring sequences are sorted as two 1D numeric arrays as statistical histograms corresponding to 1D maps, and the 2-tuple measuring sequence is sorted as a 2D integer array as statistic histograms being a 2D map. Under the controlling operations on the changes of shift displacement, multiple results of the three measuring sequences are transformed into 1D statistic histograms and 2D pseudo-color maps to show effective patterns from the generated sequence under various positions and conditions on a list of shift operations.

1.11 Organization of the Chapter

This chapter describes a testing system for a stationary random sequence on diagrams of the system architecture and the core modules with input/output and processing functions in Sect. 2. In Sect. 3, the relationships among measuring sequences and the three statistical distribution maps are analyzed. In Sect. 4, four random sequences are generated from {AES, DES, A5, RC4} ciphers and two quantum cryptographic sequences collected from the Key Laboratory of Quantum Information, USTC, CAS, and ANU quantum number site. From the results of the visual maps in section IV, numeric analysis and brief comparison are carried out in Sect. 5. And finally in Sect. 6, the main results are summarized.

2 Testing System

To describe the testing system, diagrams are shown in Fig. 3.

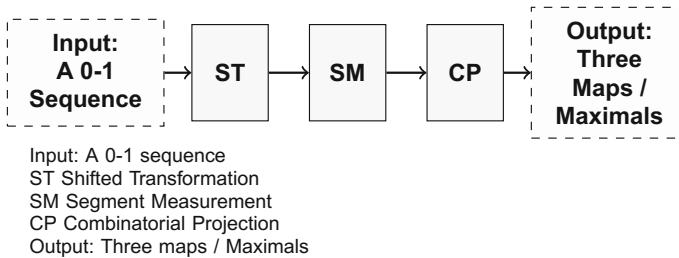


Fig. 3 The architecture of testing stationary random sequences

2.1 System Architecture

This system is composed of five parts: Input, Shifted Transformation (ST), Segment Measurement (SM), Combinatorial Projection (CP), and Output.

The input of the testing system is a selected 0–1 sequence, and its output is composed of three maps, two in 1D and one in 2D for visual distributions, and three maximals to be processed by ST, SM, and CP modules, respectively.

2.2 Core Modules

The testing system consists of three modules: {ST, SM CP}.

Input: X $N = m * M$ bit sequence; m segment length; M total segments; r shift length;

Output: Three maps {1DP, 1DQ, 2DPQ}; Three maximals {1DP_x, 1DQ_x, 2DPQ_x}

Process: Shifting r position from X to be $Y = X(r)$ in ST. Making segment measuring sequences in SM and then projecting three measuring sequences as three maps and extracting three maximals in CP.

Let X, Y be 0–1 sequences with N elements, and the ST module takes the sequence X as input, then shift r position on the whole sequence to be the shifted sequence $Y = X(r)$ (i.e., a cyclic shift right + or shift left –).

$$Y = X(r), Y[I] = X[I \pm r], I \pm r(\text{mod } N),$$

$$0 \leq I < N; X[I], Y[I] \in \{0, 1\} \quad (1)$$

In the SM module, the shifted vector is inputted and will be divided from a long sequence into M segments. For the i -th sub-vector, $0 \leq i < M$ on the j -th position $0 \leq j < m$, denoted as $Y_{i,j}$.

This sequence at the end of sub-vectors after the segmenting operation forms an $m * M$ matrix, m positions for the i -th complete row vector in the sequence correspond to a pair of 2-tuple measures: (p_i, q_i) .

$$Y = \{Y_i\}_{i=0}^{M-1} \quad (2)$$

$$Y_i = \{Y_{i,0}, Y_{i,1}, \dots, Y_{i,j}, \dots, Y_{i,m-1}\} \quad (3)$$

$$0 \leq i < M, 0 \leq j < m$$

$$Y_i \rightarrow (p_i, q_i), 0 \leq i < M \quad (4)$$

$$\{Y_i\}_{i=0}^{M-1} \rightarrow \{(p_i, q_i)\}_{i=0}^{M-1} \quad (5)$$

The pair of 2-tuple measures (p_i, q_i) is determined by the following formula:

$$Y_{i,j} = Y[J] \in \{0, 1\}; J = i * m + j, \\ 0 \leq i < M, 0 \leq j < m, 0 \leq J < m * M \quad (6)$$

$$p_i = \sum_{j=0}^{m-1} Y_{i,j}, Y_{i,j} \in \{0, 1\}, 0 \leq p_i \leq m; \quad (7)$$

$$q_i = \sum_{j=0}^{m-1} [(Y_{i,j-1}, Y_{i,j}) == (0, 1)], \\ j - 1(\text{mod } m), 0 \leq q_i \leq \lfloor m/2 \rfloor; \quad (8)$$

That is, $X = 0011010010$, $N = 10$, $M = 2$, $m = 5$; $(p_0 = 2, q_0 = 1)$; $(p_1 = 2, q_1 = 2)$.

The SM outputs the ordered M pairs of 2-tuple measures $\{p_i, q_i\}_{i=0}^{M-1}$.

The CP module consists of two units: Split and projection. The split adapts the SM's output as the input, and the 2-tuple measuring sequence $\{(p_i, q_i)\}_{i=0}^{M-1}$ will be splitted into two independent measuring sequences: $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$ to keep the original order invariant.

Three measure sequences are $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$, $\{(p_i, q_i)\}_{i=0}^{M-1}$.

The projection unit consists of three steps: Project Array (PA), Color Map (CM), and Get Maximal (GM). For three measuring sequences, two types of 1D and 2D measures will be processed separately.

The PA processes measuring sequences to transform them into integer arrays and the CM will organize them on either normalized histograms (1D measures) or color maps (2D measures), respectively.

The 1D measures involve two measuring sequences: $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$. Let $P[m+1]$, $Q[\lfloor m/2 \rfloor + 1]$ and $NP[m+1]$, $NQ[\lfloor m/2 \rfloor + 1]$ be two 1D (integer, float) arrays to represent the corresponding elements.

The 1DP statistic histogram is generated from a sequence $\{p_i\}_{i=0}^{M-1}$, NP , P two arrays (floating point, integer) with $(m+1)$ elements. For the j -th element $NP[j]$, $P[j]$, $0 \leq j \leq m$, and $1DP_x$ the maximal element, the output can be obtained by following procedure:

$$\begin{aligned} \text{Initialization: } & \forall NP[j] = 0.0, \\ & P[j] = 0, 0 \leq j \leq m; \\ \text{Calculation: } & \text{for}(i = 0; i < M; i++) \\ & \{P[p_i] ++;\} \\ \text{Normalization: } & \text{for}(j = 0; j \leq m; j++) \\ & \{NP[j] = P[j]/M;\} \\ \text{Get Maximal: } & 1DP_x = \max\{NP[j] | 0 \leq j \leq m\} \end{aligned}$$

In the 1DP map, the PA corresponds to initialization and calculation; the MA handles normalization and the GM identifies the maximal element of the map.

The 1DQ statistic histogram is generated from a sequence $\{q_i\}_{i=0}^{M-1}$, NQ , Q two arrays (floating point, integer) with $(\lfloor m/2 \rfloor + 1)$ elements. For the j -th element $NQ[j]$, $Q[j]$, $0 \leq j \leq \lfloor m/2 \rfloor$, and $1DQ_x$ the maximal element, the output can be obtained from following procedure:

Initialization: $\forall NQ[j] = 0.0$,
 $Q[j] = 0, 0 \leq j \leq \lfloor m/2 \rfloor$;
 Calculation: *for* ($i = 0; i < M; i ++$)
 $\{Q[q_i] ++;\}$
 Normalization: *for* ($j = 0; j \leq \lfloor m/2 \rfloor; j ++$)
 $\{NQ[j] = Q[j]/M;\}$
 Get Maximal: $1DQ_x = \max\{NQ[j] | 0 \leq j \leq \lfloor m/2 \rfloor\}$

Using P , NP , Q , NQ arrays, it is possible to generate corresponding 1D statistical histograms as 1D maps.

In the 1DQ map, the PA corresponds to initialization and calculation; the MA handles normalization and the GM identifies the maximal element of the map.

The 2D measures specially processes one measuring sequence: $\{(p_i, q_i)\}_{i=0}^{M-1}$. Let $PQ[m+1 : \lfloor m/2 \rfloor + 1]$ be a 2D integer array.

2DPQ statistic histogram is generated from a sequence $\{(p_i, q_i)\}_{i=0}^{M-1}$, PQ a 2D integer array with $(m+1) * (\lfloor m/2 \rfloor + 1)$ elements; For the i, j -th element $PQ[i, j]$, $0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$, and $1DPQ_x$ the maximal element, their values can be obtained by following procedure:

Initialization: $\forall PQ[i, j] = 0$,
 $0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$;
 Calculation: *for* ($i = 0; i < M; i ++$)
 $\{PQ[p_i, q_i] ++;\}$
 Pseudo-color: Matching proper color for
 $\forall PQ[i, j], 0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$
 Get Maximal: $1DPQ_x = \max\{PQ[i, j] | 0 \leq i \leq m,$
 $0 \leq j \leq \lfloor m/2 \rfloor\}$

In the 2DPQ map, the PA corresponds to initialization and calculation; the MA handles pseudo-color and the GM identifies the maximal element of the map.

Through the CP module, three measuring sequences are transformed into two 1D arrays and one 2D array with $(m+1)$, $(\lfloor m/2 \rfloor + 1)$ and $(m+1) * (\lfloor m/2 \rfloor + 1)$ clusters.

The outputs of the testing system are three maps $\{1DP, 1DQ, 2DPQ\}$ and three maximals $\{1DP_x, 1DQ_x, 2DPQ_x\}$ as expected statistic distributions and representatives of the input 0–1 sequence, respectively.

3 Association Analysis

It is a counting scheme to sort the $\{p_i\}_{i=0}^{M-1}$ measuring sequence as a 1D histogram. When the measuring sequence meets ideal conditions, the 1D statistical distribution is a binomial distribution.

Lemma 1 *For an input 0–1 sequence, if the total number of segments is equal to $M = 2^m$, and each segment of m bits appears only once in the sequence, then the 1DP array satisfies the binomial distribution*

$$p[i] = \binom{m}{i}, 0 \leq i \leq m \quad (9)$$

Corollary 1 *If the input sequence meets the conditions of Lemma 1, then the total number of items in the 1DP array is equal to*

$$\sum_{i=0}^m p[i] = 2^m = M \quad (10)$$

Lemma 2 *If the input sequence meets the conditions of Lemma 1, then the 1DQ array satisfies following relation:*

$$Q[i] = 2 \binom{m}{2i}, 0 \leq i \leq \lfloor m/2 \rfloor \quad (11)$$

Corollary 2 *If the input sequence meets the conditions of Lemma 1, then the total number of items in the 1DQ array is equal to*

$$\sum_{i=0}^{m/2} Q[i] = 2^m = M \quad (12)$$

Corollary 3 *For any 0–1 sequence with N elements, a 2DPQ projection in two directions is corresponding to either a 1DP array or a 1DQ array, respectively.*

Proof A 2DPQ array is generated from a measuring sequence $\{p_i, q_i\}_{i=0}^{M-1}$ and the 2DPQ array is sorted by $\{PQ[i, j]\}_{i=0}^m \binom{\lfloor m/2 \rfloor}{j=0}$, from two directions $P[i] = \sum_{j=0}^{\lfloor m/2 \rfloor} PQ[i, j], 0 \leq i \leq m; Q[j] = \sum_{i=0}^m PQ[i, j], 0 \leq j \leq \lfloor m/2 \rfloor$. So two projections are corresponding to an either 1DP or 1DQ array.

Corollary 4 *For an arbitrary 0–1 input sequence, the total number of items in the 2DPQ array is equal to*

$$\sum_{i=0}^m \sum_{j=0}^{\lfloor m/2 \rfloor} PQ[i, j] = \sum_{i=0}^m P[i] = \sum_{j=0}^{\lfloor m/2 \rfloor} Q[j] = M \quad (13)$$

In Corollaries 3 and 4, the total number of each component on three statistic arrays is equal to the total number of segments M , and the 2DPQ array occupies a central position in the projection to other two arrays.

Let $\{1DP_x(r), 1DQ_x(r), 2DPQ_x(r)\}$ denote three maximal on the selected sequence for $0 \leq r \leq m$; three maximal sequences are $\{1DP_x(r)\}_{r=0}^m$, $\{1DQ_x(r)\}_{r=0}^m$, $\{2DPQ_x(r)\}_{r=0}^m$.

For a 0–1 sequence with M segments, if each segment of m bits is composed of a state and only one state is involved, then the sequence is a circular sequence.

Lemma 3 *For a sequence $0 \leq r \leq m$, the sequence is a circular sequence, iff $1DP_x(r) = 1DQ_x(r) = 1$ and $2DPQ_x(r) = M$.*

Proof For a circular sequence, shift operations do not change the pair of measures, only a single (p, q) value is possible.

Theorem 1 *For a sequence with stationary random properties, it has*

$$\begin{aligned} 1DP_x(0) &\simeq \dots \simeq 1DP_x(r) \simeq \dots \simeq 1DP_x(m) \ll 1, \\ 1DQ_x(0) &\simeq \dots \simeq 1DQ_x(r) \simeq \dots \simeq 1DQ_x(m) \ll 1, \text{ or} \\ 2DPQ_x(0) &\simeq \dots \simeq 2DPQ_x(r) \simeq \dots \simeq 2DPQ_x(m) \ll 1. \end{aligned}$$

Proof In any random condition, it is necessary for pairs of $\{(p, q)\}$ to have certain states significantly different from a circular sequence in either $\ll 1$ or $\ll M$ condition. Under the stationary random condition, all maximal satisfy only \simeq relations under shift operations.

For a G map, let G_x be an average variation, ΔG_x be a region of variations, and $G_x^R = \Delta G_x / G_x$ be a variation ratio.

Theorem 2 *For two $\{i, j\}$ -th G maps G^i and G^j on $G_x^i \simeq G_x^j$ with variation ratios $G_x^{i,R}$ and $G_x^{j,R}$, if a variation ratio has a minimal value, then the relevant map has a better stationary random property than the maximal one.*

Proof Since $G_x^R = \Delta G_x / G_x$ and $G_x^i \simeq G_x^j$, it is a relative measure on $\forall r(\max\{G_x(r)\} - \min\{G_x(r)\}) / G_x \geq 0$. So $\min\{\Delta G_x^i, \Delta G_x^j\} \leq \max\{\Delta G_x^i, \Delta G_x^j\}$, the minimal variation ratio indicates the better stationary random property.

Corollary 5 *For different maps, it is better to compare various variation ratios relevant to the same type of distributions.*

Proof For various maps in the same type of distributions, relevant $\{G_x\}$ should satisfy the similar–equal condition.

4 Testing Results

Four pseudorandom sequences are generated by {A5,RC4,DES, AES} ciphers, and two quantum cryptographic sequences are selected from both ANU and USTC resources.

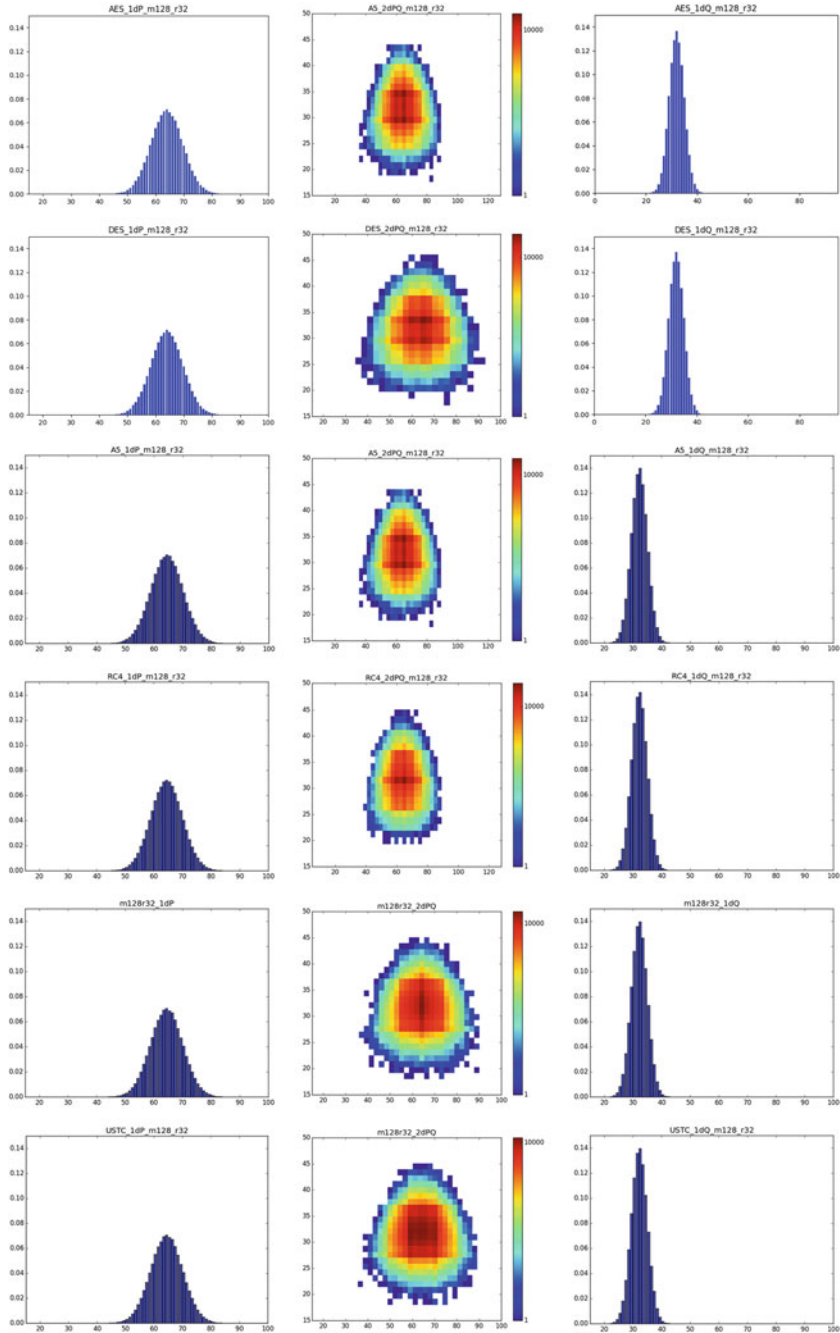


Fig. 4 Six cryptographic sequences on $r = 32$ 1DP, 2DPQ, and 1DQ maps

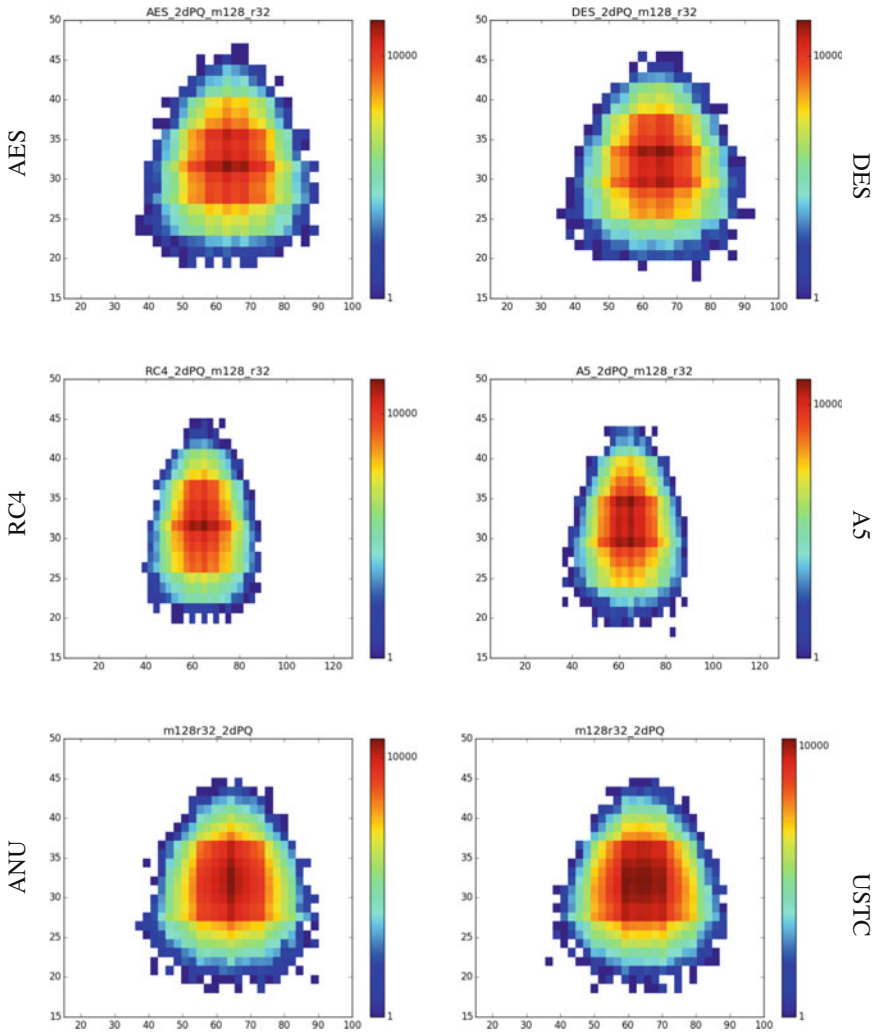


Fig. 5 Six cryptographic sequences on $r = 32$ 2DPQ maps

Typical results of testing stationary properties for six sequences on 18 maps of $\{1DP, 2DPQ, 1DQ\}$ are shown in Fig. 4. Each position contains nine shift values of $r = 32$ selected. A total number of 18 maps are included. Six 2DPQ maps are shown in Fig. 5 as enlarged maps. Each map has shift values of $r = 32$, respectively.

Three variation measures $\{G_x, \Delta G_x, G_x^R\}$ for maps $\{1DP, 2DPQ, 1DQ\}$ of six sequences are shown in Table 1, and their sorted orders are listed in Table 2. Twenty-four 2D maps of maximal curves for $r = 0 - 128$ are shown in Table 3. Three left columns contain 18 enlarged variation maps of $\{1DQ, 1DP, 2DPQ\}$ and the last column contains six variation regions of $1DQ + 1DP + 2DPQ$ in six 2D maps. Six enlarged 2D maps are shown in Table 4 and six larger 2D maps are shown in Table 5.

In Table 6, 49 pairs of differences for variation ratios are listed in three 7×7 tables to illustrate refined quantity measures on three levels. There are seven entries on diagonals with seven trivial 0 values. For other 42 nontrivial values, let $dG_x^R\%$ denote differences of $G_x^R\%$ based on the basic variation ratios in Table 1, and various differences of variation ratios among six samples are listed. Differences of three variation ratios $\{dQ_x^R\%, dP_x^R\%, dPQ_x^R\%, \}$ on seven items $\{\emptyset, \text{AES}, \text{DES}, \text{A5}, \text{RC4}, \text{ANU}, \text{USTC}\}$ are illustrated.

5 Result Analysis

Eighteen maps in Fig. 4 are composed of three groups. Six 1DP maps have similar distributions in bell shapes to illustrate Poissonian distributions. Six 2DPQ maps are

Table 1 Comparisons on three variation measures for six samples

1DQ:	$Q_x\%$	$\Delta Q_x\%$	$Q_x^R\%$
AES:	14.05	0.42	3.0
DES:	14.05	0.36	2.55
A5:	13.953	0.19725	1.4136
RC4:	14.210	0.21985	1.5471
ANU:	13.961	0.17761	1.2722
USTC:	13.944	0.19664	1.4102
1DP:	$P_x\%$	$\Delta P_x\%$	$P_x^R\%$
AES:	7.07	0.42	3.96
DES:	7.05	0.25	3.5
A5:	7.02650	0.17665	2.51409
RC4:	7.19459	0.16223	2.25498
ANU:	7.0352	0.15472	2.1992
USTC:	7.0289	0.13542	1.9265
2DPQ:	$PQ_x\%$	$\Delta PQ_x\%$	$PQ_x^R\%$
AES:	1.0	0.09	9.02
DES:	1.0	0.08	8.21
A5:	0.98690	0.05508	5.5818
RC4:	1.02754	0.05106	4.96913
ANU:	0.99245	0.04791	4.8276
USTC:	0.98675	0.04691	4.7544

Table 2 Possible sorted orders of three types of variation measures; (a) $G_x\%$, (b) $\Delta G_x\%$, (c) $G_x^R\%$

$G_x\%$	min	max	min - max sorted	min-max range
1DQ:	USTC	RC4	USTC-A5-ANU-AES-DES-RC4	$13.944 \leq Q_x\% \leq 14.21$
1DP:	A5	RC4	A5-USTC-ANU-DES-AES-RC4	$7.0289 \leq P_x\% \leq 7.19459$
2DPQ:	USTC	RC4	USTC-A5-ANU-DES-AES-RC4	$0.98675 \leq PQ_x\% \leq 1.02754$

(a)

$\Delta G_x\%$	min	max	min - max sorted	min-max range
1DQ:	ANU	AES	ANU-USTC-A5-RC4-DES-AES	$0.17761 \leq \Delta Q_x\% \leq 0.42$
1DP:	USTC	AES	USTC-ANU-RC4-A5-DES-AES	$0.13542 \leq \Delta P_x\% \leq 0.42$
2DPQ:	USTC	AES	USTC-ANU-RC4-A5-DES-AES	$0.04691 \leq \Delta PQ_x\% \leq 0.09$

(b)

$G_x^R\%$	min	max	min - max sorted	min-max range
1DQ:	ANU	AES	ANU-USTC-A5-RC4-DES-AES	$1.2722 \leq Q_x^R\% \leq 3.0$
1DP:	USTC	AES	USTC-ANU-RC4-A5-DES-AES	$1.9265 \leq P_x^R\% \leq 3.96$
2DPQ:	USTC	AES	USTC-ANU-RC4-A5-DES-AES	$4.7544 \leq PQ_x^R\% \leq 9.02$

(c)

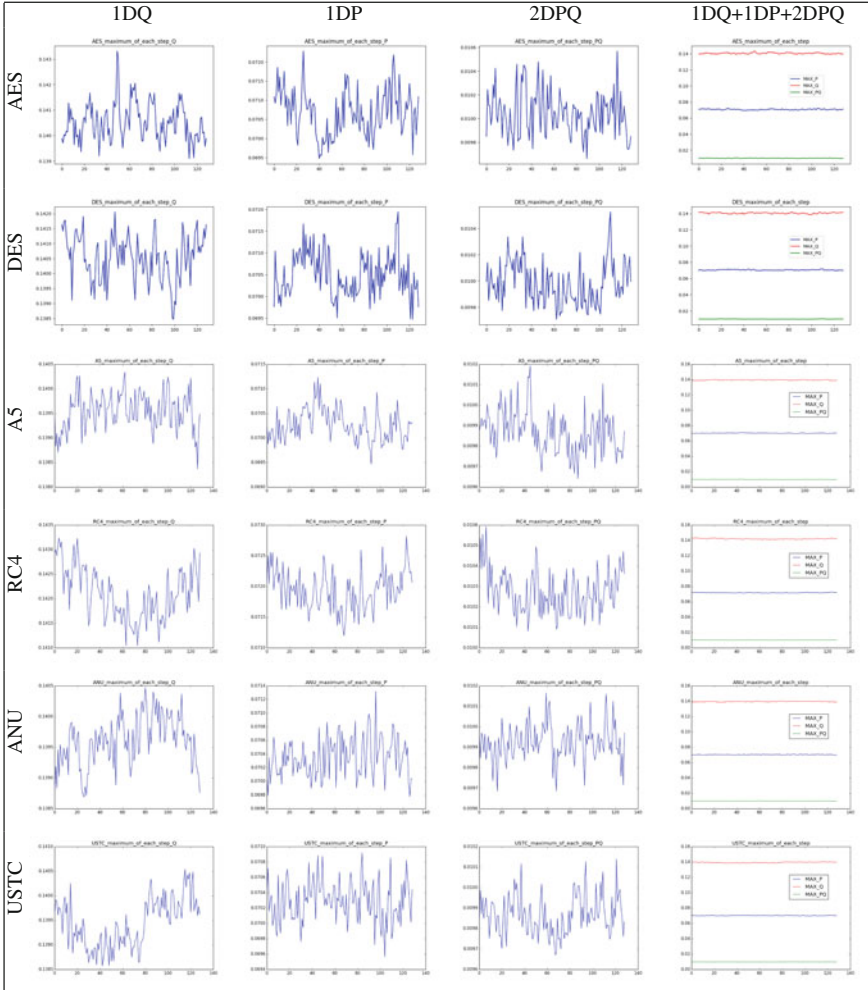
2D distributions. They have a symmetry on left/right directions and have a broken symmetry on up/down directions. Pseudo-color pixels on six maps indicate relevant 3D shapes. Compared with six 1DP maps, six 1DQ maps have similar distributions and more narrow bell shapes to illustrate sub-Poissonian distributions. It is possible to illustrate different maps on shift $r = 32$ for each map.

In Table 1, three pairs of maximal and minimal variation ratios are identified and three full orders are sorted in Table 2. Compared with G_x sorted orders, both $\{\Delta G_x, G_x^R\}$ variation ratios, six samples keep the same sorted orders as two groups: 1DQ and {1DP, 2DPQ} for their min-max variation ratios. Six enlarged 2DPQ maps on shift $r = 32$ are shown in Fig. 5 to form three pairs {AES:DES, RC4:A5, ANU:USTC}. Three pairs of six maps have similar visual distributions.

Twenty-four variation maps are shown in Table 3 as four groups. Each group contains six 2D maps. For three groups of {1DQ, 1DP, 2DPQ} variation distributions, eighteen enlarged 2D maps are shown in significant waveforms. For the group of 1DQ + 1DP + 2DPQ distributions, six maps are shown in three average variations satisfying $1DQ_x > 1DP_x > 2DPQ_x$, respectively. The fourth group of variation measures combines three variations of 1DQ + 1DP + 2DPQ in one unified 2D maps. From the six 2D maps, their stationary randomness of global variations are clearly illustrated.

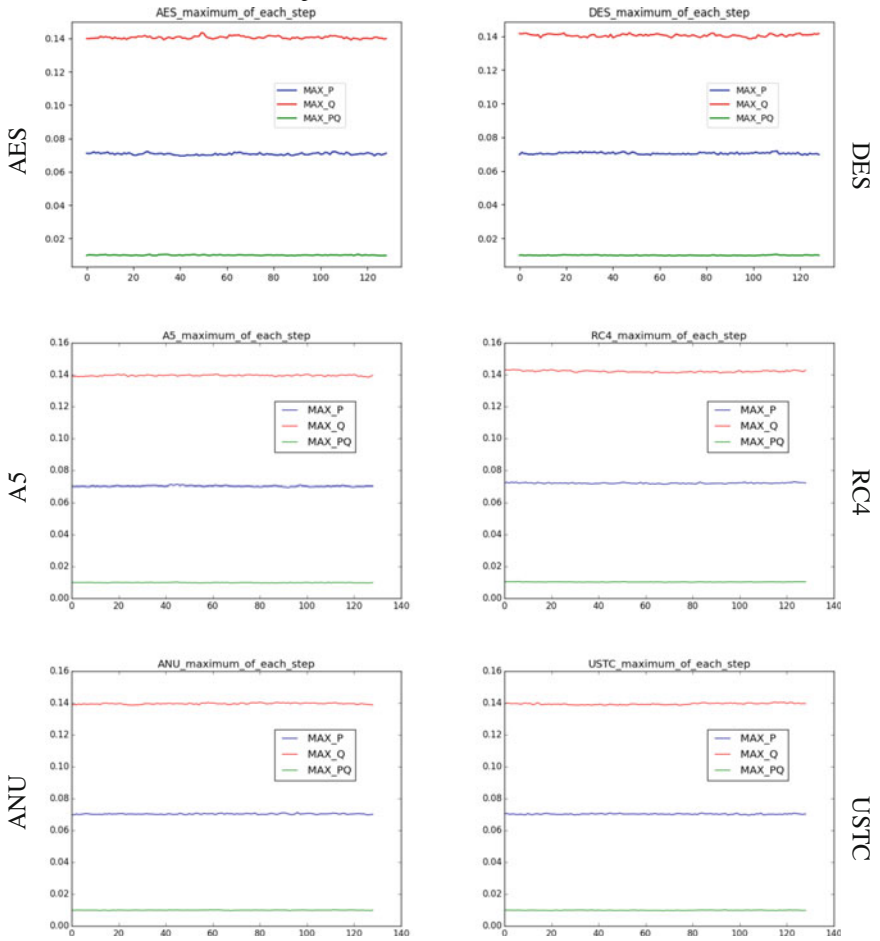
In Table 4, AES and DES map may have high frequent waves, and other enlarged 2D maps have stationary properties. In Table 5, larger waves appear and more details could be identified. Although significant variations are appeared in different 2D maps, it is difficult to make classification depending on their variation behaviors.

Table 3 Variation distributions of six samples



In Table 6, three variation ratios of differences are bounded in $0.0034 \leq |dQ_x^R\%| \leq 1.73$, $0.056 \leq |dP_x^R\%| \leq 3.96$, and $0.073 \leq |dPQ_x^R\%| \leq 4.27$, respectively. In general, three groups of variation ranges on differences meet $\{dQ_x^R\% \} \subset \{dP_x^R\% \} \subset \{dPQ_x^R\% \}$. From a stationary testing viewpoint, 2DPQ shows the strongest distinct property, 1DQ has the weakest numeric property, and 1DP provides the middle identifying property.

Table 4 Six variations on 2D maps



Since three groups can be identified by {AES, DES} block ciphers, {A5, RC4} stream ciphers, and {ANU, USTC} quantum ciphers, stationary randomness quantities can be classified as three {AES, DES}-highest, {A5, RC4}-middle, and {ANU, USTC}-lowest categories to provide distinct variation measures in the testing. Three quantity categories may correspond to distinguish artificial, semi-artificial, and natural designs for various generating mechanisms of cryptographic resources.

Considering all differences of variation ratios on six samples listed in Table 6, there are only 0.0034–4.27% differences (thirty-four in one million to four percent) are recognized. From a measuring viewpoint, all six samples are showing distinct stationary randomness properties.

Table 5 Larger six variations on 2D maps

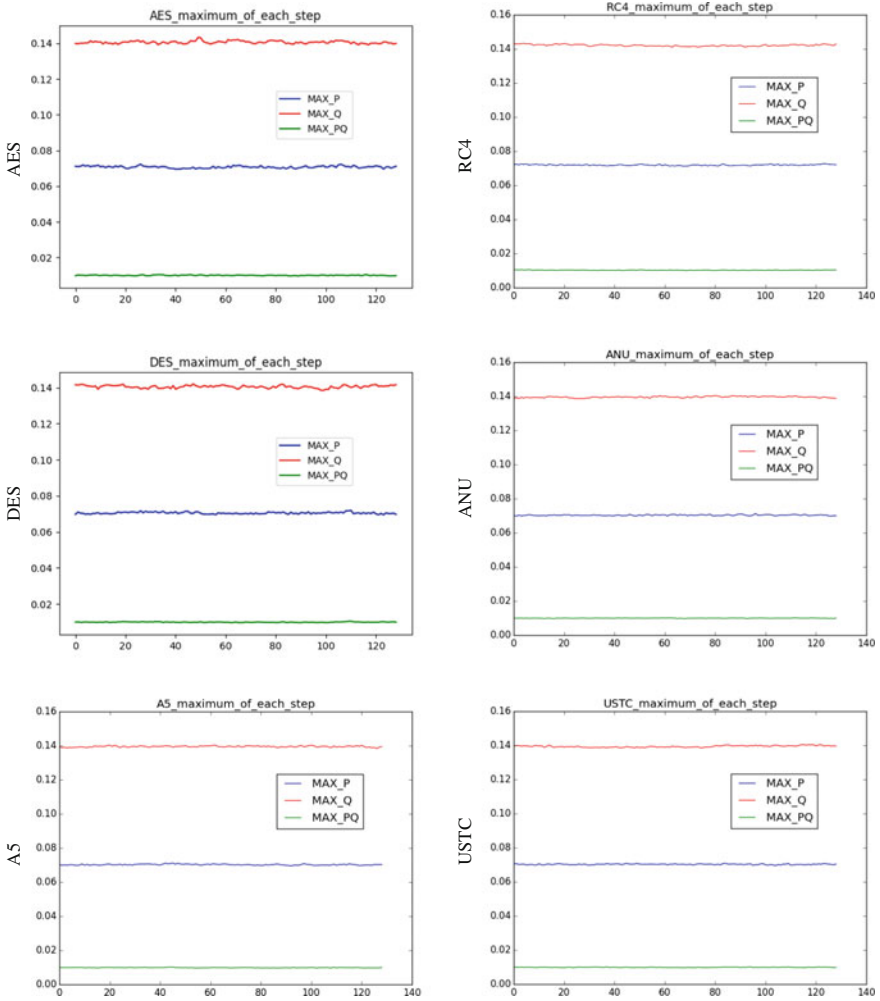


Table 6 Differences of variation ratios among three maximals of six samples

$dQ_x^R\%$	\emptyset	AES	DES	A5	RC4	ANU	USTC
\emptyset	0	-3.0	-2.55	-1.4136	-1.5471	-1.2722	-1.4102
AES	3.0	0	0.45	1.5864	1.4529	1.7278	1.5808
DES	2.55	-0.45	0	1.1364	1.0029	1.2778	1.1398
A5	1.4136	-1.5864	-1.1364	0	-0.1335	0.1414	-0.0034
RC4	1.5471	-1.4529	-1.0029	0.1335	0	0.2749	0.1369
ANU:	1.2722	-1.7278	-2.2778	-0.1414	-0.2749	0	-0.138
USTC:	1.4102	-1.5898	-1.1398	-0.0034	-0.1369	0.138	0

$dP_x^R\%$	\emptyset	AES	LFSR	A5	RC4	ANU	USTC
\emptyset	0	-3.96	-3.5	-2.51409	-2.25498	-2.1992	-1.9265
AES	3.96	0	0.46	1.44591	-0.54996	1.7608	2.0335
DES	3.5	-0.46	0	0.98591	1.24502	1.3008	1.5735
A5	2.51409	-1.44591	-0.98591	0	0.25911	0.31489	0.58759
RC4	2.25498	0.54996	-1.24502	-0.25911	0	0.05578	0.32848
ANU:	2.1992	-1.7608	-1.3008	-0.31498	-0.05578	0	0.2727
USTC:	1.9265	-2.0335	-1.5735	-0.58759	-0.32848	-0.2727	0

$dPQ_x^R\%$	\emptyset	AES	DES	A5	RC4	ANU	USTC
\emptyset	0	-9.02	-8.21	-5.5818	-4.96913	-4.8276	-4.7544
AES	9.02	0	0.81	3.4382	4.05087	4.1924	4.2656
DES	8.21	-0.81	0	2.6282	3.24087	3.3824	3.4556
A5	5.5818	-3.4382	-2.6282	0	0.61267	0.7542	0.8274
RC4	4.96913	-4.05087	-3.24087	-0.61267	0	0.14153	0.21473
ANU:	4.8276	-4.1924	-3.3824	-0.7542	-0.14153	0	0.0732
USTC:	4.7544	-4.2656	-3.4556	-0.8274	-0.21473	-0.0732	0

6 Conclusion

It is feasible to evaluate stationary properties for a random sequence using the testing system. Using three maps {1DP, 1DQ, 2DPQ}, a series of variation measures and their ratios are illustrated. Extracting maximal measures is identified for shift $r : 0 - m$. For each sample, three 2D maps of variation curves provide refined characteristics to evaluate stationary randomness properties in global. Sample variation maps are shown in exactly similar–equal relationships among the same group of average variations. Further explorations and applications are required to check

the testing system on other applications of cryptographic streams. Three quantity categories of artificial, semi-artificial, and natural designs may be explored to get intrinsic stationary randomness information from refined testing and future explorations.

Acknowledgements Thanks to National Science Foundation of China (61362014) and High Level Overseas Professional Project of Yunnan Province for financial supports to this project. Thanks to the Key Laboratory of Quantum Information, USTC, CAS and the ANU Quantum Random Numbers Server for quantum cryptographic sequences.

References

1. Cyberspace: <https://en.wikipedia.org/wiki/Cyberspace>
2. Communications Security: https://en.wikipedia.org/wiki/Communications_security
3. Quantum satellite: <https://qz.com/760804>
4. Quantum key distribution: https://en.wikipedia.org/wiki/Quantum_key_distribution
5. Random number generation: https://en.wikipedia.org/wiki/Random_number_generation
6. S. Golomb, *Shift-Register Sequences*, Revised edn. (Aegean Park Press, Laguna Hills, California, 1982)
7. E. Barkam, E. Biham, N. Keller, Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication. *Journal of Cryptology* **21**(3), 392429 (2008)
8. Y. Lu; W. Meier; S. Vaudenay. The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption. *Crypto 2005*. 3621: 97117 (2005)
9. eStream: <https://en.wikipedia.org/wiki/ESTREAM>
10. P. Junod & A. Canteaut (2011). *Advanced Linear Cryptanalysis of Block and Stream Ciphers*. IOS Press. p. 2. ISBN 9781607508441
11. ZUC. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 and 128-EIA3: Document 2: ZUC Specification
12. A. Poorghanad, A. Sadr, A. Kashanipour, generating high quality pseudo random number using evolutionary methods. *IEEE Congress on Comput. Intell. Security* **9**, 331–335 (2008)
13. A. de Queiroz, J. Schechtman, Elimination of nonlinear clock feedthrough in component-simulation switched-current circuits in *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, pp. II378–II381 (1998)
14. A. Fuster-Sabater and F. Vitini. Classes of Nonlinear Filters for Stream Ciphers, Chapter *Geometry, Algebra and Applications: From Mechanics to Cryptography*,
15. S. Ronjom, C. Cid, Nonlinear Equivalence of Stream Ciphers. in *Proceeding of Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, Lecture Notes in Computer Science*, vol. 6147, Springer, pp. 40–54 (2010)
16. J. Nechvatal, E. Barker, L. Bassham, et al., Report on the development of the advanced encryption standard (AES), in *National Institute of Standards and Technology (NIST)* (2000). <http://csrc.nist.gov/archive/aes/round2/r2report.pdf>
17. G. Paul, S. Maitra. *RC4 Stream cipher and Its Variants*. (CRC Press, 2012)
18. S.D. Cardell, A. Fuster-Sabater, Linear models for the self-shrinking generator based on CA. *Journal of Cellular Automata* **11**(23), 195211 (2016)
19. N. Nagaraj, One-time pad as a nonlinear dynamical system. *Commun. Nonlinear Sci. Numer. Simul.* **17**, 4029–4036 (2012)
20. E. Dubrova, M. Teslenko and H. Tenhunen. On analysis and synthesis of (n,k)-non-linear feedback shift registers, *Proceedings of the conference on Design, automation and test in Europe*, 1286-1291, 2008

21. E. Dubrova, A List of Maximum Period NLFSRs. Cryptology ePrint Archive, Report 2012/166, 2012
22. Y. Zhao, Y. Hu, S. Li, A new analysis method for nonlinear component of stream ciphers. *J. Inf. Comput. Sci* **10**(16), 5313–5321 (2013)
23. Meschede, D. *Optics, Light and Lasers*, 2 ed. (Wiley-VCH, 2007)
24. R. Boyd. *Nonlinear Optics* (3rd ed.). Academic Press (2008)
25. M. Nakazawa et al., QAM quantum stream cipher using digital coherent optical transmission. *Opt. Express* **22**(4), 4098–4107 (2014)
26. M. Yoshida et al., Single-channel 40 Gbit/s digital coherent QAM quantum noise stream cipher transmission over 480 km. *Opt. Express* **24**(1), 652–661 (2016)
27. J. Barry, E. Lee, *David G* (Messerschmitt. Digital Communications, Springer, 2004)
28. S. Lian et al., A chaotic stream cipher and the usage in video protection. *Chaos Solitons and Fractals* **34**(3), 851–859 (2007)
29. D. E. Knuth, *The Art of Computer Programming*, vol. 2: *Seminumerical Algorithms* (Addison-Wesley, 1969)
30. D. Makovoz, Noise variance estimation in signal processing, in *International Symposium on Signal Processing and Information Technology*, pp. 364–369 (2006)
31. Ito, Kazufumi. Gaussian filter for nonlinear filtering problems. *Conference on decision and control* (2000): 1218-1223
32. F. Orieux, O. Feron, J. Giovannelli, Sampling High-Dimensional Gaussian Distributions for General Linear Inverse Problems. *IEEE Signal Process. Lett.* **19**(5), 251–254 (2012)
33. M. Fox, *Quantum Optics: An Introduction* (Oxford University Press, New York, 2006)
34. M.B. Priestley. *Non-linear and Non-stationary Time Series Analysis* (Academic Press, 1988))
35. A. Kolmogorov, https://en.wikipedia.org/wiki/Andrey_Kolmogorov
36. A.N. Kolmogorov (1903–1987). *Royal Netherlands Academy of Arts and Sciences*. Retrieved 22 July 2015
37. D.C. Champeney, *Power spectra and Wiener's theorems* (Cambridge University Press, A Handbook of Fourier Theorems, 1987)
38. N. Wiener, Generalized harmonic analysis. *Acta Math.* **55**, 117258 (1930). <https://doi.org/10.1007/bf02546511>
39. N. Wiener, *Time Series Press* (M.I.T Press, Cambridge, 1964)
40. Stationary process: https://en.wikipedia.org/wiki/Stationary_process
41. Stationary: <https://www.otexts.org/fpp/8/1>
42. A5/1 stream cipher: <https://en.wikipedia.org/wiki/A5/1>
43. R. Rivest, J. Schuldt, Spritz: a spongy RC4-like stream cipher and hash function. Retrieved 26 October 2014
44. A.E. Ivanova et al., Using optical splitters in quantum random number generators based on fluctuations of vacuum. *J. Phys.: Conf. Ser.* **735**, 012077 (2016)
45. X.T. Song et al., Phase-Coding Self-Testing Quantum Random Number Generator. *Chin. Phys. Lett.* **32**(8), 080302–080310 (2015)
46. T. Symul, S.M. Assad, P.K. Lam, Real time demonstration of high bitrate quantum random number generation with coherent laser light. *Appl. Phys. Lett.* **98**, 231103 (2011)
47. W. Chen et al., Active phase compensation of quantum key distribution system. *Chinese Science Bulletin* **53**(9), 1310–1314 (2008)
48. M. Sasaki et al., Field test of quantum key distribution in the Tokyo QKD network. *Opt. Express* **19**(11), 10387–10409 (2011)
49. Quantum random number generator of ANU: <http://photonics.anu.edu.au/qoptics/Research/qrng.php>
50. Z.J. Zheng. *Conjugate transformation of regular plan lattices for binary images*, Ph.D. Thesis, Monash University (1994)
51. Z.J. Zheng, C.H.C. Leung, Visualising global behaviour of 1D cellular automata image sequences in 2D Map. *Phys. A* **3–4**, 785–800 (1996)
52. D.E. Knuth. *The Art of Computer Programming*, vol. 4A: *Combinatorial Algorithms Part 1* (Addison-Wesley, 2011)

53. J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Frontiers of Electr. Electron. Eng. China*, **5**(2), 163–172. Higher Educational Press and Springer-Verlag (2010). 10.1007/s11460-010-0011-4 <http://link.springer.com/article/10.1007/s11460-010-0011-4>
54. H. Wang, J. Zheng, 3D Visual Method of Variant Logic Construction for Random Sequence. *Aust. Inf. Warfare Security* 16–27 (2013)
55. W.Z. Yang, J. Zheng, Variant pseudo-random number generator, *Hakin9 Extra. Timing Attack* **06**(13), 28–31 (2012)
56. J. Zheng, Novel pseudo-random number generation using variant logic framework, in *2nd International Cyber Resilience Conference*, 10bit04. 2011. <http://igneous.scis.ecu.edu.au/proceedings/2011/icr/zheng.pdf>
57. J. Zheng, C. Zheng, Variant simulation system using quaternion structure. *J. Modern Opt.* **59**(5), 484–492 (2012) Taylor & Francis Press
58. J. Zheng, C. Zheng, T.L. Kunii, Interactive maps on variant phase space, in *Emerging Application of Cellular Automata*, pp. 113–196 (InTech Press, 2013)
59. J. Zheng, W. Zhang, J. Luo, W. Zhou, R. Shen, Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Mathe.* **3**(7A), 5–24 (2013)
60. D.M. Heim, O. Heim, P.A. Zeng, J. Zheng, Successful creation of regular patterns in variant maps from bat echolocation calls. *Biological Systems: Open Access* **5**, 2 (2016). <https://doi.org/10.4172/2329-6577.1000166>
61. Daemen, Joan; Rijmen, Vincent (March 9, 2003). *AES Proposal: Rijndael*. National Institute of Standards and Technology. p. 1. Retrieved 21 February 2013
62. LFSR scheme: https://en.wikipedia.org/wiki/Linear-feedback_shift_register
63. NIST. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* (NIST, Special Publication, 2010)
64. M. Soltanalian, P. Stoica, Computational design of sequences with good correlation properties. *IEEE Trans. Signal Process.* **60**(5), 2180–2193 (2012)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part IV

Theoretical Foundation—Meta Model

TAO produced the First—[Heaven].
The First produced the Second—[Earth].
These Two produced the Third.
The Third produced all things,
and these turn their back upon the Yin and embrace the Yang.
The intermingling of these two Affliti results in harmony.
—Lao Tzu (*Tao Te Ching*)

Knowledge has the form of a tree, and since metaphysics is the most fundamental one of the theoretical disciplines, it represents the roots of the tree.

—Gonzalo Rodriguez-Pereyra

Meta-design is much more difficult than design; it's easier to draw something than to explain how to draw it.

—Donald Knuth

From a historical viewpoint, the meta model was developed early than variant logic that provides useful concept and hierarchical organization to support this new logic framework. The core paper of concept cell (Concept Cell Model for Knowledge Representation) was published in *Int. J. Inf. Acquisition* 01, 149–168 (2004), World Scientific Press. In relation to multiple probability approach, a research paper (Voting Theory for Multiple Candidates to Resolve Intrinsic Uncertain Problems of Election) was published in *Journal of System Engineering Theory and Practices (Chinese)* 1000-6788(2002)12-0101-10. This paper proposed a useful multiple probability model to resolve intrinsic uncertain properties in election.

Part IV is composed of two chapters (9 and 10).

Chapter “[Meta Model on Concept Cell](#)” outlines a meta model on concept cell for knowledge representation to provide a brief core structure on this network topology scheme for three levels of knowledge clusters.

Chapter “[Voting Theory for Two Parties Under Approval Rule](#)” describes voting theory for two parties under approval rule to show multiple probability model also useful in two-party conditions.

Meta Model on Concept Cell



Jeffrey Zheng and Chris Zheng

Abstract Applying network topology schemes, two types of three levels of meta knowledge representations have been established. This chapter proposes a meta model on concept cell that provides a meta organisation of knowledge in natural and artificial intelligent systems structurally.

Keywords Knowledge model · Meta representation · Three levels of concept lattice · Description · Procedure · Core organisation

1 Introduction

A meta model on concept cell is outlined to represent knowledge in knowledge systems (KSs). This model has novel features that are of considerable interest for knowledge representation (KR).

Polanyi proposed a knowledge model in the 1940s. Knowledge is composed of two categories: tacit and explicit [1, 2]. In the 1970s, Anderson from a cognitive psychology identified knowledge with another two categories: declarative and procedural [3–5]. In the early 1990s, a procedural model was proposed by Nonaka who identified four transformations: tacit → tacit (socialisation), tacit → explicit (externalisation), explicit → explicit (combination) and explicit tacit (internalisation) [6, 7]. In 2000, a model was proposed by Nickols to arrange four classes (tacit, explicit, procedural and declarative) into three categories: tacit, explicit and implicit. In my opinion, the Nickols model is unsatisfactory for three reasons:

This work was supported by Australian Commercialising Emerging Technologies, (COMET) program.

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

C. Zheng

Tahto, Sydney, Australia
e-mail: z@caudate.me

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_9

- (i) it is a triangle of categories without a fixed order,
- (ii) there is uncertainty in implicit category and
- (iii) there is no structural correspondence to other KR methodologies.

To improve the first two weaknesses of Nickols approach, an executable knowledge model was proposed. A triplet (tacit, implicit and explicit) is constructed as a procedural structure. Implicit in it is the middle node linked with two other nodes in four transformations: tacit \rightarrow implicit (externalisation), implicit \rightarrow explicit (retrieval), explicit \rightarrow implicit (category) and implicit \rightarrow tacit (internalisation). In addition, the model provides distinguishable foreground/background and human/machine knowledge interfaces [8].

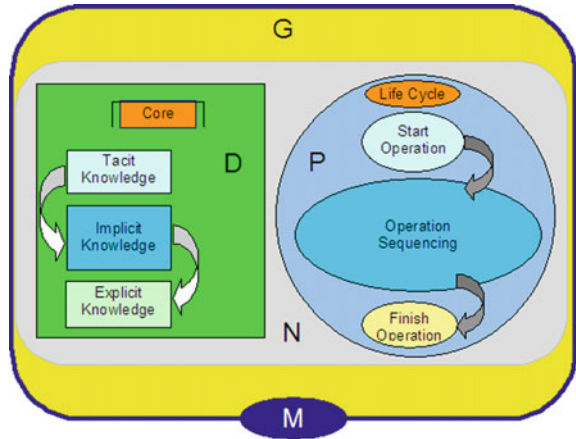
To explore different KS applications from philosophy, logic and digital libraries, to gene, chemistry, software and system engineering [9–11], people arrange common concepts to construct ontology libraries and procedures as core structures [12, 13]. Advanced system modelling tools such as ARIS [14], CIMOSA [15] and IDEF [16] provide function, data and process models and ontology description capture methodologies for constructing modern intelligent knowledge systems [17]. Because many contradictions, confusions, difficulties and unclear properties exist in KR foundation levels [13, 18, 19], consistently categorising practical knowledge into tacit/explicit and procedural/declarative is extremely hard for researchers, scientists, philosophers, psychologists and knowledge workers [14–17, 20, 21].

Practical computer-aided modelling systems use pragmatic approaches to manipulate simple structures (list, tree, stack, class and component) in real applications [14–17, 21]. Usually, declarative concepts seem easier to capture than procedural concepts. Based on this, many people believe that declarative knowledge is explicit and procedural knowledge is tacit [16, 17, 22]. A radical extension of a knowledge model in KR is proposed in a concept cell that arranges knowledge in KS for natural and artificial organisation. This model can fully support the above-mentioned knowledge models to consistently identify four categories of knowledge: tacit, explicit, declarative and procedural. The model also provides a core ontology to distinguish a hierarchy of structures within the core of a concept. According to convention, the word concept is used as an equivalent to knowledge in this chapter.

2 Concept Cell Model

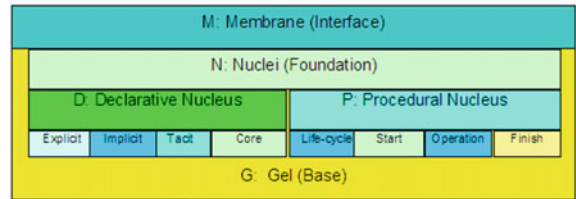
Let K denote a cell of concepts (a concept cell) that is composed of three parts: M membrane, N nuclei and G gel. M is a frame that provides a container to hold both N and G . G is a base description of the content and N establishes a foundation of the cell. M inputs provide external concepts (externals) for N from deeper levels, and then output current content to other upper level cells. N is composed of two components: D declarative nucleus and P procedural nucleus. To illustrate this organisation, a cell $K = M, N, G$ is shown in Fig. 1.

Fig. 1 A concept cell K. **a** A slice, **b** hierarchy



$$K = \{M, N, G\}$$

(a) A cell K



Here it is

M = interface; N = {D, P}; G = base;

D = {explicit, implicit, tacit, core};

P = {life-cycle, start, operation, finish}

Above it is

M membrane; N Nuclei; G Gel;

D Declarative Nucleus;

P Procedural Nucleus

(b) Ontology of K

For the convenience of construction, a special lattice is employed [23]. Only directed graphs are used similar to the most popular signal flow graphs [24] to analyse and syntheses process control [9], computer architecture [10], electric circuits [25], network topology [26–28] and dynamic systems [25, 29]. However, no lattices allow containing a loop and all lattices are composed of directed acyclic graphs [26, 28]. In a lattice, a node represents a cell and lattice links are determined by dependencies among nodes. Because the most complex part of a cell is its nuclei structures, detailed interior organisation is necessary to explore meanings of knowledge. To simplify, a simple cell (or a cell, if there is no confusion) is studied here, where nuclei of the cell are composed of only one declarative lattice and one procedural lattice.

Using lattice language, a cell K is described in Fig. 2. Different graphic symbols represent distinct forms of concepts as nodes. A rounded rectangle represents a general node; an octagon is a specific node; a rectangle shows a declarative node and an oval corresponds to a procedural node. A simple lattice cell is composed of

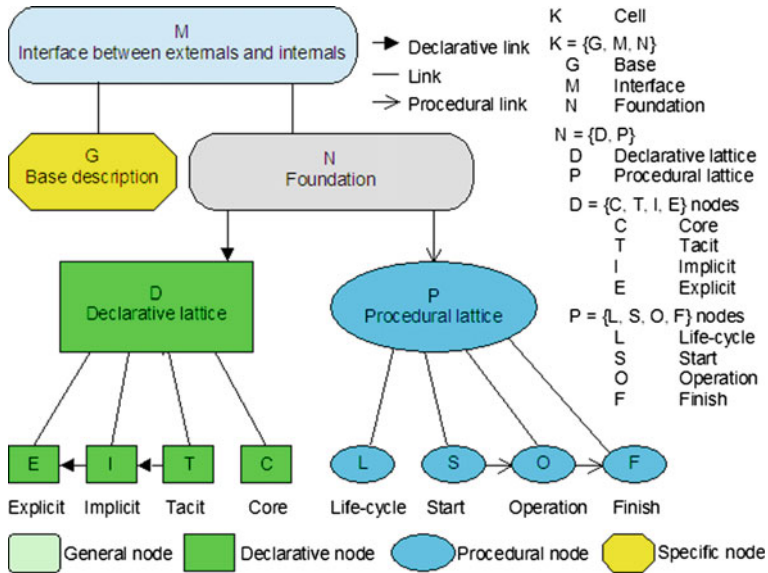


Fig. 2 A concept cell in lattices

four levels: A node M that interfaces between externals and internals are the first level. Two nodes of G and N link with an M node is the second level. The node G contains the base description and the node N plays a foundation role in the cell. Two nodes of D and P link with node N on the third level. Node D contains one lattice in declarative dependency and node P contains one lattice that assumes procedural dependency. Finally, two sets of nodes linked with nodes D and P at the fourth level. Each node of D or P contains four nodes, respectively. Among each four nodes, two links are associated with three nodes.

3 Core Components

The following four conditions can create the content of a concept cell:

- (i) M acts as an interface to import a finite number of externals into nuclei and to export the content to other cells.
- (ii) G provides the base description of the cell and N collects all externals from M for development.
- (iii) Two lattices D, P are constructed from Ns externals to carry out two dependencies.

An N external corresponds to a D node. A declarative dependency is employed to order all nodes of D as a declarative lattice. If two distinct nodes have declarative dependency, then the node with more general meanings is located at the first node and a declarative link connects from the first to the second. After building up declarative dependency among all nodes, D becomes a directed acyclic lattice.

Instances of an N external correspond to nodes of P satisfying procedural dependency. P is composed of sequences of nodes by instances of externals. If two instances represent two nodes, then the node that has to be handled earlier is specified as the first node and a procedural link connects two nodes from the first to the second. After all procedural dependencies are established among nodes, P is converted into a directed acyclic lattice.

(iv) Two lattices are composed of eight distinguishable node sets:

Four sets of declarative nodes C, T, I, E are identified: C core, T tacit, I implicit and E explicit, respectively.

Four sets of procedural nodes L, S, O, F are identified: L life cycle, S start, O operation and F finish.

The meanings of the construction process can be explained as: In the first level of kernel, M collects all externals to provide extra knowledge for its nuclei. The second level has two parts: G, N. The G node provides the base description. To map each external as a node, the number of N externals has the same number of nodes in D. A declarative dependency is valid for all D nodes that create a directed acyclic declarative lattice. Using instances of N externals as nodes, P has been assembled using procedural dependency linked with selected nodes and finally to form P itself as procedural lattice. Since both declarative and procedural lattices are organised by ordered dependencies, declarative and procedural lattices are directed acyclic to support wider requirements from theoretical foundations to practical applications. A simple construction example is shown in Fig. 3(i–v).

For an acyclic lattice, four distinct node sets are notable in Fig. 3(vi). They are (singleton, source, branch and sink) node sets, respectively, borrowed from network topology [23, 26, 30]. A singleton node provides an isolated concept. A source node exports a concept. A sink node imports concept(s) and a branch node transfers concept(s) from input link(s) to output link(s). If there is only one external in N, then the singleton set contains one single node and the other three sets are empty. If there is more than one node in N, then the singleton set is empty. In this case, the source set is composed of nodes that have at least one link to another node; however, a source node does not have a link from other nodes. Each node must have at least one in branch, or sink set consequently. In contrast to the source set, a sink set collects all nodes with links from other nodes, without a link to a node. A sink node has to be the last node in a node path of a lattice to which at least one node is linked, from branch or source set. Unlike source and sink sets, a node in a branch set may link with at

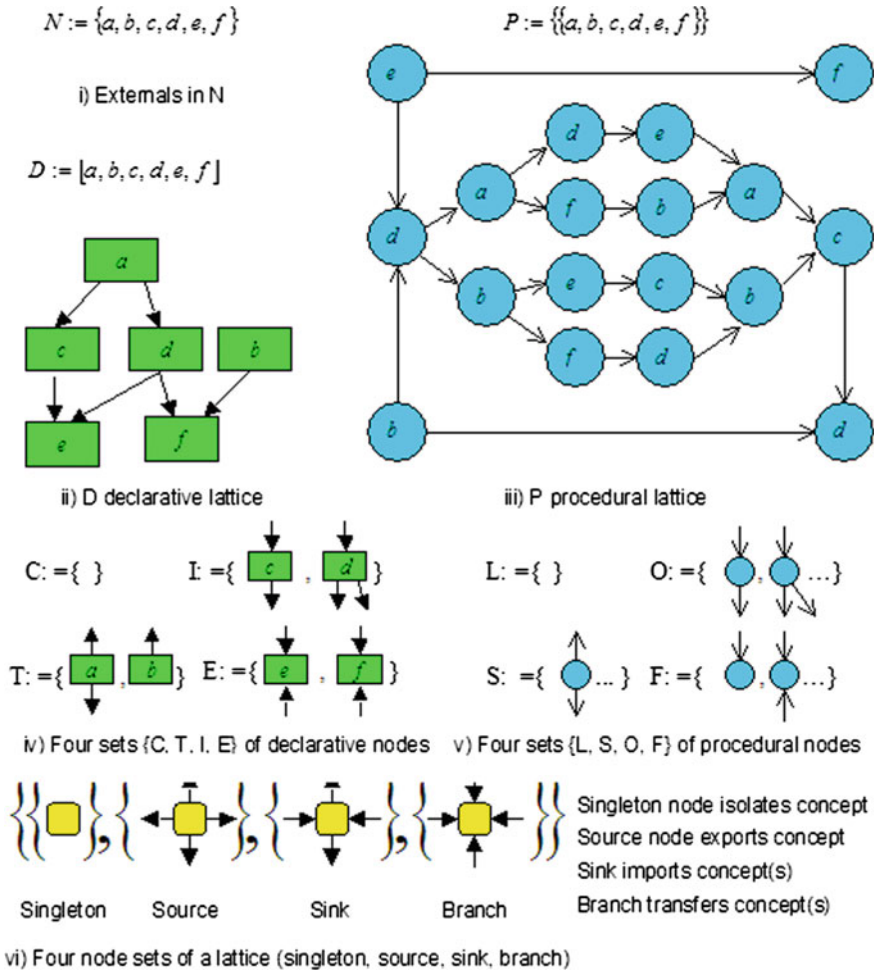


Fig. 3 External concepts, declarative and procedural lattices and node sets

least two nodes to and from source, sink, and branch sets. A branch node receives from other node(s) and outputs to other node(s). These sequence nodes provide connectivity among nodes. Although four node sets can be identified by their different connectivity, it is not convenient to use the same vocabulary to describe two distinct lattices under different dependencies. For convenience, each node set includes a proper name to indicate its specific relationship in familiar KR terms. D lattice represents an invariant structure (the simplest cases: tree, list) similar to a traditional data structure hierarchy. Because a sink node is equivalent to a factor data at the leaf level (at the lowest location) of data structure, the sink node has to be represented as an explicit knowledge. Therefore, the sink set of D is explicit. In contrast, a source node provides invaluable knowledge from the highest level of externals. There is

no link to this node and anyone wanting to explain the meaning of the node must capture knowledge from other sources far beyond the node itself. Consequently, a source node always contains deeper meanings than those can be articulated. Hence, the source set of D is tacit. Different from sink and source sets, a node in a branch set has connectivity from higher tacit node(s) and to higher explicit node(s). The branch set of D represents a typical intermediate property. Consequently, the branch set of D is implicit. A singleton node provides a complete concept. The node itself is the central of the D lattice. Therefore, the singleton node set of the D lattice is a core. Four node sets of P lattice satisfy different properties. The P lattice has a close relationship to process modelling that provides a time arrow as controllable sequences. A node in the P lattice is an instance of a node in the D lattice. The singleton node set of the P lattice is not empty if only one node is in the P lattice. The singleton node set of procedural lattice represents a complete procedure of P itself. Logically, the procedural singleton node set is a life cycle. When two or more nodes are included, three node sets of the P lattice have to link together in sequential relationships. Time relevant sequences in finite numbers of connected nodes, must have distinguishable commence and end nodes that correspond to start and finish conditions respectively. In addition, all intermediate nodes provide operational capacities to deliver knowledge to consequent nodes. Consequently, three node sets of the P lattice are called: start, finish and operation, respectively. The relative properties of the cell model with other schemes are compared in Table 1. In the table, TM represents Theoretical Model that is used in KS applications. ST denotes Structural Theory that uses structured organisations to represent complex dependency among members. ES indicates Engineering Systems that provide mixed theories, experiences and skills with commercial system modelling tools for pragmatic applications especially in enterprise management, manufacturing and building industries, software and hardware systems, global communication networks, web and Internet environment. ES applies advanced TM methodologies plus business experiences and engineering skills to solve practical problems efficiently using system engineering methodologies in global business explorations.

From this comparison, it is clear that existing systems that are the most similar to the cell concept model come from enterprise modelling that provides all functionality for ten meta nodes from engineering practices. However, other theoretical models cannot support full functionality. This property indicates the potential capacity for applying the cell concept model from theoretic foundations to practical applications. Details of the concept cell have published [31] to represent further classifications, recursive constructions, non-simple cells and sample applications for knowledge construction systems.

Table 1 Comparisons on different models

Model	D	T	I	E	C	P	S	O	F	L	Notes
Concept Cell	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	A hierarchy of four levels Tacit and Explicit
Polanyi[1,2]		TM		TM							
Anderson[3-5]	TM					TM					Declarative and Procedural
Nonaka[6,7]							ST		ST		Four Transformations
Nickols[8]		TM	TM	TM							Tacit, Explicit and Implicit
Zheng et al.[9]							ST	ST	ST		Four Transformations
Lattice Theory[17,28]		TM	TM	TM			TM	TM	TM		Theoretical Model
Ontology Metalogic[13-15]	TM	TM	TM	TM	TM						Logic, Metaphysics, ...
Conceptual graphs[15]						TM	TM	TM	TM	TM	Logic reasoning in graphs
/First order logic[15]							ES	ES	ES	ES	/Symbol notations
Enterprise Modelling [11-14,27]	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES	ARIS, IDEF ...
Object Oriented[13,27]	ES				ES	ES	ES	ES	ES	ES	OTM, UML, C++, Java
Function/.../Logic[13,27]						ES	ES	ES	ES	ES	Algol, Fortran, Lisp, Prolog

Ten basic symbols: {D, T, I, E, C}, { P, S, O, F, L}

D: Declarative; T: Tacit, I: Implicit, E: Explicit, C: Core;

P: Procedural; S: Start, O: Operation, F: Finish, L: Life cycle

Three types of models: {ST, TM, ES}

ST: Structural theory

TM: Theoretical model

ES: Engineering system

References

1. M. Polanyi, *Knowing and Being* (The University of Chicago Press, Chicago, 1969)
2. M. Polanyi, *Tacit knowledge (the tacit dimension)*, in *Knowledge in Organizations* (Butterworth-Heinemann, Boston, 1997), pp. 135-146
3. J.R. Anderson, *Language, Memory and Thought* (Erlbaum, Hillsdale, 1976)
4. J.R. Anderson, *Rules of the Mind* (Erlbaum, Hillsdale, 1993)
5. J.R. Anderson, *Cognitive Psychology and Its Implications* (W.H. Freeman and Company, New York, 1995)
6. I. Nonaka, *The Knowledge Creating Company*, Harvard Business Review (November-December 1991), pp. 96-104
7. I. Nonaka, *The Knowledge Creating Company* (Oxford University Press, 1995)
8. J.Z.J. Zheng, M. Zhou, J. Mo, A. Tharumarajah, Background and foreground knowledge in knowledge management, in *Global Engineering, Manufacturing and Enterprise Networks* (Kluwer Academic Publisher, 2001), pp. 332-339
9. K. Hartmann, K. Kaplick, *Analysis and Synthesis of Chemical Process Systems* (Elsevier, Amsterdam, 1990)
10. D. Agrawal, *Advanced Computer Architecture* (IEEE Computer Society Press, Los Alamitos, 1986)
11. J. Cuena, *Knowledge Oriented Software Design* (North-Holland, Amsterdam, 1993)
12. A.P. Sage, W.B. Rouse, *Handbook of Systems Engineering and Management* (Wiley, New York, 1999)
13. W. Ziarko (ed.), *Rough Sets* (Fuzzy Sets and Knowledge Discovery (Springer, Berlin, 1994)
14. A.W. Scheer, *Architecture of Integrated Information Systems: Foundation of Enterprise Modelling* (Springer, 1992)

15. P. Bernus, L. Nemes, T.J. Williams, *Architecture for Enterprise Integration* (Chapman & Hall, New York, 1996)
16. IDEF Family of Methods A Structured Approach to Enterprise Modeling and Analysis (IDEF0 5). <http://www.idef.com>
17. P. Bernus, K. Mertins, G. Schmidt (eds.), *Handbook on Architectures of Information Systems* (Springer, New York, 1998)
18. P. Hjek, T. Havrnek, R. Jirouek, *Uncertain Information Processing in Expert Systems* (CRC Press, Boca Raton, 1992)
19. I. Graham, P. Jones, *Expert Systems-Knowledge, Uncertainty and Decision* (Chapman and Hall, London, 1988)
20. T. Davenport, L. Prusak, *Working Knowledge* (Harvard Business School Press, Boston, 1998)
21. J. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations* (Brooks Cole Publishing, Pacific Grove, 2000)
22. F. Nickols, *Knowledge in Knowledge Management, Knowledge Management Yearbook* (Butterworth-Heinemann, 2000), pp. 12–21. http://home.att.net/~nickols/Knowledge_in_KM.htm
23. G. Birkhoff, *Lattice Theory* (Providence, 1967)
24. J.R. Abrahams, G.P. Coverley, *Signal Flow Analysis* (Pergamon Press, 1965)
25. G. Lago, L.M. Benningfield, *Circuit and System Theory* (Wiley, New York, 1979)
26. S. Chan, *Introductory Topological Analysis of Electrical Networks* (Holt, Rinehart and Winston Inc, 1969)
27. R. Clay, *Nonlinear Networks and Systems* (Wiley, New York, 1971)
28. W. Chen, *Linear Networks and Systems* (Brooks/Cole Engineering Division, New York, 1983)
29. W.E. Lewis, D.G. Pryce, *The Application of Matrix Theory to Electrical Engineering* (E&FN Spon, London, 1965)
30. H. Hopf, *Differential Geometry in the Large* (Springer, Berlin, 1983)
31. J. Zheng, C. Zheng, T.L. Kunii, *Int. J. Inf. Acquisition* **01**, 149 (2004). <https://doi.org/10.1142/S021987890400015X>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Voting Theory for Two Parties Under Approval Rule



Jeffrey Zheng

Abstract The Simple Ballot Model (SBM) and the Component Ballot Model (CBM)—are proposed for solving uncertainty in an election when two candidates gain the same number of votes under the approval rule. The SBM establishes a framework to support counting. In separating the two candidates, it is essential to extract additional information from dominantly valid votes. The CBM uses probability matrices, vectors and permutation group as components. A stable-voting mechanism under permutation invariant can be created to distinguish candidates. The result of the chapter establishes a voting authority to resolve uncertainty of two candidates under the approval rule.

Keywords Approval rule · Permutation invariant · Feature vector · Uncertainty Voting system

JEL Classifications D72 · D81 · C34 · C31

1 Introduction

As a common practice in a modern democratic society, voting is a practical way to resolve a contest where each candidate seeks to gain maximal support from the electors. Approval voting is a voting procedure in which electors can vote for as many candidates as they wish. Each candidate approved of receives one vote and the candidate with the most votes wins. Approval voting, unlike more complicated ranking systems, is easier and simpler for electors to understand and use. This voting

This work was supported by Yunnan Advanced Overseas Scholar Project and Yunnan National Science & Technology Foundation(2004F0009R).

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_10

method has been widely used today by various governments and organizations around world (including the use by the United Nations to elect the secretary-general).

To keep healthy economic and political progress in modern democracy societies, it is necessary to apply reliable and convenient voting methodologies and tools to ensure fairness, efficiency and transparency and to overcome paradoxes and difficulties in elections.

1.1 Brief Review of Voting Systems

We can find interesting voting-based models and practices in many ancient stories from Chinese literature to Roman and Greek history. Just before the French revolution in the French Academy, de Borda [1] and de Condorcet [2] proposed the *Borda rule* and the *Condorcet procedures*. They wanted to use new voting methods to resolve difficulties and unfair results under traditional plurality-based voting rules in elections for the Academy. In 1920s, Hotelling [3] investigated the *equilibrium* of spatial economic competition for two firms between location and price. During World War II, von Neumann and Morgenstern [4] developed *Theory of Games* using differential equations to investigate complicated competition behaviors. This theoretical foundation has a superior influence to develop analytical methodologies and tools from applying pre-designed strategic policies to predicting practical election outcomes. Under fairness conditions, Arrow [5] proved his famous *Impossibility Theorem* which claims that there is no single election procedure to fairly decide the outcome of an election involving more than three candidates. Various ideas, methods and technologies have emerged to resolve voting difficulties [6–9].

1.2 Problems in the 2000 American Election

The most debatable problem in the 2000 American election, the 2K-election, is that

Whether the machine-rejected ballots need to be manually recounted?

The practical solution of the 2K-election problem was finally decided by the nine judge's votes in the US Supreme Court on the lawsuits from the Florida Supreme Court.

This indicates that current voting theories and vote-counting models are all faults to be an authority resolving the problem.

Although the 2K-election is under the plurality rule, not under the approval rule, however the approval rule cannot guarantee to avoid the similar uncertainty when a large number of electors are involved. It is necessary to establish relevant theoretical structure to avoid possible problems in the future.

1.3 Structure of the Chapter

This chapter proposes two models constructing a voting theory to resolve the 2K-election-like problems and other paradoxes in voting practices. Only one voting system under approval rule is concerned.

In Sect. 2, a Simple Ballot Model (SBM) is proposed. Using the SBM, the separable and uncertain conditions for the ballot papers are established. To show some practical strategies and relevant problems in current voting methodologies, four additional rules (reducing error probability, merging other candidate votes, re-election, and court decision) that are commonly used in practical voting processes are discussed.

In Sect. 2.8, the error margin for the 2K-election problem is analyzed. Through voting practice is not an accurate science, but the error margin of 0.233% in the event still cannot be acceptable as an accurate measure. Although almost 99.8% of the valid votes were counted, there is still no way of determining that who is the winner. Therefore, the attentions shifts to the 0.2% votes which were already deemed invalid. This problem highlights that the voting system needs to improve, and a method of extracting additional information from valid votes to separate the two candidates under uncertainty conditions becomes essential.

In Sect. 3, a new voting model—the Component Ballot Model (CBM)—is defined and constructed to provide the essential construction for extracting more information from votes for comparisons. Based on multiple feature matrices (similar to contingency tables in classical statistics), probability feature vectors and permutation invariant group and other advanced mathematical tools, multiple pair sets of feature index families for two candidates are constructed. This mechanism establishes a voting authority to make a decision for an election. After the mathematical definitions and constructions to feature matrix, feature vector, probability feature vector and feature index, the most important results are summarized in *Two-D Separable Proposition* and *Voting Authority Proposition*.

Taking into account only the valid votes, the election model will have intrinsic stability for the reliable results immediately after the election. Confusion, frustration and dissatisfaction as those experienced in the 2K-election can be avoided.

In the light of this research, some further research directions are suggested in Sect. 4.

2 Simple Ballot Model

2.1 Key Words in Election

Key words used in an election event can be defined as follows.

- **Election**—a special event based on counting votes for a winner (normally whoever attracts the most votes wins the election)
- **Candidate**—a person who has been nominated in an election
- **Elector**—a person who may legally vote in an election
- **Ballot**—a pre-designed form used to record choices of an elector
- **Vote**—a ballot on which the choices of an elector are recorded
- **Poll**—the collections of votes from all legal electors
- **Decision**—A result on who wins the election.

The Simple Ballot Model simulates the simplest case scenario of whole voting procedure based upon all ballots directly collected from an election under approval rule. In this scenario, one elector can only create one vote for as many candidates selected from a list of candidates.

2.2 Definitions

For an ideal **election** involving n (≥ 2) **candidates**, let $C = \{c_1, c_2, \dots, c_n\}$ be a set of the selected candidates. A **ballot** $B = \langle c_1, c_2, \dots, c_n \rangle$ is a pre-designed form containing the list of candidates for whom the electors may vote.

A **vote** is a record of a ballot B . Let a vote denote v . It is valid if $v = \langle v_1, v_2, \dots, v_n \rangle$, $v_i \in \{0, 1\}$, $i \in [1, n]$, $\sum_{i=1}^n v_i > 0$, otherwise if $\exists v_i = x \notin \{0, 1\}$, $i \in [1, n]$ or $\sum_{i=1}^n v_i = 0$ (null selection), then the vote v is invalid; where $v_i = 1$ indicates selected the candidate c_i , $v_i = 0$ indicates not selected c_i and $v_i = x$ indicates invalid selection to c_i . Normally a vote v has a value region from $\langle 0, 0, \dots, 0 \rangle$ to $\langle 1, 1, \dots, 1 \rangle \dots \langle x, x, \dots, x \rangle$.

An elector can only create one vote and there are a total number of N ($\gg n$) votes in the election.

A poll V is a vote collection in which all votes can be arranged as an array with N entries:

$$V = (v(1), \dots, v(t), \dots, v(N)), \quad t \in [1, N]. \quad (2.1)$$

where $v(t)$ denotes the vote of the t th elector. As each candidate has a number, let $k \in v(t)$ denote the t th elector selected the k th candidate on the vote.

For example, $n = 6$, $N = 8$, a poll V is: $V = (v(1), \dots, v(t), \dots, v(8))$, $t \in [1, 8]$

$$\begin{aligned} v(1) &= \langle 0, 0, 1, 1, 0, 0 \rangle, v(2) = \langle 0, 1, 0, 1, 0, 0 \rangle, v(3) = \langle 0, 1, 0, 1, 1, 0 \rangle, \\ v(4) &= \langle 1, 0, x, 1, 1, 0 \rangle, v(5) = \langle 0, 1, 0, 1, 0, 0 \rangle, v(6) = \langle 0, 0, 1, 1, 1, 0 \rangle, \\ v(7) &= \langle 0, 0, 1, 0, 0, 0 \rangle, v(8) = \langle 0, 0, 0, 0, 0, 0 \rangle \end{aligned}$$

In this poll, $\{v(1), v(2), v(3), v(5), v(6), v(7)\}$ are valid votes ($v_3(1) = v_4(1) = 1$ indicates the 1-st vote selected the third and fourth candidates). In addition, $v(4)$

contains an uncertain selection ($v_3(4) = x$) and $v(8)$ is a null selection, both votes are invalid.

Let V_0 denote the invalid-poll in the election. It collects all invalid votes from the poll V . Let V_c denote a valid sub-poll in the election. Both sub-polls V_c and V_0 partition the poll V . i.e.

$$V = V_c \cup V_0.$$

Let V_k denote a sub-poll in the election. For any $k \in [1, n]$, V_k collects all valid votes from the poll V for the k th candidate.

$$V_k = \{v(t) | v_k(t) = 1, k \in [1, n], t \in [1, N], v(t) \in V_c\}.$$

Let \tilde{V} denote a poll vector,

$$\tilde{V} = (V_0, V_1, \dots, V_k, \dots, V_n), k \in [1, n]. \quad (2.2)$$

A SBM is a collection of a ballot form, all votes, poll and poll components for an election.

$$SBM = (B | V; \tilde{V}) \quad (2.3)$$

Let N_{V_c} denote the number of votes in the valid poll V_c , $N_{V_c} = |V_c|$. Let N_k denote the number of votes in the valid poll V_k , $N_k = |V_k|$, $k \in [1, n]$ and N_0 denote the number of votes in the invalid poll V_0 .

The total number of votes in an election, N , is equal to the sum of the number of the valid votes N_{V_c} plus the number of all invalid votes N_0 , i.e.

$$N = N_{V_c} + N_0. \quad (2.4)$$

Let $p_{V_c} = |V_c|/|V| = N_{V_c}/N$ denote a measure of the valid votes.

For any poll vector \tilde{V} , let $p_k = |V_k|/|V| = N_k/N$, $1 \leq k \leq n$ denote a measure of the k th candidate and $p_0 = |V_0|/|V| = N_0/N$ denote the measure of the invalid votes.

Under the approval rule, there are many overlaps among different sub-polls. Considering two candidate sub-polls and their common parts, if $\exists k, l \in [1, n]$, $V_k, V_l \subseteq V_c$, $V_k \cap V_l \neq \emptyset$, then

$$|V_k \cup V_l| = |V_k| + |V_l| - |V_k \cap V_l| \quad (2.5)$$

In general, we have

$$|V_k \cup V_l| \leq |V_k| + |V_l| \quad (2.6)$$

Let $\tilde{\Psi}$ denote a frequency vector,

$$\tilde{\Psi} = (p_0, p_1, \dots, p_k, \dots, p_n), \quad k \in [1, n] \quad (2.7)$$

2.3 One-Dimensional Feature Distribution

The frequency vector $\tilde{\Psi}$ corresponds to a density distribution. There are equations as follows.

$$1 = p_{V_c} + p_0; \quad (2.8)$$

$$1 \geq p_k \geq 0, \quad k \in [1, n]. \quad (2.9)$$

Because there is no further partition among sub-polls, the vector $\tilde{\Psi}$ is composed of a one-Dimension frequency feature histogram.

Considering inequalities (2.6), (2.8) and (2.9), there is an inequality.

$$1 \leq \sum_{k=0}^n p_k \leq n. \quad (2.10)$$

If sub-polls partition the poll, then there is $1 = \sum_{k=0}^n p_k$. In the worst case scenario, if all valid votes select all candidates without invalid votes, then

$$p_0 = 0, p_1 = \dots = p_n = 1, \quad \sum_{k=0}^n p_k = n$$

2.4 Separable Condition

When $\exists i, j \in [1, n], p_i, p_j > p_0$, a decision between the candidates i and j can be made if and only if

$$|p_i - p_j| > p_0 \quad (2.11)$$

This is the separable condition.

2.5 Uncertain Condition

However, there will be intrinsic difficulties to make a decision between the candidates i and j simply from their measures p_i and p_j , if

$$|p_i - p_j| \leq p_0 \quad (2.12)$$

This is the uncertain condition.

Under the uncertain condition, there are no simple solutions to distinguish signals clearly between p_i and p_j under the interference of p_0 .

2.6 *Balanced Opposites*

It is extremely hard to make any decision when both candidates gain the same number of votes in an election. However, for any equilibrium dynamic system involving two balanced opposites in competition, the most probable trends are $p_j = p_i$. In general, more complicated feedback mechanisms are involved and balanced events occur more frequently [10, 11].

2.7 *Four Additional Policies*

To resolve conflicts in an election, four additional policies may be useful: reducing error probability ($p_0 \rightarrow 0$), merging other candidate votes ($V_i \cup V_l \rightarrow V_i$ or $V_j \cup V_l \rightarrow V_j$; $i, j, l \in [1, n]$), re-election (new p_i, p_j) and court decision.

The reducing error probability policy works well in certain conditions involving only a small number of electors. Using various controlled methods, e.g., the total number of seats in Parliament being an odd number or some additional votes allowed by Parliament Leaders, the worst case scenario where both candidates hold equal votes without a decision can be eliminated. However, when an election involves a large number of electors like sizes of the 2K-election, the voting system becomes a naturally complex dynamic system and there is no way to make the error margin ($p_0 \rightarrow 0$) negligible.

The merging other votes policy works in simple conditions at a single location. To combine votes for candidates from multiple locations under approval rule would be more difficult than under plurality rules since there are many overlaps among sub-polls. There is no guarantee to ensure the policy work. In the best cases, old difficulties may be temporarily solved, but new similar uncertainties could immediately emerge.

From a complex-dynamic system, re-election is as same as the original election. Therefore, the re-election policy cannot provide improved separable property between two candidates.

If other solutions can not be found by timing or other issues, then it is feasible to use Courts to make decision. The court decision policy uses Courts to make decision, it results in efficient decision-making but breaks down the election procedure and it may lose fairness, transparency, self-determination and other advantages of the election process.

2.8 *How Accurate Is Accurate?*

It is well known that all measurements in physics and in all exact science are inaccurate in some degree. So, what then is sufficient to be deemed accurate for an election? Can we accept a 10% margin of error to be accurate? What about 1% or even 0.1%?

In real life, an error margin of 1% would be highly commendable and one of 0.1% would be considered highly accurate.

Although, voting and polling were not meant to be an exact science, polls and other pre-election statistics had error margin of almost 5–10%. Yet in the actual election, the margin of error was less in the disputed counties, e.g. Miami-Dade and Palm Beach, only 14,000 votes from a total number of six million votes were rejected. The margin of error was only 0.233%. Usually, this would be deemed a negligible number, as almost 99.8% of votes were valid. However, it was not enough to separate the two candidates, this margin would have to reduce the rejected votes from 14,000 to 100. In the condition, at least an error margin of 0.00016666% is required. This is highly improbable due to the cost, time and other factors.

2.9 *Shifting Attentions from Invalid Votes to Valid Votes*

Almost 99.8% votes are valid. This indicates that in order to determine who will be the winner under the uncertain condition, it is necessary to fetch additional information to determine a victor from valid votes instead of reducing the error margin by handling invalid votes. The total number of votes is far greater than the number of candidates. This makes possible to extract additional information using cross-classification methods based on contingency table-like techniques among multiple categories. The cross-classified technique is a powerful toolkit in modern statistics [12, 13, 14, 15].

Under additional categories such as location, age group and sex, valid votes will be categorized as two-dimensional classified feature distributions in respective contingency tables. Such spatial or histogram-like feature distributions provide invaluable information to support improving separable properties between two uncertain candidates. To represent this idea, a new model is proposed in next chapter.

3 **Component Ballot Model**

To overcome the intrinsic complexities and uncertain problems in approval voting practices, a new model—the Component Ballot Model—is proposed in this chapter to use multiple variables on a ballot for a better description and an easier comparison.

3.1 Definitions

To be consistent with the previous notation, similar symbols (ballot paper) are used. However, the contents of the ballot paper and other notations will be compounded into vector forms.

Let $C = \{C_1, C_2, \dots, C_m\}$ be a set of the selected conditions. The i -th item contains n_i distinct values for selections, $C_i = \langle c_1^i, \dots, c_j^i, \dots, c_{n_i}^i \rangle$, $j \in [1, n_i]$, $i \in [1, m]$.

A **ballot** B (or a **component ballot**) is a vector composed of m items:

$$B = \begin{pmatrix} C_1 \\ \dots \\ C_i \\ \dots \\ C_m \end{pmatrix} = \begin{pmatrix} \langle c_1^1, \dots, c_{n_1}^1 \rangle \\ \dots \\ \langle c_1^i, \dots, c_j^i, \dots, c_{n_i}^i \rangle \\ \dots \\ \langle c_1^m, \dots, c_{n_m}^m \rangle \end{pmatrix}, \quad j \in [1, n_i], i \in [1, m] \quad (3.1)$$

Component items in a ballot provide additional information about elector to the paper such as sex, voting time, location, age group, and minority, living area, social security and employ situations.

For example, the first item contains 10 candidates, the second item presents 100,000 locations, the third item has 3 sex groups (male, female, neutral), the fourth item contains 150 age groups, and the fifth item indicates 10^{10} social security number. Under above conditions, a ballot paper could be

$$B = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{pmatrix} = \begin{pmatrix} \langle c_1^1, \dots, c_{10}^1 \rangle \\ \langle c_1^2, \dots, c_{100000}^2 \rangle \\ \langle c_1^3, c_2^3, c_3^3 \rangle \\ \langle c_1^4, \dots, c_{150}^4 \rangle \\ \langle c_1^5, \dots, c_{10^{10}}^5 \rangle \end{pmatrix},$$

$$m = 5, n_1 = 10, n_2 = 100000, n_3 = 3, n_4 = 150, n_5 = 10^{10}.$$

A **vote** v (or a component vote) is a record of a component ballot B for which at least one value for each m items has been assigned:

$$v = \begin{pmatrix} v^1 \\ \dots \\ v^i \\ \dots \\ v^m \end{pmatrix} = \begin{pmatrix} \langle v_1^1, \dots, v_{n_1}^1 \rangle \\ \dots \\ \langle v_1^i, \dots, v_l^i, \dots, v_{n_i}^i \rangle \\ \dots \\ \langle v_1^m, \dots, v_{n_m}^m \rangle \end{pmatrix}, \quad v_l^i \in \{0, 1, x\}, l \in [1, n_i], i \in [1, m]. \quad (3.2)$$

where n_i is the upper limit of v^i ; $v_l^i = 1$ (or 0) means c_l^i candidate selected (or not selected), $v_l^i = x$ indicates c_l^i being an invalid value.

More items are provided for each ballot to include more information. Further distinctions of their valid regions are necessary. If for a vote v , the first item satisfies $i = 1$, $\sum_{l=1}^{n_i} v_l^1 \geq 1$ (more than one values selected) and all additional items satisfy $v_l^i \in \{0, 1\}$, $l \in [1, n_i]$, $i \in [2, m]$, $\sum_{l=1}^{n_i} v_l^i = 1$ (one and only one value selected), then the vote v is a **valid vote**. However, if $\exists i, l, v_l^i \in \{x\}$, $i \in [1, m]$, $l \in [1, n_i]$ or there is one v^i in additional items assigned multiple values, ($\exists i, v_l^i \in \{0, 1\}$, $\sum_{l=1}^{n_i} v_l^i > 1$, $l \in [1, n_i]$, $i \in [2, m]$) then v is an **invalid vote**.

Normally the valid first item in a vote has a value region from $\langle 0, 0, \dots, 0, 1 \rangle$ to $\langle 1, 1, \dots, 1 \rangle$. A total number of $2^{n_1} - 1$ combinations are valid to allow one, two or more candidates selected. However, for other additional items there is one and only one value selected from $\langle 0, 0, \dots, 0, 1 \rangle$ to $\langle 1, 0, \dots, 0, 0 \rangle$. There are only $n_i, i \in [2, m]$ selections allowed.

Additional information for electors may be accessed from existing election databases somewhere, there is no any technical difficulty to merge them to be a compound vote automatically using modern information technology.

There are enough rooms for an elector with various parameters on a vote and a total number of N electors in voting.

A **poll** V is a vote collection in which all votes can be arranged as an array with N entries:

$$V = (v(1), \dots, v(t), \dots, v(N)), \quad t \in [1, N]. \quad (3.3)$$

Considering each vote has m items, a poll V can be represented as a 2D $m \times N$ array.

$$\begin{aligned} V &= (v(1), \dots, v(t), \dots, v(N)) \\ &= \left(\begin{pmatrix} v^1(1) \\ \dots \\ v^i(1) \\ \dots \\ v^m(1) \end{pmatrix}, \dots, \begin{pmatrix} v^1(t) \\ \dots \\ v^i(t) \\ \dots \\ v^m(t) \end{pmatrix}, \dots, \begin{pmatrix} v^1(N) \\ \dots \\ v^i(N) \\ \dots \\ v^m(N) \end{pmatrix} \right) \quad t \in [1, N], i \in [1, m]. \end{aligned} \quad (3.4)$$

3.2 Feature Partition

Let V_c denote a valid poll and V_0 denote an invalid poll, V_c and V_0 partition the poll V i.e.

$$\begin{aligned} V_c &= \{\forall v|v \text{ is a valid vote, } v \in V\}; \\ V_0 &= \{\forall v|v \notin V_c, v \in V\}; \\ V &= V_c \cup V_0. \end{aligned} \quad (3.5)$$

Let V^i denote a sub-poll in the election. For any $i \in [1, m]$, V^i collects all valid votes of the poll V for the i th item.

$$\begin{aligned} V^i &= \left\{ \forall v(t)|v(t) \in V_c, v_l^i(t) \in \{0, 1\}, \sum_{l=1}^{n_i} v_l^i(t) \geq 1, \right. \\ &\quad \left. l \in [1, n_i], t \in [1, N], i \in [1, m] \right\} \end{aligned} \quad (3.6)$$

Zero-D Feature Lemma All $\{V^i\}_{i=1}^m$ sub-polls contain the same votes as in the poll V_c :

$$V_c = V^1 = V^2 = \dots = V^i = \dots = V^m \quad (3.7)$$

Proof Using Eqs. (3.5) and (3.6), a valid vote contains at least one valid value in each category. No difference exists to project all valid votes as one group. \square

Let V_k^i denote a sub-poll in the election. For any $i \in [1, m]$, V_k^i collects all valid votes of the poll V_c for the i th item in a special location k .

$$V_k^i = \{\forall v(t)|v(t) \in V_c, v_k^i(t) = 1, t \in [1, N], i \in [1, m], k \in [1, n_i]\} \quad (3.8)$$

One-D Feature Lemma All $\{V_k^i\}_{k \in [1, n_i]}$ sub-polls dissect a sub poll V^i :

$$V^i = \bigcup_{k=1}^{n_i} V_k^i \quad (3.9)$$

Proof By Eqs. (3.5)–(3.8), each vote has at least an identified value. To collect all votes with the value, we have the result. \square

One-D Feature Corollary If each vote contains only one value in the category item, then all sub-polls $\{V_k^i\}_{k \in [1, n_i]}$ partition a sub poll V^i :

$$|V^i| = \sum_{k=1}^{n_i} |V_k^i| \quad (3.10)$$

Proof By Eq. (3.9), each vote has an identified value. There is no overlap among possible sub-polls in relation to the category item. \square

It can be noticed that only candidate category does not satisfy one-D feature corollary under approval voting rule. Other additional categories satisfied the condition.

Different from the Zero-D feature lemma, the One-D feature corollary provides non-trivial partition of the votes into multiple sub polls.

Let V^0 denote an invalid-poll in the election. It collects all invalid votes of the poll V .

$$V^0 = \{\forall v(t)|v(t) \notin Vc, t \in [1, N]\} \quad (3.11)$$

Since there is no any further distinction for votes in V^0 , all votes in this poll correspond to discarded votes.

Let $V_{k,l}^{i,j}$ denote a sub poll. It can be described as

$$V_{k,l}^{i,j} = \left\{ \forall v(t)|v(t) \in Vc, v_k^i(t) = 1, v_l^j(t) = 1; \right. \\ \left. t \in [1, N], i, j \in [1, m], k \in [1, n_i], l \in [1, n_j] \right\} \quad (3.12)$$

For any $i, j \in [1, m], k \in [1, n_i], l \in [1, n_j]$, collected votes of $V_{k,l}^{i,j}$ are the same as the votes in $V_{l,k}^{j,i}$.

If $l \neq k$, then votes in $V_{k,l}^{i,j}$ are different from the votes in $V_{k,l}^{j,i}$.

Two-D Feature Lemma All votes in $\left\{ V_{k,l}^{i,j} \right\}_{k \in [1, n_i], l \in [1, n_j]}$ dissect either V_k^i or V_l^j .

$$V_k^i = \bigcup_{l=1}^{n_j} V_{k,l}^{i,j}; \quad (3.13a)$$

or

$$V_l^j = \bigcup_{k=1}^{n_i} V_{k,l}^{i,j}. \quad (3.13b)$$

Proof By Eq. (3.12) and one-D feature lemma, each vote in the sub-polls has other identified values. To collect all votes with the value in relevant sub-polls, we have the result. \square

Two-D Feature Corollary If a valid vote contains a single value in the selected category item, then all votes in $\left\{ V_{k,l}^{i,j} \right\}_{k \in [1, n_i], l \in [1, n_j]}$ partition either V_k^i or V_l^j . For j category,

$$|V_k^i| = \sum_{l=1}^{n_j} |V_{k,l}^{i,j}|; \quad (3.13c)$$

Or for i category,

$$|V_l^j| = \sum_{k=1}^{n_i} |V_{k,l}^{i,j}|. \tag{3.13d}$$

Proof When each vote in the sub-polls has only a single value in relation to the selected category item, the sub-polls partition the selected poll. \square

Under this construction, all votes in $\left\{ V_{k,l}^{i,j} \right\}_{k \in [1, n_i], l \in [1, n_j]}^{i, j \in [1, m]}$ dissect the valid poll V_c . When single value condition satisfied, sub-polls can partition the valid poll.

3.3 Feature Matrix Representation

For a given pair $i, j \in [1, m]$, let k corresponding to row number and l corresponding to column number, for a given $\left\{ V_{k,l}^{i,j} \right\}_{k \in [1, n_i], l \in [1, n_j]}$ sub polls, there is a unique feature matrix representation.

3.3.1 Feature Matrix

Let $V^{i,j}$ denote a feature matrix,

$$V^{i,j} = \begin{pmatrix} V_{1,1}^{i,j} & \dots & V_{1,l}^{i,j} & \dots & V_{1,n_j}^{i,j} \\ \dots & \dots & \dots & \dots & \dots \\ V_{k,1}^{i,j} & \dots & V_{k,l}^{i,j} & \dots & V_{k,n_j}^{i,j} \\ \dots & \dots & \dots & \dots & \dots \\ V_{n_i,1}^{i,j} & \dots & V_{n_i,l}^{i,j} & \dots & V_{n_i,n_j}^{i,j} \end{pmatrix}, \quad k \in [1, n_i], l \in [1, n_j]. \tag{3.14}$$

Using a statistical language, a feature matrix $V^{i,j}$ may correspond to a contingency table based on cross-classified categorical data under two selected categories [13, 16, 17]. Each element of the matrix collects a sub-set of votes in a respective cross-categorical meaning.

3.3.2 Feature Matrix Set

For a given $\left\{ V_{k,l}^{i,j} \right\}_{k \in [1, n_i], l \in [1, n_j]}^{i, j \in [1, m]}$, there are a total number of $2 * \binom{m}{2} = m * (m - 1)$ distinction feature matrixes. It is composed of a matrix set VS ,

$$VS = \{V^{i,j} | i, j \in [1, m]\}. \quad (3.15)$$

For a given pair $i \neq j, i, j \in [1, m]$ in the set, each $\left\{V_{k,l}^{i,j}\right\}_{k \in [1, n_i], l \in [1, n_j]}$ or $\left\{V_{k,l}^{j,i}\right\}_{k \in [1, n_j], l \in [1, n_i]}$ corresponds to a unique matrix or its translation matrix. However a given pair $i = j, i, j \in [1, m]$, the matrix is equal to its translation matrix. So there are a total of $m * m - m$ different matrix representations.

For a fixed item (e.g. $i = 1$) as the first index, there are a total number of $m = \binom{m}{1}$ different matrices in the system to record different relations among

$\left\{V_{k,l}^{i,j}\right\}_{k \in [1, n_i], l \in [1, n_j]}^{i, j \in [1, m]}$ sub polls.

Let $VSC(i)$ denotes the matrix set with first index fixed at i ,

$$VSC(i) = \{V^{i,j} | j \in [1, m]\}. \quad (3.16)$$

Selecting one category for both row and column values, for a given $VSC(i)$, if $V_{k,l}^{i,i} \in V^{i,i}$ in $VSC(i)$, a vote in the i th category contains only one valid value, then $V_{k,l}^{i,i}$ can be determined as following.

$$V_{k,l}^{i,i} = \begin{cases} \emptyset, & \text{if } k \neq l; \\ V_k^i, & \text{if } k = l; \end{cases} \quad k, l \in [1, n_i], i \in [1, m]. \quad (3.17a)$$

In this case, the matrix $V^{i,i}$ is a diagonal matrix.

However, if $V_{k,l}^{i,i} \in V^{i,i}$ in $VSC(i)$, a vote in the i th category contains multiple distinguishable values, then $\left\{V_{k,l}^{i,i}\right\}$ provides cross-classified sub-polls.

$$V_{k,l}^{i,i} = V_{l,k}^{i,i}, \quad V_k^i = \bigcup_{l=1}^{n_i} V_{k,l}^{i,i} = \bigcup_{l=1}^{n_i} V_{l,k}^{i,i}, \quad k, l \in [1, n_i], i \in [1, m]. \quad (3.17b)$$

In this case, the matrix $V^{i,i}$ is a symmetric matrix.

For a given $VSC(i)$, $V_{k,l}^{i,j} \in V^{i,j}$ in $VSC(i)$, following equation is true.

$$V_k^i = \bigcup_{l=1}^{n_j} V_{k,l}^{i,j} \quad k \in [1, n_i], l \in [1, n_j], i, j \in [1, m]. \quad (3.18)$$

3.3.3 Probability Feature Matrix

Let $P^{i,j}$ denote a probability feature matrix corresponding to the matrix $P^{i,j}$ and $\left\{p_{k,l}^{i,j}\right\}$ denote its element set, for any $p_{k,l}^{i,j} \in P^{i,j}$,

$$p_{k,l}^{i,j} = \begin{cases} |V_{k,l}^{i,j}|/|V_k^i|, & V_k^i \neq \emptyset; \\ 0, & V_k^i = \emptyset. \end{cases} \quad (3.19)$$

$$P^{i,j} = \begin{pmatrix} p_{1,1}^{i,j} & \cdots & p_{1,l}^{i,j} & \cdots & p_{1,n_j}^{i,j} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{k,1}^{i,j} & \cdots & p_{k,l}^{i,j} & \cdots & p_{k,n_j}^{i,j} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{n_i,1}^{i,j} & \cdots & p_{n_i,l}^{i,j} & \cdots & p_{n_i,n_j}^{i,j} \end{pmatrix}, \quad k \in [1, n_i], l \in [1, n_j] \quad (3.20)$$

For example, $n_1 = 6, n_2 = 4$, a probability feature matrix can be as follows:

$$P^{1,2} = \begin{pmatrix} 0.04 & 0.26 & 0.1 & 0.6 \\ 0.42 & 0.2 & 0.3 & 0.18 \\ 0.14 & 0.21 & 0.42 & 0.23 \\ 0 & 0 & 0 & 0 \\ 0.008 & 0.022 & 0.75 & 0.22 \\ 0.33 & 0.01 & 0.23 & 0.43 \end{pmatrix}. \quad (3.21)$$

3.4 Probability Feature Vector

For any $P^{i,j}$, only at most n_i row vectors in the matrix need to satisfy Eq. (3.22).

$$1 = \sum_{l=1}^{n_j} p_{k,l}^{i,j}, \quad k \in [1, n_i], l \in [1, n_j], i, j \in [1, m]. \quad (3.22)$$

The Eq. (3.22) can be established from Eq. (3.13c), if the column items partition the sub-polls for the given row.

Because there is not any restriction among the columns of the probability feature matrix $P^{i,j}$, such properties make flexible select different categories partitioning a given vote set $\{p_{k,l}^{i,j}\}$ into multiple distributions in larger selection spaces to satisfy complicated dynamic system requirements.

For a given $P^{i,j}$, if the i th item is a categorical index of candidates, then any candidate $k \in [1, n_i]$ has a probability feature vector corresponding to its probability densities relevant to item j and denoted by $\Psi_k^{i,j}$.

$$\Psi_k^{i,j} = (p_{k,1}^{i,j}, \dots, p_{k,l}^{i,j}, \dots, p_{k,n_j}^{i,j}), \quad k \in [1, n_i], l \in [1, n_j], i, j \in [1, m] \quad (3.23)$$

3.5 Differences Between Two Probability Vectors

Let $\{V_l^i\}_{l \in [1, n_i]}$ sub-polls denote a vector $\tilde{V}^i = (V_0, V_1^i, \dots, V_l^i, \dots, V_{n_i}^i)$, $l \in [1, n_i]$, this vote vector corresponds to a probability vector

$$\tilde{\Psi}^i = (\tilde{p}^0, \tilde{p}_1^i, \dots, \tilde{p}_l^i, \dots, \tilde{p}_{n_i}^i), l \in [1, n_i], \text{ let}$$

$$\tilde{p}_l^i = |V_l^i|/(|V^i|+|V_0|) = N_l/N, l \in [1, n_i] \quad (3.24)$$

and

$$\tilde{p}^0 = |V_0|/(|V^i|+|V_0|) = N_0/N, i \in [1, m]. \quad (3.25)$$

Let $\{V_l^i\}_{l \in [1, n_i]}$ sub-polls denote a vector $V^i = (V_1^i, \dots, V_l^i, \dots, V_{n_i}^i)$, $l \in [1, n_i]$ and

$$p_l^i = |V_l^i|/|V^i| = N_l/(N - N_0), l \in [1, n_i] \text{ and } i \in [1, m]. \quad (3.26)$$

A vector V^i is corresponding to a probability vector Ψ^i ,

$$\Psi^i = (p_1^i, \dots, p_l^i, \dots, p_{n_i}^i), l \in [1, n_i]. \quad (3.27)$$

If the i th item of a vote indicates an ordinal number of candidates in an election, a probability vector $\tilde{\Psi}^i$ is a special case of a linear spectral distribution.

For any l th candidate, if $1 \geq \tilde{p}_l^i \gg \tilde{p}^0 \geq 0$, then $\tilde{p}_l^i \cong p_l^i$.

Considering the difference between the two probability measures,

$$\begin{aligned} p_l^i - \tilde{p}_l^i &= N_l/(N - N_0) - N_l/N \\ &= N_l N_0 / N(N - N_0) \\ &= N_l / (N - N_0) \times N_0 / N \\ &= p_l^i \times \tilde{p}^0 \geq 0 \rightarrow 0. \end{aligned} \quad (3.28)$$

Equation (3.28) indicates that the probability measure of invalid votes is small compared with the candidate measures. There is no significant difference for both probability measures \tilde{p}_l^i and p_l^i for a candidate in two probability vectors $\tilde{\Psi}^i$ and Ψ^i respectively.

If any l th and g th candidates gain a similar number of votes in an election to satisfy the uncertain condition, then the difference between both probability measures p_l^i and p_g^i are restricted by the uncertain condition too.

Considering probability measure difference under uncertain condition, their difference is

$$\begin{aligned}
|\tilde{p}_l^i - \tilde{p}_g^i| &= |\tilde{p}_l^i - p_l^i + p_l^i - \tilde{p}_g^i + p_g^i - p_g^i| \\
&= |p_l^i - p_g^i - (\tilde{p}_l^i - p_l^i) - (\tilde{p}_g^i - p_g^i)| \\
&= |p_l^i - p_g^i + (p_l^i - \tilde{p}_l^i) + (p_g^i - \tilde{p}_g^i)| \tag{3.29}
\end{aligned}$$

$$\begin{aligned}
&\rightarrow \\
&\because (p_l^i - \tilde{p}_l^i) + (p_g^i - \tilde{p}_g^i) = (p_l^i + p_g^i) \times \tilde{p}^0 \geq 0, \tag{3.30}
\end{aligned}$$

$$\begin{aligned}
|p_l^i - p_g^i| + (p_l^i + p_g^i) \times \tilde{p}^0 &\leq |\tilde{p}_l^i - \tilde{p}_g^i| + (p_l^i + p_g^i) \times \tilde{p}^0 \leq \tilde{p}^0 + (p_l^i + p_g^i) \times \tilde{p}^0 \\
\therefore |p_l^i - p_g^i| &\leq 3 \times \tilde{p}^0. \tag{3.31}
\end{aligned}$$

Equation (3.31) indicates that the new probability vector does not solve the uncertain problem. To overcome the difficulty, other techniques need to be employed.

3.6 Permutation Invariant Group

For any $\Psi_k^{i,j}$, a permutation invariant group $\Psi(i, j|k)$ can be constructed to collect vectors using all elements in $\psi_k^{i,j}$ as constructors of possible permutations.

3.6.1 Feature Index and Permutation Invariant Family

For a vector $\Xi \in \Psi(i, j|k)$, if it is feasible to define a numeric measure (or feature index) and all vectors $\forall \Phi \in \Psi(i, j|k)$ have the same index, then the feature index λ is an **invariant** of $\Psi(i, j|k)$.

For $\forall \Phi \in \Psi(i, j|k)$,

$$\{\exists \lambda | \lambda(\Phi) = \lambda(\Xi) = c, \Phi \neq \Xi; \Phi, \Xi \in \Psi(i, j|k), k \in [1, n_i], l \in [1, n_j], i, j \in [1, m]\} \tag{3.32}$$

3.6.2 Polynomial Feature Index Family

For any probability vector $\Psi = (p_1, \dots, p_j, \dots, p_m)$ with m items and $\exists k \in [1, m]$, $p_k > 0$ a family of polynomial indexes $\{\lambda_n\}$ is defined by Eqs. (3.33)–(3.36).

$$\lambda_0(\Psi) = \sum_{l=1}^m (p_l)^0 = m; \tag{3.33}$$

$$\lambda_1(\Psi) = \sum_{l=1}^m (p_l)^1 = 1; \tag{3.34}$$

$$\lambda_2(\Psi) = \sum_{l=1}^m (p_l)^2; \tag{3.35}$$

...

$$\lambda_n(\Psi) = \sum_{l=1}^m (p_l)^n, n \geq 0. \tag{3.36}$$

For example, using the sample probability matrix $P^{1,2}$ of Eq. (3.21), its polynomial indexes $\{\lambda_n\}$ are

$$\lambda_0(P^{1,2}) = \begin{pmatrix} 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \end{pmatrix}; \quad \lambda_1(P^{1,2}) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}; \quad \lambda_2(P^{1,2}) = \begin{pmatrix} 0.437616 \\ 0.3388 \\ 0.293 \\ 0 \\ 0.611448 \\ 0.3468 \end{pmatrix};$$

$$\lambda_3(P^{1,2}) = \begin{pmatrix} 0.23464 \\ 0.11492 \\ 0.090664 \\ 0 \\ 0.43253416 \\ 0.127612 \end{pmatrix}; \dots$$

3.6.3 Entropy Feature Index

For a probability vector $\Psi = (p_1, \dots, p_j, \dots, p_m)$ with m items, an entropy feature index λ_E is defined by Eq. (3.37).

$$\lambda_E(\Psi) = - \sum_{l=1}^m p_l * \ln(p_l). \tag{3.37}$$

In polynomial index family $\{\lambda_n(\Psi)\}_{n \geq 0}$, $\lambda_0(\Psi)$ indicates the length of vector and $\lambda_1(\Psi)$ provides the normalized measure. In addition to $\{\lambda_n(\Psi)\}_{n \geq 0}$ family, $\lambda_E(\Psi)$ provides another type of indexes in relation to the entropy measurement. Using one of these indexes, it is feasible to distinguish two probability vectors in different permutation groups.

For example, using the same probability matrix $P^{1,2}$ of Eq. (3.21), its entropy index λ_E is

$$\lambda_E(P^{1,2}) = \begin{pmatrix} 1.015748065 \\ 1.356003379 \\ 1.305367539 \\ 0 \\ 0.6714638476 \\ 1.113842971 \end{pmatrix}.$$

3.7 Two Probability Vectors and Their Feature Indexes

Two probability vectors $\Psi_k^{i,j}$ and $\Psi_l^{i,j}$, have two distinct index families $\{\lambda_n(\Psi_k^{i,j})\}_{n \geq 0}$, $\{\lambda_n(\Psi_l^{i,j})\}_{n \geq 0}$ and $\exists \tau, \lambda_\tau(\Psi_k^{i,j}) \neq \lambda_\tau(\Psi_l^{i,j}), 1 < \tau \leq \lambda_0(\Psi_l^{i,j})$ then the two vectors belong to two different permutation groups.

For two probability vectors $\Psi_k^{i,j}$ and $\Psi_l^{i,j}$, each vector belongs to one permutation group and cannot be generated from another vector then $\exists n > 1, \lambda_n(\Psi_k^{i,j}) \neq \lambda_n(\Psi_l^{i,j}), 1 < n \leq \lambda_0(\Psi_l^{i,j})$.

Under such conditions, if two vectors have different index families, then they are in different permutation groups. In another way, when two vectors cannot be generated from another one, at least one indexes is distinguishable.

3.8 CBM Construction

Let CBM denote a Component Ballot Model. A CBM is a collection of a ballot form, vote sequences, poll and poll component matrix collection, probability matrix collections with normalized probability vectors plus the selected indexing family for an election.

$$CBM = (B|V, VS, \{P^{i,j}\}, \{\lambda_i\}). \tag{3.38}$$

Compared with SBM (Eq. 2.3) and CBM (Eq. 3.38), it is clear that the SMB is the simplest case of CBM and CBM provides more powerful properties for refined descriptions and comparisons in complicated voting applications.

Two-D Separable Proposition For two candidates to gain similar number of votes in the uncertain condition, it is always feasible to use other categorical information (i.e. location, age group) to re-partition sub polls for each candidate. If the two refined probability feature vectors belong to two permutation groups, then the uncertain problem can be solved in most case scenarios by using the polynomial feature index family or the entropy future index.

Proof For most case scenarios, cross-classified categorical data make corresponding probability feature vectors with significant differences in relation to respective density distributions. Under different categories without simple correspondences, this mechanism makes it possible to use the same strategy to handle votes for candidates. Since one party may be very strong in certain polices and relative weak in other strategies, those differences create various probability feature vectors easier located in different permutation groups. Even in the most balanced election events from a global viewpoint, hugely distinguishable distributions exist in local regions. This is the most important reason for two probability feature vectors making a pair of significantly distinct feature indexes. \square

In a complex dynamic system, equilibrium is the most probable state when the system is in dynamic balance. However, there are significant differences among local areas even in the most equilibrium conditions. This is the most powerful part of proposed model for solving uncertainty in general for complex dynamic systems.

For an election to avoid uncertainty and frustrations due to the voting result in uncertainty, it is necessary to pre-select additional odd $m - 1 \geq 1$ categories different from candidates. Following main conclusion can be statement.

Voting Authority Proposition If two candidates in an election under approval rule are in uncertainty, then additional categories (odd $m - 1 \geq 1$) under pre-agreed conditions could be used. These create the $m - 1$ pairs of feature indexes for making the decision for who will be the winner.

Proof According to the two-D separable proposition, each additional category can provide a pair of significantly distinct feature indexes to separate the two candidates, and all selected $m - 1$ pairs have such properties. Considering $m - 1$ an odd number, each pair of indexes acts as an authority vote. So, there is no problem using the majority rule to make the decision. \square

4 Conclusion and Further Work

In the proposed Component Ballot Model, multiple probability-feature matrix collections are employed and component categories other than the candidate are proposed on ballot papers to overcome confusion and frustration when two candidates are in uncertainty.

Applying advanced invariant constructions to probability feature vectors and also distinguishable properties among measurements in polynomial and entropy feature index families, voting authority provides a stable indexing mechanism to make the whole calculation based on valid votes. Distinguishable properties and invariant properties among feature index families provide reliable measurements for election outcomes.

The basic ideas, tools and technologies in the chapter are originated and created from the author's research works in 1990s for advanced content-based information retrieval and image feature indexing [18–20].

Because the approval rule is only one of the rules in practical voting systems, reader may read author's other paper discussing related aspects of voting theory under plurality and majority rules [21]. It is interesting to know whether the proposed new model can apply to other voting systems (such as Borda rules, proportional-representation system and preference voting systems) consistently. Similar uncertainty exists in other voting mechanisms. This will be a natural extension of current study.

To satisfy practical voting systems, it is essential to establish testing frameworks to make recommendations for the specific invariant properties contained in the proposed or new indexing families. There is no doubt that different voting systems may require various combinations of different feature indexing schemes to satisfy their optimal properties. More case studies linking between theoretical models and practical applications should be conducted to solve complicated voting paradoxes and other similar problems.

Acknowledgements and Disclaimer The author would like to express his gratitude to Dr. Wilson Wen for distinguishing the relationship between a feature matrix and a contingency table. Sincerely thanks also go to Dr. Gangjun Liu, Dr. Grahame Smith, Ms. Wilna Macmillan and Dr. Wen Dai for their invaluable comments, suggestions, modifications and careful proofreading of the manuscript. The constructions and conclusions contained in the chapter are merely the author's personal opinion of a scientist from a complex-dynamic system view. The author would like to take full responsibility for the contents. No government agent or company should bear the responsibility for the chapter.

References

1. J.C. de Borda, *Mémoire sur les élections au scrutin* (Histoire de l'Académie Royal des Sciences, Paris, 1781)
2. M.-J. Condorcet, *Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, Paris (1785)
3. H. Hotelling, Stability in competition. *Econ. J.* **39**, 41–57 (1929)
4. J. von Neumann, O. Morgenstern, *Theory of Games and Economic Behaviour*, Princeton (1944)
5. K.J. Arrow, in *Social Choice and Individual Values*. Cowles Commission Monographs, no. 12, 2nd edn., 1963 (New York and London, 1951)
6. S.J. Brams, P.C. Fishburn, Approval voting. *Am. Polit. Sci. Rev.* **72**(3), 831–847 (1978)
7. S. Galam, Application of statistical physics to politics. *Phys. A* **274**, 132–139 (1999)
8. V. Merlin et al., On the probability that all decision rules select the same winner. *J. Math. Econ.* **33**, 183–207 (2000)
9. M. Regenwetter, Probabilistic preferences and topset voting. *Math. Soc. Sci.* **34**, 91–105 (1997)
10. C. Robinson, *Dynamical System – Stability, Symbolic Dynamics and Chaos*, 2nd edn. (CRC Press, Boca Raton, 1999)
11. M.J. Zechman, in *Dynamic Models of Voting Behavior and Spatial Models of Party Competition*. Working papers in Methodology (No. 10, Institute for Research in Social Science University of North Carolina at Chapel Hill, 1978)
12. M.R. Anderberg, *Cluster Analysis for Applications*. (Academic Press, 1973)
13. B.S. Everitt, *The Analysis of Contingency Tables*, 2nd edn. (Chapman & Hall, London, 1992)
14. J.L. Devore, *Probability and Statistics for Engineering and the Sciences* (Duxbury Press, 1995)
15. N.L. Johnson, N. Balakrishnan, *Advances in the Theory and Practice of Statistics* (Wiley, Hoboken, 1997)
16. W.J. Conover, *Practical Nonparametric Statistics*, 2nd edn. (Wiley, Hoboken, 1980)

17. S.E. Fienberg, *The Analysis of Cross-Classified Categorical Data*, 2nd edn (The MIT Press, Cambridge, 1994)
18. Z.J. Zheng, in *Conjugate Visualisation of Global Complex Behaviour*, ed. by R. Stocker, H. Jelinek, B. Durnota and T. Bossomaier. Complex Systems: From Local Interactions to Global Phenomena (IOS Press, Amsterdam, 1996), pp. 57–67
19. Z.J. Zheng, C.H.C. Leung, Visualising global behaviour of 1D cellular automata image sequence in 2D map. *Phys. A* **233**(3–4), 785–800 (1996)
20. Z.J. Zheng, C.H.C. Leung, Graph indexes of 2D-thinned images for rapid content-based image retrieval. *J. Vis. Commun. Image Represent.* **8**(2), 121–134 (1997)
21. J.Z.J. Zheng, *Voting Theory for Two Parties*, submitted to SIAM Review (2001)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part V

Applications—Global Variant Functions

The only thing permanent is change.

—Immanuel Kant

The scientist needs an artistically creative imagination.

—Max Planck

The thought: A logical inquiry.

—Gottlob Frege

Extensive researches were focus on global function and their distributions published in the period of 2000–2010. Conjugate transformation and content-based image retrievals are typical examples for development. Using a hierarchical architecture of knowledge model, multiple levels of balanced structures were developed in both image analysis and processing, e.g., Towards Automated Mammographic Image Analysis, Proceedings of the 2005 IEEE International Conference on Information Acquisition 85–90, and content-based retrievals, e.g., Mixed Query Image Retrieval System, Proceedings of the 2007 IEEE International Conference on Information Acquisition DOI:<https://doi.org/10.1109/ICIA.2007.4295776>.

Associated with variant logic and various applications, wider explorations were carried out in the fields of cellular automata functions under different symmetric conditions that were examined. For example, Permutation and Complementary Algorithm to Generate Random Sequences for Binary Logic, International Journal of Communications, Network and System Sciences 4(5):345–350, 2011.

This part of global variant functions is composed of five chapters (11–15).

Chapter “[Biometrics and Knowledge Management Information Systems](#)” describes a hierarchical framework to use concept cell model on Biometrics & KMIS applications. Searching for brides and fingerprints was samples of typical applications in addition to process on SARS and fingerprint images.

Chapter “[Recursive Measures of Edge Accuracy on Digital Images](#)” uses recursive measures to handle image edges under different conditions to compare various edge algorithms, edge quality, and their accuracies. Conjugate maps and four

other edge schemes {Gradient, Laplacian, Gaussian, Mathematical Morphology} were selected.

Chapters “[2D Spatial Distributions for Measures of Random Sequences Using Conjugate Maps](#)” to “[3D Visual Method of Variant Logic Construction for Random Sequence](#)” use variant logic framework to illustrate 2D/3D and visual maps of variant logic operations on $n = 2$ conditions to show global visual distributions in their configurations of functional spaces.

Biometrics and Knowledge Management Information Systems



Jeffrey Zheng and Chris Zheng

Abstract Biometrics and knowledge management information systems are two important fields in recent years to attract wider attentions from different social groups. This chapter explores the use of hierarchical construction linking with biometrics applications and knowledge management information systems. The key issues are discussed and a sample case of information acquisition in content-based image retrieval system has been illustrated.

Keywords Biometrics · Complexity · Hierarchical organization · Feature classification · Content-based image retrieval

1 Introduction

Biometrics has attracted people attention in recent years due to terrorist attack and rapid scientific development and advanced information technology. In the twenty-first century, one of the most significant achievements in biology decodes a full list of gene codes of human DNA sequences. Using advanced pattern recognition technology, it is now convenient to make real-time face verification and fingerprint identification.

This work was supported by Australian Commercialising Emerging Technologies, (COMET) program.

J. Zheng (✉)
Key Laboratory of Software Engineering of Yunnan, Yunnan University,
Kunming, China
e-mail: conjugatelogic@yahoo.com

C. Zheng
Tahto, Sydney, Australia
e-mail: z@caudate.me

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_11

In general, all quantitative measures of living objects and activities from different sources including biology, anatomy, sound, photo, electronics and nerve pulse could link to biometrics. In such extremely complicated fields and areas, if we can efficiently acquire essential information to be manipulated by knowledge management information systems, then this mechanism will play an important role in the practices of applied biometrics. Useful concepts, methodologies and software/hardware toolkits in the direction will be invaluable helpful biometric applications in practical environments.

To resolve real-world problems, it is useful to apply system engineering schemes using analysis and synthesis mechanisms. In this chapter, hierarchical construction will be used as a framework to represent biometrics and knowledge management information systems. The original concepts and methodologies used in the chapter come from an established theoretical construction of dynamic systems conjugate classification and transformation [1–3]. Main algorithms and methods from the concepts have been implemented into software packages in advanced image analysis, content-based image retrieval and image understanding systems.

Using these concepts and methodologies in biometrics is a new application. The author would like to have this opportunity to sincerely discuss the possibility with other experts of the field in detail.

2 Different Complexity Issues in Biometrics Applications

Different measurement may have variant forms and contents in practical biometrics applications. In a measure space, measure data set can be relevant to length, position, angles, time and other basic measurable quantitative. Using dimension number of geometric spaces representing different biometrics objects has been shown extremely useful in many applications. Very rich contents can be observed through representatives of biometrics measures.

Infrared Detector for SARS detection (1D body temperature > 38 °C)

In protecting SARS virus distribution process, infrared detectors installed on the major channels of airports, stations and customs played active roles in indirectly measure body temperature whether higher than 38°. This process has significantly reduced the SARS virus fatal distributions.

DNA sequence (1.5D sequence)

A DNA sequence is composed of four types of gene codes forming of conjugate pair linear structure. Since the sequence itself has very complicated combination characteristics and also local grouping properties, this makes structure much more complex than simple 1D linear sequence [4].

Face identification and early breast cancer detection (2D)

In most image analysis systems, especially face identification and early breast cancer detection systems use of 2D features in manipulations. In larger applications or data sets, those feature spaces are very complicated.

CT scanning and reconstruction (3D and higher D)

Using modern CT scan medical imaging equipments, it is feasible to reconstruct 3D images from multiple 2D image slice sequences to represent complicated projection and dynamic properties of interested areas and organs. 3D visualization has much more complicated properties than 2D image visualization process.

Retinal analysis and synthesis (higher D nerve network)

The detailed principles of retinal nerve network in human vision is not fully understood. But their biological structures are well recognized by interconnected nerve networks. This type of connectivity is much higher than three dimensions. The corresponding symptoms of distributions among brain surfaces and visual simulations indicate hierarchical structures in optical nerve systems naturally [5].

Abstract Thinking (Super Hypercomplex Cells)

The capacity of abstract thinking may belong to super hierarchical organizations of nerve systems. If there are real nerve objects, this structure could be super hypercomplex cells or their superposition on extensive hierarchy [5].

From a certainty viewpoint, lower dimension cases have more certain properties than higher dimensions. In addition, higher dimension structure expressed abstract properties with more variables and richer possibilities in real-world cases.

3 Proper Concepts, Methods and Useful Toolkits

Using modern mathematical toolkits, concepts and methods such as geometric topology and combinatorial topology, it is feasible to use basic analysis on neighbourhood relationship of kernel structure to partition complicated systems into non-reducible invariant characteristics base family. Using non-reducible bases as generators, it is possible to apply synthesis techniques to rebuild complicated systems in certain forms [6]. In invariant and singularity analysis relevant applications, global topology characteristics play core roles using modern mathematics analysis toolkits [7]. Since connectivity belongs to one of the topological properties, higher dimensional geometric problems could be represented as graph problems or other forms to use common probability and statistical methods for practical calculations to resolve the equivalent problems in certain degree [8]. It does not matter how to represent a certain problem in detail, and abstract concepts could be always represented as lattice structures.

After systematic analysis of modern knowledge management information systems in concepts, principles and operational levels, a useful kernel structure Concept Cell Model for knowledge management using directed acyclic lattices in hierarchical constructions has been proposed for base construction toolkits of representation [9, 10]. The model can distinguish two similar lattices of three essential concept levels in different abstract structures as building lattice constructions:

Time Invariant Structure: Descriptive Knowledge Lattice (Tacit, Implicit, Explicit)
 Time Variable Structure: Procedure Knowledge Lattice (Start, Operation, Finish).

Undertaken hierarchical construction, it is convenient and efficient to represent knowledge systems in information request, abstract representation, categories, organization and other statistic and dynamic application requirements.

Concept cells in hierarchies can efficiently represent from real measurement data sets to higher levels of conceptual networks to represent application systems as multiple levels of organizations. This provides an operational knowledge management framework to flexibly support from user cases, abstract design, and implementation and operation requirements for system engineering practices. By applying conceptual categories, it is feasible to construct useful application systems with powerful self-organization and self-learning capacities in wider engineering and social environments.

To easily understand the main point, it is convenient to show an example to represent a partial structure in implemented content-based image retrieval systems using hierarchical concept structures shown in Fig. 1.

In the construction, a single index represents specific content-based information extracted from an image. A set of images needs to correspond to a set of indexes, respectively, and is organized as a list. It is convenient to use a multiple hierarchy to organize the list of single indexes as its end nodes. Each intermediate node can be established as a group of multiple indexes with strong similarity properties in their contents as a combined index. By this way, a root node can be established by combined individual nodes and intermediate nodes to be the representative of the whole set of indexes. Three types of information can be distinguished as follows:

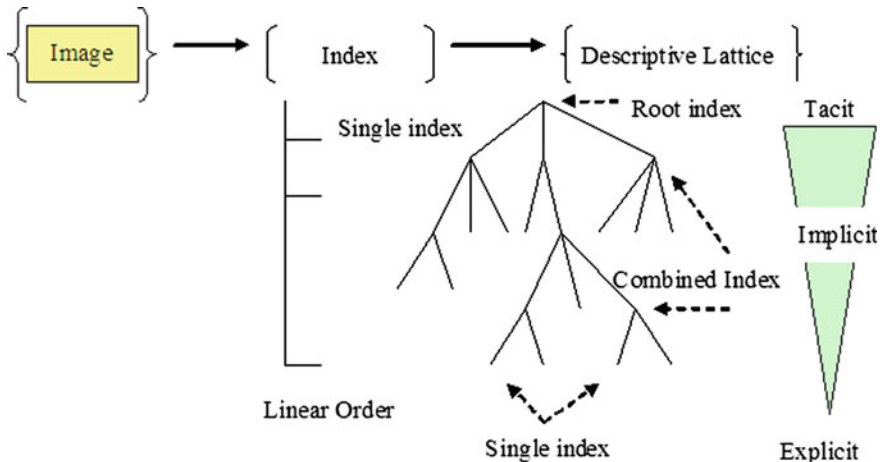


Fig. 1 Descriptive lattice in hierarchical representation

Single index: individual information explicit

Combined index: group information implicit

Root index: whole information tacit.

Using descriptive lattice structure in multiple levels of representations, complicated content-based image retrieval system can be mapped to a multiple layout network structure. It provides efficient organization to do information acquisition and organization linking with individuals, groups and the whole in information network construction.

While search operation, the current index will check from root (tacit node) to get the best match through combined indexes (implicit nodes) and single indexes (explicit nodes) to obtain the best-matched cases in hierarchy. Using best match information, a selected image group will be determined as output results.

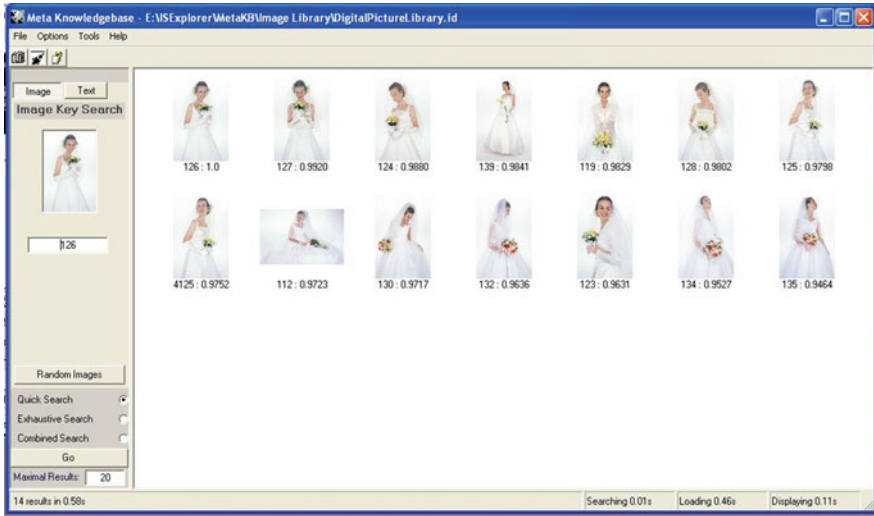
In Fig. 2, two sets of implemented results on brides and fingerprint verification are provided to illustrate visual qualities of retrieved output results. The 125th bride image is selected and a list of similar brides as retrieved results. The 194th fingerprint image has been selected as a query example, and the output result is shown in right panel and arranged by similarity from higher to lower values in relation to the best 20 matched images from the image database in which the 194th, 193rd and 195th images are strong relative fingerprints from the same person.

Two sets of image processing results are shown in Figs. 3, 4 and 5. In Fig. 3, four enhanced results on an original SARS image are selected. In Figs. 4 and 5, various results of a fingerprint image are processed in different parameters under special enhanced functions.

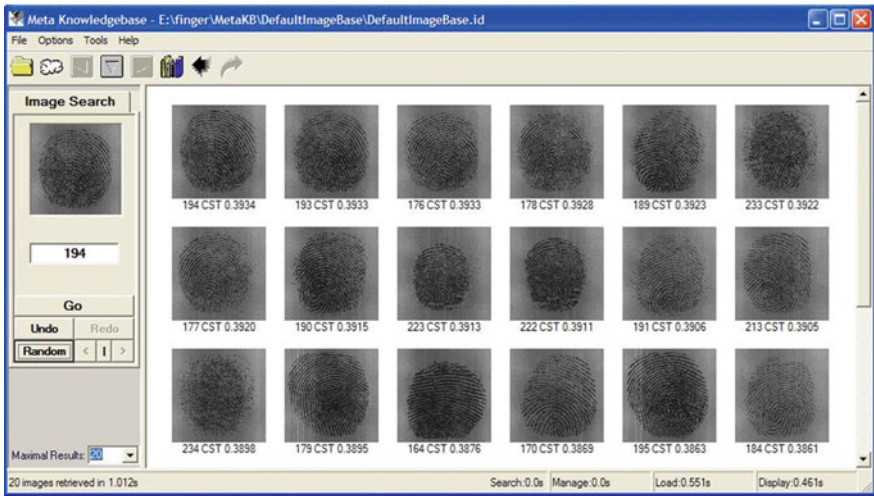
4 Demand in Future Society

From biometrics measure viewpoint, measure data itself can be very accurate and crystal certain as numeric values. However, through hierarchical construction, more uncertainty will appear as higher level contents. Complicated interconnections will be linked with simply single measures to complicated global organization. Using hierarchical construction, it is feasible to organize single, group and whole information through network construction to cover wider applications.

In rapid development of web-based network, high-speed interactive facility and quick connections have changed traditional concepts and methods significantly. It is a convenient approach to use knowledge management information system to do information acquisition, intelligent analysis, combination and synthesis.



(a)



(b)

Fig. 2 Search results: a Brides; b Fingerprints

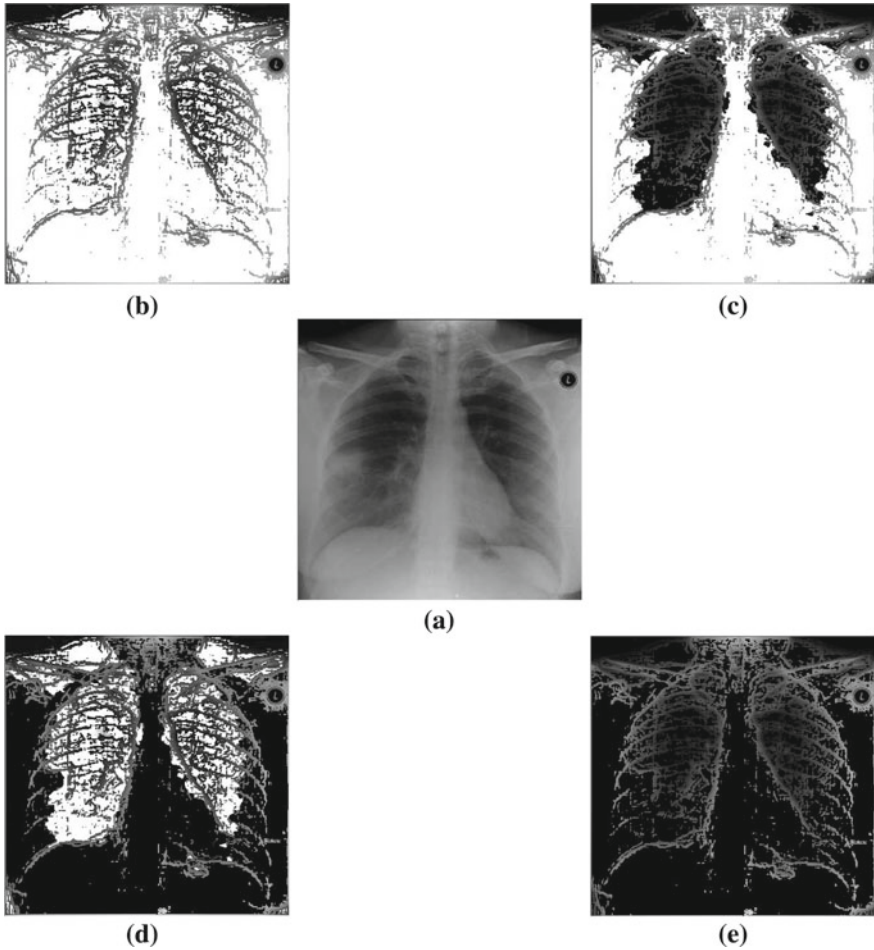


Fig. 3 Four image enhancements on SARS image (a–e); **a** Original; **b** Positive enhanced; **c** Valley enhanced; **d** Hill enhanced; **e** Negative enhanced

Hierarchical operations become the most advanced parts of optimal control and best operational strategies. In the current application environment, fast, convenient and efficient design and implementation can get wider applications in many fields. It can be expected to use automatic and intelligent methodologies to complete complicated issues, especially on complex and time consumed design processes. Facing of many practical applications, simple and unified concepts can help larger dynamic system in forming stable structures. Global interactive connection and their evaluations will be helpful for social environment in high speed and sustainable development.

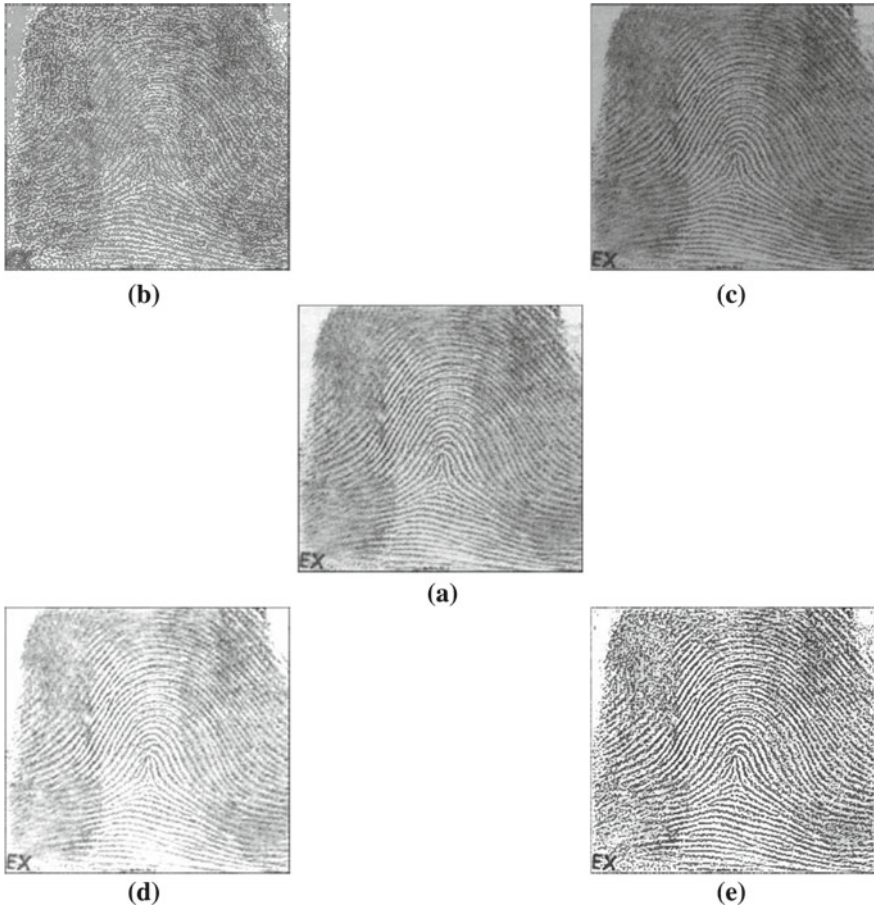


Fig. 4 Four image enhancements on fingerprint image (a–e); **a** Original; **b** Positive enhanced; **c** Valley enhanced; **d** Hill enhanced; **e** Negative enhanced

5 Base Strategy of Development

Any theoretical scheme cannot ensure itself in practice operations successfully without carefully matching environment requirements. In current social and economic conditions, it is more important for biometrics to make a positive impact on social economy to help the existing developments. Market-oriented mechanism can be used to resolve key problems in applications. It is most important to identify core technology in the application and collect the required energies and resources to attack it resulting in significant impact.

In knowledge management information systems, content-based acquisition, representation, indexing and retrieval components are the core components for automatic

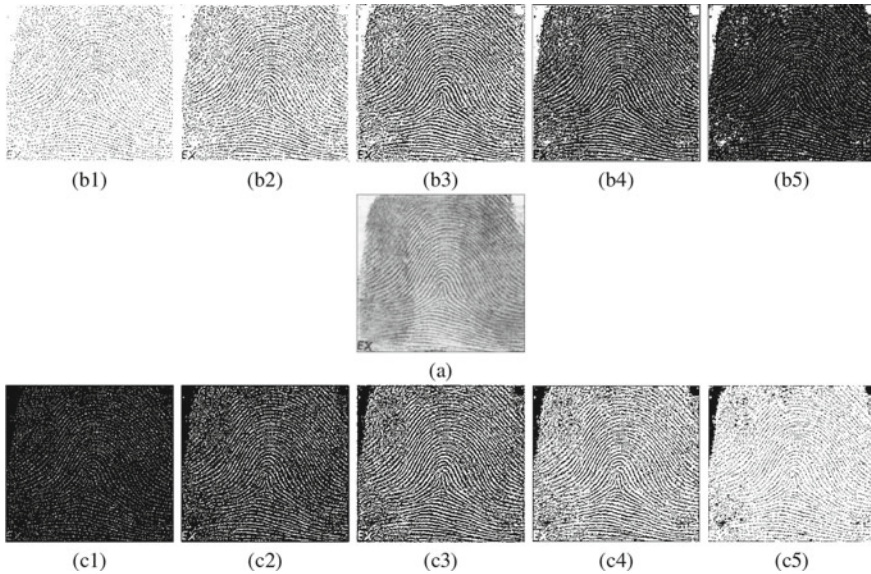


Fig. 5 Ten enhanced results of a fingerprint image (a–c); **a** Original; **b1–b5** Hill enhanced; **c1–c5** Valley enhanced; **b1/c1** $\alpha = 30$; **b2/c2** $\alpha = 80$; **b3/c3** $\alpha = 128$; **b4/c4** $\alpha = 160$; **b5/c5** $\alpha = 220$

organization and high-efficient retrieval. Ultra-fast and accurate retrieval technology for databases and meta-knowledge bases can be widely used in many applications to satisfy information acquisition, extraction, categories, and organization, storage and retrieval requirements. Under global web-based environment, hierarchical organization of knowledge management systems and biometrics will be further refined and developed in health environment.

References

1. Z.J. Zheng, Conjugate Transformation of Regular Plane Lattices for Binary Images, Ph.D. Thesis, Department of Computer Science, Monash University, 1994
2. Z.J. Zheng, A.J. Maeder, The elementary equation of the conjugate transformation for hexagonal grid, in *Modeling in Computer Graphics*, ed. by B. Falcidieno, T.L. Kunii (Springer, Berlin, 1993), pp. 21–42
3. Z.J. Zheng, A.J. Maeder, The conjugate classification of the kernel form of the hexagonal grid, in *Modern Geometric Computing for Visualization*, ed. by T.L. Kunii, Y. Shinagawa (Berlin, 1992), pp. 73–89
4. F.W. Stahl, *Genetic Recombination* (W.H Freeman and Company, New York, 1979)
5. 荆其诚, 焦书兰, 纪桂萍, 人类的视觉, 科学出版社 1987
6. R.S. Palais, C.-L. Terng, *Critical Point Theory and Submanifold Geometry* (Springer, Berlin, 1989)
7. H. Hopf, *Differential Geometry in the Large* (Springer, Berlin, 1983)
8. V.V. Nikulin, I.R. Shafarevich, *Geometries and Groups* (Springer, Berlin, 1989)

9. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, Concept cell model for knowledge representation. *Int. J. Inf. Acquisition* **1**(2), 149–168 (2004) (World Scientific Publishing Company)
10. J. Zheng, M. Zhou, J. Mo, A. Tharumarajah, Background and foreground knowledge in knowledge management, in *Global Engineering, Manufacturing and Enterprise Networks*, ed. by J. Mo, L. Nemes (Kluwer Academic Publisher, Dordrecht, 2001), pp. 332–339

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Recursive Measures of Edge Accuracy on Digital Images



Jeffrey Zheng and Chris Zheng

Abstract In this chapter, an edge accuracy model is proposed on digital images and five types of edge detection methods are discussed as examples to investigate their edge maps undertaken recursive operations. Using invariant criterion, it is possible to compare different schemes in accuracy, consistency, completeness and simplicity. This provides general mechanism in relation to accurate edge extractions from digital images.

Keywords Edge detection · Accuracy · Invariant · Digital image

1 Introduction

Edge detection plays a fundamental importance in image analysis, processing and computer vision applications. As the first step of visual perception, extensive R&D has being focused for 40 years (more than forty thousand years—drawing arts in human civilization). Many useful edge detection operators have been invented and applied in wider applications.

From an operational viewpoint, edge detection creates edge maps from images shown in Fig. 1a. Edge detection operators identify significant changes from visual objects as their edges or contours. From a historical viewpoint, common edge detec-

This work was supported by Australian Commercialising Emerging Technologies, (COMET) program.

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, Yunnan, China
e-mail: conjugatelogic@yahoo.com

C. Zheng

Tahto, Sydney, Australia
e-mail: z@caudate.me

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_12

203

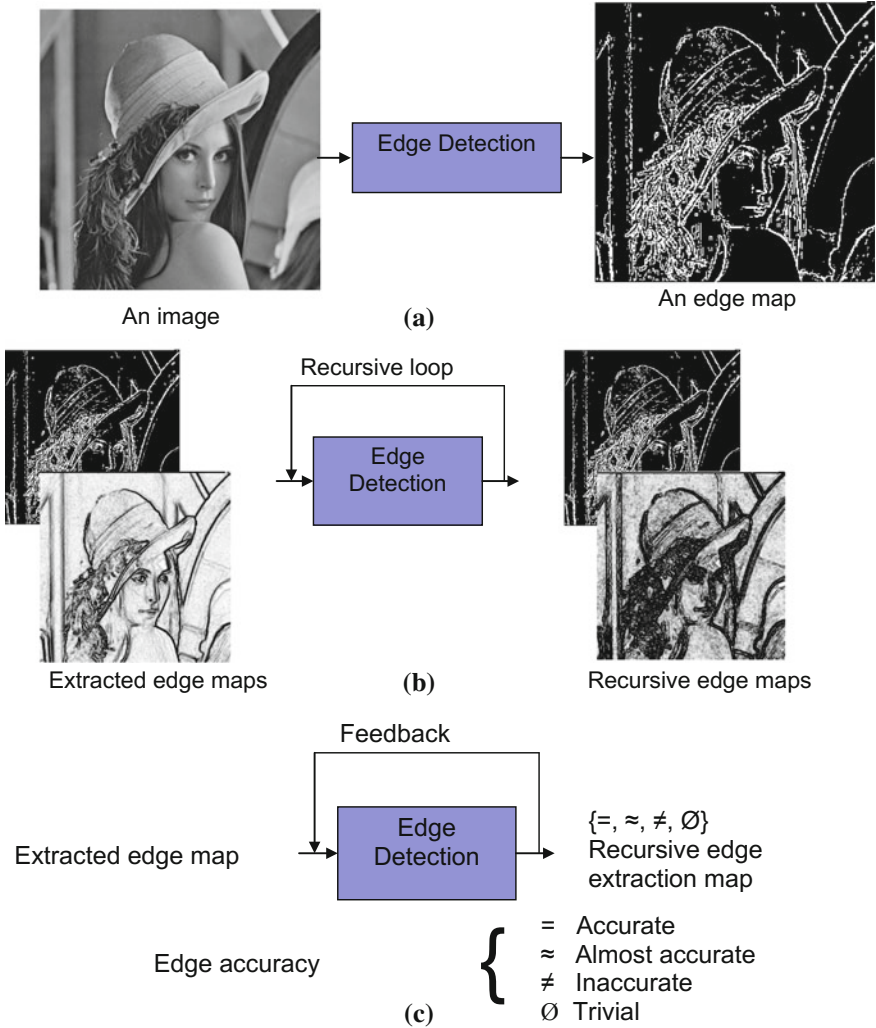


Fig. 1 Recursive edge extraction. **a** Edge detection; **b** Recursive edge maps; **c** Edge accuracy measures

tion approaches are divided into five approaches. Traditional edge detections have three main categories: Gradient, Laplacian and Gaussian; another two categories are mathematical morphology and conjugate. The five categories will be briefly introduced as follows.

1.1 Gradient

Gradient scheme has a direction corresponding to convolution operations; we can use 2×2 , 3×3 matrices or more complicated schemes to construct relevant operators, for example, Roberts operator uses 2×2 matrix to detect edges on main diagonal or anti-diagonal directions. Prewitt, Sobel and Isotropic schemes take 3×3 matrices using different parameters to extract horizontal or vertical edges from digital images shown in Fig. 2a.

1.2 Laplacian

A typical Laplacian scheme is Marrs–Hildreth’s zero crossings. This scheme uses the second differential information to determine zero crossings of the edges shown in Fig. 2b.

1.3 Gaussian

Canny edge filter plays a significant role in advanced edge detection applications from late of 1980s. This scheme applies Gaussian smoothing filter first, then gradient operations and finally thinning processes and its final results shown in Fig. 2c. Different from Gradient and Laplacian schemes, Canny edge detection provides controllable parameters to balance noise levels and significant edge components. Because of its controllable properties, this scheme widely used in many practical applications in relation to significant edge components.

1.4 Mathematical Morphology

Mathematical morphology plays an important role in advanced image analysis and processing applications from 1980s. Using discrete patterns as morphological masks, the method applies erosion and dilation, opening and closing operations on the processed images. This method distinguishes edge and non-edge masks. In general, only translation invariant can be retained in operations. Each time of basic operation uses one mask on either erosion or dilation corresponding to reduce or extend boundaries of the visual objects. There is no simple relationship between the selected mask states and edge states. Two edge maps using a crossing mask under either erosion or dilation are shown in Fig. 2d. Each edge map has been calculated by either dilated or eroded output image subtracted by the input edge map.

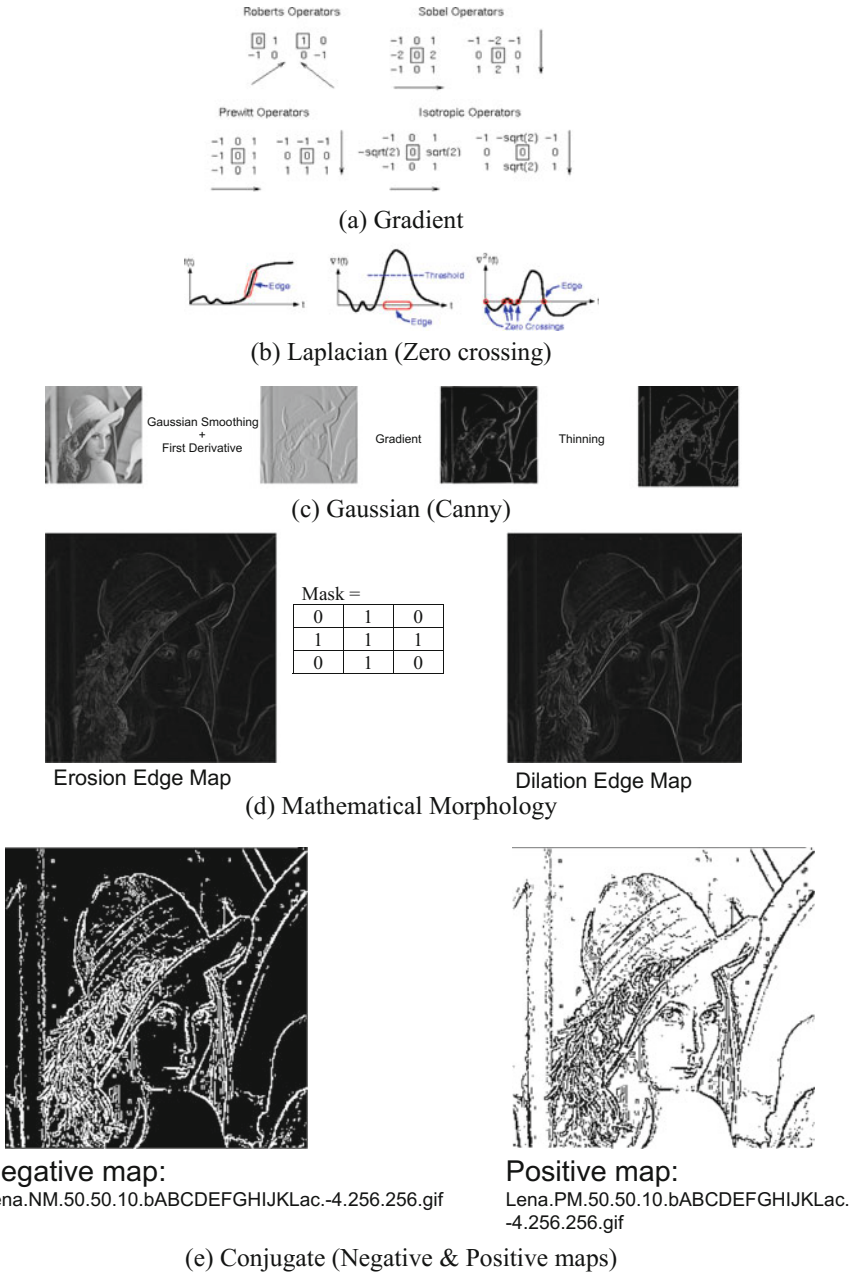


Fig. 2 Different edge detection methods. **a** Gradient; **b** Laplacian; **c** Gaussian; **d** Mathematical morphology; **e** Conjugate

1.5 Conjugate

Conjugate scheme has been developed from 1990s and based on a full pattern classification of nearest neighbourhood relationship of discrete states on regular plan lattices under rotation, reflection and translation invariants. This approach can express local patterns into invariant groups such as isolated, inner, block edge and intersection to organise whole pattern space as a hierarchical construction. Both background and foreground information need to be represented as balanced structures in conjugate phase space. Under certain conditions, it is feasible to use two types of edge maps in representations. In Fig. 2e, two typical edge maps are illustrated to use conjugate scheme:

- Negative (White edge map on black background) and
- Positive (Black edge map on white background).

From edge detection considerations, different operations provide special properties to be emphasised by various visual information from images. Simple convolution filters may provide fast process; however, it is highly possible to be sensitively influenced by minor noise levels. Among three traditional edge detection schemes, Canny edge detector shows an important characteristic with a series of controllable edge maps in reliable properties. Because distinct edge detectors have different behaviours, it is very hard for applicants to make simple selections apply the best one among schemes. Mathematical morphology applies discrete masks in operations. Since edge maps normally do not correspond to masks themselves directly, it is difficult to establish a link from relevant operations and edge detection results.

Considering edge detection operation extracts edge map from digital images. Under this viewpoint, we need to establish a proper model in determining invariant properties among edge detection schemes.

2 Recursive Model of Edge Accuracy

Different edge detection methods cover various applications with advantages in many aspects. From a practical viewpoint, it is hard for users to make proper judgment on which method provides the best edge map to satisfy suitable applications. From history of edge detection research, no model can provide general mechanism in systematic comparison among distinguished methods. Since the target of different edge detections creates edge maps, it is natural for us to determine under which conditions the edge maps can represent true edge.

2.1 Question

Could an extracted edge map be a true edge representation?

From a morphological viewpoint, true edge map needs to have invariant properties relevant to their geometric and topological constrains. In many theories and practices in relation to dynamic systems and cybernetics, recursive methods and models have been approved to be a foundational importance in detailed analysis tasks. A recursive model has been applied in testing edge detection operators to explore their refined properties shown in Fig. 1b. Using this feedback mechanism, edge map needs to be looped back again undertaken the same type of edge detection operators. The recursive loop shows an important magnification to identify dynamic behaviours among input and output pairs directly.

3 Four Types of Edge Accuracy Measures

Under the recursive approach, a true edge representation must be the recursive edge map itself. Such invariant of recursive operations can be observed as intrinsic properties in relation to the edge detection operators themselves. In addition to invariant properties, many rich effects among input and output pairs need to be concerned. To make proper judgment among recursive results, it is essential to apply four different accurate measures shown in Fig. 1c. They are $\{=, \approx, \neq, \emptyset\}$ representing accurate, almost accurate, inaccurate and trivial behaviours, respectively, between input and output edge maps. From matched results between extracted edge map and its recursive edge extraction map, it is feasible to determine the category in which generated results need to be belonging to. This provides a general model independent of a specific edge detection scheme. If anyone would like to check which category could be belonging to a special scheme, the person can simply apply this recursive mechanism to check specific method itself directly in explorations.

4 Four Sample Groups of Recursive Edge Maps

In Fig. 3a–d, four groups of recursive edge maps are generated in illustration. Two operators are selected from Photoshop: Find edge (Gradient) Fig. 3a and trace contour (Zero crossings) Fig. 3b. Find edge operation has a clear variant property, and trace contour will have a flip-flop behaviour after certain operations. One example is selected from Canny edge detector shown in Fig. 3c. Recursive results of Canny operation show that two sets of examples are shown in Fig. 3d for mathematical morphology. It is interesting to see dilation representing almost invariant properties and erosion creating edge map similar to zero crossing effects. To show different recursive properties of conjugate scheme, four sub-operators are illustrated in Fig. 3e1–e4.

Table 1 Edge detection schemes and their accuracy properties

Operator	Edge quality	Noise sensitivity	Accuracy	Recursive maps
Find edge	Good, fair	Very high	± 2 pixels	\neq, \emptyset
Trace contour	Better, good, fair	High	± 1 pixels	\neq, \approx
Canny edge	Better, good, fair	Controllable	± 2 pixels	\neq
Mathematical morphology	Better, good, fair	High	± 1 pixels	\approx, \neq, \emptyset
Conjugate map	Best, better, good, fair	Full controllable	≤ 1 pixel True edge	$=, \approx, \neq, \emptyset$

Each group shows a specific category among three non-trivial results. In conjugate edge detection operators, there are two types of controllable parameters that are available corresponding to meta-shape parameters $\{A, \dots, L, a, \dots, l\}$ and enhanced ratio control $\{-8, \dots, 8\}$. Both controllable parameters can provide universal edge representation on true edge map to support various edge representations undertaken selected operations.

5 Comparison

Using the five categories, it is feasible to make summary in Table 1. This provides a systematic way in comparison.



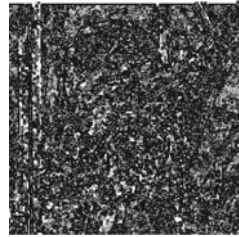
(A1) The first edge map



(A2) The second edge map



(A3) The third edge map



(A4) The fourth edge map

(A1) ≠ (A2) ≠ (A3) ≠ (A4) No invariant edge map available!

Recursive condition: Directly use find edges filter to each map

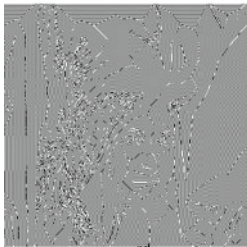
(A). Photoshop: Find Edges (Gradient)



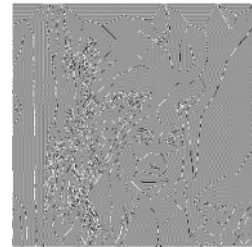
(B1) The first map



(B2) The third map



(B3) The 53rd map



(B4) The 54th map

(B1) ≠ (B2) ≠ (B3) ≈ (B4) Flip flap variations after the 53rd operation

Recursive condition: Trace contour filter (level = 119, edge = low)

(B). Photoshop: Trace Contour (Zero Crossing)

Fig. 3 Recursive maps of different edge detection operators. **a** Find edges; **b** Trace edge; **c** Canny edge detection; **d** Morphology; **e** Conjugate edge detection

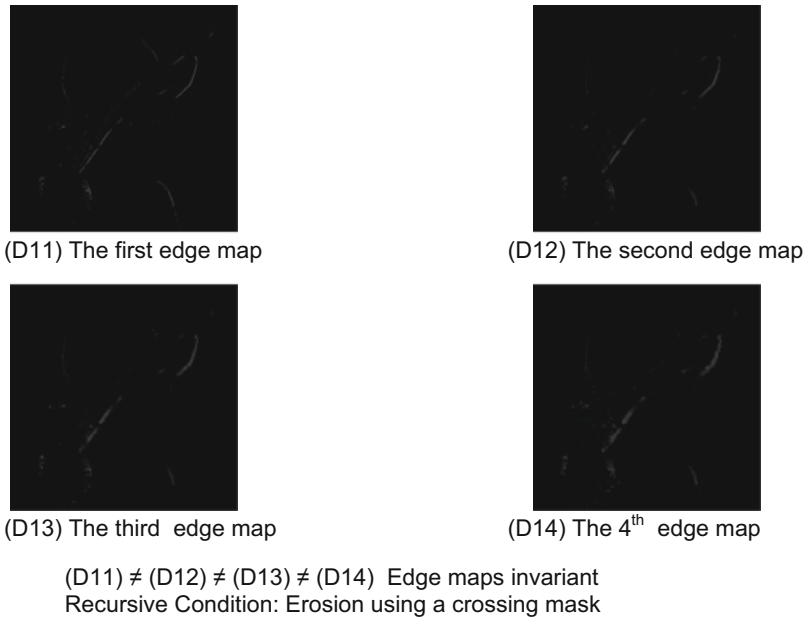
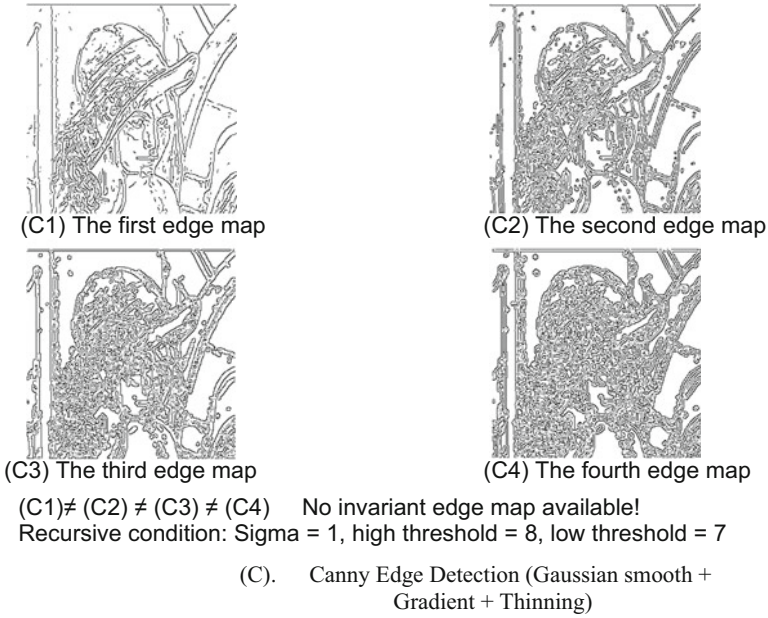
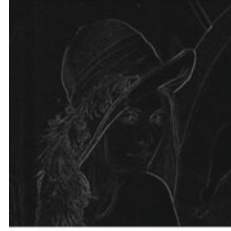


Fig. 3 (continued)



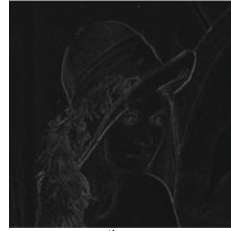
(D21) The first edge map



(D22) The second edge map



(D23) The third edge map



(D24) The 4th edge map

(D11) ≈ (D12) ≈ (D13) ≈ (D14) Edge maps almost invariant
Recursive Condition: Dilation using a crossing mask

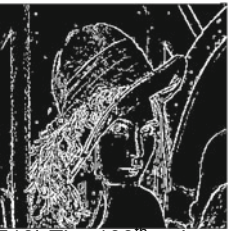
(D). Mathematical Morphology



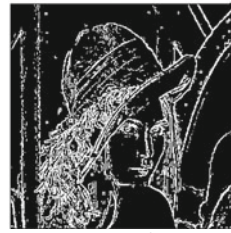
(E11) The first edge map



(E12) The fifth edge map



(E13) The 100th edge map



(E14) The 1000th edge map

(E11) = (E12) = (E13) = (E14) Edge maps invariant
Recursive Condition: NM 50 50 10 abcdefghijkl -2

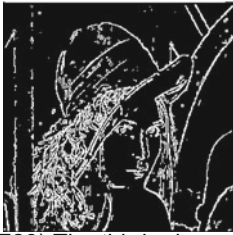
Fig. 3 (continued)



(E21) The first edge map



(E22) The second edge map



(E23) The third edge map

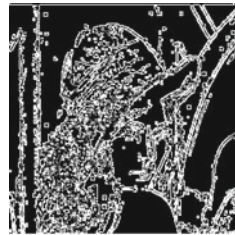


(E24) The fourth edge map

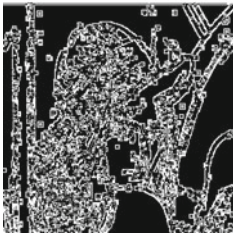
(E21) \approx (E22) \approx (E23) \approx (E24) Similar edge maps with noise removing
Recursive Condition: NM.50.50.10.abcdefghijkiABC.-2



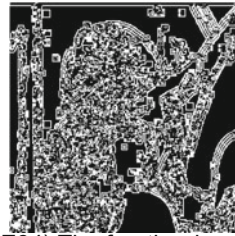
(E31) The first edge map



(E32) The second edge map



(E33) The third edge map



(E34) The fourth edge map

(E31) \neq (E32) \neq (E33) \neq (E34) Changed edge maps
Recursive Condition: PM.50.50.10. cdefCDEF.-2

Fig. 3 (continued)



(E41) The first edge map



(E42) The second edge map



(E43) The third edge map



(E44) The fourth edge map

(E41) \approx (E42) \approx (E43) \approx (E44) Similar edge maps with noise removing
Recursive Condition: PM.50.50.10.ABCDEFGHIJKLabc.-2

(E). Conjugate Edge Detection

Fig. 3 (continued)

6 Conclusion

Existing edge detections are without unique recursive maps as their representations. Conjugate technology provides full controls to create true edge maps in accuracy and invariance.

True edge maps contain unique shape information in fundamental importance to support all visual applications.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



2D Spatial Distributions for Measures of Random Sequences Using Conjugate Maps



Qingping Li and Jeffrey Zheng

Abstract Advanced visual tools are useful to provide additional information for modern information warfare. 2D spatial distributions of random sequences play an important role to understand properties of complex sequences. This chapter proposes time sequences from a given logical function of 1D cellular automata in both Poincare map and conjugate map. Multiple measure sequences of Markov chains can be used to display spatial distributions using conjugate maps. Measure sequences are recursively produced by different logical functions generating maps. Possible complementary feature exists between pair functions. Conjugate symmetry relationships between a pair of logical functions in conjugate maps can be observed.

Keywords Time sequence · Random property · Cellular automata
Spatial distribution · Conjugate symmetry

1 Introduction

Random sequences are widely used in many security-based applications such as security communication, cryptology coding, and information security systems [1]. To make proper analysis, Markov chain methodologies and technologies provide a series of important methods and tools to help analyzers decoding process [2–4]. In modern information warfare, it is essential for analyzers to detect and decrypt the opponent's communications using information acquisition toolkits from real coding sequences [5].

This work was supported by Yunnan Advanced Overseas Scholar Project.

Q. Li
School of Software, Yunnan University, Kunming, China
e-mail: lqpbupt@126.com

J. Zheng (✉)
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_13

Information Warfare describes terms of “actions” executed to achieve a sought outcome—denial, exploitation, corruption and destruction of an opponent’s “information” and related functions, and prevention of such “actions” executed by an opponent [6].

The battle between the obscurers and those who sought to break the codes has been a continual one, but it reached a new level of stature and importance during World War II with its decryption of Germany’s Enigma messages. Historic events are approved that statistical and probability tools are extremely important in Information Warfare applications. This battle of wits fought by British mathematicians and statisticians shortened World War II and ushered in the age of information warfare [7].

Prerequisite of executing these attack actions is thoroughly understood by the mechanism of information encryption that opponent uses [8]. In information warfare, secured communications among opposite parts may use public networks. It is feasible to capture relevant information for further analysis. Different quantitative tools and methods are useful to provide additional information in decoding process. Variant features play an important role for measurement and analysis of random sequences [9].

Because of the implicated expression of functions that generate random sequences, it is hard to get the characteristic of random sequences from the function and coding sequences themselves [10]. Traditionally, time sequence map and Poincare map are the two most popular methods to take the measure features of a random sequence in two dimensions [11]. From a visual viewpoint, current Markov chain schemes do not provide efficient visual mechanism to display multiple measurement sequences from the spatial characteristic of complex random sequences.

To extract further information from random sequences, this chapter establishes a visual system to illustrate multiparameter measurement sequences of Markov chains as conjugate maps. For a given set of measurement sequences, the conjugate map proposed in this chapter can provide refined information of distributed structure than present map technologies [12].

In the second section, respective characteristics of traditional methods and conjugate method are discussed. The measurement mechanism of logical function’s spatial characteristics, disposal model, measuring model, and visualizing model, is described in the third section. The results of maps and analysis of the results are discussed in the fourth and fifth sections, and then, concluding remarks are provided in the last section.

2 Traditional Methods and Conjugate Method

In this section, two typically traditional methods, time sequence map and Poincare map, are discussed for comparison.

Time sequence map generates a 2D coordinate; X -axis is determined by the time scale t , and Y -axis is determined by the value of measured parameter $f(t)$, as shown in Fig. 1a.

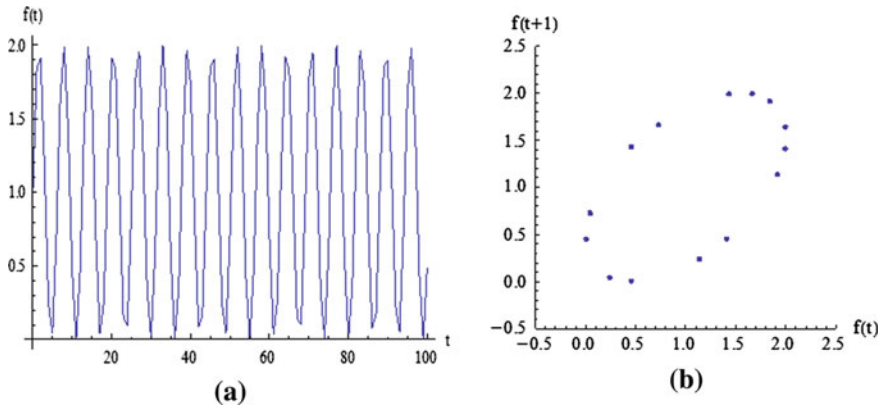


Fig. 1 Simple time sequence map and Poincare map; **a** Time sequence map, **b** Poincare map

The measure sequence $\{f(t)\}_{t=0}^{T-1}$ with length of T can form Poincare map according to the matching pattern considering data correlation. Poincare method maps one group of measures of time sequence to a 2D map. It detects spatial distribution of sequence through the distribution of point cluster. In Poincare map, X-axis is determined by the value of $f(t)$ while Y is $f(t + l)$. It is vicinity-related patterns map when $l = 1$, as shown in Fig. 1b.

Different from Poincare method based on one group of measures, new map proposed in this chapter chooses two groups of measures from relevant parallel measures sequences. As two different groups of measures are acted simultaneously, the value of each axis is determined by these two groups of measurements. It is convenient to name new map as conjugate map to present this kind of multiple parameter measurement map.

3 Generate and Measure Mechanism of Time Sequence

In this section, the Cellular Automata (CA) method is applied to generate time sequence and then to make concomitant measurement sequence. First, the initial sequence inputted, and the output sequence is generated by a given logical function using 1D cellular automata. Using this data sequence, measurements are formed by probability measurement according to pairs of input and output sequences. Finally, the generated measure sequences can be used to construct a 2D conjugate map showing 2D spatial distribution of the time sequence. The processing flow of the mechanism is shown in Fig. 2.

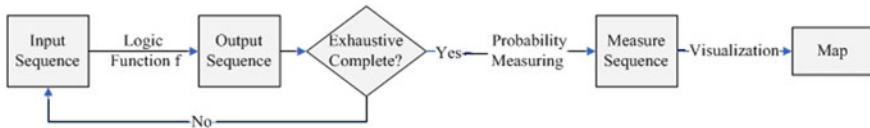


Fig. 2 Flow sheet of the produce and detect mechanism of time sequences

Table 1 I/O pattern of disposal model

Function f	Input sequence	$X_0, X_1, \dots, X_i, \dots, X_{N-1}, X_i \in \{0, 1\}$
	Output sequence	$Y_0, Y_1, \dots, Y_i, \dots, Y_{N-1}, Y_i \in \{0, 1\}$

Table 2 Exhaustion of initial input sequences

Serial number	Input sequences
0	0 0 0 ... 0 0 0
1	0 0 0 ... 0 0 1
...	...
$2^N - 2$	1 1 1 ... 1 1 0
$2^N - 1$	1 1 1 ... 1 1 1

3.1 Disposal Model

Consider a logical function f as a function of CA. The function generates equal-length output sequence $\{Y_i\}_{i=0}^{N-1}$ for any initial input sequence $\{X_i\}_{i=0}^{N-1}$ with N -length bits. The I/O pattern is shown in Table 1.

A total of 2^N states of N -length initial input sequence are exhaustively generated, and the corresponding sequence under the logical function $f : X \rightarrow Y$ can be generated. The input and the output sequences are in the same group corresponded to each other; there are 2^N groups of corresponding relationship [13]. Exhaustion of all the initial input sequences is shown in Table 2.

3.2 Measure Model

The basic model of measurement can be confirmed to establish the transformation relation between the input sequence $\{X_i\}_{i=0}^{N-1}$ and the output sequence $\{Y_i\}_{i=0}^{N-1}$ for each group.

In the transformation of $f : X_i \rightarrow Y_i, 0 \leq i < N$, there are a total of four types of transformations, each type determines a number, and corresponding relationships are shown in Table 3. This type of measurement structure has a directly corresponding relationship to the Markov chain mechanism [4].

Table 3 Measure parameters

Transform type	Number of types	Number of 0, 1 in input sequences	Total number
0 → 0	N_{00}	$N_0 = N_{00} + N_{01}$	$N = N_0 + N_1$ $= N_{00} + N_{01} + N_{10} + N_{11}$
0 → 1	N_{01}		
1 → 0	N_{10}	$N_1 = N_{10} + N_{11}$	
1 → 1	N_{11}		

Table 4 Probability measure

Measure parameters	Value of parameter
$P_{00}(j)$	$N_{00}(j)/N_0(j)$
$P_{01}(j)$	$N_{01}(j)/N_0(j)$
$P_{10}(j)$	$N_{10}(j)/N_1(j)$
$P_{11}(j)$	$N_{11}(j)/N_1(j)$

Consider $j \in \{0, 1, 2, \dots, 2^N - 1\}$ as the serial number of different initial input sequences. There are four measurements that can be identified by the measurement parameters above shown in Table 4 with Markov chain properties, respectively.

For different initial input sequences, there can be generated four groups of measurements on the corresponding I/O sequences: $\{P_{00}(j)\}_{j=0}^{2^N-1}$, $\{P_{01}(j)\}_{j=0}^{2^N-1}$, $\{P_{10}(j)\}_{j=0}^{2^N-1}$, and $\{P_{11}(j)\}_{j=0}^{2^N-1}$.

3.3 Visualization Model

Based on the probability measurements presented above, two measurements are chosen to construct 2D map, as two different groups of measurements are used simultaneously, to name this kind of map conjugate map, of which the value of each axis is determined by these two groups of measurements.

According to the construction pattern introduced above, there are $C_4^2 = 6$ kinds of different combinations as below: $\{P_{00}(j), P_{01}(j)\}$, $\{P_{00}(j), P_{10}(j)\}$, $\{P_{00}(j), P_{11}(j)\}$, $\{P_{10}(j), P_{11}(j)\}$, $\{P_{01}(j), P_{11}(j)\}$, and $\{P_{01}(j), P_{10}(j)\}$.

On the same group of sequences, construct 2D conjugate maps, respectively, by using the combinations above as shown in Fig. 3.

This chapter chooses the typical combination $\{P_{01}(j), P_{10}(j)\}$ constructing 2D conjugate map to detect the special distribution of time sequences for $N = 13$ condition.

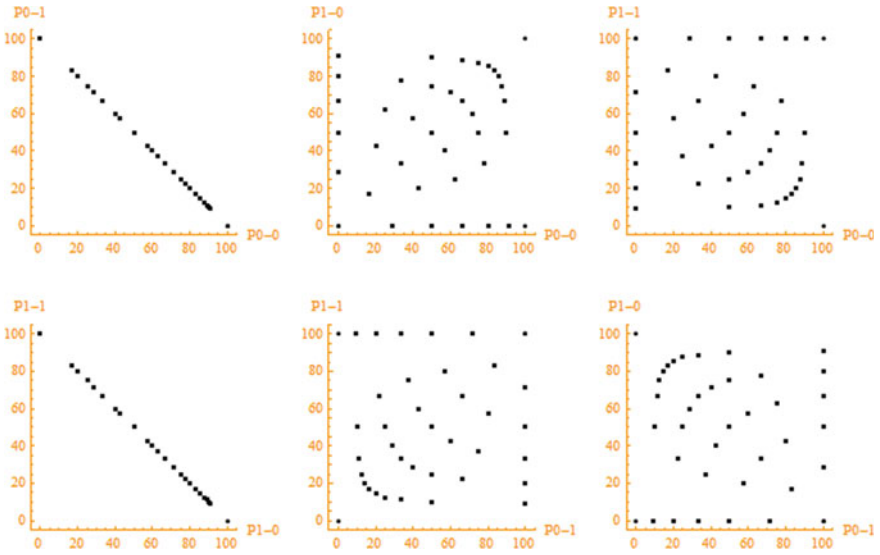


Fig. 3 2D conjugate maps constructed by separate six pairs of measures of No. 6 function; $N = 13$

4 Visualization Result

Because of the restriction of the structural complexity of the logical function, 16 functions of 2 variables are used to describe them in the way of exhaustion [14]. Output sequences are generated by different initial input sequences under the given logical function and then obtaining various measure data from the corresponding I/O sequence based on probability method. Then, the map is constructed using these measurement data.

This chapter chooses No. 1, 5, 6, and 13 functions which are typical functions as an example, observing the characteristic of three kinds of maps which are given in Fig. 4.

In (a) group of time sequence maps, only one measurement sequence transforms with time.

In (b) group of Poincare maps, different functions form different point clusters.

In (c) group of conjugate maps, the distribution of the points cluster has clear polarized properties.

According to the variable-value logic theory, three kinds of encoding model can be distinguished: W, F, and C [15].

The visualization information that can be acquired from a single function’s map is rather limited. In order to compare the spatial property of different logical functions, a 4×4 array is constructed using the maps that are generated from 16 logical functions in different encoding patterns as shown in Fig. 5.

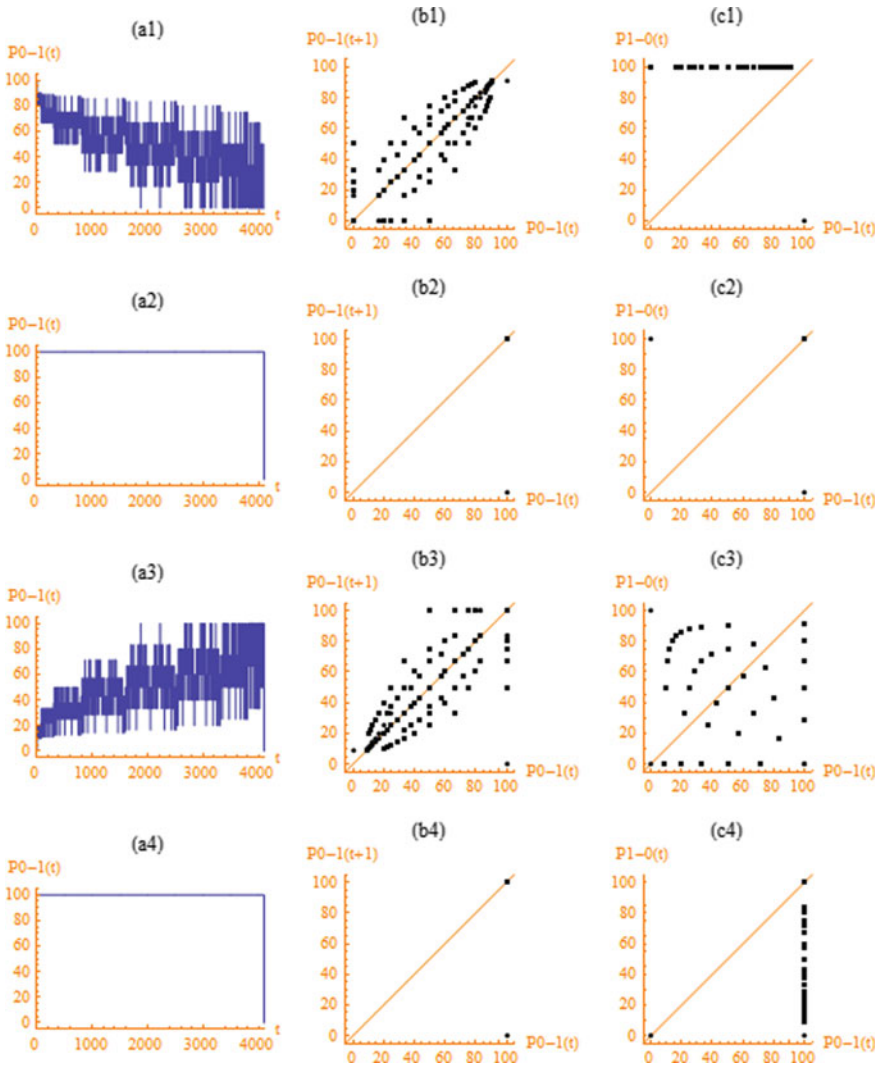


Fig. 4 Time sequence maps, Poincaré maps, and 2D conjugate maps. **a** Time sequence map; **b** Poincaré map; **c** 2D conjugate map

By assemble maps of total 16 logical functions under the models, the entire structure information among logical functions themselves can be observed.

To compare conveniently, combinations of 16 recursive images which generated from 16 functions are given in this chapter under different codes. Recursive images in W-code, F-code, and C-code from a given initial sequence are shown in Figs. 6, 7, and 8, respectively.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

W

0	2	1	3
4	6	5	7
8	10	9	11
12	14	13	15

F

0	4	1	5
2	6	3	7
8	12	9	13
10	14	11	15

C

Fig. 5 Assemble pattern of maps in W-code, F-code, and C-code

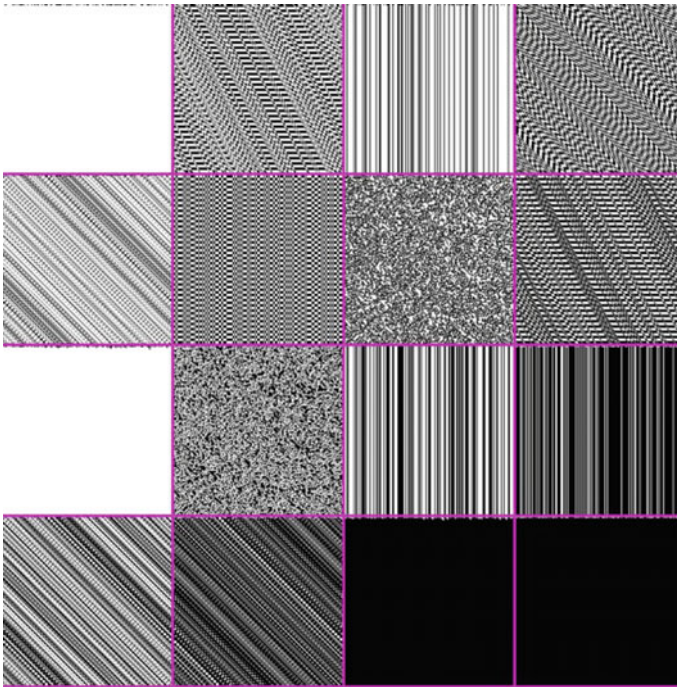


Fig. 6 Recursive images in W-code

The combination of time sequence map is shown in Fig. 9. The figure shows that different functions have different distribution properties, and also reveals the trend of single measurement's transforming with time.

The combination of Poincare map in W-code is shown in Fig. 10. Different distribution properties of functions can be observed from the figure. It is clear that there are four groups of configurations appeared in the figure: $\{0, 8, 2, 10\}$, $\{1, 3, 9, 11\}$, $\{4, 6, 12, 14\}$, $\{5, 7, 13, 15\}$.

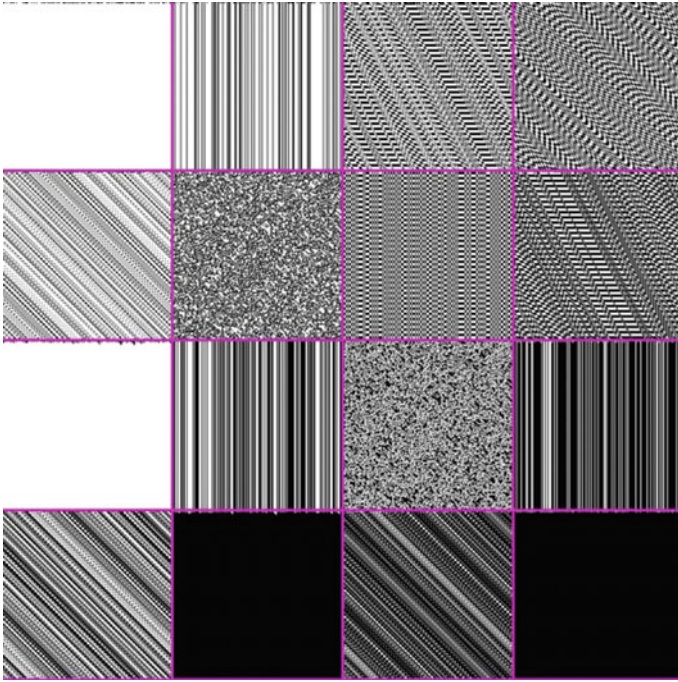


Fig. 7 Recursive images in F-code

For W-code, Poincare maps are shown in Fig. 10 and corresponding 2D conjugate maps are shown in Fig. 11. Conjugate maps have polarized properties, and their function pairs of 0:15, 1:7, 2:11, 4:13 and 8:14 have conjugate symmetry. In general, 16 conjugate maps are different from relevant maps generated by Poincare maps.

To arrange 16 Poincare maps and conjugate maps by F-code structure, F-code maps are shown in Figs. 12 and 13, respectively.

Under C-code structure, Poincare maps and conjugate maps are shown in Figs. 14 and 15.

In the above maps, 2D conjugate maps not only show spatial distributions of different logical functions but also have special holistic symmetries under the F- and C-code conditions.

5 Analyze

Through three types of different maps, three different coding schemes can be observed.

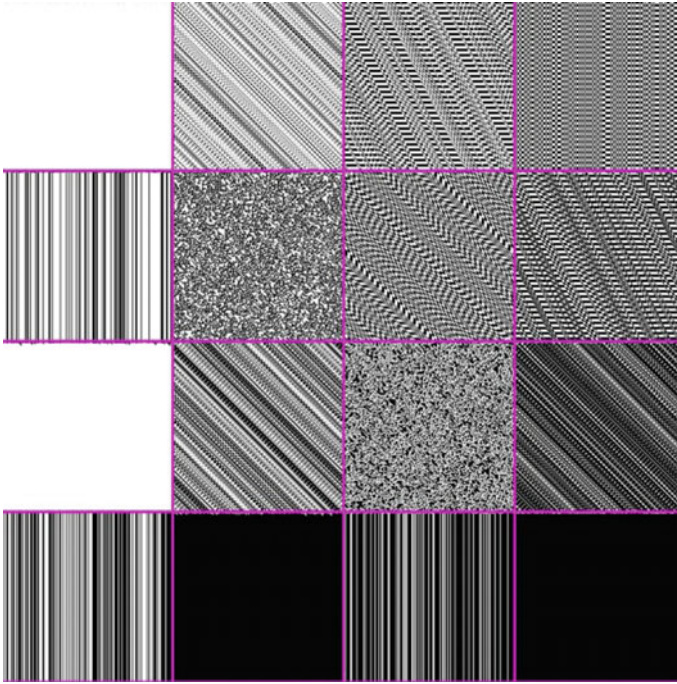


Fig. 8 Recursive images in C-code

Time sequence map can show the simple trend of single measurement series with time variations, but it was difficult for the scheme to describe spatial distributions of time sequence.

Poincare map can apply a single measurement sequence; although the map can be generated under different lengths in a correlation, information of distribution is naturally limited by the selected measurement sequence.

A 2D conjugate map uses two groups of independent measurements simultaneously; this scheme can show differences and connections between spatial distributions of logical functions; furthermore, through different coding models, it can illustrate holistic relationships among different functions, i.e., function pairs of 0:15, 1:7, 2:11, 4:13, and 8:14 have clear conjugated symmetry in conjugate maps. In addition, for C-code condition, the points of four functions on each edge of maps are located on the same side of edge. For example, points clusters of (0, 4, 1, 5), (0, 2, 8, 10), (10, 14, 11, 15), and (5, 7, 13, 15) functions are separately located on four sides of the 2D map space.

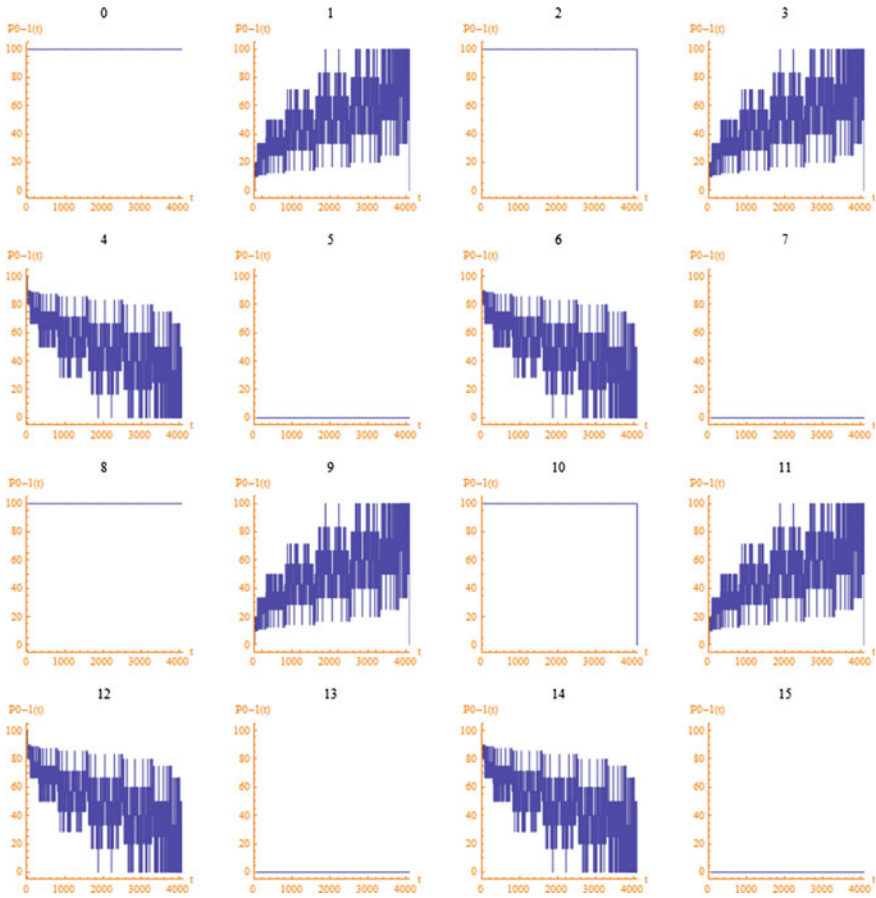


Fig. 9 Time sequence maps of 16 functions constructed by $\{t, P_{0-1}(t)\}$ sequences

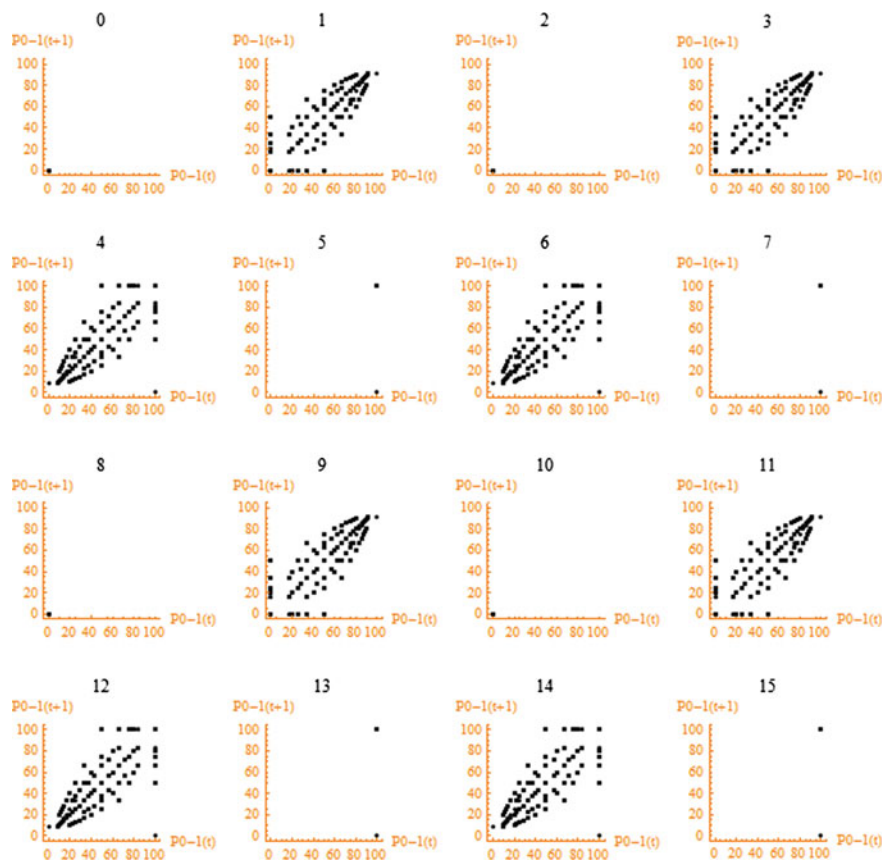


Fig. 10 Poincaré maps in W-code

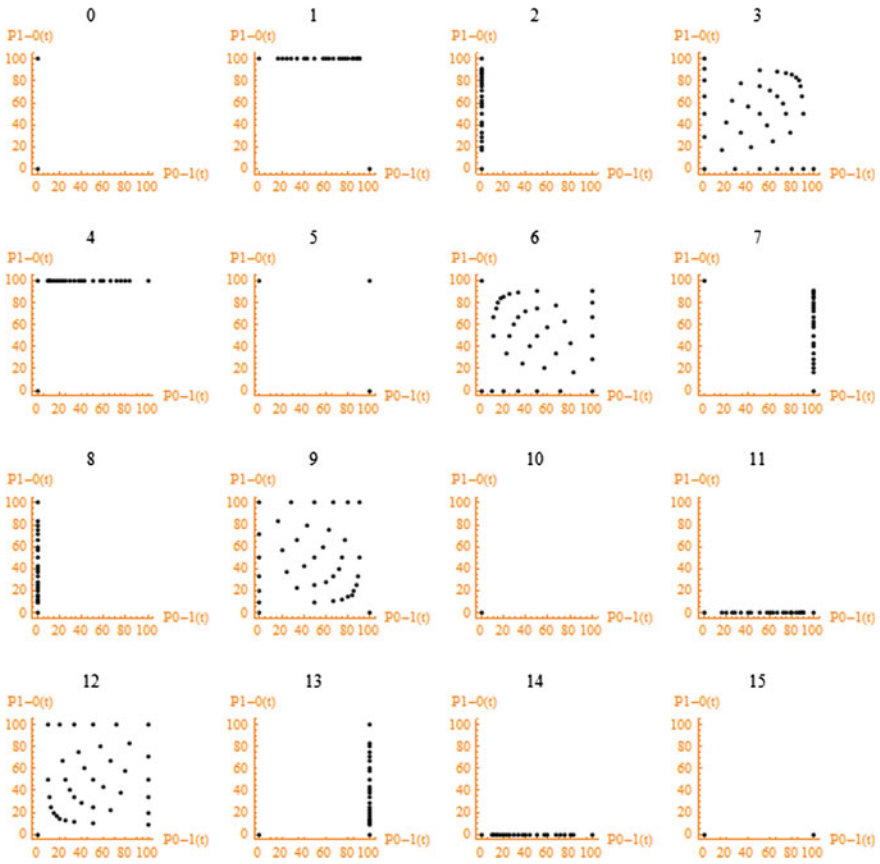


Fig. 11 Conjugate maps in W-code

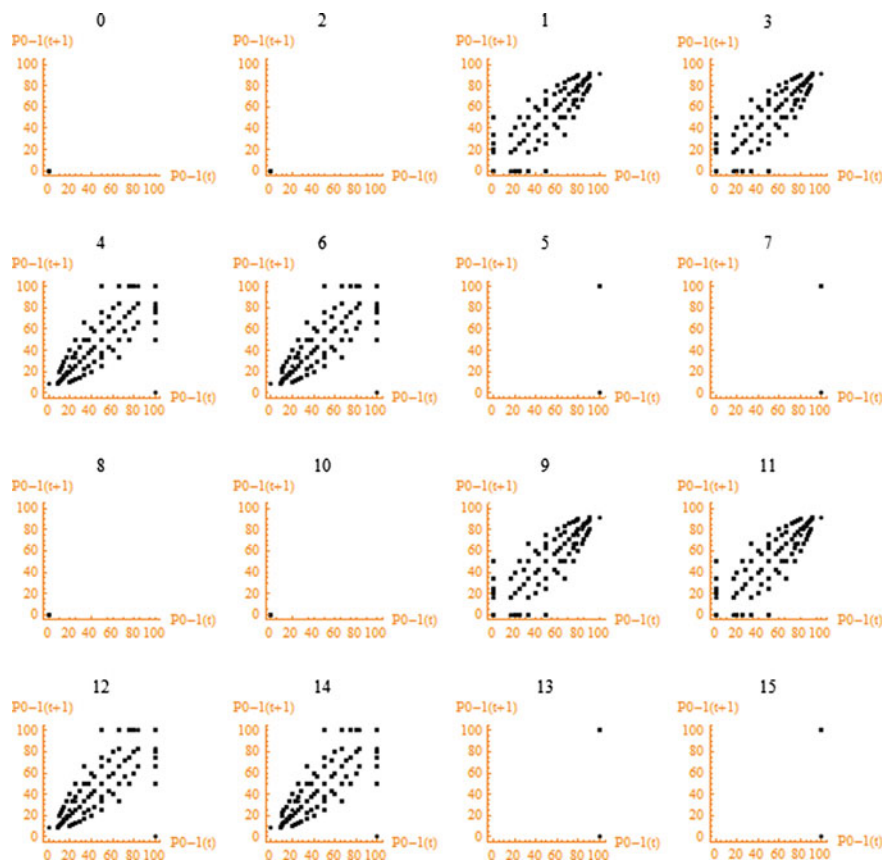


Fig. 12 Poincaré maps in F-code

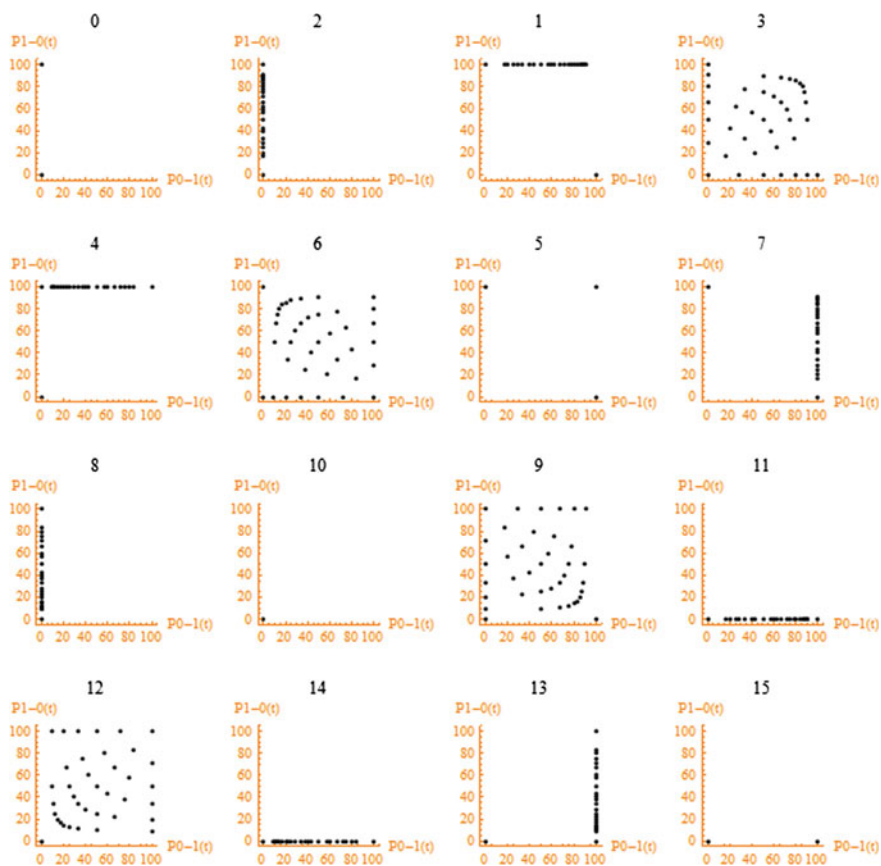


Fig. 13 Conjugate maps in F-code

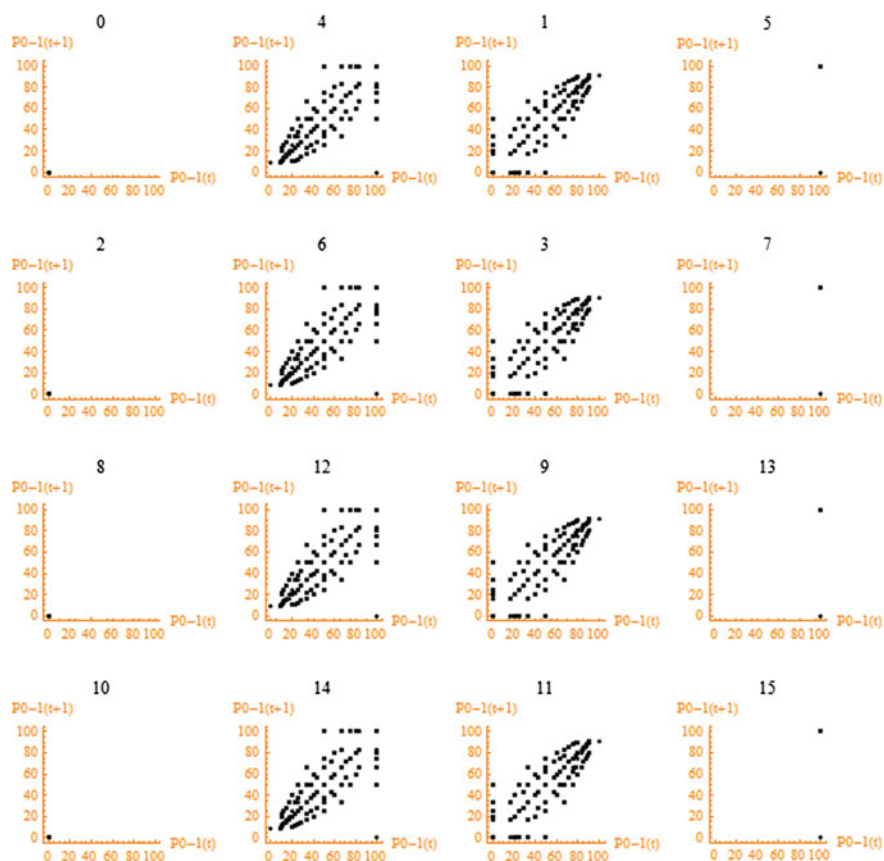


Fig. 14 Poincare maps in F-code

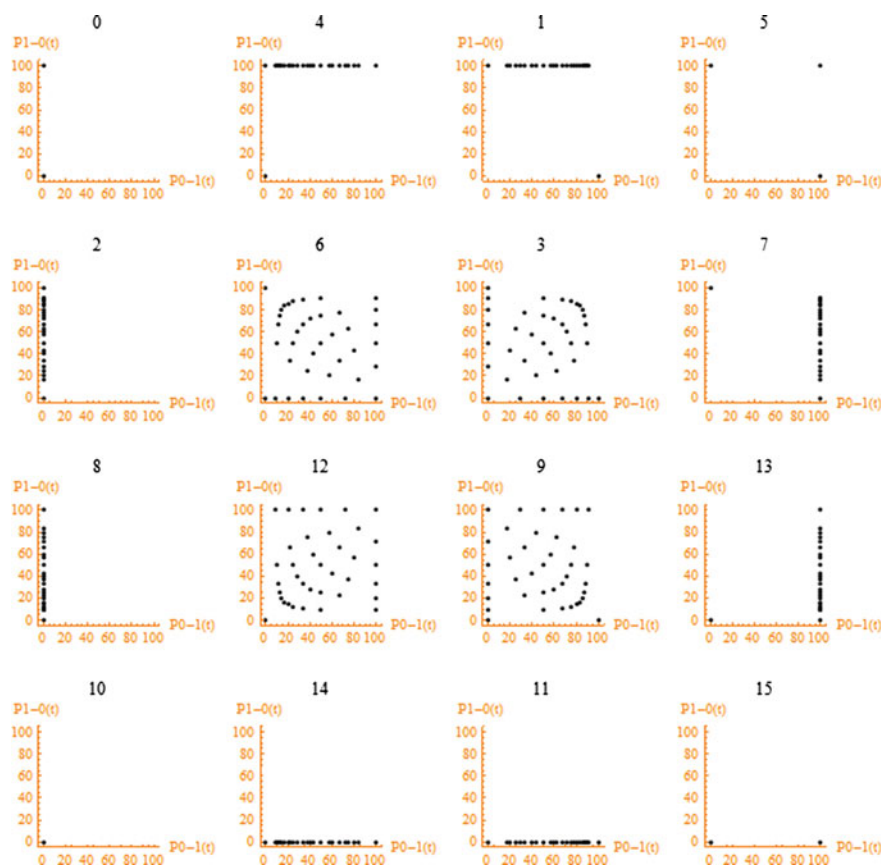


Fig. 15 Conjugate maps in C-code

6 Conclusion

Refined property of various time sequences can be identified from 2D conjugate maps to illustrate multiple measurement sequences under Markov chain mechanism. Spatial property of time sequence plays an important role in the study of dynamic sequence's behavior. The stable distribution under visualization method can help people understand relevant issues.

In comparison with Poincare maps and conjugate maps, there are additional properties in the complex dynamic sequences. Conjugate map method uses multiple parameters of Markov chains to make independent measurements simultaneously.

Proposed technology can provide further structural information among multiple measurements, and refined relationship via spatial distributions can be established. It is possible for the scheme to use statistical and probability methodologies to enhance visual tools of Markov chain mechanisms to resolve real problems and requirements for modern information warfare and information security applications in near future.

Acknowledgements Thanks goes to Mr. Jie Wan for helping him to generate data for this study and the special fund of Information Security (No. 2010KS06), Software School of Yunnan University to fund the project.

References

1. E.L. Key, An analysis of the structure and complexity of nonlinear binary sequence generators. *IEEE Trans. Inf. Theory* **IT-22**(6), 732–736 (1976)
2. D. Haccoun, A Markov chain analysis of the sequential decoding metric. *IEEE Trans. Inf. Theory* **26**(1), 109–113 (1980)
3. J.L. Massey, M.K. Sain, Certain infinite Markov chains and sequential decoding. *Discrete Math.* **3**(1–3), 163–175 (1972)
4. O.B. Sheynin, Markov's work on probability. *Artch. History Exact Sci.* **39**(3), 337–377 (1989)
5. S.E. Widnall, R.R. Fogelman, *Cornerstones of Information Warfare* (Doctrine/Policy Document, United States Air Force, 1997)
6. A. Borden, What is Information Warfare? *Aerospace Power Chronicles*, United States, Air Force, Air University, Maxwell AFB, Contributor's Corner (1999). <http://www.airpower.maxwell.af.mil/airchronicles/cc/borden.html>
7. S. Budiansky, *Battle of Wits: The Complete Story of Codebreaking in World War II* (Free Press, New York, 2000)
8. C. Kopp, B. Mills, Information warfare and evolution, in *Proceedings of the 3rd Australian Information Warfare & Security Conference*, ECU (2002)
9. D.E. Denning, *Information Warfare and Security* (Addison Wesley, Reading, MA, 1999)
10. S. Li, X. Tian, *Nonlinear Study and Complexity Study* (Harbin Institute of Technology Press, Harbin, 2006)
11. G. Pye, M. Warren, Appraising critical infrastructure systems with visualisation, in *10th Australian Information Warfare and Security Conference*, pp. 5–12 (2009)

12. Q. Li, J.Z.J. Zheng, Spacial distributions for measures of random sequences using 2D conjugate maps, *Proceedings of Asia-Pacific Youth Conference on Communication*, pp. 64–68 (2010)
13. S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific Press, Singapore, 1986)
14. J. Wan, J.Z.J. Zheng, Showing exhaustive number sequences of two logic variables for variant logic functional space, in *Proceedings of Asia-Pacific Youth Conference on Communication*, pp. 69–73 (2010)
15. J.Z.J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* **5**(2), 163–172 (2010). (Higher Education Press & Springer Press)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Permutation and Complementary Algorithm to Generate Random Sequences for Binary Logic



Jie Wan and Jeffrey Zheng

Abstract Randomness number generation plays a key role in network, information security, and IT applications. In this chapter, a permutation and complementary algorithm is proposed to use vector complementary and permutation operations to extend n -variable logic function space from 2^{2^n} functions to $2^{2^n} \times 2^n!$ configurations for variant logic framework. Each configuration contains 2^{2^n} functions that can be shown in a $2^{2^{n-1}} \times 2^{2^{n-1}}$ matrix. A set of visual results can be represented by their symmetric properties in W, F, and C codes, respectively, to provide the essential support on the variant logic framework.

Keywords Logic function · Permutation and complementary · Variant logic
Symmetric distribution · Random sequence

1 Introduction

Random numbers play an important role in many network protocols and encryption schemas on various network security applications [1], for example, digital signatures, authentication protocols, key generation for PKI, RSA/AES [2], nonce frustrate, and symmetric stream encryption. A better random number algorithm will enhance encryption schemas, to do other applications. To satisfy different requirements, the NIST has published a series of statistical tests as standards [3] to determine whether a random number generator is suitable for a cryptographic application. After using the

Project supported by Yunnan Advanced Overseas Scholar Project, NSF of China (61362014).

J. Wan
Yunnan University, Kunming, China
e-mail: wanjiech@163.com

J. Zheng (✉)
Key Laboratory of Yunnan Software Engineering, Yunnan University, Kunming 650091, Yunnan, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_14

vector complementary and the permutation operations on binary logic, the variant logical framework extends the traditional Logic function space from 2^{2^n} functions to $2^{2^n} \times 2^n!$ configurations [4]. Under the new extension conditions, it is possible to use simple transformation to generate huge numbers of random sequences for future applications.

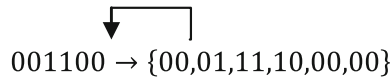
Permutation and complementary algorithm is described in the chapter to express different random properties through a series of binary image sequences undertaking typical recursive operations.

2 Method

Cellular automata perform a natural way to generate random sequence. The principle of binary cellular automata [5, 6] can be explained by an example as follows:

First, a sequence 001100 and a function $f : \{00 \rightarrow 0, 01 \rightarrow 1, 10 \rightarrow 1, 11 \rightarrow 0\}$ are selected.

Second, the sequence can be decomposed from left to right. The last bit is composed to the first bit



Third, according to the decomposed sequences and the generating function, a new sequence 010100 can be generated, i.e., $f : 001100 \rightarrow 010100$.

Followed the algorithm, the space of the generation function can be extended further; large numbers of random sequences can be generated. This mechanism can increase the complexity of code breaking.

In variant logic framework, the logic function space has been extended from 2^{2^n} to $2^{2^n} \times 2^n!$ by the permutation and the complementary operations. In two variable functions of cellular automata, there are 16 generated functions, and the 16 functions can be described in a truth table (Fig. 1a) with 16 entries.

2.1 Permutation Operation

The bit string of states {00, 01, 10, 11} in generating function can be converted to decimal number {0, 1, 2, 3}. An example in Fig. 1b is shown to permute 3210 to 1320 of the table.

(a).The Truth Table of 3210							(b).The Permutation Table of 1320						
J	P Status				K		J	P Status				K	
	3	2	1	0				1	3	2	0		
	11	10	01	00			01	11	10	00			
0	0	0	0	0	0		0	0	0	0	0	0	
1	0	0	0	1	1		1	0	0	0	1	1	
2	0	0	1	0	2		2	1	0	0	0	8	
3	0	0	1	1	3		3	1	0	0	1	9	
4	0	1	0	0	4		4	0	0	1	0	2	
⋮	⋯	⋯	⋯	⋮	⋮		⋮	⋯	⋯	⋯	⋮	⋮	
⋮	⋯	⋯	⋯	⋮	⋮		⋮	⋯	⋯	⋯	⋮	⋮	
13	1	1	0	1	13		13	0	1	1	1	7	
14	1	1	1	0	14		14	1	1	1	0	14	
15	1	1	1	1	15		15	1	1	1	1	15	

$$P \begin{pmatrix} 3210 \\ 1320 \end{pmatrix} \longrightarrow$$

Fig. 1 Permutation example

2.2 Complementary Operation

In the complementary operation, the complementary vector σ is applied to operate the truth table.

It can be described as

$$y^\delta = \begin{cases} y, & \delta = 1 \\ \bar{y}, & \delta = 0 \end{cases}$$

In two-variable variant logic, σ is a binary sequence of 4 bits in {0000, ..., 1111}. In the example, the original table is $\sigma = 1111$ and shown in Fig. 2a given $\sigma = 1100$ in Table 2 which can be described as $1320^{(1100)} = 1^13^12^00^0$. Under such operation, the sequence values of state 1 and 3 columns are invariant. But the values of columns whose index is 0 and values of the permutation sequence in state 2 and 0 are changed to their revised values, respectively.

After the complementary operation, Fig. 2a changes to Fig. 2b.

2.3 Visualization

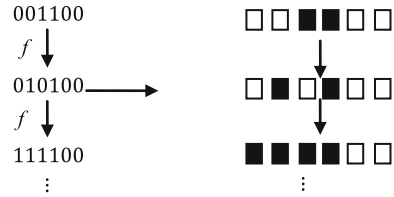
For function f , once applied on the sequence 001100 to output 010100, then this function can be applied on the sequence 010100 to output 111100. For such binary sequence, select black for 1 and white for 0 to generate the visual patterns as follows (Fig. 3).

(a).The Permutation Table of 1320 ⁽¹¹¹¹⁾							(b).The Complementary Table of 1320 ⁽¹¹⁰⁰⁾							
σ						σ								
1 1 1 1						1 1 0 0								
J	P Status				K	J	P Status				K			
	1	3	2	0		1	3	2	0		1	3	2	0
	01	11	10	00		01	11	10	00		01	11	10	00
0	0	0	0	0	0	0	0	0	1	1	3			
1	0	0	0	1	1	1	0	0	1	0	2			
2	1	0	0	0	8	2	1	0	1	1	11			
3	1	0	0	1	9	3	1	0	1	0	10			
4	0	0	1	0	2	4	0	0	0	1	1			
⋮	⋯	⋯	⋯	⋮	⋮	⋮	⋯	⋯	⋯	⋯	⋮			
⋮	⋯	⋯	⋯	⋮	⋮	⋮	⋯	⋯	⋯	⋯	⋮			
13	0	1	1	1	7	13	0	1	0	0	4			
14	1	1	1	0	14	14	1	1	0	1	13			
15	1	1	1	1	15	15	1	1	0	0	12			

$\sigma = 1100 \longrightarrow$

Fig. 2 Complementary example

Fig. 3 Visualize the random sequence



2.4 Matrix Representation

For example (Fig. 2b), the truth value of third function is 1010. It can be converted to a binary coordinate $\langle 10|10 \rangle$ distinguished by left two and right two bits, respectively. So the decimal coordinate is $\langle 2|2 \rangle$. Then Fig. 2b can be converted to Table 1.

Under such conversion, the 2D matrix can be represented in Table 2.

3 Algorithm and Properties

3.1 Permutation and Complementary Algorithm

Using permutation and complementary operations, an algorithm is extended to express the n -ary variant logic functional space.

Table 1 Coordinate map of $1320^{(1100)}$

J	σ				Transformed bracket
	1	1	0	0	
P Status					
	1	3	2	0	
	01	11	10	00	
0	0	0	1	1	$\langle 0, 3 \rangle$
1	0	0	1	0	$\langle 0, 2 \rangle$
2	1	0	1	1	$\langle 2, 3 \rangle$
3	1	0	1	0	$\langle 2, 2 \rangle$
4	0	0	0	1	$\langle 0, 1 \rangle$
⋮	⋮	⋮
⋮	⋮	⋮
13	0	1	0	0	$\langle 1, 0 \rangle$
14	1	1	0	1	$\langle 3, 1 \rangle$
15	1	1	0	0	$\langle 3, 0 \rangle$

Table 2 2D matrix of the $1320^{(1100)}$

$\langle 0, 0 \rangle$ 5	$\langle 0, 1 \rangle$ 4	$\langle 0, 2 \rangle$ 1	$\langle 0, 3 \rangle$ 0
$\langle 1, 0 \rangle$ 13	$\langle 1, 1 \rangle$ 12	$\langle 1, 2 \rangle$ 9	$\langle 1, 3 \rangle$ 8
$\langle 2, 0 \rangle$ 7	$\langle 2, 1 \rangle$ 6	$\langle 2, 2 \rangle$ 3	$\langle 2, 3 \rangle$ 2
$\langle 3, 0 \rangle$ 15	$\langle 3, 1 \rangle$ 14	$\langle 3, 2 \rangle$ 11	$\langle 3, 3 \rangle$ 10

Algorithm: Permutation and Complementary:

Input: variable n

Output: a set of truth table of $P^\sigma, \forall P \in S(2^n), \forall \sigma \in B_2^{2^n}$.

Method:

Step 1. Initial $T = \{2^n 2^n - 1 \dots \dots 10\}$

Step 2. Generate a permutation P for T

Step 3. From $\sigma = 000 \dots 0$ to $111 \dots 1$ do vector complementary operation.

Step 4. Any new permutation?

Yes go to Step 2.

Step 5. End

where $S(N)$ is a symmetry group with N member and B_2^M is an M variable Boolean structure with 2^M members.

Table 3 2D matrix for n-ary logic functions

$\langle 0, 0 \rangle$	$\langle 0, 2^{2^{n-1}} - 1 \rangle$
$\langle 1, 0 \rangle$	$\langle 1, 2^{2^{n-1}} - 1 \rangle$
\vdots	\vdots	\vdots	\vdots
$\langle 2^{2^{n-1}} - 2, 0 \rangle$	$\langle 2^{2^{n-1}} - 2, 2^{2^{n-1}} - 1 \rangle$
$\langle 2^{2^{n-1}} - 1, 0 \rangle$	$\langle 2^{2^{n-1}} - 1, 2^{2^{n-1}} - 1 \rangle$

Table 4 The number of W, F, and C codes in 2-ary variant functional space

Code system	No
W	384
F	128
C	16

3.2 Representation Scheme

Every truth table has a 2D matrix to arrange visual results of random sequence. The $\langle X, Y \rangle$ is the coordinate to allocate each visual result. So for n-ary logic function space, the 2D matrix has a size of $2^{2^{n-1}} \times 2^{2^{n-1}}$ as shown in Table 3.

3.3 W, F, and C

Three coding schemes can be distinguished in the algorithm.

W code [4] is a binary sequence of 2^n bits. It separates into two parts, $(J^1|J^0)$. Each part has 2^{n-1} bits.

F code is a subset of W code, and it is a symmetry code. In F code, if the I th meta-state in J^1 is 1 or 0, the I th meta-state in J^0 is the negative state.

If a code is F code, the I th meta-state in J^1 has the same value. Besides, four corners of its matrix are included in $\{0, x, \bar{x}, 1\}$; it is C code [4].

For example, $(32|10)(1110|0100)$ is an element of W code. In the sequence, 1 is not the negative sequence of 3, and the 0 is not also the negative sequence of 2. $(32|01)(1110|0001)$ is an F code. It has the symmetry property. In the sequence, 0 is the negative sequence of 3 and 1 is the negative sequence of 2. $(13|20)(0111|1000)$ is a C code. It has the symmetry property of F code, and four comers of 1320's matrix are included in $\{0, x, \bar{x}, 1\}$.

The further definition of W, F, and C codes can be found in [4].

From the exhaustive of the binary variant function space, the number of W, F, and C codes in binary variant function space [7] is shown in Table 4.

4 Coding Simples

W Code:
 Permutation sequence: 3210
 The value of σ :1011

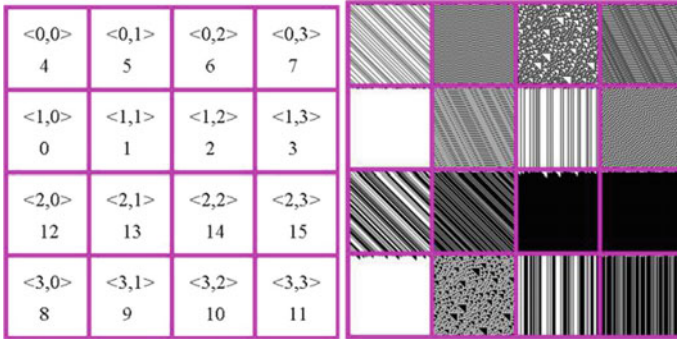


Fig. 4 The 2D matrix diagram and the visual result of 3210¹⁰¹¹

F Code:
 Permutation sequence: 3201
 The value of σ : 1111

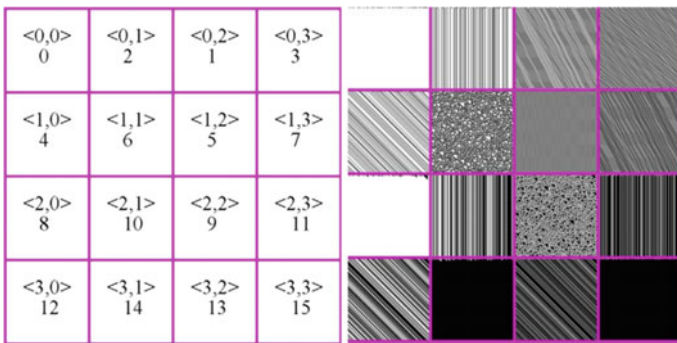


Fig. 5 The 2D matrix diagram and the visual result of 3201¹¹¹¹

C Code:
 Permutation sequence: 1320
 The value of σ :1100

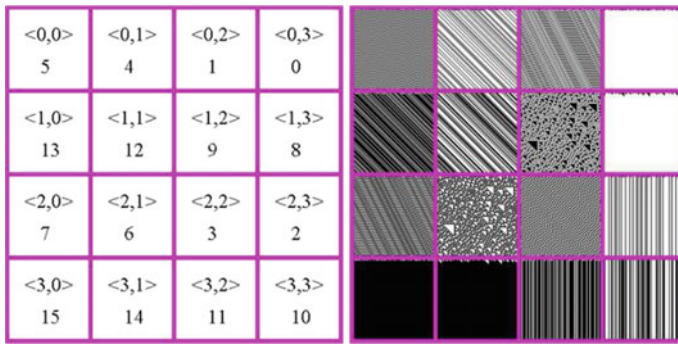


Fig. 6 The 2D matrix diagram and the visual result of 1320^{1100}

5 Result Analysis

In Fig. 4, W code is shown as a general code. Majority W code does not have apparent symmetry property. W code covers all the code spaces which are formed from binary input variable. These properties can be seen in Fig. 4.

All the F codes have overall symmetry in 2D distribution. Obvious symmetry among functions in the 2D matrix can be observed in Fig. 5.

Simple is shown in a C code in Fig. 6. It is a small set of F code with complete symmetry property. C code has the four-constant vertex property. The group of the four vertexes in C code are located by 0, 15, 10, and 5 functions, respectively.

In the n -ary logical function permutation and complementary algorithm, the permutation is operated for $2^n!$; the complementary exhaustive needs 2^{2^n} operation for each permutation operation. A total of computational complexity of an n -ary variant logical function using permutation and complementary algorithm is $O(2^n! \times 2^{2^n})$.

6 Conclusion

A permutation and complementary algorithm has been proposed for n -ary logical function, and sample results are visualized. The visual results of W, F, and C codes in the variant and invariant properties support the variant logic system through experimentation to use an algorithmic mechanism to generate a series of huge random number sequences.

References

1. W. Stallings, *Cryptography and Network Security: Principles and Practice* (Pearson Education, 2006)
2. J. Soto, L. Bassham, *Randomness Testing of the Advanced Encryption Standard Finalist Candidates* (NIST, 2000)
3. *Random number generation* (NIST, 2008), <http://csrc.nist.gov/groups/ST/toolkit/rng/index.html>
4. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* **5**(2) (2010) (Higher Education Press & Springer Press)
5. S. Wolfram, *Theory and Applications of Cellular Automata* (Word Scientific, Singapore, 1986)
6. S. Wolfram, Cellular automata as models of complexity. *Nature* **311** (4 October 1984)
7. J. Wan, J. Zheng, Showing exhaustive number sequences of two logic variables for variant logic functional space, in *Proceedings of Asia-Pacific Youth Conference on Communication (APYCC)*, p. 4 (October 2010)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



3D Visual Method of Variant Logic Construction for Random Sequence



Huan Wang and Jeffrey Zheng

Abstract As Internet security threats continue to evolve, in order to ensure information transmission security, various encrypts and decrypts have been used in channel coding and decoding of data communication. While cryptography requires a very high degree of apparent randomness, random sequences play an important role in cryptography. Both Cellular Automata (CA) and RC4 contain pseudorandom number generators and may have intrinsic properties, respectively. In this chapter, a 3D visualization model 3DVM is proposed to display spatial characteristics of the random sequences from CA or RC4 keystream. Key components of this model and core mechanism are described. Every module and their I/O parameters are discussed, respectively. A serial of logic function of CA is selected as examples to compare with some RC4 keystreams to show their intrinsic properties in three-dimensional space. Visual results are briefly analyzed to explore their intrinsic properties including similarity and difference. The results provide support to explore the RC4 algorithm by using 3D dimensional visualization tools to organize its interactive properties as visual maps.

Keywords Pseudorandom sequence · CA · Stream cipher · RC4 keystream
3D maps

1 Introduction

Wireless Sensor Networks WSN and Wireless Networks WN are most popular and widely used types of network of this era. Because of the openness these types of

Project supported by NSF of China (613620214), the Key R&D project of Yunnan Higher Education Bureau (K1059178) and Yunnan Advanced Overseas Scholar Project (W8110305).

H. Wang
Yunnan University, Kunming, China
e-mail: lights127@gmail.com

J. Zheng (✉)
Key Laboratory of Yunnan Software Engineering, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_15

networks are not very much secure. To provide the security over the WSN and WN, algorithm used must be fast enough which can encrypt and decrypt data comparatively in less amount of time to require less resource too. In this concern, Wi-Fi Protected Access WPA and Wired Equivalent Privacy WEP protocols are used as standard. These standards have adopted the RC4 stream cipher algorithm to secure the data over the WN environment. These standard adopted RC4 algorithms because RC4 algorithm gives speedy encryption and decryption of data, utilize less hardware resource during processing, and easy to implement [1, 2]. Presently, RC4 algorithm is not secure in many aspects. Lots of weaknesses and attacks have been detected by the cryptanalysis [3, 4].

1.1 The Weakness of RC4

RC4 algorithm is a stream cipher under the symmetric ciphers algorithms. Typically, in a stream cipher, the keystream is the sequence which is combined digit by digit to the plaintext sequence for obtaining the ciphertext sequence. However, the data encryption is equivalent to a simple XOR with keystream. The keystream is generated by a finite state automaton called the keystream generator [5, 6]. The encryption can be broken if the plaintexts are encrypted using the same keystream. RC4 keystream generated by RC4 keystream generator is completely compromising the security of RC4.

Because it is very hard to trace the characteristics of keystream generators, random characteristics of keystream can be investigated on spatial characteristics of the keystream generator to test pseudorandom sequences. This chapter is the expansion work of [7] by Qingping Li from 2D to 3D. In this chapter, random sequences from given keystreams are collected in comparison with random sequences generated by sample logical function of 1D Cellular Automata to show their intrinsic properties in three-dimensional space of relationships.

1.2 CA

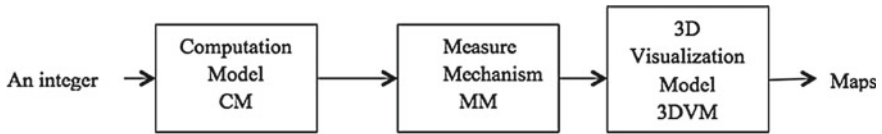
Cellular Automata is a great discovery in the twentieth century, and it forms a time series according to a given function in an iterations process by introducing logic function and related calculation methods in the natural pattern [8]. In 1985, S. Wolfram formed the sequential cipher from pseudorandom sequence generated from logic calculation using cellular automata. Because of the implicated expression of the logic function, the spatial characteristic cannot be directly observed from the function formula [9].

2 Architecture

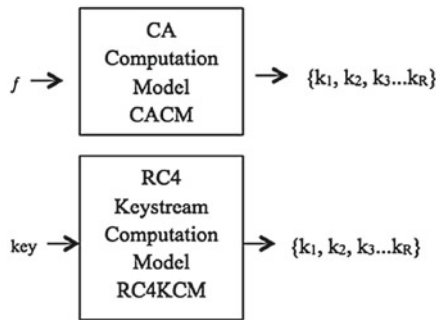
2.1 Architecture

The architecture is shown in Fig. 1a. The three main components and their modules are shown in Fig. 2b–d, respectively.

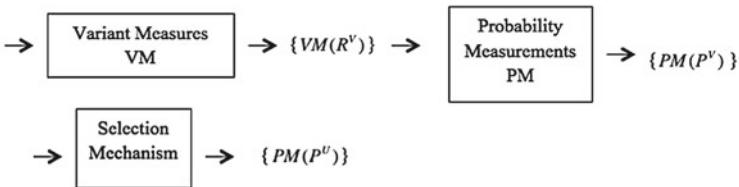
In the first part of this system, two types of data sets are generated by CACM and RC4KCM, respectively. The data sets on either CACM or RC4KCM get into



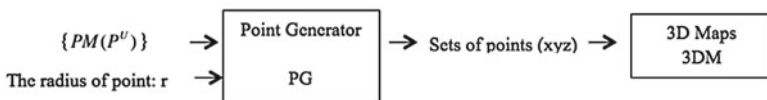
(a) Architecture



(b) CM



(c) MM



(d) 3DVM 3D Visualization Model

Fig. 1 Variant 3D visualization system and key components

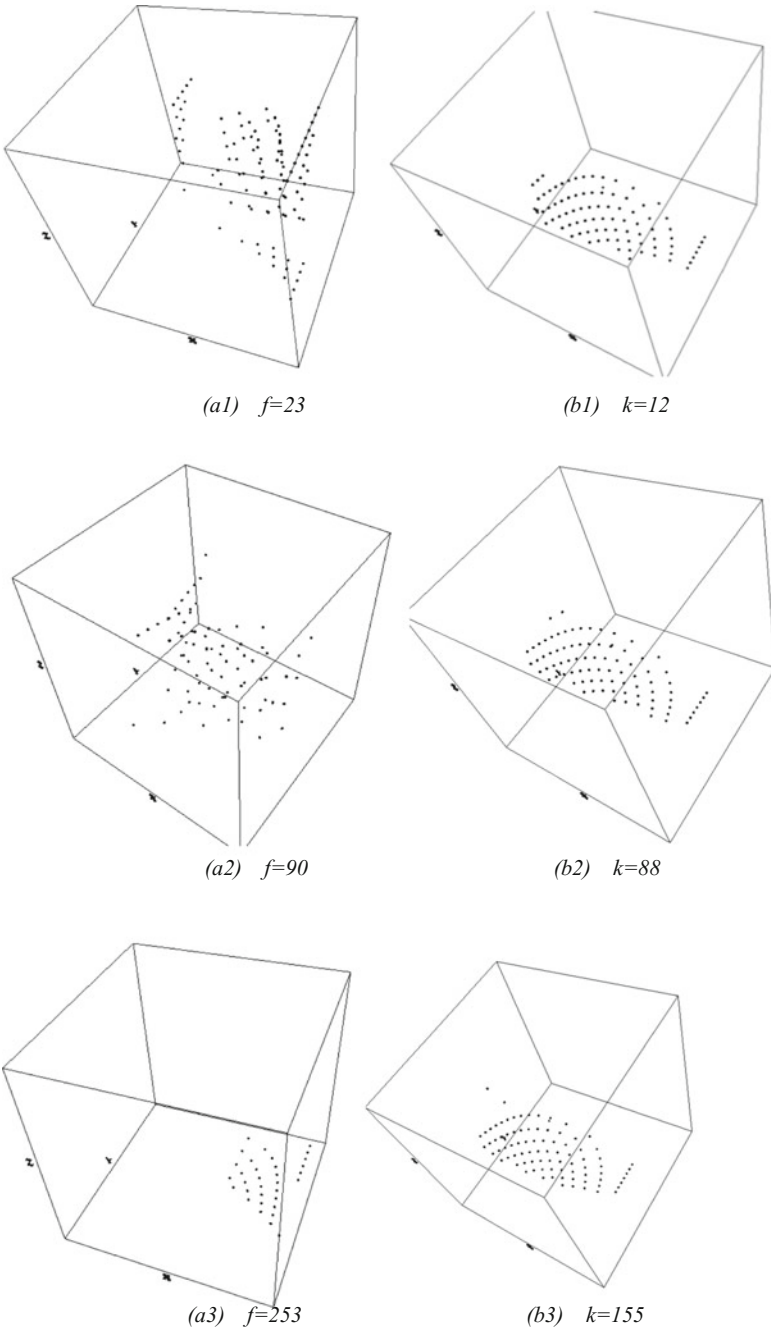


Fig. 2 Two sets of six 3D maps based on unified model in different conditions; **a1–a3** for the file CA; **b1–b3** for the file RC4

the MM module as input data. The main function of the VM is to output the four vectors of variant measurements. Using unified or non-unified method, six probability measurements are created by PM module. In order to establish 3D maps, three vectors of probability measurements are selected from the six probability measurements by the SM module. Three vectors determine a 3D spatial position. All vectors generate a 3D map using 3DVM.

There are six parameters in an input group, three sets of parameters in the intermediate group, and one set of parameters in the output group.

Input Group:

An integer indicates the serial number of logic function or the value of the key selected

An integer indicates which model is selected

An integer indicates the number of elements in the binary sequence

An integer indicates the number of elements in a segment

An integer indicates the method of selection mechanism

An integer indicates the control parameter for mapping

Intermediate Group:

A 0-1 vector generated by CA logic function or RC4 keystream generator

A set of four variant measures

A set of six probability vectors

Output Group:

3D maps

2.2 Computation Model of CA (CMCA)

CMCA module is used to measure the features of a logic function based on Cellular Automata (CA). Consider a logic function $f: Y = f(X)$ as a function of CA, the output sequence Y can be generated by the given initial input sequence X with 2 states. For N bits initial input sequence, a total of 2^N states are generated under the logic function $f: X \rightarrow Y$. A pair of vectors (X, Y) could be collected for their correspondences on the pair of input-output relationships. There are 2^N groups of this corresponding relationship.

Input Group:

X A 0-1 vector with N elements, $X \in B_2^N$

n An integer indicating a 0-1 vector with n elements,

f A function with 2 variables

Intermediate Group:

Y A 0-1 vector with N elements, $Y \in B_2^N$

Output Group:

$\forall Y$ Exhaustive set of all states of N bit vectors with 2^n elements

2.3 Computation Model of RC4 Keystream (RC4KCM)

For an L bits input keystream K , divided into G segments and $W = L/G$ bits of each segment with $G < L$. The value of parameter G determines the amount of points and W determines the spatial distribution for the output keystream in the phase space.

Input Group:

A 0-1 vector with L elements generated by RC4 keystream generator

- L An integer indicates the number of elements in an input sequence,
- G An integer indicates the number of segments divided,
- W An integer indicates the number of elements in a segment.

Output Group:

G sets of W bits 0-1 vectors

The CMRC4 component uses an input vector as input, under different segment strategies to divide into several segments. The output of this component is G sets of W bits 0-1 vectors.

2.4 Measure Mechanism (MM)

The MM component shown in Fig. 1c is composed of three modules: Variant Measure (VM), Probability Measurement (PM), and Selection Mechanism (SM). Three parameters are listed as input signals; four variant measures are outputted from VM module, six probability measurements are created from variant measures by Probability Measurement (PM), under the Selection Mechanism (SM) module, and a set of triples interactive projections is selected.

Input Group:

- V A symbol is selected from four types of transformations $\{\perp, +, -, T\}$,
- N An integer indicates the number of elements in an input vector

A 0-1 data vector

Intermediate Group:

- $VM(R^V)$ A set of four variant measures
- $PM(P^V)$ A set of four probability vectors

Output Group:

$U \subset V$ A set of three interactive projections under the SM condition, $U \subset V$
 $PM(P^U)$ A set of three probability vectors

2.5 Variant Measure (VM)

Considering the transformation of every bit between input sequence $\{X_i\}_{i=0}^{N-1}$ and output sequence $\{Y_i\}_{i=0}^{N-1}$, there are a total of four types of transformations: $0 \rightarrow 0$, $0 \rightarrow 1$, $1 \rightarrow 0$, and $1 \rightarrow 1$ [10, 11].

Define the variant representation as follows:

$$V = \begin{cases} \perp, X_i = 0, Y_i = 0; \\ +, X_i = 0, Y_i = 1; \\ -, X_i = 1, Y_i = 0; \\ T, X_i = 1, Y_i = 1; \end{cases} \quad 0 \leq i \leq N, \quad X_i, Y_i \in B_2$$

For any N bit 0-1 vector $X, X = X_0X_1 \dots X_i \dots X_{N-1}X_N, 0 \leq i \leq N, X_i \in B_2, X_i \in B_2^N$ under 2-variable function f , N bit 0-1 output vector $Y, Y = Y_0Y_1 \dots Y_i \dots Y_{N-1}Y_N, 0 \leq i \leq N, Y_i \in B_2, Y_i \in B_2^N$. Let Δ be the variant measure function.

$$\Delta(X \rightarrow Y) = \sum_{i=0}^{N-1} \Delta(X_i \rightarrow Y_i) = \langle R_{\perp}, R_+, R_-, R_T \rangle, \quad N = R_{\perp} + R_+ + R_- + R_T, R_0 \\ = R_{\perp} + R_+, R_1 = R_- + R_T$$

Example

$N = 13, Y = f(X)$.

$$X = 1001011100101$$

$$Y = 0010110101100$$

$$\Delta(X \rightarrow Y) = -\perp + - + T - T\perp + T-$$

$$\langle R_{\perp} + R_+ + R_-, R_T \rangle = \langle 3, 3, 4, 3 \rangle, R_0 = 6, R_1 = 7, N = 13$$

Input and output pairs are 0-1 variables for only four combinations. For any given function, the quantitative relationship of $\{\perp, +, -, T\}$ is directly derived from the input/output sequences. Four meta measures are determined [12].

Input Group:

V A symbol is selected from four types of transformations $\{\perp, +, -, T\}$,

N An integer indicates the number of elements in an input vector

A 0-1 data vector

Output Group:

- $VM(R^V)$ A set of four variant measures
- R_0 An integer indicates the number of 0 in an input vector
- R_1 An integer indicates the number of 1 in an input vector

2.6 Probability Measurement (PM)

Variant measure parameters and the other three parameters are listed as input signals; the output of probability signals is calculated as eight measurements in two groups by following the given equations.

The first group of probability signal vectors ρ is called a non-unified model and defined as follows:

$$\left\{ \begin{array}{l} \rho = \frac{R^V}{N} = R_{\perp}, R_{+}, R_{-}, R_{\top} \\ \rho_{\alpha} = \frac{R_{\alpha}}{N}, \alpha \in \{\perp, +, -, \top\} \end{array} \right. \quad \& \quad \left\{ \begin{array}{l} \rho_0 = \frac{R_0}{N} \\ \rho_1 = \frac{R_1}{N} \end{array} \right.$$

The second group of probability signal vectors $\tilde{\rho}$ is called a unified model and defined as follows:

$$\left\{ \begin{array}{l} \tilde{\rho} = \frac{R^V}{R_0 R_1} = R_{\perp}, R_{+}, R_{-}, R_{\top} \\ \rho_{\alpha} = \frac{R_{\alpha}}{R_0}, \alpha \in \{\perp, +\} \\ \rho_{\beta} = \frac{R_{\beta}}{R_1}, \beta \in \{-, \top\} \end{array} \right. \quad \& \quad \left\{ \begin{array}{l} \rho_0 = \frac{R_0}{N} \\ \rho_1 = \frac{R_1}{N} \end{array} \right.$$

Under such condition, the output signals of the PM module can be expressed as a pair of probability vectors in quaternion forms $PM(P^V) = \{\rho, \tilde{\rho}\}$.

Input Group:

- V A symbol is selected from four types of transformations $\{\perp, +, -, \top\}$,
- N An integer indicates the number of elements in an input vector
- $VM(R^V)$ A set of four variant measures
- R_0 An integer indicates the number of 0 in an input vector
- R_1 An integer indicates the number of 1 in an input vector

Output Group:

- $PM(P^V)$ A set of four probability vectors

2.7 Selection Mechanism Module

The SM Module is composed of two models: Non-unified Model and Unified Model. Under different constructions, two models are established respectively as follows.

Non-unified Model

Selecting two measurements from four combinations $\{\tilde{\rho}_\perp, \tilde{\rho}_+, \tilde{\rho}_-, \tilde{\rho}_T\}$, there will be C_4^2 choices. And then selecting one measurement from two combinations $\{\rho_0, \rho_1\}$, there will be C_2^1 choices. A 3-tuple S is defined as follows:

$$\begin{cases} S = (\rho_\alpha, \rho_\beta, \rho_\gamma) \\ S' = (\rho_\beta, \rho_\alpha, \rho_\gamma), & \alpha, \beta \in V, \gamma \in \{0, 1\}, \alpha \neq \beta \\ S = S' \end{cases}$$

Unified Model

Selecting two measurements from four combinations $\{\tilde{\rho}_\perp, \tilde{\rho}_+, \tilde{\rho}_-, \tilde{\rho}_T\}$, there will be C_4^2 choices. And then selecting one measurement from two combinations $\{\rho_0, \rho_1\}$, there will be C_4^2 choices. A 3-tuple \tilde{S} is defined as follows:

$$\begin{cases} \tilde{S} = (\tilde{\rho}_\alpha, \tilde{\rho}_\beta, \tilde{\rho}_\gamma) \\ \tilde{S}' = (\tilde{\rho}_\beta, \tilde{\rho}_\alpha, \tilde{\rho}_\gamma), & \alpha, \beta \in V, \gamma \in \{0, 1\}, \alpha \neq \beta \\ \tilde{S} = \tilde{S}' \end{cases}$$

Under such condition, the output signals of the SM module can be expressed as a 3D visual model in 3-tuples forms S or \tilde{S} . Specifically ρ_α or $\tilde{\rho}_\alpha$ determines the value of X-axis, ρ_β or $\tilde{\rho}_\beta$ determines the value of Y-axis, and ρ_γ or $\tilde{\rho}_\gamma$ determines the value of Z-axis.

Input Group:

$PM(P^V)$ A set of four probability vectors

Output Group:

$U \subset V$ A set of three interactive projections under the SM condition, $U \subset V$

$PM(P^U)$ A set of three probability vectors

2.8 Visualization Model

Using a visual model, *all possible measurements are calculated exhaustively on all G-1 vectors*. Each 3-tuple can be drawn as a point in three-dimensional space (xyz-space). All G-1 points are constructed in the phase space for the selected keys.

3 Sample Results on 3D Maps

In this section, two types of data sets are selected to illustrate their differences on 3D maps for comparison. The first type of data sets is generated by CA. The second type of data sets is generated by RC4.

3.1 Visualization Results of Unified Model

See Fig. 2.

3.2 Visualization Results of Non-unified Model

See Fig. 3.

3.3 Visualization Results of CA with Different Length of Initial Sequence

See Fig. 4.

3.4 Visualization Results of RC4 Keystream with Different Segment Strategies

See Fig. 5.

4 Analysis of Results

The above 27 3D maps contain different information. Some important conclusions will be discussed in detail in this section.

The first group of results shown in Fig. 2 presents two sets of six 3D maps constructed by the unified model from two data files: CA and RC4 to illustrate their 3D spatial characteristics. Three 3D maps of each group in Fig. 2a1–a3 show 3D spatial characteristics of CA with different logic functions. In this group, No. 23, 90, 253 functions are selected as examples to compare each other. And three 3D maps of each group in Fig. 2b1–b3 show 3D spatial characteristics of RC4 with 20

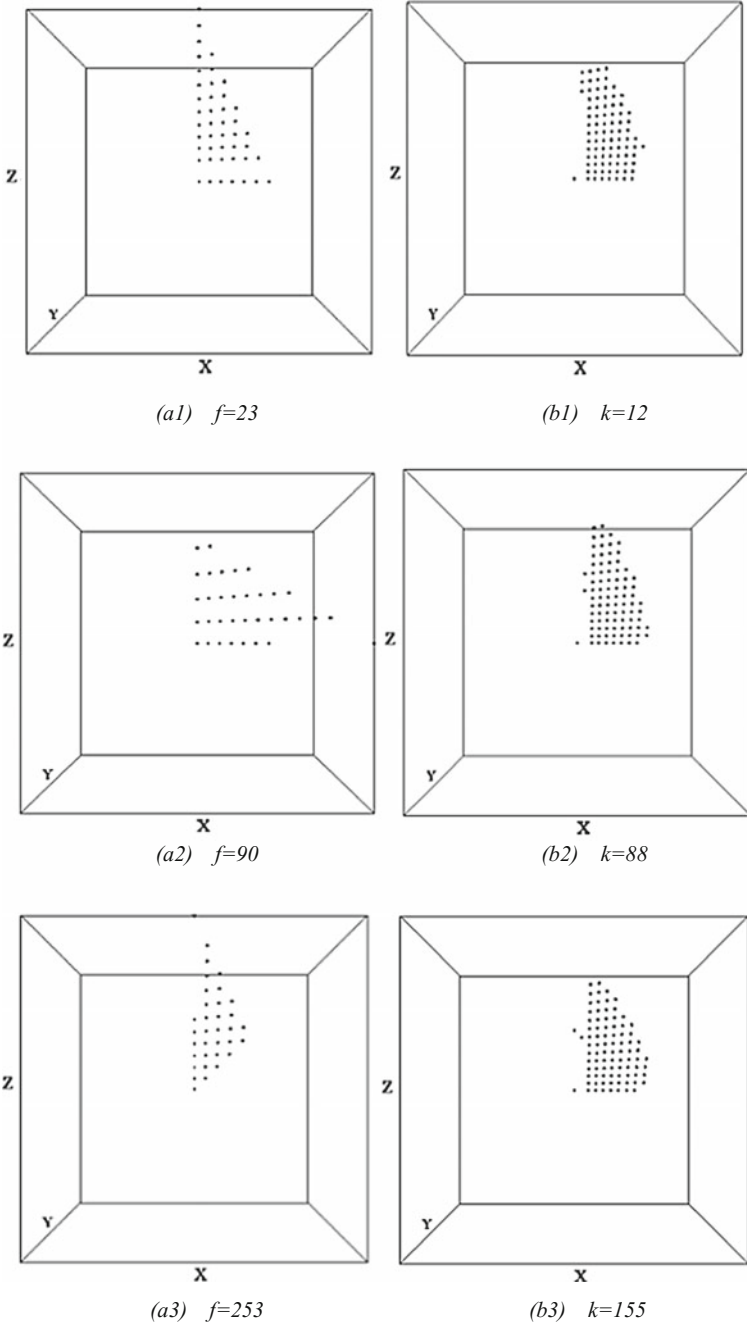


Fig. 3 Two sets of six 3D maps based on non-unified model in different conditions; **a1–a3** for the file CA; **b1–b3** for the file RC4

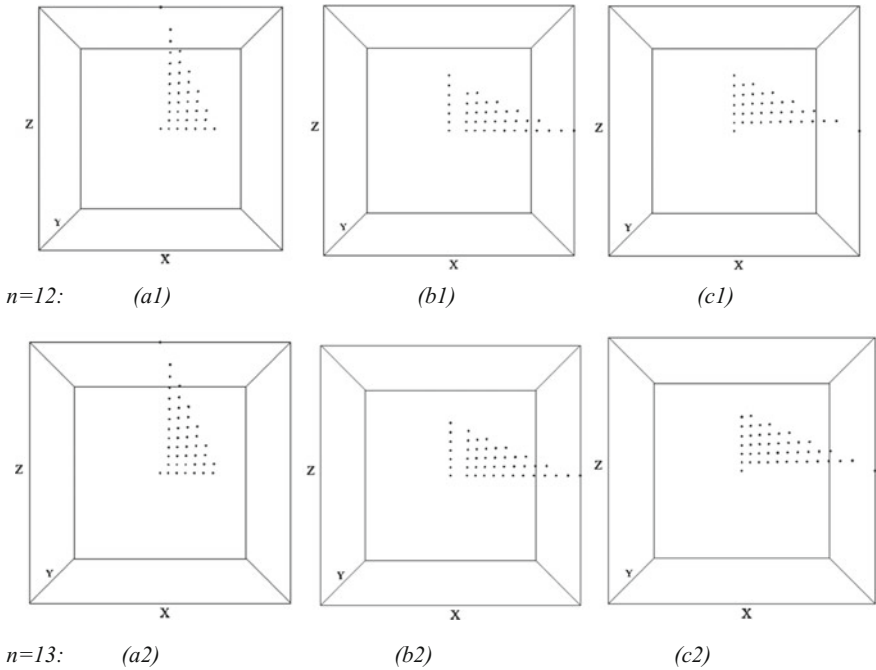


Fig. 4 Three sets of nine 3D maps under different conditions; **a1–a2** for the logic function $f = 15$ and non-unified model; **b1–b2** for the logic function $f = 100$ and non-unified model; **c1–c2** for the logic function $f = 170$ and non-unified model

bits of every segment and different given keys. In this group, keys: 12, 88, and 155 are selected as examples to compare each other. From a distribution viewpoint, different logic function can be distinguished by their three-dimensional spatial characteristics from CA files, e.g., (a1–a3). Different from CA, for RC4 keystream, all spatial distributions are always in a plane, e.g., (b1–b3).

The second group of results shown in Fig. 3 presents two sets of six 3D maps constructed by non-unified model. It is interesting to observe that all maps (no matter CA data files or RC4 keystream data files) have planar distribution, e.g., (a1–a3) and (b1–b3).

The third group of results shown in Fig. 4 presents three sets of six 3D maps constructed by non-unified model from CA data files with different lengths of the initial sequence and given logic functions. Figure 4a1–a2 shows 3D maps for the No. 15 function, (b1–b2) shows 3D maps for the No. 100 function, and (c1–c2) shows 3D maps for the No. 170 function. The overall relationship of multiple-variable logic functions for spatial characteristics can be shown clearly. For example, under the non-unified model, no matter what logic functions are, all spatial distributions are always in a plane, e.g., (a1–a2), (b1–b2), and (c1–c2). Different lengths of initial

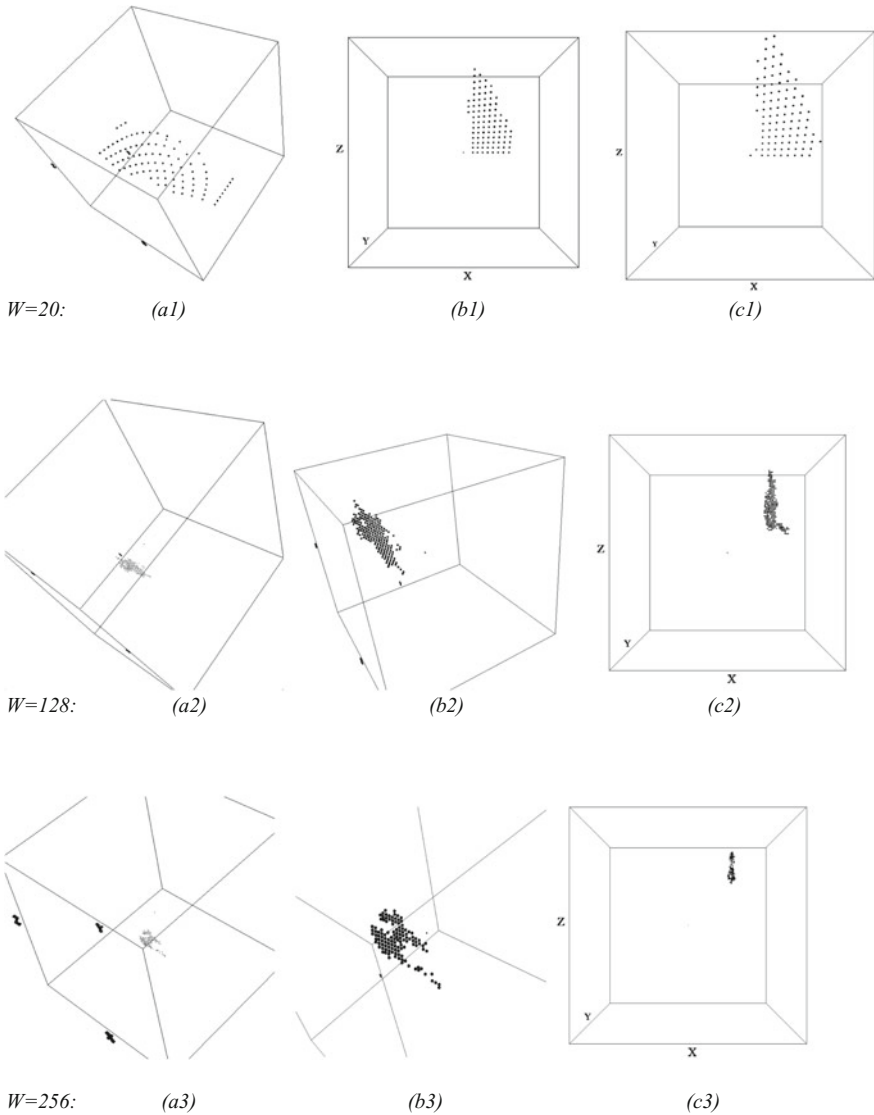


Fig. 5 Three sets of nine 3D maps under different conditions; **a1–a3** for the key=90 and unified model; **b1–b3** for the key=90 and non-unified model; **c1–c3** for the key=123 and non-unified model

sequence ($n = 12, 13$) have different spatial characteristics distribution with the same given logic function, e.g., (a1–a2), (b1–b2) and (c1–c2).

The fourth group of results shown in Fig. 5 presents three sets of nine 3D maps for the different conditions including segments strategies and keys. In this group, three types of segment strategies ($W = 20, 128, 256$) are proposed to compare.

Combinations of three set use the same key e.g., (a1–a3), (b1–b3), and (c1–c3) to observe them conveniently. The dispersity of points increased with reducing the bit length of each segment. Obviously, the spatial distribution of points with 256 bits of each segment is more concentrated than the distribution of points with 20 bits, as shown in (a1–a2), (b1–b2), and (c1–c2). 3D map shows some commonalities of the spatial distribution of different keys and different segment strategies. First, under this construction, different keys can be distinguished by their three-dimensional spatial characteristics in the model, e.g., (b1–c1), (b2–c2), and (b3–c3). Second, no matter what keys or segment strategies are, all spatial distributions are always in a plane. Third, the distribution features are varying from key to key and segment strategy to segment strategy.

5 Conclusions

Both the similarities and the differences may indicate those maps with comparable mechanism to express keystream with different given keys and in their high levels of relationships applying to the stream cipher mechanism. The spatial property of random sequence can be detected from the distribution of cluster point in the 3D maps discussed in details. Different spatial distributions are illustrated to show various distributions on each phase space for relevant logic function or keystream. For example, no matter what keys or segment strategies are, all spatial distributions are always in a plane. And all maps (no matter CA data files or RC4 keystream data files) are planar distribution under non-unified model. Spatial distribution properties like this provide useful information for further exploring the RC4 stream cipher. This construction could provide remarkable insights to spatial information on stream cipher construction via 3D maps. Further explorations are required on this scheme.

Acknowledgements Thanks to the school of software Yunnan University, to the key laboratory of Yunnan software engineering for excellent working environment, to the Yunnan Advanced Overseas Scholar Project (W8110305), to the Key R&D project of Yunnan Higher Education Bureau (K1059178), and to National Science Foundation of China (61362014) for the financial support to this project.

References

1. H. Brandon, A.J. Patricia, Information Warfare. Inf. Syst. Educ. J. **4**(49) (2006). <http://isedj.org/4/49/>. ISSN: 1545-679X
2. O.S. Suhaila, S.P. Mansoor, Performance analysis of Stream Cipher algorithms, in *The 3rd International Conference on Advanced Computer Theory and Engineering (ICATE)*, Chengdu, China (2010)
3. J.K. Fahime, V.M. Mohammad, R.N. Hamid, H. Payman, A new symmetric cryptographic algorithm to secure E-commerce transactions, in *The International Conference on Financial Theory and Engineering*, Dubai, United Arab Emirates (2010)

4. C.S. Lamba, Design and analysis of Stream Cipher for Network security, in *The 2nd International Conference on Communication Software and Networks*, Singapore (2010)
5. M.J.B. Robshaw, *Stream Cipher*. RSA Laboratories Technical Report TR-701. Retrieved from <http://citeseerx.ist.psu.edu/> (1995)
6. S. Bruce, *Applied cryptography* (Wiley, CRC Press, 1997)
7. Q. Li, J. Zheng, 2D spatial distributions for measures of random sequences using conjugate maps, in *The Proceedings of the 11th Australian Information Warfare and Security Conference*, Perth 1–9, 2010. <http://ro.ecu.edu.au/isw/34>
8. S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific Press)
9. L. Shiyong, T. Xinhua, *Nonlinear Study and Complexity Study* (Harbin Institute of Technology Press)
10. J. Zheng, C. Zheng, T.L. Kunii, A framework of Variant Logic Construction for Cellular Automata, in *Cellular Automata—Innovative Modelling for Science & Engineering* (InTech Press, 2011)
11. S. Alejandro, *Cellular Automata-Innovative Modelling for Science and Engineering* (InTech Press, 2011)
12. J. Zheng, C. Zheng, T.L. Kunii, Interactive maps on variant phase spaces—from measurements—micro ensembles to ensemble matrices on statistical mechanics of particle models, in *Emerging Applications of Cellular Automata* (InTech Press, 2013)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part VI

Applications—Quantum Simulations

The best way to understanding is a few good examples.

—Isaac Newton

The true logic of this world is in the calculus of probabilities.

—James Clerk Maxwell

A deep truth is a truth so deep that not only is it true but it's exact opposite is also true.

—Niels Bohr

In the direction of quantum information, several papers were published in the period of 2011–2013. For example, Variant simulation system using quaternion structures, *Journal of Modern Optics* 59(5):484–492, 2012, “Chapter Interactive Maps on Variant Phase Spaces”, *Emerging Applications of Cellular Automata*, <https://doi.org/10.5772/51635>, In Tech Press 2013. In the Afshar experiment, variant scheme has been cited, https://en.wikipedia.org/wiki/Afshar_experiment.

This part of quantum simulation is composed of two chapters (16 and 17).

Chapter “[Synchronous Property—Key Fact on Quantum Interferences](#)” describes synchronous property in quantum interferences simulation on double path experiment.

Chapter “[The \$n\$ th Root of NOT Operators of Quantum Computers](#)” proposes a typical operator on the n th root of NOT operators as an algebraic solution.

Synchronous Property—Key Fact on Quantum Interferences



Particle Simulation on Double Path Experiment

Jeffrey Zheng

Abstract Double-slit experiment plays a key role in Quantum Theory to distinct particle and wave interactions according to Feynman's claims. In this chapter, double path model and variant logic principle are applied to establish a simulation system for exhaustive testing targets. Using Einstein quanta interaction, different measure quaternion structures are investigated. Under Symmetry/Anti-symmetry and Synchronous/Asynchronous interaction conditions, eight groups of statistical results are generated as eight histograms to show their distributions. From this set of simulation results, it can be recognized that the synchronous condition is the key fact to generate quantum wave interference patterns and, in addition, the asynchronous condition is the key fact to make classic particle distributions. Sample results are illustrated and explanations are discussed.

Keywords Double path · Interaction · Probability · Statistics · Simulation

1 Introduction

Feynman explored quantum measurement puzzles deeply [1, 2] and emphasized: "The entire mystery of quantum mechanics is in the double-slit experiment." This experiment directly illustrates both classical and quantum interactive results. Under single and double slit conditions, dual visual distributions are shown in particle and wave statistical distributions linked to von Neumann's measure theory [3].

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), and Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)
Key Laboratory of Quantum Information of Yunnan, Yunnan University,
Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_16

From the 1970s, piloted by CHSH [4], Aspect used experiments to test Bell inequalities [5–7]. After 40 years of development, many accurate experiments [8–10] have been performed successfully worldwide using Laser, NMRI, large molecular, quantum coding, and quantum communication approaches [5–8, 11–26].

In this chapter, a double path model is established using the Mach–Zehnder interferometer. Different approaches of quantum measures: Einstein, CHSH, and Aspect are investigated by quaternion structures. Under multiple-variable logic functions and variant principle, logic functions can be transferred into variant logic expression as variant measures. Under such conditions, a variant simulation model is proposed. A given logic function f can be represented as two meta-logic functions f_+ and f_- to simulate single and double path conditions. N bits of input vectors are exhausted by 2^N states for measured data, recursive data are organized into eight histograms. Results are determined by symmetry/anti-symmetry properties evident in these histograms. Both results are obtained consistently from this model on synchronous/asynchronous conditions. Based on this set of simulation results, synchronous condition shows significant relationship linked to interference properties.

2 Double Path Model and Their Measures

2.1 Mach–Zehnder Interferometer Model

The Mach–Zehnder interferometer is the most popular device [6, 20] to support Young’s double-slit experiment.

In Fig. 1a, a double path interferometer is shown. An input signal X under control function f causes Laser LS to emit the output signal ρ under BP (Bi-polarized filter) operation output a pair of signals: ρ^+ and ρ^- . Both signals are processed by SW output ρ_L^+ and ρ_R^- , and then IM to generate output signals $IM(\rho_L^+, \rho_R^-)$. In Fig. 1b, a representation model has been described with the same signals being used.

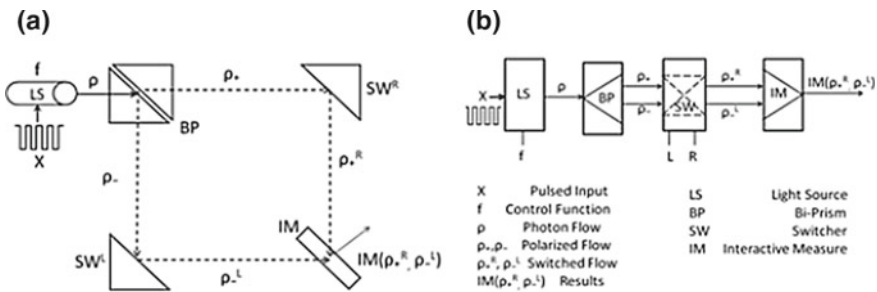


Fig. 1 Double path model **a** Mach–Zehnder double path model, **b** Description model

2.2 Emission and Absorption Measures of Quantum Interaction

Einstein established a model to describe atomic interaction [27–30] with radiation in 1916. For two-state systems, let a system have two energy states: the ground state E_1 and the excited state E_2 . Let N_1 and N_2 be the average numbers of atoms in the ground and excited states, respectively. The numbers of states are changed from an emission state E_2 to E_1 with a rate $\frac{dN_{21}}{dt}$, in the same time; the numbers of ground states are determined by absorbed energies from E_1 to E_2 with a rate $\frac{dN_{12}}{dt}$, respectively. Let N_{12} be the number of atoms from E_1 to E_2 and N_{21} be the numbers from E_2 to E_1 . In Einstein's model, a measurement quaternion is $\langle N_1, N_2, N_{12}, N_{21} \rangle$.

CHSH proposed spin measures testing Bell inequalities [4, 6]. They applied $\perp \rightarrow +$ and $\parallel \rightarrow -$ to establish a measurement quaternion

$$\langle N_{++}(a, b), N_{+-}(a, b), N_{-+}(a, b), N_{--}(a, b) \rangle.$$

Experimental testing of Bell inequalities was performed by Aspect [5] in 1982. Four parameters are measured: transmission rate N_t , reflection rate N_r , correspondent rate N_c , and the total number N_ω in ω time period. This set of measures is a quaternion $\langle N_t, N_r, N_c, N_\omega \rangle$. Among these, N_c is a new data type not in Einstein and CHSH methods, this parameter could be an extension of synchronous/asynchronous time-measurement.

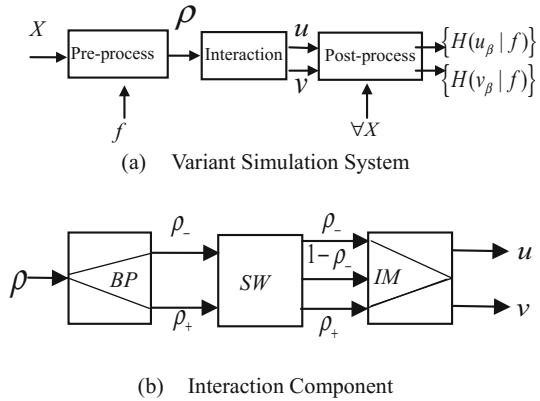
3 Simulation Systems

3.1 Simulation Model

Using variant principle described in the next subsections, a N bit 0-1 vector X and a given logic function f , all N bit vectors are exhausted, variant measures generate two groups of histograms. This variant simulation system is composed of three components: Pre-process, Interaction, and Post-process, respectively, and shown in Fig. 2.

In Fig. 2a, three components of the variant simulation model are presented. At the pre-process, a N bit 0-1 vector X and a function f feed in to output a signal ρ . After interactive component process, two groups of signals are the output: u for symmetry group and v for anti-symmetry group. In the post-process, all N bit vectors are processed by pre-processing and interactive components until all of the 2^N data set has been processed to transform symmetry and anti-symmetry signals into eight histograms: four for symmetry distributions and another four for anti-symmetry distributions.

Fig. 2 Variant simulation system; **a** Variant simulation system; **b** interactive component



In Fig. 2b, only the interaction component is selected, input signal ρ processed by BP to generate two signals $\{\rho_-, \rho_+\}$. SW output triple signals $\{\rho_-, 1 - \rho_-, \rho_+\}$ though IM to generate two groups of signals u and v .

3.2 Variant Principle

The variant principle is based on n -variable logic functions [31–33]. For any n -variables, $x = x_{n-1} \dots x_i \dots x_0$, $0 \leq i < n, x_i \in \{0, 1\} = B_2$. Let a position j be the selected bit $0 \leq j < n$, $x_j \in B_2$ be the selected variable. Let output variable y and n -variable function f , $y = f(x)$, $y \in B_2, x \in B_2^n$. For all states of x , a set $S(n)$ composed of the 2^n states can be divided into two sets: $S_0^j(n)$ and $S_1^j(n)$.

$$\begin{cases} S_0^j(n) = \{x | x_j = 0, \forall x \in B_2^n\} \\ S_1^j(n) = \{x | x_j = 1, \forall x \in B_2^n\} \\ S(n) = \{S_0^j(n), S_1^j(n)\} \end{cases}$$

For a given logic function f , there are input and output pair relationships to define four meta-logic functions $\{f_\perp, f_+, f_-, f_T\}$:

$$\begin{cases} f_\perp(x) = \{f(x) | x \in S_0^j(n), y = 0\} \\ f_+(x) = \{f(x) | x \in S_0^j(n), y = 1\} \\ f_-(x) = \{f(x) | x \in S_1^j(n), y = 0\} \\ f_T(x) = \{f(x) | x \in S_1^j(n), y = 1\} \end{cases}$$

Two logic canonical expressions: AND-OR form is selected by $\{f_+(x), f_T(x)\}$ as $y=1$ items, and OR-AND form is selected from $\{f_-(x), f_\perp(x)\}$ as $y=0$ items. Considering $\{f_T(x), f_\perp(x)\}$, $x_j = y$ items, they are invariant themselves.

To select $\{f_+(x), f_-(x)\}$; $x_j \neq y$ forming variant logic expression. Let $f(x) = \langle f_+|x|f_- \rangle$ be a variant logic expression. Any logic function can be expressed as a variant logic form. In $\langle f_+|x|f_- \rangle$ structure, f_+ selected 1 item in $S_0^j(n)$ as same as the AND-OR standard expression, and f_- selecting relevant parts as same as the OR-AND expression 0 items in $S_1^j(n)$. For a convenient understanding of variant representation, two-variable logic structures are illustrated for its 16 functions shown in Table 1.

For example, checking two functions $f = 3$ and $f = 12$:

$$\begin{aligned} \{f = 3 := \langle 0 | 3 \rangle, f_+ = 11 := \langle 0 | \phi \rangle, f_- = 2 := \langle \phi | 3 \rangle\} \\ \{f = 12 := \langle 2 | 1 \rangle, f_+ = 14 := \langle 2 | \phi \rangle, f_- = 8 := \langle \phi | 1 \rangle\} \end{aligned}$$

3.3 Variant Measures

Let Δ be variant measure function [23, 33].

$$\Delta = \langle \Delta_\perp, \Delta_+, \Delta_-, \Delta_T \rangle$$

$$\begin{aligned} \Delta f(x) &= \langle \Delta_\perp f(x), \Delta_+ f(x), \Delta_- f(x), \Delta_T f(x) \rangle \\ &= \langle \Delta f_\perp(x), \Delta f_+(x), \Delta f_-(x), \Delta f_T(x) \rangle \end{aligned}$$

$$\Delta f_\alpha(x) = \begin{cases} 1, & \text{if } f(x) = f_\alpha(x), \alpha \in \{\perp, +, -, T\} \\ 0, & \text{others} \end{cases}$$

For any given n-variable state there is one position in $\Delta f(x)$ to be 1 and other three positions are 0.

For any N bit 0-1 vector X ; $X = X_{N-1} \dots X_J \dots X_0$, $0 \leq J < N$, $X_J \in \beta_2$, $X \in \beta_2^N$ under n-variable function f , n bit 0-1 output vector Y , $Y = f(X) = \langle f_+|X|f_- \rangle$, $Y = Y_{N-1} \dots Y_J \dots Y_0$, $0 \leq J < N$, $Y_J \in \beta_2$, $Y \in \beta_2^N$.

For the J th position, be $x^J = [\dots X_J \dots] \in \beta_2^n$ to form $Y_J = f(x^J) = \langle f_+|x^J|f_- \rangle$, let N bit positions be cyclic linked. Variant measures of $f(X)$ can be decomposed

$$\Delta \langle X : Y \rangle = \Delta f(X) = \sum_{J=0}^{N-1} \Delta f(x^J) = \langle N_\perp, N_+, N_-, N_T \rangle$$

as a quaternion $\langle N_\perp, N_+, N_-, N_T \rangle$.

For example, $N = 10$, given f , $Y = f(X)$.

Table 1 Two variable logic functions and variable logic representation ($n = 2, j = 0$)

f No.	$f \in$ $S(2)$	3 11	2 10	1 01	0 00	$f_+ \in$ $S_0^0(2)$	3^0 11^0	2^1 10^1	1^0 01^0	0^1 00^1	$f_- \in$ $S_1^0(2)$
0	$\{\emptyset\}$	0	0	0	0	$\langle \emptyset $	1	0	1	0	$ 3,1\rangle$
1	$\{0\}$	0	0	0	1	$\langle 0 $	1	0	1	1	$ 3,1\rangle$
2	$\{1\}$	0	0	1	0	$\langle \emptyset $	1	0	0	0	$ 3\rangle$
3	$\{1,0\}$	0	0	1	1	$\langle 0 $	1	0	0	1	$ 3\rangle$
4	$\{2\}$	0	1	0	0	$\langle 2 $	1	1	1	0	$ 3,1\rangle$
5	$\{2,0\}$	0	1	0	1	$\langle 2,0 $	1	1	1	1	$ 3,1\rangle$
6	$\{2,1\}$	0	1	1	0	$\langle 2 $	1	1	0	0	$ 3\rangle$
7	$\{2,1,0\}$	0	1	1	1	$\langle 2,0 $	1	1	0	1	$ 3\rangle$
8	$\{3\}$	1	0	0	0	$\langle \emptyset $	0	0	1	0	$ 1\rangle$
9	$\{3,0\}$	1	0	0	1	$\langle 0 $	0	0	1	1	$ 1\rangle$
10	$\{3,1\}$	1	0	1	0	$\langle \emptyset $	0	0	0	0	$ \emptyset\rangle$
11	$\{3,1,0\}$	1	0	1	1	$\langle 0 $	0	0	0	1	$ \emptyset\rangle$
12	$\{3,2\}$	1	1	0	0	$\langle 2 $	0	1	1	0	$ 1\rangle$
13	$\{3,2,0\}$	1	1	0	1	$\langle 2,0 $	0	1	1	1	$ 1\rangle$
14	$\{3,2,1\}$	1	1	1	0	$\langle 2 $	0	1	0	0	$ \emptyset\rangle$
15	$\{3,2,1,0\}$	1	1	1	1	$\langle 2,0 $	0	1	0	1	$ \emptyset\rangle$

$$\begin{array}{rcccccccccc}
X & = & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
Y & = & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\Delta(X : Y) & = & + & - & T & \perp & + & - & T & - & + & \perp
\end{array}$$

$$\Delta f(X) = \langle N_{\perp}, N_{+}, N_{-}, N_T \rangle = \langle 2, 3, 3, 2 \rangle, N = 10$$

Input and output pairs are 0-1 variables on the four combinations. For any given function f , the quantitative relationship of $\{\perp, +, -, T\}$ is determined directly from input/output sequences.

3.4 Measurement Equations

Using variant quaternion, signals are calculated by following equations. For any N bit 0-1 vector X , function f , under Δ measurement: $\Delta f(x) = \langle N_{\perp}, N_{+}, N_{-}, N_T \rangle$, $N = N_{\perp} + N_{+} + N_{-} + N_T$ Signal ρ is defined by

$$\rho = \frac{\Delta f(x)}{N} = \langle \rho_{\perp}, \rho_{+}, \rho_{-}, \rho_T \rangle$$

$$\rho_{\alpha} = \frac{N_{\alpha}}{N}, 0 \leq \rho_{\alpha} \leq 1, \alpha \in \{\perp, +, -, T\}$$

Using $\{\rho_{+}, \rho_{-}\}$, a pair of signals $\{u, v\}$ are formulated:

$$\begin{cases}
u = \langle u_0, u_{+}, u_{-}, u_1 \rangle = \{u_{\beta}\} \\
v = \langle v_0, v_{+}, v_{-}, v_1 \rangle = \{v_{\beta}\}
\end{cases}$$

$$\beta \in \{0, +, -, 1\}$$

$$\begin{cases}
u_0 = \rho_{-} \oplus \rho_{+} \\
v_0 = (1 - \rho_{-})/2 \oplus (1 + \rho_{+})/2 \\
u_{+} = \rho_{+} \\
v_{+} = (1 + \rho_{+})/2 \\
u_{-} = \rho_{-} \\
v_{-} = (1 - \rho_{-})/2 \\
u_1 = \rho_{-} + \rho_{+} \\
v_1 = (1 - \rho_{-} + \rho_{+})/2
\end{cases}$$

where $0 \leq u_\beta, v_\beta \leq 1, \beta \in \{0, +, -, 1\}, \oplus$: Asynchronous addition, $+$: Synchronous addition.

Using $\{u, v\}$ signals, each $u_\beta (v_\beta)$ determines a fixed position in the relevant histogram to make vector X on a position. After complete 2^N data sequences, eight symmetry/anti-symmetry histograms of $\{H(u_\beta|f)\}(\{H(v_\beta|f)\})\beta \in \{0, +, -, 1\}$ are generated.

4 Simulation Results

The simulation provides a series of output results. In this section, two cases are selected: $N = \{12, 13\}, n = 2, j = 0, \{f = 3, f_+ = 11, f_- = 2\}$, and $\{f = 12, f_+ = 14, f_- = 8\}$. Corresponding to double path, right path, left path, symmetric and nonsymmetric conditions, respectively. For the convenience of comparison, sample cases are shown in Fig. 3a–c. In Fig. 3a, representation patterns are illustrated. Figure 3b represents $f = 3$ conditions and Fig. 3c represents $f = 12$ conditions, respectively. Eight histograms of $H(u_+|f) = H(u_-|f)$ are shown with results represented by symmetric meta-functions in four groups.

5 Analysis of Results

5.1 Visual Distributions

In $H(u_+|f) = H(u_-|f)$ conditions, $\{H(u_1|f), H(v_1|f)\}$ have significant interference patterns different from other conditions. Output results are balanced.

5.2 Particle Statistical Distributions

For all symmetric or nonsymmetric cases under \oplus asynchronous addition operations, relevant values meet $0 \leq u_0, v_0, u_-, v_-, u_+, v_+ \leq 1$.

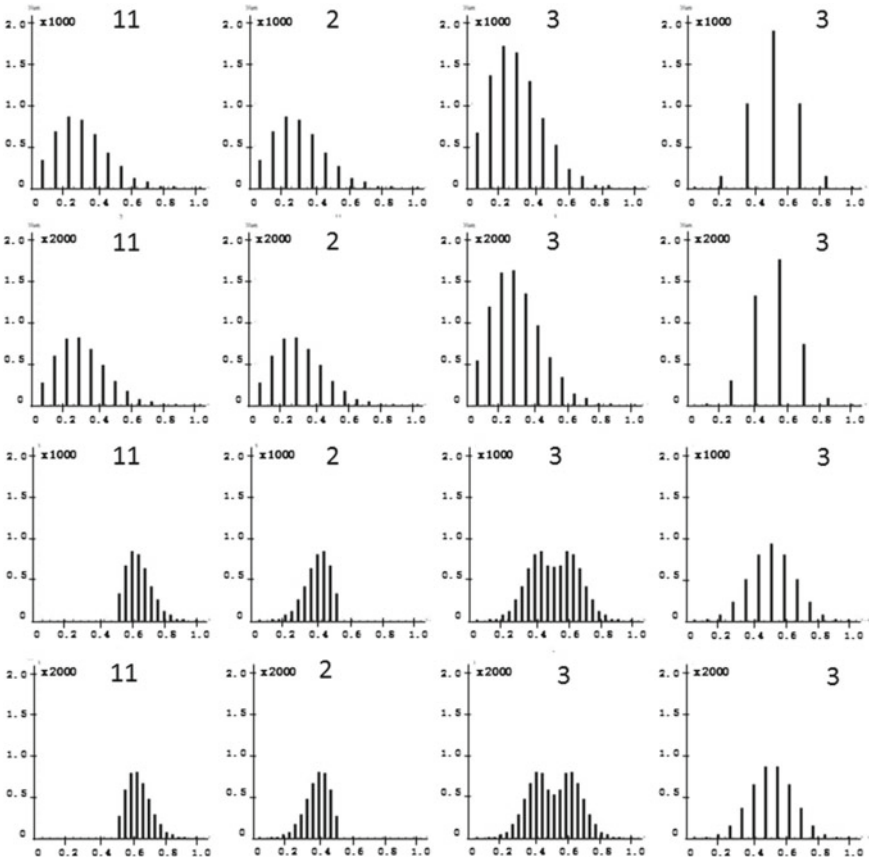
Checking $\{H(u_0|f), H(v_0|f)\}$ series, $\{H(u_+|f), H(u_-|f)\}$ and $\{H(v_+|f), H(v_-|f)\}$ satisfy the following equation:

$$\begin{cases} H(u_0|f) = H(u_-|f) + H(u_+|f) \\ H(v_0|f) = H(v_-|f) + H(v_+|f) \end{cases}$$

The equation is true even N and n in different values.

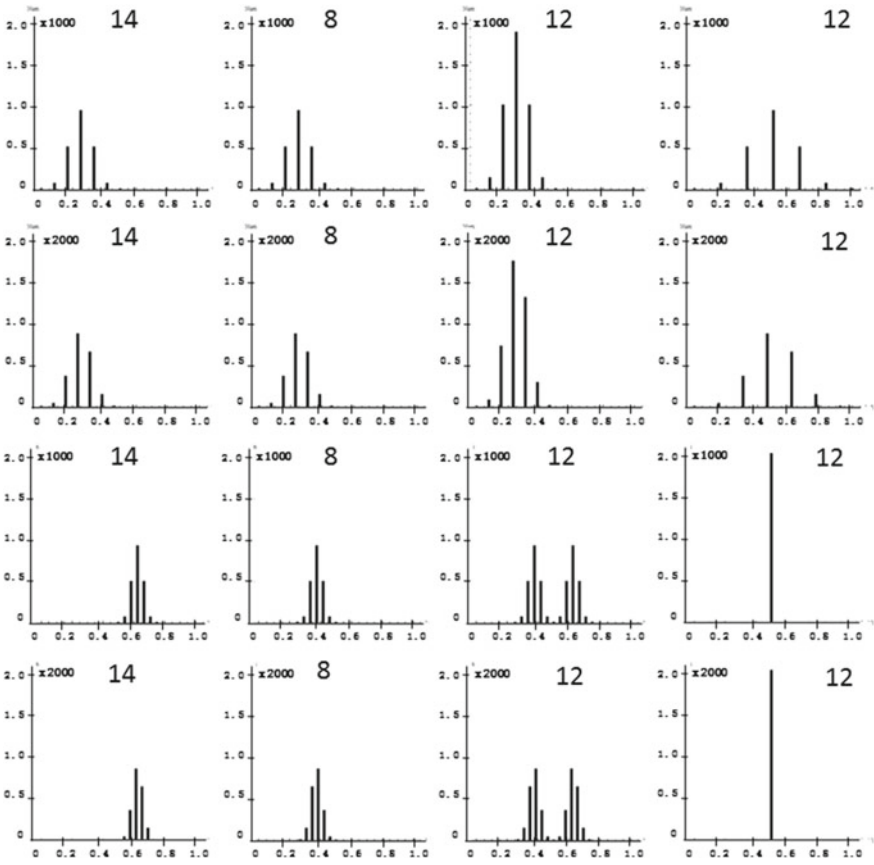
N	Left Path	Right Path	Double-Particle	Double-Wave	Conditions
12	$H(u_+ f)$	$H(u_- f)$	$H(u_0 f)$	$H(u_1 f)$	Symmetric Meta Distributions $H(u_+ f) = H(u_- f)$
13	$H(u_+ f)$	$H(u_- f)$	$H(u_0 f)$	$H(u_1 f)$	
12	$H(v_+ f)$	$H(v_- f)$	$H(v_0 f)$	$H(v_1 f)$	Anti-symmetric Meta Distributions $H(v_+ f) = H(1 - v_- f)$
13	$H(v_+ f)$	$H(v_- f)$	$H(v_0 f)$	$H(v_1 f)$	

(a) Statistical Histogram Patterns



(b) $N=\{12, 13\}$, $f = 3$, Histograms of Symmetric Meta Distributions

Fig. 3 Results of symmetric meta distributions



(c) $N=\{12, 13\}$, $f = 12$, Histograms of Symmetric Meta Distributions

Fig. 3 (continued)

5.3 Wave Interference Patterns

Different interference properties are observed clearly in $H(u_+|f) = H(u_-|f)$ and $H(v_+|f) = H(1 - v_-|f)$ conditions. Under synchronous addition operations, relevant values meet $0 \leq u_1, v_1, u_-, v_-, u_+, v_+ \leq 1$.

Checking $\{H(u_1|f), H(v_1|f)\}$ distributions especially in Fig. 3b-c $\{u_1, v_1\}$ cases extremely strong interferences appeared and compared with $\{H(u_+|f), H(u_-|f)\}$ and $\{H(v_+|f), H(v_-|f)\}$, there are significant differences. Spectra in different cases illustrate wave interference properties. From listed histogram distributions, they are all satisfied

$$\begin{cases} H(u_1|f) \neq H(u_-|f) + H(u_+|f) = H(u_0|f) \\ H(v_1|f) \neq H(v_-|f) + H(v_+|f) = H(v_0|f) \end{cases}$$

Single and double peaks are shown in interference patterns as classical double-slit distributions.

5.4 Quaternion Measures

It is interesting to see the relationship between the variant quaternion and other measures.

In the variant quaternion, $\Delta f(x) = \langle N_{\perp}, N_{+}, N_{-}, N_T \rangle$, $N = N_{\perp} + N_{+} + N_{-} + N_T$.

In Einstein's two-state system of interaction $\langle N_1, N_2, N_{12}, N_{21} \rangle$ allows the following equations to be established:

$$\begin{cases} N_1 = N_{\perp} + N_{+} \\ N_2 = N_{-} + N_T \\ N_{12} = N_{+} \\ N_{21} = N_{-} \\ N = N_1 + N_2 \end{cases}$$

From the equations, the measured pair $\{N_{21}, N_{12}\}$ has a 1-1 correspondence to $\{N_{-}, N_{+}\}$.

Selecting $+ \rightarrow 1, - \rightarrow 0$, CHSHs $N_{\pm, \mp}(a, b)$ measures meet

$$\begin{cases} N_{+,+}(a, b) \rightarrow N_T \\ N_{+,-}(a, b) \rightarrow N_{-} \\ N_{-,+}(a, b) \rightarrow N_{+} \\ N_{-,-}(a, b) \rightarrow N_{\perp} \end{cases}$$

$$(N_{++}, N_{+-}, N_{-+}, N_{--}) \rightarrow (N_T, N_{-}, N_{+}, N_{\perp}),$$

Let $N = N_{++} + N_{+-} + N_{-+} + N_{--}$, CHSH quaternion is a permutation of the variant quaternion.

Aspect's quaternion $(N_t, N_r, N_c, N_{\omega})$ have the following corresponding:

$$\begin{cases} N_t \rightarrow N_{-} \\ N_r \rightarrow N_{+} \\ N_{\omega} \rightarrow N \end{cases}$$

There is no parameter in the variant quaternion for the parameter N_c . It indicates joined action numbers to distinguish single and double paths, corresponding to

$\{u_0, v_0\}$ and $\{u_1, v_1\}$ times. In an actual experiment, this parameter is significant. In a simulated system, the parameter is a control coefficient that separates two types of measured paths $\{u_0, v_0\}$ and $\{u_1, v_1\}$ in the integration of comparisons on real experiments.

6 Conclusions

Analyzing N bit 0-1 vector and its exhaustive sequences for variant measurement, this system simulates double path interference properties through different accurate distributions. Using this model, two groups of parameters $\{u_\beta\}$ and $\{v_\beta\}$ describe the left path, right path, double paths for particle, and double path for wave with distinguished symmetry and anti-symmetry properties.

Only synchronous conditions, double path system provides wave-like interference patterns different from classical ones.

Compared with the variant quaternion and other quaternion structures, it is helpful to understand possible properties of usages and limitations for variant simulation systems.

The complexity of n -variable function space has a size of 2^{2^n} . Whole simulation complexity is determined by $O(2^{2^n} \times 2^N)$ as ultra exponent productions. How to overcome the limitations imposed by such complexity and how best to compare and contrast such simulations with real-world experimentation will be key issues in future work.

Acknowledgements Thanks to Mr. Colin W Campbell for making English edition, Mr. Jie Wan for generating the simulation data, and Mr. Qingping Li for making the statistical histograms.

References

1. R. Feynman, R. Leighton, M. Sands, *The Feynman Lectures on Physics*, vol. 3 (Addison-Wesley, Reading, MA, 1965, 1989)
2. R. Feynman, *The Character of Physical Law* (MIT Press, 1965)
3. J. von Neumann, *Mathematical Foundations of Quantum Mechanics* (Princeton University Press, 1932, 1996). ISBN 0-691-02893-1
4. J. Clauser, N. Home, A. Shimony, R. Holt, PRL **23**, 880–884 (1969)
5. A. Aspect, P. Grangier, G. Roger, Phys. Rev. Lett. **49**, 91–94 (1982)
6. A. Aspect, *Quantum [Un]speakables—From Bell to Quantum Information*, ed. by R.A. Bertlmann, A. Zeilinger (Springer, Berlin, 2002)
7. A. Aspect, Nature **446**, 866–867 (2007)
8. F. Lindner, M.G. Schätzle, H. Walther, A. Baltuska, E. Goulielmakis, F. Krausz, D.B. Milosevic, D. Bauer, W. Becker, G.G. Paulus, Phys. Rev. Lett. **95**, 040401 (2005)
9. H.D. Zeh, *Foundation of Physics* **1**, 69–76 (1970)
10. A. Zeilinger, G. Weihs, T. Jennewein, M. Aspelmeyer, Nature **433**, 230–238 (2005)
11. S. Afshar et al., *Found. Phys.* **37**, 295–305 (2007). <http://www.springerlink.com/content/q110r82074w03277/fulltext.pdf>

12. M. Arndt, O. Nairz, J. Vos-Andreae, C. Keller, G. van der Zouw, A. Zeilinger, *Nature* **401**, 680–682 (1999)
13. S.M. Barnett, *Quantum Information* (Oxford University Press, Oxford, 2009). ISBN 978-0-19-852762-6
14. J.D. Barrow, P.C.W. Davies, J.E. Charles, L. Harper, *Science and Ultimate Reality: Quantum Theory, Cosmology and Complexity* (Cambridge University Press, Cambridge, 2004)
15. M. Fox, *Quantum Optics* (Oxford University Press, Oxford, 2006). ISBN 0-19-856672-7
16. J.C. Garrison, R. Chiao, *Quantum Optics* (Oxford University Press, Oxford, 2008). ISBN 978-0-850886-1
17. P. Grangier, G. Roger, A. Aspect, *Europhys. Lett.* **1**, 173–179 (1986)
18. S. Hawking, L. Mlodinow, *The Grand Design* (Bantam Books, 2010)
19. R. Healey, G. Hellman (eds.), *Quantum Measurement: Beyond Paradox* (University of Minnesota Press, 1998). ISBN 0-8166-3065-8
20. M. Horne, A. Shimony, A. Zeilinger, *Nature* **347**, 429–430 (1990)
21. V. Jacques et al., *Science* **315**, 966 (2007). <http://www.arxiv.org/abs/quant-ph/0610241>. 13
22. M. Jammer, *The Philosophy of Quantum Mechanics* (Wiley-Interscience Publication, 1974). ISBN 0-471-43958-4
23. Q. Li, J. Zheng, in *11th Australian Information Warfare Conference* (2010). <http://ro.ecu.edu.au/isw/34>
24. P. Mittelstaedt, A. Prieur, R. Schieder, *Found. Phys.* **17**(9), 891–903 (1987). <https://doi.org/10.1007/bf00734319>
25. R. Penrose, *The Road to Reality* (Vintage Books, London, 2004)
26. W.P. Schleich, H. Walther (eds.), *Elements of Quantum Information* (Wiley-VCH Verlag GmbH & Co KGaA Weinheim, 2007). ISBN 978-3-527-40725-5
27. N. Bohr, *Discussion with Einstein on Epistemological Problems in Atomic Physics* (Evanston, 1949), pp. 200–241
28. L. de Broglie, *Nature* **112**, 540 (1923)
29. A. Einstein, *Ann. Phys.* 891–921 (1905)
30. A. Einstein, *Mit. Phys. Ges. Zurich* **16**, 47 (1916)
31. J. Zheng, C. Zheng, *Front. Electr. Electron. Eng. China* **5**, 163 (2010). <http://www.springerlink.com/content/91474403127n446u/> (Higher Education Press and Springer)
32. J. Zheng, C. Zheng, T. Kunii, *Cellular Automata—Innovative Modelling for Science and Engineering*, ed. by A. Salcido (InTech Press, 2011). <http://www.intechopen.com/articles/show/title/a-framework-of-variant-logic-construction-for-cellular-automata>
33. J. Zheng, C. Zheng, Variant measures and visualized statistical distributions. *Acta Photonica Sin.*, to appear 2011

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



The n th Root of NOT Operators of Quantum Computers



Jeffrey Zheng

Abstract This chapter proposes a novel approach to resolve the n th root of NOT problem for quantum computers using $(-1, 0, 1)$ permutation matrices. Only logic NOT and exchange operations are required. This result provides a complete solution to design and implement the n th root of NOT operators of quantum computers.

Keywords Quantum simulator · Quantum computation · Square root of NOT n -th root of NOT · Permutation matrix · Quantum logic gate

1 Introduction

Feynman [1] first proposed ‘universal quantum simulator’ towards a true quantum computer. Since then, research and development activities of quantum computation and quantum computers have become the new frontal of next-generation computers for two decades [2, 3]. Classical quantum mechanics use complex number vectors in Hilbert space to represent quantum states [4]. Any complex number is composed of two parts: a real part and an imaginary part. The imaginary number $i = \sqrt{-1}$ plays the essential role in the quantum mechanics construction. However, the mystery of the imaginary number causes severe difficulties for its manipulation, imagination and understanding [4–6]. Considering that modern computers are constructed by Boolean logic principles, how traditional logic structure is used to implement $\sqrt{-1}$ has been puzzling and deeply entangled in quantum computing for at least two decades [7–10]. Nothing in the published literature has described a way to implement this untamed operator using traditional logic operations [2, 11, 12].

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_17

279

1.1 The Square Root of NOT Problem

Following traditional logic, negation corresponds to logic NOT (\neg). Initiated by Feynman [1] and further developed by Deutsch [9, 13], this problem has been represented as $\sqrt{\neg}$ ‘the Square Root of NOT’ as one of the most difficult issues in quantum computation especially in general quantum gates. They suggested resolving $\neg = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ equation using logic operations for the solution. Maglicki and Wang [11] provided an example of how to resolve the problem this way.

Let \neg operation reverse two quantum spin states $|0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$,

$$\neg|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |1\rangle$$

$$\neg|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |0\rangle$$

To apply unitary rotational matrices, $\sqrt{\neg}$ operator can be expressed as

$$\sqrt{\neg} = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{i\pi/4} & e^{-i\pi/4} \\ e^{-i\pi/4} & e^{i\pi/4} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$

In the equations, both $e^{i\pi}$ and i symbols are involved. From a representative viewpoint, equations are useless because the symbols i and $\sqrt{\neg}$ are both logic equivalent. The equations are in circular definitions.

To explore how to use traditional logic implementing $\sqrt{\neg}$, it is necessary to analyse what has been established at the foundation levels of modern complex number construction.

1.2 Complex Number in History

The origin and development of complex number has a long and mysterious history [14–16]. In the nineteenth century, Gauss and Euler [15] made their foundation contributions to formally identifying imaginary parts as the most essential components to resolve solutions from n th algebraic equations. After their work, the imaginary number has been gradually accepted by mainstream mathematicians to be one of the most important parts of mathematics [15]. Hamilton established consistent operations on complex number in 1837 [17]. He constructed a complex number $a + bi$ as an ordered number pair (a, b) .

For example, let $a + bi$ and $c + di$ be two complex numbers. Four essential operations: $\{\pm, \bullet, /\}$ can be expressed as

$$\begin{aligned} (a, b) \pm (c, d) &= (a \pm c, b \pm d) \\ (a, b) \bullet (c, d) &= (ac - bd, ad + bc) \\ \frac{(a, b)}{(c, d)} &= \left(\frac{ac + bd}{c^2 + d^2}, \frac{bc - ad}{c^2 + d^2} \right) \end{aligned}$$

Using ordered pair representation, complex number operations are firmly established on real number operations. No further mysterious characteristics of imaginary numbers remain in the equations because all operations are well defined in real number construction.

2 Solution of the Square Root of NOT Problem

If we apply an imaginary number to an ordered pair, we have

$$i : (a, b) \rightarrow (-b, a)$$

When we do not restrict $\sqrt{-1}$ solution in $\{0, 1\}$ field but extend the field to $\{-1, 0, 1\}$. A permutation matrix can be constructed.

Let

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, I_2^+ = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, I_2^- = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, Z_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, Z_2^\perp = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

$$\begin{aligned} Z_2 : (a, b) &\rightarrow (-b, a) \\ (-b, a) &= (a, b) \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{aligned}$$

Because Z_2 provides the same result as the imaginary number when applied to the pair, it is necessary for us to explore Z_2 features in details.

Two eigenvalues of Z_2 can be determined from its determinant.

$$\begin{aligned} |\lambda I_2 - Z_2| &= \begin{vmatrix} \lambda - 1 & -1 \\ 1 & \lambda \end{vmatrix} = 0 \\ \lambda^2 + 1 &= 0, \quad \lambda^2 = -1, \quad \lambda = \pm\sqrt{-1} \end{aligned}$$

This corresponds to either $\begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}$ or $\begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix}$ as the solution. There are two unitary matrices U_+, U_- and two Hermite conjugate matrices U_+^*, U_-^* undertaken similarity transformation on Z_2 to produce the two diagonal matrices:

$$iI_2^\pm = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} = U_+ \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} U_+^*;$$

$$iI_2^\mp = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = U_- \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} U_-^*$$

Although three matrices belong to one matrix group under similarity transformation, five matrices can be distinguished without any direct equality.

$$iI_2 \neq iI_2^\pm \neq Z_2 \neq iI_2^\mp \neq -iI_2$$

To apply the five matrices twice separately, they all equal to $-I_2$.

$$(\pm iI_2)^2 = \begin{pmatrix} \pm i & 0 \\ 0 & \pm i \end{pmatrix} \begin{pmatrix} \pm i & 0 \\ 0 & \pm i \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = -I_2$$

$$(iI_2^\pm)^2 = (iI_2^\mp)^2 = \begin{pmatrix} \pm i & 0 \\ 0 & \mp i \end{pmatrix} \begin{pmatrix} \pm i & 0 \\ 0 & \mp i \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = -I_2$$

and

$$Z_2^2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = -I_2$$

Therefore, the Z_2 matrix is an equivalent form of the imaginary number under the transformation.

For any ordered pair (a, b) ,

$$(Z_2)^2 : (a, b) \rightarrow (-a, -b)$$

$$(Z_2)^2 : (a, b) \xrightarrow{Z_2} (-b, a) \xrightarrow{Z_2} (-a, -b)$$

$$(Z_2)^2 = -I_2$$

$$Z_2 = \sqrt{-I_2}$$

So, $\sqrt{-}$ operation can be constructed originally from one-one correspondences from the Z_2 matrix.

Let $\langle x|$ be a quantum state, $\neg\langle x| = \langle \bar{x}|$. For a non-zero element of Z_2 , two values $\{-1, 1\}$ of the elements map $\begin{cases} -1 : \langle x| \xrightarrow{\neg} \langle \bar{x}| \\ 1 : \langle x| \rightarrow \langle x| \end{cases}$ then a $\sqrt{-}$ operator is generated from a Z_2 operator.

For an ordered state pair $(\langle x|, \langle y|)$,

$$(\langle x|, \langle y|) \xrightarrow{\sqrt[n]{-}} (\langle \bar{y}|, \langle x|) \xrightarrow{\sqrt[n]{-}} (\langle \bar{x}|, \langle \bar{y}|) = -(\langle x|, \langle y|)$$

Therefore, Z_2 is a homologous form of the $\sqrt[n]{-}$ operator.

Under this construction, the square root of NOT problem in quantum computation is solved entirely. Only two elementary operations are involved in the transformation: logic NOT operation and pair-state exchange, respectively. They can be implemented readily using traditional logic constructions.

3 General Solution of the n th Root of NOT Operation

In this part, a general solution of $\sqrt[n]{-}$ ‘the n th root of NOT’ for quantum computers is explored.

Let J_n denote a conjugate permutation matrix which contains n columns and n rows and each row (column) has one non-zero element.

$$J_n = (J_{i,j}), \quad 1 = \sum_{i=1}^n |J_{i,j}| = \sum_{i=1}^n |J_{i,j}|, \quad J_{i,j} \in \{-1, 0, 1\}, \quad i, j \in [1, n]$$

Let I_n be a unit matrix, $I_{i,j} = 1, i = j; I_{i,j} = 0, i \neq j, i, j \in [1, n]$.

For example, matrices $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

are J_n matrices.

Let P_n be a (0, 1)-permutation matrix in which each column (row) contains only one element, and $PS(n)$ denote a permutation space containing all P_n matrices.

Let $JS(n)$ denote a conjugate permutation space.

Lemma For a given n , $PS(n)$ contains a total number of $n!$ distinguishable matrices, that is, $|PS(n)| = n!$.

Theorem For a given n , $JS(n)$ contains a total number of $2^n n!$ distinguishable matrices, $|JS(n)| = 2^n n!$.

Proof Each non-zero element of J_n has two values $\{-1, 1\}$, and n different elements have 2^n selections. The n elements can select a total number of $n!$ different positions. Both symbol and position selections are independent, and each combination determines a J_n matrix. So there are $2^n n!$ distinguishable matrices.

Corollary $JS(n)$ is a matrix space that is 2^n times larger than $PS(n)$.

Theorem A matrix group of simple rotation in $JS(n)$ may contain $2n$ distinguishable matrices.

Proof Using a rotation matrix $Z_n \in JS(n)$,

$$Z_n = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ & & & & \dots & & \\ \dots & & & & \dots & & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ -1 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}, J_{i,i+1} = 1, i \in [1, n], J_{n,1} = -1 \text{ and a vector}$$

$$X = (1 \ 2 \ 3 \ \dots \ n - 1 \ n).$$

To apply $2n$ Z_n matrices sequentially to the vector X , the following $2n$ vectors are produced:

$$\begin{pmatrix} X = XZ_n^{2n} \\ XZ_n \\ \dots \\ XZ_n^n \\ XZ_n^{n+1} \\ \dots \\ XZ_n^{2n-1} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ -n & 1 & 2 & \dots & n-3 & n-2 & n-1 \\ \dots & \dots & \dots & & \dots & \dots & \dots \\ -1 & -2 & -3 & \dots & -n+2 & -n+1 & -n \\ n & -1 & -2 & \dots & -n+3 & -n+2 & -n+1 \\ \dots & \dots & \dots & & \dots & \dots & \dots \\ 2 & 3 & 4 & \dots & n-1 & n & -1 \end{pmatrix}.$$

That is, $2n$ distinguishable matrices $\{Z_n^j\}_{j=1}^{2n}, Z_n^0 = Z_n^{2n} = I_n$ are included.

Because of $X \xrightarrow{Z_n} -X \xrightarrow{Z_n} X$, there are $Z_n^n = -I_n$ and $Z_n^{2n} = I_n$, that is, $Z_n^n = -I_n$.

Theorem For a Z_n , there are n eigenvalues $\{\lambda_i\}_{i=1}^n, \lambda_i = \sqrt[n]{-1}, i \in [1, n]$.

Proof

$$|\lambda I_n - Z_n| = \begin{vmatrix} \lambda - 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda & -1 & \dots & 0 & 0 \\ \dots & \dots & \dots & & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda & -1 \\ 1 & 0 & 0 & \dots & 0 & \lambda \end{vmatrix} = \lambda^n + 1 = 0.$$

Therefore, $Z_n = \sqrt[n]{-I_n}$.

For non-zero values, $\begin{cases} 1 : \langle x | \rightarrow \langle x | \\ -1 : \langle x | \rightarrow \langle \bar{x} | \end{cases} \text{ map } Z_n \rightarrow \sqrt[n]{-1}.$

Theorem For any state vector X , $X \left(\sqrt[n]{-1} \right)^n = -X$.

Proof

$$\begin{pmatrix} X \\ X \sqrt[n]{-1} \\ \\ X \sqrt[n]{-1}^{n-1} \\ X \sqrt[n]{-1}^n = -X \end{pmatrix} = \begin{pmatrix} \langle 1 | \langle 2 | \langle 3 | \dots \langle n | \\ \langle \bar{n} | \langle 1 | \langle 2 | \dots \langle n-1 | \\ \dots \dots \dots \dots \dots \\ \langle \bar{2} | \langle \bar{3} | \langle \bar{4} | \dots \langle 1 | \\ \langle \bar{1} | \langle \bar{2} | \langle \bar{3} | \dots \langle \bar{n} | \end{pmatrix}.$$

4 Conclusion

Using $(-1,0,1)$ permutation matrices as basic tools, the n th root of NOT operators for quantum computers can be constructed and implemented by the traditional logic structure. Considering that this problem has puzzled advanced research of quantum computer for 20 years, this solution can provide quantum computer designers to practically implement quantum computers using traditional logic. The details of this construction will investigate in other places and the relationships among conjugate logic, quantum logic, quantum gates and complex number structures will be explored for foundation of Quantum computers and quantum computation of future computers.

Acknowledgements Thanks to Dr. G. Liu, Mrs. W. Macmillan, Dr. C. Liu, Dr. A. Tharumarajah and Dr. S. Yang for their invaluable comments, suggestions and careful corrections. Supported, in part by CRC for Intelligent Manufacturing Systems and Technologies.

References

1. R.P. Feynman, Simulating physics with computers. *Int. J. Theor. Phys.* **21**(6/7), 467–488 (1982)
2. M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000)
3. J. Preskill, S.P. Trivedi, M.B. Wise, *Phys. Lett. B* **223**, 26 (1989)
4. R.P. Feynman, R.B. Leighton, M.S. Sands, *The Feynman Lectures on Physics*, vol. 3 (Addison-Wesley, 1989)
5. C. Bennett, Logic reversibility of computation. *IBM J. Res. Dev.* **17**, 525–532 (1973)
6. D. Bouwmeester, A. Ekert, A. Zeilinger, *The Physics of Quantum Information* (Springer, 2000)
7. A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, H. Weinfurter, Elementary gates for quantum computation. *Phys. Rev. A* **52**(5), 3457–3467 (1995)

8. D.W. Leung, I.L. Chuang, F. Yamaguchi, Y. Yamamoto, Efficient implementation of coupled logic gates for quantum computation. *Phys. Rev. A* **61** (2000)
9. D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A* **400**, 97–117 (1985). <http://doi.org/10.1098/rspa.1985.0070>. Published 8 July 1985
10. A.C.C. Yao, Quantum circuit complexity, in *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society Press, 1993), pp. 352–360
11. Z. Meglicki, Z. Wang, *Quantum Computing and Topological Quantum Computing* (2001)
12. C.P. Williams, S.H. Clearwater, *Explorations in Quantum Computing* (Springer, Berlin, 1998)
13. D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond.* **439A**, 553–558 (1992)
14. F.J. Budden, *Complex Numbers and their Applications* (Longmans Green and Co Ltd. 1968)
15. M. Kline, *Mathematical Thought From Ancient to Modern Times*, (Oxford University Press, 1972)
16. J.M. Olmsted, *Calculus with Analytic Geometry*, vol. II (Meredith Publishing Company, 1966)
17. W.R. Hamilton, Theory of conjugate functions or algebraic couples; with a preliminary essay on algebra as the science of pure time. *Trans. Royal Irish Academy* Vol. XVII, 293–422. (The Mathematical Papers of Sir William Rowan Hamilton, Vol. III Algebra, edited for Royal Irish Academy, 3–100) (1837)
18. J. Preskill, A. Kiteav, *Quantum Information and Computation*, Lecture Notes for Physics 229, <http://www.theory.caltech.edu/people/preskill/ph229>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part VII

Applications—Binary Sequences

Unity can only be manifested by the binary.
Unity itself and the idea of Unity are already two.

—*Buddha*

Every axiomatic (abstract) theory admits, as is well known, an unlimited number of concrete interpretations besides those from which it was derived.

Thus we find applications in fields of science which have no relation to the concepts of random event and of probability in the precise meaning of these words.

—*Andrey Kolmogorov*

At its most fundamental, information is a binary choice, in other words,

a single bit of information is one yes-or-no choice.

—*James Cleick*

Various approaches of variant construction on binary sequences were developed from 2011 on cellular automata data sequences to construct 2D/3D maps. From 2014, different binary sequences generated from stream ciphers have been extensively examined and combinatorial maps were developed. For example, Variant Pseudo-Random Number Generator, Hakin9 Extra, Issue 6, 2012 (13), 28–31. <http://hakin9.org/hakin9-extra-62012/>, Interactive Maps on Variant Phase Spaces in Emerging Application of Cellular Automata, InTech Press, 113–196, 2013. <http://dx.doi.org/10.5772/51635>.

Further results were published, e.g., Cryptographic Sequence on Variant Maps, ASONAM 2017: 1065–1071. <https://doi.org/10.1145/3110025.3110152>, and Stationary Randomness of Quantum Cryptographic Sequences on Variant Maps, the 2017 IEEE/ACM International Conference, ASONAM 2017:1041–1048. <https://doi.org/10.1145/3110025.3110151>.

This part of binary sequences is composed of five chapters (18–22).

Chapter “[Novel Pseudorandom Number Generation Using Variant Logic Framework](#)” proposes a novel PRNG using variant logic framework to apply mixed operations of permutation and complement in variant tables to generate random sequences under various control parameters.

Chapter “[RC4 Cryptographic Sequence on Variant Maps](#)” uses binary sequences of RC4 stream cipher on 1DP and 2DP variant maps. Different characteristics of visual distributions can be observed.

Chapter “[Refined Stationary Randomness of Quantum Random Sequences on Variant Maps](#)” checks three quantum random sequences {ANU, USTC, USTC₀} stationary randomness, significant measuring differences identified.

Chapter “[Using Information Entropy to Measure Stationary Randomness of Quantum Random Sequences](#)” uses information entropy to measure stationary randomness of quantum random sequences. Data streams from USTC are selected and their quantitative measurements are compared.

Chapter “[Visual Maps of Variant Combinations on Random Sequences](#)” proposes visual maps of variant combinations on random sequences that provide a flexible framework to support various projections under complicated combinations. Typical maps are illustrated.

Novel Pseudorandom Number Generation Using Variant Logic Framework



Jeffrey Zheng

Abstract Cybersecurity requires cryptology for the basic protection. Among different ECRYPT technologies, stream cipher plays a central role in advanced network security applications; in addition, pseudorandom number generators are placed in the core position of the mechanism. In this chapter, a novel method of pseudorandom number generation is proposed to take advantage of the large functional space described using variant logic, a new framework for binary logic. Using permutation and complementary operations on classical truth table to form relevant variant table, numbers can be selected from table entries having pseudorandom properties. A simple generation mechanism is described and shown, and pseudorandom sequences are analyzed for their cycle property and complexity. Applying this novel method, it can play a useful role in future applications for higher performance of cybersecurity environments.

Keywords Pseudorandom number generation · Variant logic · Cryptology

1 Introduction

In advanced cyber environment, cybersecurity mechanism plays a guider role to protect the secure information communicated and stored in network facilities [1, 2]. To achieve adequate network security effects, cryptology has to be placed in the essential position [1]. Different from block ciphers that operate with a fixed transformation on a large block of plaintext, stream ciphers operate with a time-varying transformation on individual plaintext digits. Under the stream cipher methodology, Pseudorandom Number Generator (PRNG) is placed in the central part of the mechanism.

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_18

From 2000 to 2003, New European Schemes for Signatures, Integrity, and Encryption (NESSIE) were started [3]. During 2004–2008, another European stream cipher project: eSTREAM selected four software and three hardware schemes for ECRYPT stream ciphers [4]. Such extensive international activities on ECRYPT methodologies are showing the ultra-importance of stream cipher technologies in cyber environments for wider security applications.

From a cyber resilience viewpoint [5–7], a set of researchers focus attention on leakage-resilient pseudorandom generator. This direction has shown interesting results to protect valuable information from side-channel attack aspects.

Since PRNG plays a key role in stream cipher applications and is the heart of cryptology [1, 8–10]. Many mathematical methodologies are applied to this field such as linear automata, cellular automata, Galois fields, and other algebraic constructions [1, 9, 11–14]. In cryptology, Boolean logic operations are essential to create highly effective cryptology systems [1, 9, 15, 16] as binary logic generates the greatest efficiency through manipulation of only 1’s and 0’s. Therefore, it is advantageous to investigate potential mechanisms in binary logic due to the follow-on effect it has in cryptology.

2 Classical Logic Function Table

A classic logic function in n variables can be represented as a truth table [8, 9]. For a classic sequence in an ordinary number sequence, each table contains 2^n columns and 2^n rows with a total of $2^n \cdot 2^n$ bits, respectively. An example of the standard truth table can be seen in Fig. 1a.

N	2^n-1	...	i	...	0	$\Delta P(2^n-1)$..	$\Delta P(i)$..	$\Delta P(0)$	K
0	0	...	0	...	0	$\Delta P(0)_{2^n-1}$..	$\Delta P(0)$..	$\Delta P(0)_0$	K_0
...
J	J_{2^n-1}	...	J_i	...	J_0	$\Delta P(J)_{2^n-1}$..	$\Delta P(J)$..	$\Delta P(J)_0$	K_J
...
$2^{2^n}-1$	1	...	1	...	1	$\Delta P((2^{2^n}-1)_{2^n-1})$..	$\Delta P((2^{2^n}-1)_i)$..	$\Delta P((2^{2^n}-1)_0)$	$K_{2^{2^n}-1}$

(a) Truth Table Example

(b) Variant Table Example

Fig. 1 n variable truth table and variant table under P and Δ operators

3 Variant Logic Function Table

Variant logic construction is a new proposed theoretical structure [17, 18] to extend classical logic from the three basic operators: $\{\cap, \cup, \neg\}$. Two additional vector operators: permutation P and complementary Δ are included with the original three to form the five basic operators within the novel framework. Let $S(N)$ denote a permutation group with N elements, then $S(N)$ contains a total of $N!$ permutation operators. Let $B_2^N = \{0, 1\}^N$ denote a binary group with N elements, then B_2^N contains a total of 2^N complementary operators.

The permutation (P) and complementary (Δ) operators are two vector operators performed on each column vector of 2^{2^n} bits. For a given P and Δ , two operators transform the truth table into a variant table. Permutation operators change positions of relevant columns but do not change their values. Complementary operators (Δ) do not change the position for each column, but may change entire values of the column. Two given operators can be performed together to generate a variant table for further usages. There are 2^n columns in the table as permutation elements, so this permutation group $S(2^n)$ contains a total of $2^n!$ permutation operators, and its complementary group $B_2^{2^n}$ includes a total of 2^{2^n} complementary operators. An example of the variant table can be seen in Fig. 1b.

4 Variant Method of Pseudorandom Number Generation

Input: n, P, Δ, m, L variables, $n \in N, P \in S(2^n), \Delta, L, m \in B_2^{2^n}$

Output: $\{K_m, K_{m+1}, \dots, K_{m+L-1}\}L \cdot 2^n$ bit sequences

Method: The process for pseudorandom number generation can be seen in Fig. 2. n is the input variable number. Using n variables, a standard truth table can be constructed in 2^n columns and 2^{2^n} rows. P is a given permutation operator $P = (P_{2^n-1} \dots P_I \dots P_0)$, $P \in S(2^n)$, where P_I corresponds to the I -th column. A given complementary operator $\Delta \in B_2^{2^n}$, $\Delta = (\Delta_{2^n-1} \dots \Delta_I \dots \Delta_0)$, $\Delta_I \in B_2$ shows that the operator is performed on the I -th column, if $\Delta_I = 0$, all values of the column are reversed and if $\Delta_I = 1$, all values are invariant. $0 \leq m < 2^{2^n}$ is an initial position for output sequences; from K_m, L conditions, $\{K_{m+i}\}_{i=0}^{L-1}$ are output generated 0–1 bit sequences.

5 Sequence Generation Example

For convenient understanding procedure, an example is selected to show in the $n = 2$ case shown in Fig. 3. Parameters are initialized to arbitrary values: $n = 2, P = (1203)$, and $\Delta = (0110)$.

After the table is generated, the pseudorandom sequence can read off the table. For $m = 4$ and $L = 6$ conditions, a random number starting at position 4 of the variant table containing six elements can be found.

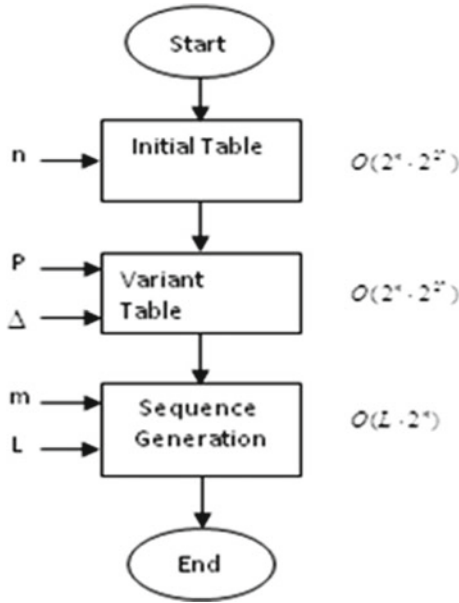


Fig. 2 Variant method of random number generation

Truth Table	P=(1203)	Permutation Table	Δ=(0110)	Variant Table																																																																																																																																																																																																																																																																												
<table border="1"> <thead> <tr> <th>N0</th> <th>11</th> <th>10</th> <th>01</th> <th>00</th> </tr> <tr> <th></th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>3</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>4</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>5</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>6</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>7</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>8</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>9</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>10</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>11</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>12</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>13</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>14</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>15</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	N0	11	10	01	00		3	2	1	0	0	0	0	0	0	1	0	0	0	1	2	0	0	1	0	3	0	0	1	1	4	0	1	0	0	5	0	1	0	1	6	0	1	1	0	7	0	1	1	1	8	1	0	0	0	9	1	0	0	1	10	1	0	1	0	11	1	0	1	1	12	1	1	0	0	13	1	1	0	1	14	1	1	1	0	15	1	1	1	1	<table border="1"> <thead> <tr> <th>01</th> <th>10</th> <th>00</th> <th>11</th> <th>K'</th> </tr> <tr> <th>1</th> <th>2</th> <th>0</th> <th>3</th> <th></th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td></tr> </tbody> </table>	01	10	00	11	K'	1	2	0	3		0	0	0	0	0	0	0	1	0	2	1	0	0	0	8	1	0	1	0	10	0	1	0	0	4	0	1	1	0	6	1	1	0	0	12	1	1	1	0	14	0	0	0	1	1	0	0	1	1	3	1	0	0	1	9	1	0	1	1	11	0	1	0	1	5	0	1	1	1	7	1	1	0	1	13	1	1	1	1	15	<table border="1"> <thead> <tr> <th>(01)⁰</th> <th>(10)¹</th> <th>(00)⁰</th> <th>(11)⁰</th> <th>K</th> </tr> <tr> <th>1⁰</th> <th>2¹</th> <th>0¹</th> <th>3⁰</th> <th></th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr> </tbody> </table>	(01) ⁰	(10) ¹	(00) ⁰	(11) ⁰	K	1 ⁰	2 ¹	0 ¹	3 ⁰		1	0	0	1	9	1	0	1	1	11	0	0	0	1	1	0	0	1	1	3	1	1	0	1	13	1	1	1	1	15	0	1	0	1	5	0	1	1	1	7	1	0	0	0	8	1	0	1	0	10	0	0	0	0	0	0	0	1	0	2	1	1	0	0	12	1	1	1	0	14	0	1	0	0	4	0	1	1	0	6
N0	11	10	01	00																																																																																																																																																																																																																																																																												
	3	2	1	0																																																																																																																																																																																																																																																																												
0	0	0	0	0																																																																																																																																																																																																																																																																												
1	0	0	0	1																																																																																																																																																																																																																																																																												
2	0	0	1	0																																																																																																																																																																																																																																																																												
3	0	0	1	1																																																																																																																																																																																																																																																																												
4	0	1	0	0																																																																																																																																																																																																																																																																												
5	0	1	0	1																																																																																																																																																																																																																																																																												
6	0	1	1	0																																																																																																																																																																																																																																																																												
7	0	1	1	1																																																																																																																																																																																																																																																																												
8	1	0	0	0																																																																																																																																																																																																																																																																												
9	1	0	0	1																																																																																																																																																																																																																																																																												
10	1	0	1	0																																																																																																																																																																																																																																																																												
11	1	0	1	1																																																																																																																																																																																																																																																																												
12	1	1	0	0																																																																																																																																																																																																																																																																												
13	1	1	0	1																																																																																																																																																																																																																																																																												
14	1	1	1	0																																																																																																																																																																																																																																																																												
15	1	1	1	1																																																																																																																																																																																																																																																																												
01	10	00	11	K'																																																																																																																																																																																																																																																																												
1	2	0	3																																																																																																																																																																																																																																																																													
0	0	0	0	0																																																																																																																																																																																																																																																																												
0	0	1	0	2																																																																																																																																																																																																																																																																												
1	0	0	0	8																																																																																																																																																																																																																																																																												
1	0	1	0	10																																																																																																																																																																																																																																																																												
0	1	0	0	4																																																																																																																																																																																																																																																																												
0	1	1	0	6																																																																																																																																																																																																																																																																												
1	1	0	0	12																																																																																																																																																																																																																																																																												
1	1	1	0	14																																																																																																																																																																																																																																																																												
0	0	0	1	1																																																																																																																																																																																																																																																																												
0	0	1	1	3																																																																																																																																																																																																																																																																												
1	0	0	1	9																																																																																																																																																																																																																																																																												
1	0	1	1	11																																																																																																																																																																																																																																																																												
0	1	0	1	5																																																																																																																																																																																																																																																																												
0	1	1	1	7																																																																																																																																																																																																																																																																												
1	1	0	1	13																																																																																																																																																																																																																																																																												
1	1	1	1	15																																																																																																																																																																																																																																																																												
(01) ⁰	(10) ¹	(00) ⁰	(11) ⁰	K																																																																																																																																																																																																																																																																												
1 ⁰	2 ¹	0 ¹	3 ⁰																																																																																																																																																																																																																																																																													
1	0	0	1	9																																																																																																																																																																																																																																																																												
1	0	1	1	11																																																																																																																																																																																																																																																																												
0	0	0	1	1																																																																																																																																																																																																																																																																												
0	0	1	1	3																																																																																																																																																																																																																																																																												
1	1	0	1	13																																																																																																																																																																																																																																																																												
1	1	1	1	15																																																																																																																																																																																																																																																																												
0	1	0	1	5																																																																																																																																																																																																																																																																												
0	1	1	1	7																																																																																																																																																																																																																																																																												
1	0	0	0	8																																																																																																																																																																																																																																																																												
1	0	1	0	10																																																																																																																																																																																																																																																																												
0	0	0	0	0																																																																																																																																																																																																																																																																												
0	0	1	0	2																																																																																																																																																																																																																																																																												
1	1	0	0	12																																																																																																																																																																																																																																																																												
1	1	1	0	14																																																																																																																																																																																																																																																																												
0	1	0	0	4																																																																																																																																																																																																																																																																												
0	1	1	0	6																																																																																																																																																																																																																																																																												

Fig. 3 Example for generation of pseudorandom sequence

6 Complexity Analysis

From an application viewpoint, it is important to have the exact complexity evaluation for the method. In the initial stage, it is necessary to manipulate 2^n columns and each column with 2^{2^n} rows; the total numbers of $2^n \cdot 2^{2^n}$ bits are required. The total complexity is of order $O(2^n \cdot 2^{2^n})$.

To generate variant table values, P operations need at least to manipulate bits once and Δ operations to manipulate the same number of bits, i.e., $O(2^n \cdot 2^{2^n})$.

Selecting $L \cdot 2^n$ bits from the variant table, it is necessary to perform $O(L \cdot 2^n)$ operations.

If a full table needs to be generated as a random resource, $O(2^n \cdot 2^{2^n})$ computational complexity is required. In general, their computational complexity is $O(L \cdot 2^n) - O(2^n \cdot 2^{2^n})$ $0 < L < 2^{2^n}$.

Maximal cycle length: under this construction, the maximal length of the pseudorandom number sequence is $2^n \cdot 2^{2^n}$ bits. For any short sequences, the output sequence has a length less than this number. No clear cycle effects can be directly observed.

7 Conclusion

It is important to design this new PRNG method to use variant logic construction. Since P and Δ potentially have a huge configuration space $2^n! \times 2^{2^n}$ times larger than classical logic function spaces. Exploring how difficulties for this mechanism to be decoded will be the main issue for coming cryptologist's theoretical targets. In addition, it is important to understand what type of distribution will be relevant to this generation mechanism. Owing to intrinsic complexity of variant logic construction, this provides potential barriers to protect this type of sequences decoded directly.

Considering PRNG placed in the central part of stream cipher mechanism, and stream cipher technologies are more and more important in advanced network security environment, higher performance methodology and relevant implementation will be useful in this field. Ongoing approaches will focus on whether this mechanism provides better PRNG methods to help different protections on side-channel attacks [1–7, 19, 20] in wider network applications to resolve practical leakage-resilient issues in the future.

References

1. M. Robshaw, *Stream ciphers*. RSA Laboratories Technical Report TR-701 (1995)
2. Y. Xiao, H. Li, S. Choi, Protection and guarantee for voice and video traffic in IEEE 802.11e Wireless LANs, in *IEEE INFOCOM* (2004), p. 11

3. NESSIE New European Schemes for Signatures, Integrity and Encryption, <https://www.cosic.esat.kuleuven.be/nessie/>
4. The eSTREAM Project, <http://www.ecrypt.eu.org/stream/index.html>
5. F.X. Standaert, T. Malkin, M. Yung, A unified framework for the analysis of side-channel key recovery attacks, in *EUROCRYPT*, (2009), pp. 443–461
6. A. Dwivedi, D. Tebben, P. Harshavardhana, Characterizing cyber-resiliency, in *The 2010 Military Communication Conference—Unclassified Program—Cyber Security and Network Management* (IEEE press, 2010), pp. 1847–1852
7. Y. Yu, F. X. Standaert, O. Pereira, M. Yung, Practical leakage-resilient pseudorandom generator, in *CCS'2010* (ACM, 2010), pp. 141–151
8. G.B. Agnew, Random source for cryptographic systems, in *Advances in Cryptology | EUROCRYPT '87 Proceedings* (Springer-Verlag, 1988), pp. 77–81
9. C. Atkinson, A family of switching algorithms for the computer generation of beta random variables. *Biometrika* **66**(1), 141–145 (1979)
10. A statistical test suite for random and pseudorandom number generators for cryptographic applications (NIST Special Publication, 800-22 2010)
11. R. Davies. Hardware random number generators, in *International 15th Australian Statistical Conference* (2000)
12. D. Eastlake, S.D. Crocker, J.I. Schiller, Randomness requirements for security, RFC 1750, Internet Engineering Task Force (1994)
13. V. Kachitvichyanukul, B.W. Schmeiser, Binomial random variate generation. *Commun. ACM* **31**(2), 216–223 (1988)
14. M. Matsumoto, T. Nishimura, Dynamic creation of pseudorandom number generators, in *Proceedings of the Third International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing: Monte Carlo and Quasi-Monte Carlo Methods 1998* (2000), pp. 56–69
15. S.K. Park, K.W. Miller, Random number generators: good ones are hard to find. *Commun. ACM* **31**(10), 1192–1201 (1988)
16. M. Santha, U.V. Vazirani, Generating quasi-random sequences from slightly random sources. *J. Comput. Syst. Sci.* **33**, 75–87 (1986)
17. J. Zheng, C. Zheng, T.L. Kunii, *A framework of variant logic construction for cellular automata* (InTech—Open Access Publisher, 2011). <http://www.intechopen.com/articles/show/title/a-framework-of-variant-logic-construction-for-cellular-automata>. ISBN 978-953-307-172-5
18. J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China.* **5**(2), 163–173 (2010). <http://www.springerlink.com/content/91474403127n446u/> (Higher Education Press & Springer)
19. G. Gong, Cryptographic properties of the welch-gong transformation sequence generators. *IEEE Trans. Inf. Theor.* **48**(11), 2837–2846 (2002)
20. B. Aissa, D. Nouredine, Designing resilient functions and bent function for stream ciphers. *Georgian Electron. Sci. J Comput. Sci. Telecommun.* **1**(18), 27–33 (2009)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



RC4 Cryptographic Sequence on Variant Maps



Zhonghao Yang and Jeffrey Zheng

Abstract In modern cyberspace environment, big data streams are the most important issue in people's daily lives, each person produces a larger number of data streams every day from personal computer, cell phone, and kinds of wearable smart device. Security risks of storage and transmission of data streams may lead to personal privacy disclosure, it is important for network security to have useful tools facing challenges. Randomness testing provides useful tools to secure results of stream ciphers. Based on multiple statistical probability distributions, this chapter presents a visual scheme, variant maps, to measure a whole cryptographic sequence into multiple 1D and 2D maps. Mapping mechanism and sample cases are provided.

Keywords Random sequence · Big data · Variant map

1 Introduction

In modern cyberspace environments, more than 2.5 EB data streams per day are generated from global network environments [1]. Huge network companies managed massive data streams in PB every day [2]. The development of artificial intelligence fields makes it easier to extract valuable information from big data [3–5]. Big data

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014) and Yunnan Advanced Overseas Scholar Project.

Z. Yang
Yunnan University, Kunming, China
e-mail: houseashley07@hotmail.com

J. Zheng (✉)
Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_19

and big data technology provide modern societies so much convenience to many places, and with several threats to network security [6, 7].

Stream ciphers are the most useful scheme to protect the security of data streams in both transmission and storage processes. Pseudorandom number sequences are generated by various algorithms based on recursive computational models, and true random number sequences are generated by different physical methods. The typical stream ciphers are RC4 and Salsa20. Stream ciphers can be built using block ciphers in OFB or CTR model. In this chapter, an RC4 stream cipher is selected to generate pseudorandom sequences for testing.

From a testing viewpoint, randomness tests focus on three aspects: probability, autocorrelation, and unpredictability. NIST 800-22 provides a list of randomness testing method based on p -value [8].

In this chapter, two types of 1D and 2D statistical probability maps are used to visualize a longer pseudorandom number sequence generated from an RC4 stream cipher.

2 Related Work

Variant map is an emerging technology proposed in 2010s to handle multiple 0–1 vectors in phase spaces on variant framework [9–11]. Different applications are explored for variant maps on ECG data sequences [12], bat echolocation call sequences [13], gene sequence [14], and cryptographic sequences [15–17].

3 Mapping Model

This chapter uses two mapping schemes on 1D and 2D statistical probability distributions as variant maps for an input N -length 0–1 sequence. The architectural diagram of the mapping model is shown in Fig. 1. It is composed of three components: segmentation, measurement, and visualization.

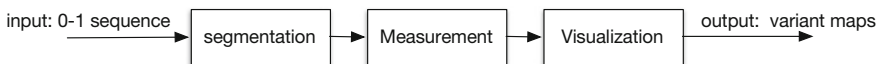
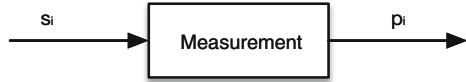


Fig. 1 Architecture of variant map for cryptographic sequence

Fig. 2 Measurement



3.1 Basic Symbol

- (1) S : an input 0–1 sequence,
- (2) s_i : the i -th segment of the input sequence,
- (3) N : length of the input sequence,
- (4) M : count of segments,
- (5) m : length of a segment, and
- (6) p : number of 1's elements in the segment.

3.2 Mapping Model

Three components can be described as follows.

- Segmentation

Input data is a 0–1 sequence S of length N . It can be divided into M segments and each segment has m elements.

$$M = \left\lfloor \frac{N}{m} \right\rfloor$$

$$S = \{s_0, s_1, \dots, s_i, \dots, s_{M-1}\}, \quad 0 \leq i < M$$

- Measurement

For each segment s_i of S , the following analysis is performed to obtain the one feature p_i of the segment, that is, the number of 1 of s_i , and $0 \leq p \leq m$. For example, for two segments $s_1 = 00011$ and $s_2 = 10110$, and two measurements are $p_1 = 2$ and $p_2 = 3$ (Fig. 2).

Calculating all segments of S , a set of p measurements are determined.

$$\{p_0, \dots, p_i, \dots, p_{M-1}\} = \{p_i\}_{i=0}^{M-1}, \quad 0 \leq i < M$$

- Visualization

From the generated sequence of measurements, two types of diagrams can be created: The first one is a 1D map, 1DP sorted from $\{p_i\}_{i=0}^{M-1}$ directly shown in Fig. 3a. The second one is a 2D map, 2DP sorted from a pair of measurements $\{p_i, p_{i+1}\}_{i=0}^{M-1}$

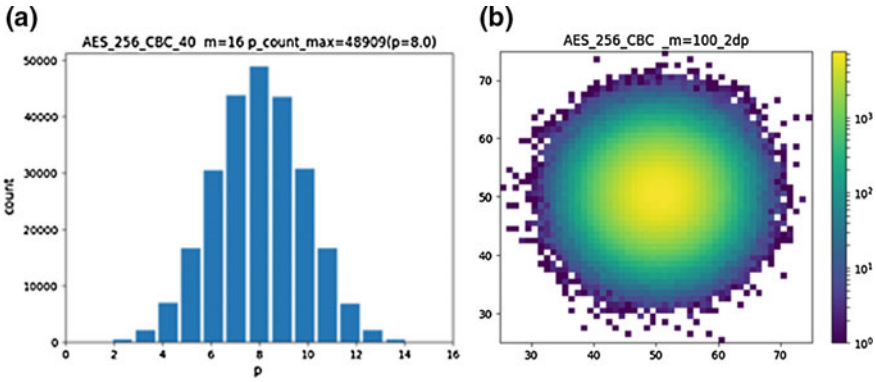


Fig. 3 Two maps; a 1DP; b 2DP

created from $\{p_i\}_{i=0}^{M-1}$ shown in Fig. 3b. This mapping scheme is one of Markov chain models.

4 Random Sequence Data Sources

In this chapter, a pseudorandom generator is based on an AES block cipher on the OFB mode. A total amount of 120 MB cryptographic sequences has been generated.

5 Mapping Results

The input sequence is mapped with a list of various lengths on different segmentations. Three sets of various m lengths are selected and two types of relevant 1DP and 2DP maps are shown in Fig. 4a–c, for (a) $m = \{8, 16, 32, 64, 128, 256\}$, (b) $m = \{80, 100, 120, 140, 160\}$, and (c) $m = \{126, 127, 128, 129, 130\}$. Four enlarged 2DP maps are shown in Fig. 5 for $m = \{126, 127, 128, 129\}$ and two enlarger 2DP maps are shown in Fig. 6 for $m = \{128, 130\}$, respectively.

6 Result Analysis

In Fig. 4, both 1DP and 2DP maps are illustrated. When the input sequence is larger enough to $m \times 2^m$, the results of 1DP maps are corresponding to binomial distributions. It is interesting to see significant changes when various lengths of segments are applied.

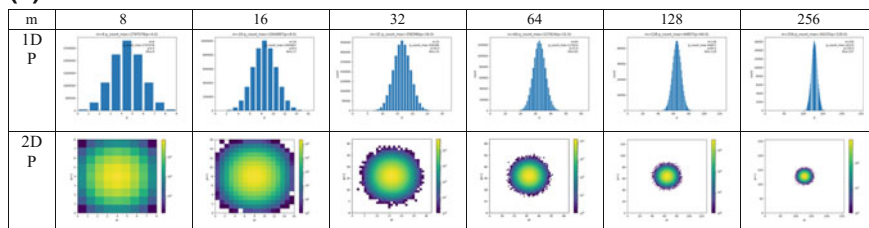
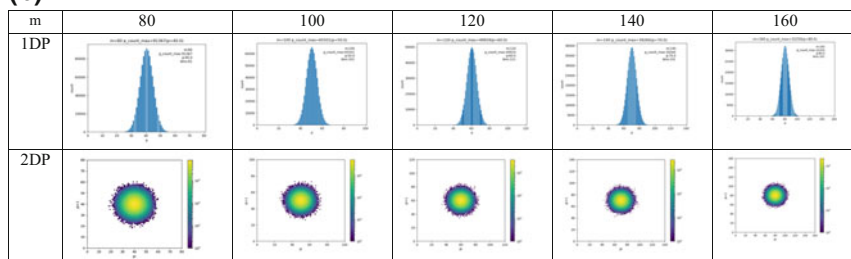
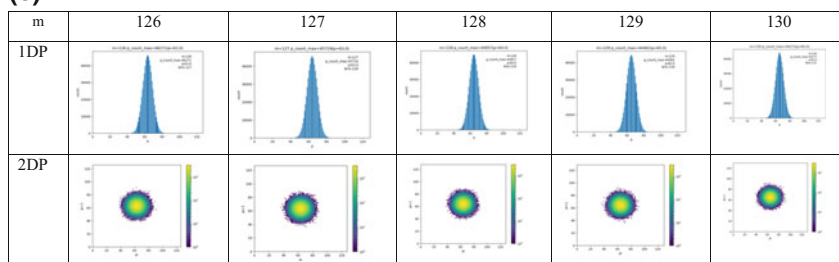
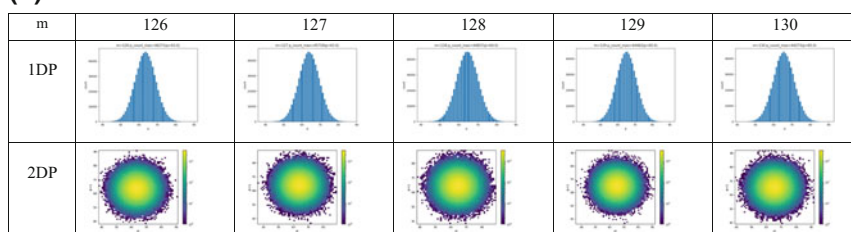
(a)**(b)****(c)****(d)**

Fig. 4 1DP and 2DP maps. **a** $m = \{8, 16, 32, 64, 128, 256\}$; **b** $m = \{80, 100, 120, 140, 160\}$; **c** $m = \{126, 127, 128, 129, 130\}$; **d** enlarged 1dp and 2dp, $m = \{126, 127, 128, 129, 130\}$

For various 2DP maps in Figs. 4, 5, and 6, 2D distributions are represented as pseudocolor to illustrate relevant 3D structures. From smaller maps to enlarged maps,

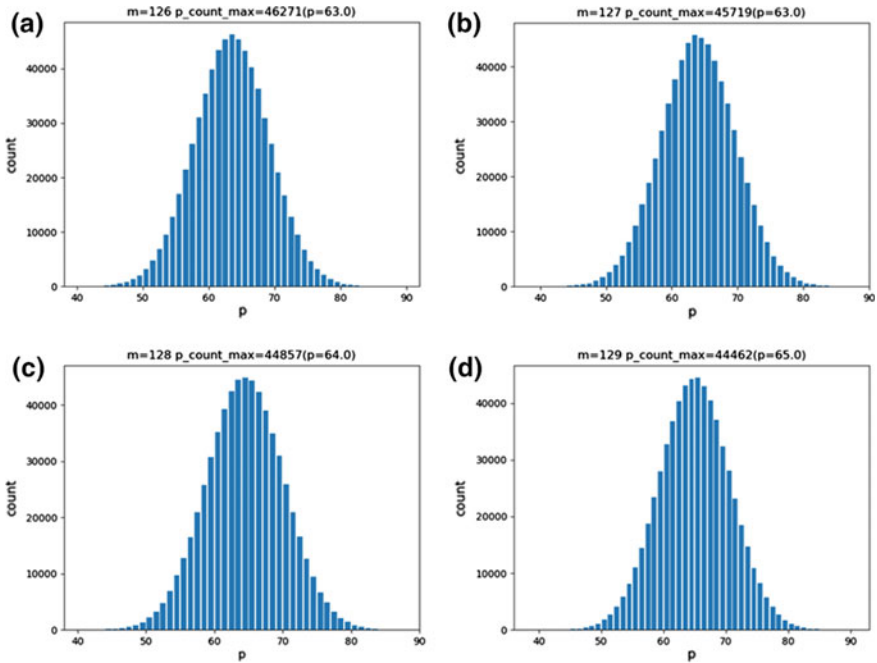


Fig. 5 Enlarger 1DP maps. **a** $m = 126$; **b** $m = 127$; **c** $m = 128$; **d** $m = 129$

many interesting features can be identified and significant symmetric or nonsymmetric properties could be identified. Enlarger maps can see further refined patterns in detail.

7 Conclusion

Mapping model in this chapter is a focus on a single sequence for two types of 1DP and 2DP maps. 1DP maps are corresponding to classical statistical maps and 2DP maps are represented as various Markov chains. Further researches and experiments are required to explore useful tools on cryptographic sequences in detail (Figs. 7 and 8).

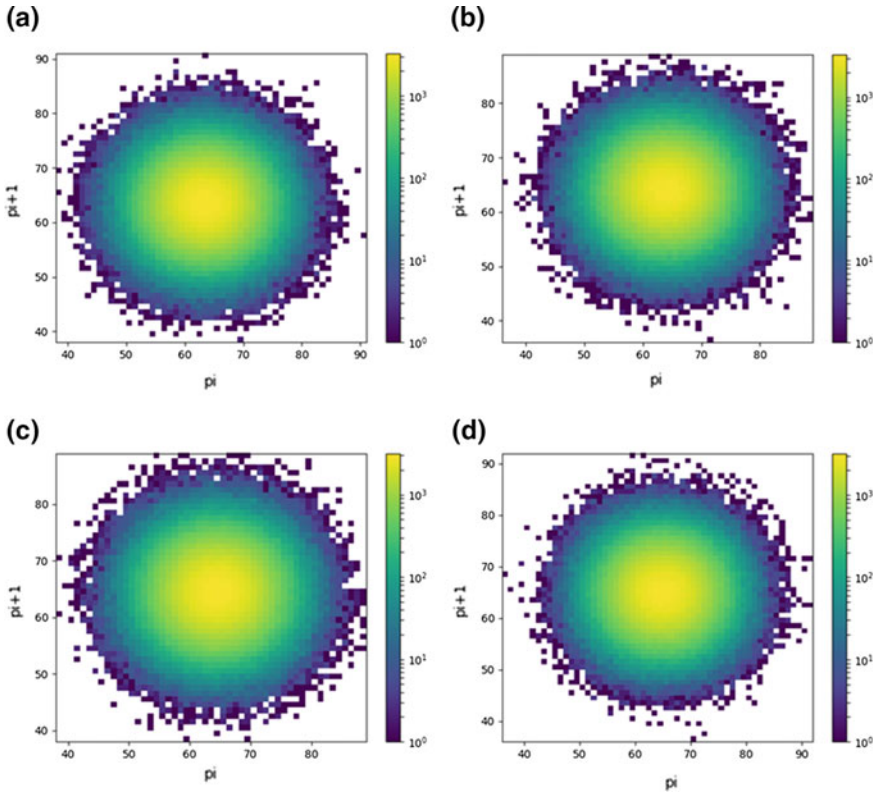


Fig. 6 Enlarged 2DP maps. **a** $m = 126$; **b** $m = 127$; **c** $m = 128$; **d** $m = 129$

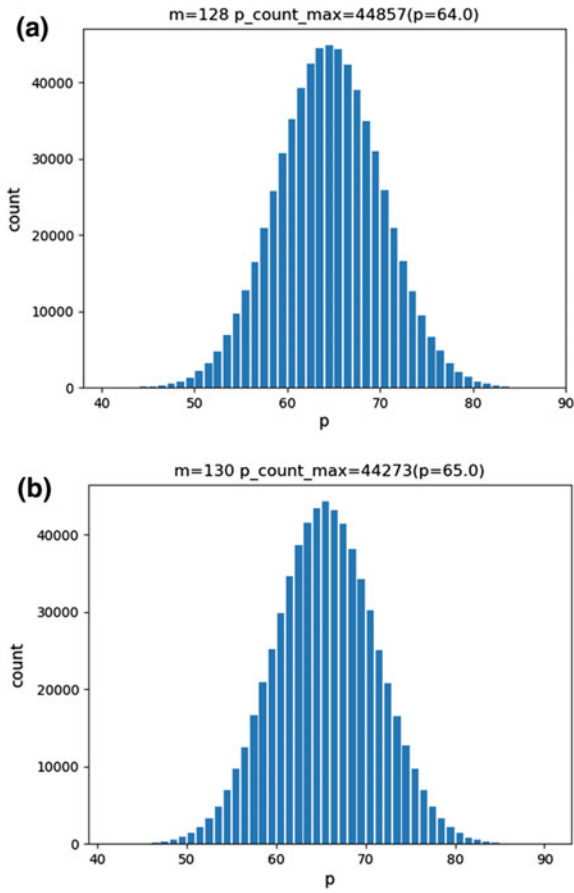


Fig. 7 Enlarger 1DP maps. **a** $m = 128$; **b** $m = 130$

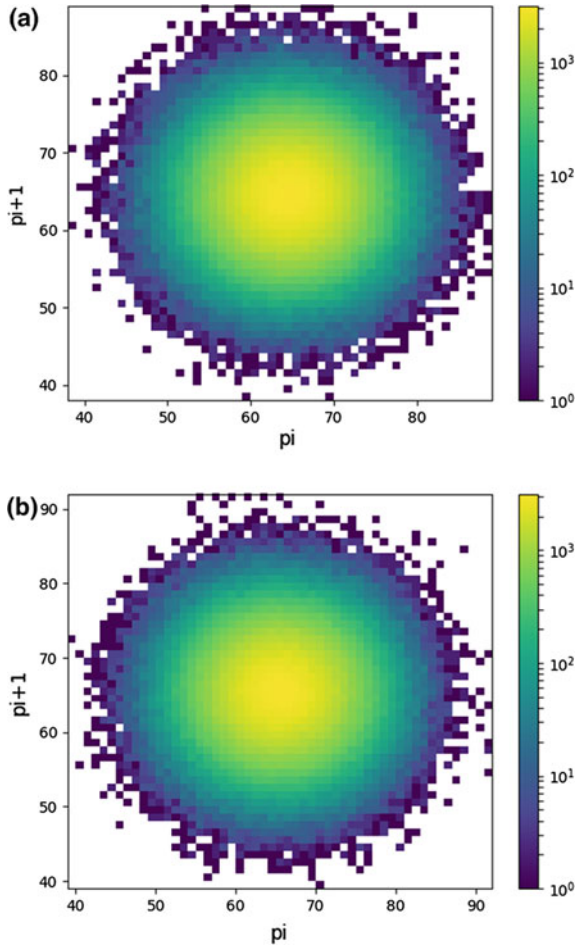


Fig. 8 Enlarger 2DP maps. **a** $m = 128$; **b** $m = 130$

References

1. A. Anwaar et al., Big data for development: applications and techniques. *Big Data Analytics* 1(1), 2 (2016)
2. V. Mayer-Schönberger, K. Cukier, *Big Data: a revolution that will transform how we live, work, and think* (Eamon Dolan/Houghton Mifflin Harcourt, 2013)
3. M.M. Najafabadi et al., Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2(1), 1 (2015)
4. R. Fang, S. Pouyanfar, Y. Yang et al., Computational health informatics in the big data age: a survey. *ACM Comput. Surv.* 49(1), 12 (2016)
5. M.D. Assunção et al., Big data computing and clouds: trends and future directions. *J. Parallel Distrib. Comput.* 79–80, 3–15 (2015)
6. L. Xu, C. Jiang, J. Wang et al., Information security in big data: privacy and data mining. *IEEE Access* 2, 1149–1176 (2014)
7. L. Lerman, G. Bontempi, O. Markowitch, A machine learning approach against a masked AES. *J. Cryptographic Eng* 5(2), 123–139 (2015)
8. L.E. Bassham III et al., *SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudo-random Number Generators for Cryptographic Applications* (Nist Special Publication 2010)
9. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* 5, 163 (2010)
10. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, *A Framework of Variant Logic Construction for Cellular Automata* (InTech, Shanghai, 2011)
11. J. Zheng, C. Zheng, Variant measures and visualized statistical distributions. *Acta Photonica Sinica* 40, 1397 (2011)
12. Y. Ji et al., Variant maps on normal and abnormal ECG data sequences. *Biol. Med.* 8(6), 1 (2016)
13. D.M. Heim, O. Heim, P.A. Zeng, J. Zheng, Successful creation of regular patterns in variant maps from bat echolocation calls. *Biol. Syst. Open Access* 5, 166 (2016)
14. J. Zheng, W. Zhang, J. Luo, W. Zhou, V. Liesaputra, Variant map construction to detect symmetric properties of genomes on 2D distributions. *J. Data Min. Genomics Proteomics* 5, 1 (2014)
15. J. Zheng, J. Luo, J. Zhou, Pseudo DNA sequence generation of non-coding distributions using variant maps on cellular automata. *Appl. Math.* 5, 153 (2014)
16. W.Z. Yang, J. Zheng, Variant Pseudo-Random Number Generator. *Hakin9 Extra* 6, 28–31 (2012). <http://hakin9.org/hakin9-extra-62012/>.
17. J. Zheng et al., Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Math.* 5(7A), 5–24 (2013)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Refined Stationary Randomness of Quantum Random Sequences on Variant Maps



Jeffrey Zheng, Yamin Luo and Zhefei Li

Abstract In this chapter, a testing model is used to apply statistical probability in multiple distributions on three maps for a selected sequence to check refined stationary randomness on quantum sequences. Three random data sequences are collected from two quantum random resources: one from Australian National University (ANU) and two (initial and secure) from University of Science and Technology of China (USTC). Multiple results are created on three maps, and measurements of stationary randomness are illustrated and compared. Three samples show distinct stationary properties.

Keywords Variant maps · Quantum random sequence · Chaotic random sequence
Ordered measures · Maximal; Stationary randomness

1 Introduction

In advanced social network environment, multimedia signal sequences of big data streams are composed of time series processes. Quantum experiments in quantum satellite using quantum key distribution (QKD) systems [1] is the most advanced ICT

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

Y. Luo · Z. Li

Yunnan University, Kunming, China
e-mail: 1047668416@qq.com

Z. Li

e-mail: 576167164@qq.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_20

development to establish ultra-secure quantum communications. For a QKD system, a truly random number generator [2] play a key role. From an analysis viewpoint, it is necessary to test stationary randomness in time variations. In this section, a list of relevant schemes: pseudo/truly random sequences, P _value, statistical probability distribution, optical statistics, stationary properties, and variant maps, are discussed.

1.1 Pseudo/True Random Sequences

1.1.1 Pseudorandom Sequences

Traditional stream ciphers [3] on linear feedback shift register structure (LFSR) are used as pseudorandom number generators. The LFSR stream ciphers are the core in classical stream ciphers.

The new generation of stream ciphers has being shifted from LFSR [3] to nonlinear modes: NLFSR, clock control [4] and nonlinear functions, etc. It is difficult to use nonlinear mathematical theories, recursive models, descriptive tools, and implementing schemes in nonlinear dynamic environments.

1.1.2 True Random Sequences

Differently from pseudorandom sequences generated by stream ciphers, high-quality stochastic oscillators of truly random sequences are generated from special hardware devices such as laser photonics [5], nonlinear optics, quantum optics [6], quantum noises, thermal noise, chaos, and fractal nonlinear dynamics [7].

1.2 Testing Schemes

1.2.1 P _value Schemes

Various statistic testing packages measure randomness properties on a given random sequence. NIST 800-22 package [8] is a typical representative to provide more than 15 testing schemes. Using the package, it is essential to check whether P _value > 0.01 for the sequence. Since such measuring scheme provides a static condition, it is difficult to use only P _value parameter to express complex dynamic behaviors involved in random sequences.

1.2.2 Multiple Statistical Probability Distributions

Measuring random sequences under segment conditions, multiple statistical probability schemes are useful to create various distributions to illustrate complex spatial relationships.

Multivariate normal probability distributions are the most important and powerful tools to test stochastic characteristics of a random data sequence under the framework of probability, stochastic process and statistics [9] for nonlinear problems. In this kind of measuring models, when a data sequence is sufficiently long, the high dimensional probability distribution of the sequence [10] is converged to a continuous Gaussian distribution. Multivariate Gaussian probability distributions support various schemes to analyze complex stochastic data set of measuring sequences in continuous conditions.

1.2.3 Photon Statistic in Quantum Optics

Photon statistics is the theoretical and experimental approach on the statistical distributions in photon counting experiments to analyze the statistical nature of photons in a light source.

Three types of distributions can be obtained by the light source [11]: Poissonian, super-Poissonian, and sub-Poissonian. The variance and average number of photon counts are identified for the corresponding distribution. Both Poissonian and super-Poissonian light are described by a semi-classical theory in which the light source is modeled as an electromagnetic wave and the atom is modeled by quantum mechanics. In contrast, sub-Poissonian light requires the quantization of the electromagnetic field for a proper description and is a direct measure of the particle nature of light.

1.2.4 Stationary Properties

In mathematics and statistics, a stationary process is a stochastic process [12] whose joint probability distribution does not change when shift operations performed. Consequently, parameters such as mean and variance, if they are present, also do not change over time. Stationarity is an interesting property in time series analysis.

In applied mathematics, the Wiener–Khinchin theorem [13], states that the Auto-correlation Function (ACF) of a wide-sense stationary process has a spectral decomposition given by the power spectrum of the process. One of the effective ways for identifying stationary times series is the ACF plot [14]. For a stationary time series, the ACF will drop to zero relatively quickly.

1.3 *Quantum Random Resources*

Quantum random numbers can be generated from a physical quantum source of a coherent laser light to be splitting a beam of light into two beams and then measuring the power in each beam. Due to the light intensity in each beam fluctuates about the mean. Those fluctuations can be converted into a source of random numbers [15–17] being a stationary Poisson distribution.

1.3.1 ANU Resource

The ANU Quantum Random Numbers Server is an open website [18] to offer true random numbers to anyone on the internet. Such random numbers are generated in real-time by measuring the quantum fluctuations of the vacuum. The electromagnetic field of the vacuum exhibits random fluctuations in phase and amplitude at all frequencies. By carefully measuring these fluctuations, ultra-high bandwidth random numbers can be generated.

About 1 GB data streams are downloaded and 100MB data streams are used for the testing.

1.3.2 USTC Resource

In the Key Laboratory of Quantum Information, USTC, and CAS, true random number sequences are generated [16]. This type of true random sequences supports advanced quantum communication devices of QKD systems [19].

More than 20GB quantum random number sequences are provided by USTC for random streams testing. Two data sequences are represented as $USTC_0$ (initial) and USTC (secure), respectively. About 100MB data streams are selected for each sequence.

1.3.3 Refined Properties

From an analysis viewpoint, a Toeplitz hash algorithm has used to get an initial sequence $USTC_0$ as input and USTC sequence as output. Checking such refined variations, this is an interesting property for us to make a detailed identification.

From a random testing viewpoint, initial sequences have some difficulties to pass NIST tests and secure sequences are ensured to pass NIST tests. Some refined differences on random characteristics could be distinguished.

1.4 Variant Framework

Various schemes following the top-down strategy are explored to use multiple measures to partition special phase spaces from a top state set to multiple bottom states via multilevels of a hierarchy in combinatorial algorithms [20], image analysis and processing for many years.

The conjugate classification [21] is proposed to apply seven measures in a hierarchy to partition the kernels of four regular plane lattices on $n = \{4, 5, 7, 9\}$ cases for 2D binary images. For 1D cellular automata sequences, global random behaviors are visualized in 2D maps.

For n -tuple bit vectors, the variant logic framework [22] is proposed, various applications are explored: 3D visual method on random number sequences [23], variant Pseudorandom Number Generator (PRNG) [24], computational simulation on quantum interactions [25], noncoding DNA analysis, bat echolocation [26], and stationary randomness [27].

1.5 Proposed Scheme

For the convenience of testing stationary randomness on random sequences, we propose a testing system for a stationary random sequence with length N , multiple segments M are divided from the sequence by a given length m , a 2-tuple pair of measures can be extracted from a 0-1 segment that are the number of 1 element and the number of 1 pattern in the segment. All paired measures are composed of a sequence of M pairs of measures as an ordered measuring set with M elements.

The pairs of the measuring sequence are directly separated as two independent measuring sequences to keep each parameter in the same order. A total of three sequences of distinct measures are constructed including two sequences on single measures and one sequence on 2-tuple measures.

Following this approach, two sets of single measuring sequences are sorted as two 1D numeric arrays as statistical histograms corresponding to 1D maps and the 2-tuple measuring sequence is sorted as a 2D integer array as statistic histograms being a 2D map. Under the controlling operations on the changes of shift displacement, multiple results of the three measuring sequences are transformed into 1D statistic histograms and 2D pseudo-color maps to show effective patterns from the generated sequence under various positions and conditions on a list of shift operations.

1.6 Organization of the Chapter

This chapter uses a testing system for a stationary random sequence on the system architecture in Sect. 2. In Sect. 3, test results are provided for two quantum random sequences. From the results of the visual maps in Sect. 3, result analysis and brief comparison are described in Sect. 4. And finally in Sect. 5, the main results are summarized.

2 Testing System

To describe the testing system, diagrams are shown in Fig. 1.

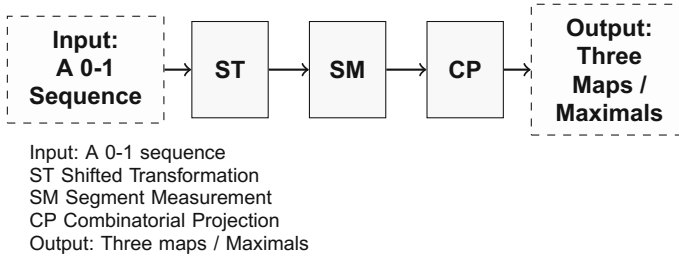


Fig. 1 The architecture of testing stationary random sequences

2.1 System Architecture

This system is composed of five parts: Input, Shifted Transformation (ST), Segment Measurement (SM), Combinatorial Projection (CP), and Output.

The input of the testing system is a selected 0-1 sequence and its output is composed of three maps, two in 1D and one in 2D for visual distributions, and three maximals to be processed by ST, SM, and CP modules, respectively.

Further technical details are described in Chapter [Stationary Randomness of Three Types of Six Random Sequences on Variant Maps](#) of this book.

3 Testing Results

Three quantum random sequences are selected from ANU and USTC resources.

Typical results of testing stationary properties for three sequences in nine maps are shown in Fig. 2. Three sets of results are shown in Fig. 3a, b. In Fig. 3a, six values of $r = \{0, 16, 32, 96, 112, 128\}$ are selected to show three pairs of corresponding maps: 1DP, 2DPQ, and 1DQ for three sequences on the top part. Nine 2D maps of maximal curves for $r = 0 - 128$ are shown to illustrate refined properties in stationary random curves on the bottom column. In Fig. 3b, three maximal curves on three 2D maps are compared. In Fig. 4a–c, three larger maps on $r = \{48, 64, 80\}$ are shown corresponding to (a) 1DP, (b) 2DPQ, and (c) 1DQ for three cases. Three larger maps of three maximal curves are shown in Fig. 5.

3.1 Quantitative Measurements

For a G map, let G_x be an average variation, ΔG_x be a region of variations and $G_x^R = \Delta G_x / G_x$ be a variation ratio. In convenient in comparison, let $\{\text{Max}, \text{Min}\}$ be the $\{\text{largest}, \text{smallest}\}$ value on a maximal curve; Max-Min is its difference and $|\text{ANU} - \text{USTC}|$ is an absolute difference between ANU and USTC measures.

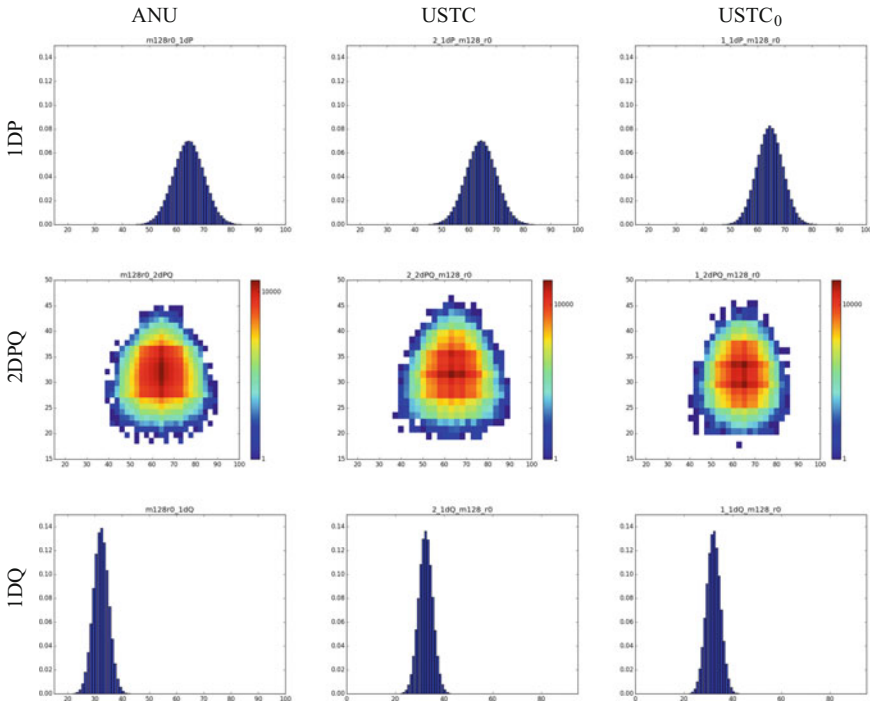


Fig. 2 ANU, USTC and USTC₀ random sequences on 1DP, 2DPQ, and 1DQ maps

Let $(Max - Min)/|ANU - USTC|$ be a relative ratio between (Max-Min) and $|ANU - USTC|$.

4 Result Analysis

Nine maps in Fig. 2 are in three columns. Three 1DP maps have similar distributions in bell shapes to illustrate Poissonian distributions. Three 2DPQ maps are 2D distributions and there are different symmetric distributions. Maximal elements in ANU, USTC, and USTC₀ maps show stronger vertical oriented features. Three maps have a symmetry on left/right directions and have a broken symmetry on up/down directions. Pseudo-color pixels on three maps are shown in 3D shapes. Compared with three 1DP maps, three 1DQ maps have similar distributions and more narrow bell shapes to illustrate sub-Poissonian distributions.

Six groups of results on shift $r : \{0, 16, 32, 96, 112, 128\}$ are shown in Fig. 3a on the top columns and each group contains nine distributions in three columns. Three random sequences have stronger stationary randomness that makes all maps in the similar style with minor changes on shift operations. Larger maps on $r = \{48, 64, 80\}$

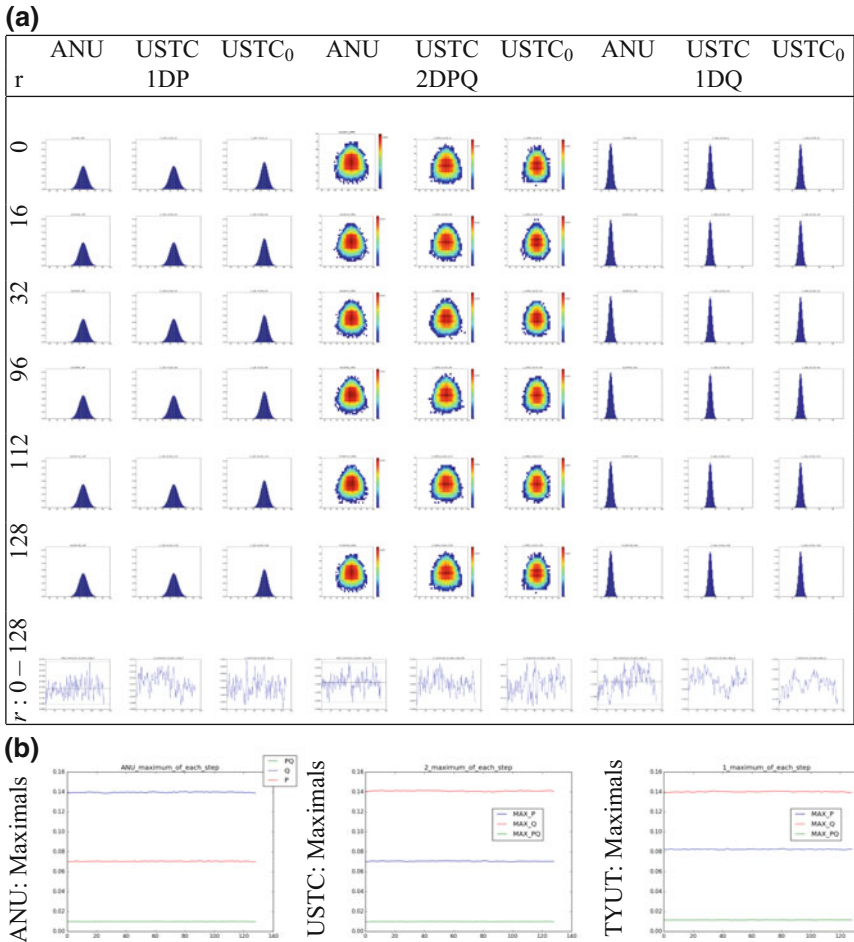


Fig. 3 ANU, USTC and USTC₀ random sequences on three maps and maximals **(a), (b)**: **a** Three pairs of nine variant maps in six groups and three pairs of nine maximal maps; **b** Three 2D maps of three maximal curves for ANU, USTC, and USTC₀

in Fig. 4a–c provide refined visual information to show their variations in details. Enlarged and larger maximal curves are shown in Figs. 3b and 5 for $r : 0 - 128$ as nine 2D maps with values of average variation and region of variations. From the maximal and minimal stationary regions, there are 1–2% variation ratios for 1DP and 1DQ and 5% variation ratios for 2DPQ observed. Three curves of maximals on three 2D maps are illustrated in Figs. 3b and 5.

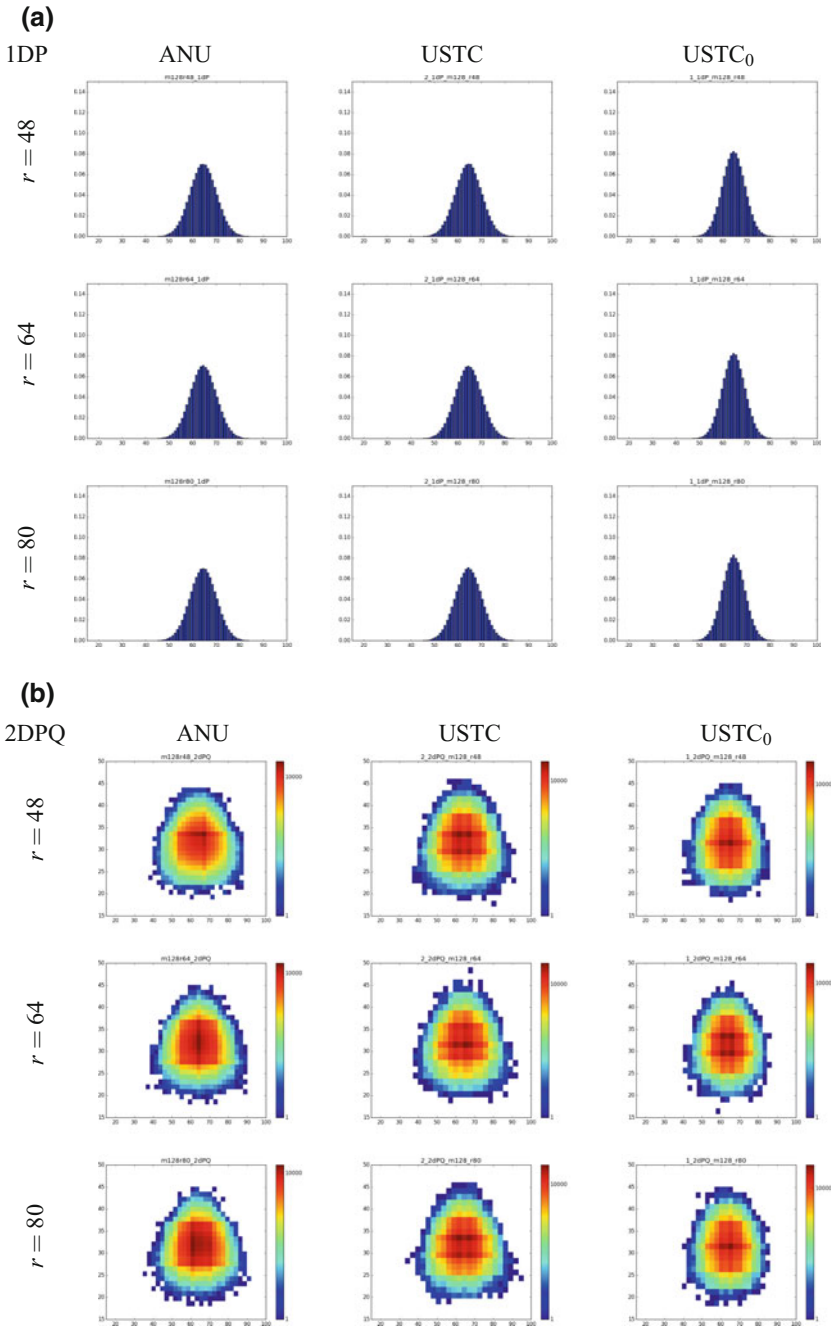


Fig. 4 ANU, USTC, and USTC₀ random sequences random sequences on enlarged maps, $r = \{48, 64, 80\}$; **a** 1DP; **b** 2DPQ; **c** 1DQ

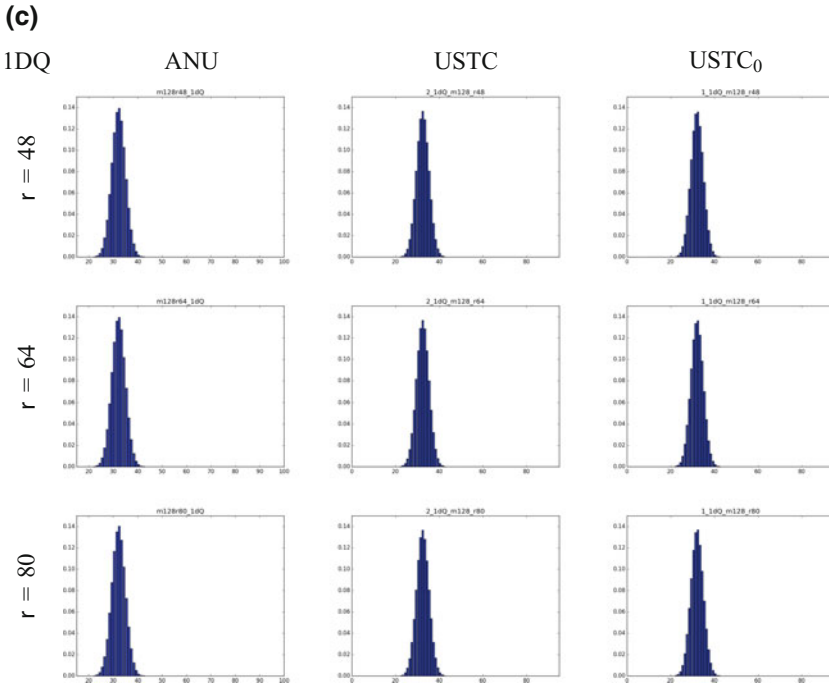


Fig. 4 (continued)

4.1 Relative Ratios on Differences

Details of three maximal measures are compared in Table 1. Three parameters $\{Q_x, \Delta Q_x, Q_x^R\}$ on 1DQ maps have 1 values on Max-Min and $|ANU - USTC|$ ratios; there are 81 on P_x and 1.6 on P_x^R and there are 65 on PQ_x and 7.9 on PQ_x^R observed.

From this set of testing results, two samples of ANU and USTC are showing similar stationary properties and USTC₀ with different stationary properties among the three sequences. Significant differences of relative ratios are observed from 2DPQ variation measurements.

Fig. 5 Three enlarged 2D maps of three maximal curves for ANU, USTC, and USTC₀

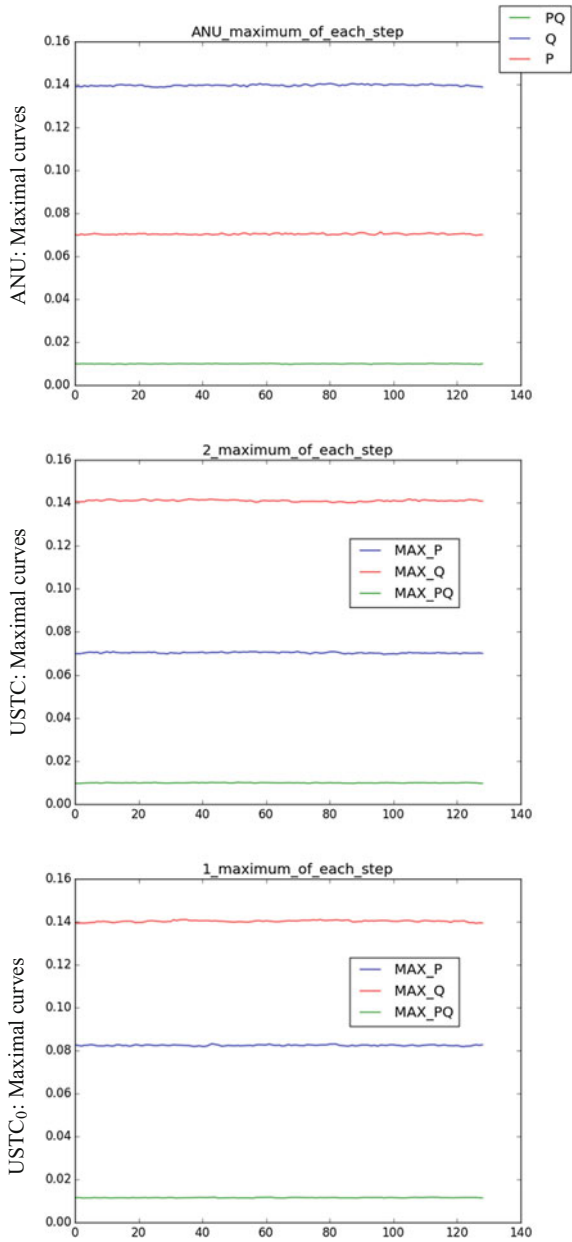


Table 1 Comparisons on three measures for ANU, USTC, and USTC₀ samples

	Q_x	ΔQ_x	Q_x^R
ANU:	13.961%	0.17761%	1.2722%
USTC:	14.09%	0.18%	1.27%
USTC ₀ :	14.02%	0.18%	1.27%
Min:	13.961%	0.17761%	1.27%
Max:	14.09%	0.18%	1.2722%
Max–Min:	0.129%	0.0239%	0.0022%
$ ANU - USTC $:	0.129%	0.0239%	0.0022%
$(Max - Min)/ ANU - USTC $:	1	1	1
	P_x	ΔP_x	P_x^R
ANU:	7.0352%	0.15472%	2.1992%
USTC:	7.05%	0.13%	1.87%
USTC ₀ :	8.24%	0.14%	1.68%
Min:	7.0352%	0.13%	1.68%
Max:	8.24%	0.15472%	2.1992%
Max–Min:	1.2048%	0.02472%	0.5192%
$ ANU - USTC $:	0.0148%	0.02472%	0.3292%
$(Max - Min)/ ANU - USTC $:	81	1	1.6
	PQ_x	ΔPQ_x	PQ_x^R
ANU:	0.99245%	0.04791%	4.8276%
USTC:	0.99%	0.05%	5.01%
USTC ₀ :	1.15%	0.05%	3.56%
Min:	0.99%	0.04691%	3.56%
Max:	1.15%	0.05%	5.01%
Max–Min:	0.16%	0.00209%	1.45%
$ ANU - USTC $:	0.00245%	0.00209%	0.1824%
$(Max - Min)/ ANU - USTC $:	65	1	7.9

5 Conclusion

It is feasible to evaluate stationary randomness for a random sequence using the testing system. From three maps {1DP, 1DQ, 2DPQ}, maximals are identified for shift $r : 0 - m$. Three 2D maps of maximal curves provide refined characteristics to evaluate stationary randomness. Further explorations and applications are required to check the testing system on other applications.

Acknowledgements Thanks to the Key project of Quantum Communication of Yunnan Province, National Science Foundation of China (61362014) and High-Level Overseas Professional Project of Yunnan Province for financial supports to this project. Thanks to the Key Laboratory of Quantum Information, USTC, CAS, and the ANU Quantum Optical Laboratory for providing quantum random sequences.

References

1. Quantum key distribution, https://en.wikipedia.org/wiki/Quantum_key_distribution
2. Random number generation, https://en.wikipedia.org/wiki/Random_number_generation
3. S. Golomb, *Shift-Register Sequences*, Revised edn. (Aegean Park Press, Laguna Hills, California, 1982)
4. de A. Queiroz, J. Schechtman, Elimination of nonlinear clock feedthrough in component-simulation switched-current circuits. in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, 1998. ISCAS '98* (II378-II381, 1998)
5. D. Meschede, in *Optics, Light and Lasers*, 2nd edn. (Wiley-VCH, 2007)
6. M. Nakazawa et al., QAM quantum stream cipher using digital coherent optical transmission. *Opt. Expr.* **22**(4), 4098–4107 (2014)
7. S. Lian et al., A chaotic stream cipher and the usage in video protection. *Chaos Solitons Fractals* **34**(3), 851–859 (2007)
8. NIST, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (NIST, Special Publication, 2010)
9. K. Ito, Gaussian filter for nonlinear filtering problems, in *Conference on decision and control*, pp. 1218–1223 (2000)
10. F. Orieux, O. Feron, J. Giovannelli, Sampling high-dimensional Gaussian distributions for general linear inverse problems. *IEEE Signal Process. Lett.* **19**(5), 251–254 (2012)
11. M. Fox, *Quantum Optics: An Introduction* (Oxford University Press, New York, 2006)
12. M.B. Priestley, in *Non-linear and Non-stationary Time Series Analysis* (Academic Press, 1988)
13. N. Wiener, *Time Series* (Cambridge, M.I.T Press, 1964)
14. Stationary process, https://en.wikipedia.org/wiki/Stationary_process
15. A.E. Ivanova, Using optical splitters in quantum random number generators based on fluctuations of vacuum. *J. Phys., Conf. Ser.* **735**, 012077 (2016)
16. X.T. Song, Phase-coding self-testing Quantum random number generator. *Chin. Phys. Lett.* **32**(8), 080302–080310 (2015)
17. T. Symul, S.M. Assad, P.K. Lam, Real time demonstration of high bitrate quantum random number generation with coherent laser light. *Appl. Phys. Lett.* **98**, 231103 (2011)
18. Quantum random number generator, <http://photonics.anu.edu.au/qoptics/Research/qrng.php>
19. W. Chen, Active phase compensation of quantum key distribution system. *Chin. Sci. Bull.* **53**(9), 1310–1314 (2008)
20. D.E. Knuth. *The Art of Computer Programming*, vol 4A: Combinatorial Algorithms Part 1 (Addison-Wesley, 2011)
21. Z.J. Zheng, *Conjugate Transformation of Regular Plan Lattices for Binary Images*, Ph.D. Thesis (Monash University, 1994)
22. J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* **5**(2), 163–172 (2010) Higher Educational Press and Springer, <https://doi.org/10.1007/Fs11460-010-0011-4>
23. H. Wang, J. Zheng, 3D visual method of variant logic construction for random sequence, in *Australian Information Warfare and Security*, pp. 16–27 (2013)
24. J. Zheng, Novel Pseudo-Random number generation using variant logic framework, in *2nd International Cyber Resilience Conference*, 10bit04 (2011). <http://igneous.scis.ecu.edu.au/proceedings/2011/icr/zheng.pdf>

25. J. Zheng, C. Zheng, T.L. Kunii, Interactive maps on variant phase space, in *Emerging Application of Cellular Automata*, (InTech Press, 2013) pp. 113–196
26. D.M. Heim, O. Heim, P.A. Zeng, J. Zheng, Successful creation of regular patterns in variant maps from bat echolocation calls. *Biol. Syst.: Open Access* 5, 2 (2016). <https://doi.org/10.4172/2329-6577.1000166>
27. J. Zheng, C. Zheng, Stationary randomness of quantum cryptographic sequences on variant maps, in *Proceedings on ASONAM '17*, (ACM, 2017). ISBN 987-1-4503-4993-2/17/07, <https://doi.org/10.1145/3110025.3110151>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Using Information Entropy to Measure Stationary Randomness of Quantum Random Sequences



Weizhong Yang, Yamin Luo, Zhefei Li and Jeffrey Zheng

Abstract Different statistical measurements can be used to determine stationary randomness for random sequences. This chapter proposes a testing scheme for random sequences using information entropy as measurements. Datasets are collected from University of Science & Technology of China (USTC), three quantum random sequences are selected for testing. Multiple results are created on three maps, entropy curves, and quantitative measurements of stationary randomness are compared. Three differences of Max-Min entropy variation ratios are bounded in $[0.08, 0.09]\%$ region. The whole structure has measurable stationary properties.

Keywords Variant maps · Quantum random sequence · Ordered measures Entropy · Stationary randomness

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

W. Yang
Shanghai Key Laboratory of Intelligent Information Processing,
School of Computer Science, Fudan University, Shanghai, China

W. Yang
Key Laboratory of Quantum Information of Yunnan, School of Software,
Yunnan University, Kunming, China
e-mail: yangweizhong@126.com

Y. Luo · Z. Li · J. Zheng (✉)
Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

Y. Luo
e-mail: 1047668416@qq.com

Z. Li
e-mail: 576167164@qq.com

1 Introduction

From a statistical viewpoint, various parameters of statistical process [2–4, 7] could be stationary invariant [6] under shift operations on random sequences. Using variant maps [8], it is a normal approach to transfer a long random sequence into 1D and 2D statistical distributions as three maps: 1DP, 1DQ, and 2DPQ [9]. For each map, it is easy to divide each number by the total number to transfer a counting number into a probability measure. By this way, three sets of probability measures can be generated. Applying information entropy function to summarize all pairs of probability parameters, one map corresponds an information entropy measurement determined by the distribution for stationary randomness.

2 Test Methodology

The test for a stationary randomness requires a sequence with length N . For the given input sequence, multiple segments M are divided from the sequence by a given length m , a 2-tuple pair of measures can be extracted from a 0-1 segment that are the number of 1 element and the number of 1 pattern in the segment. All paired measures are composed of a sequence of M pairs of measures as an ordered measuring set with M elements.

The pairs of the measuring sequence are directly separated as two independent measuring sequences to keep each parameter in the same order. A total of three sequences of distinct measures are constructed including two sequences on single measures and one sequence on 2-tuple measures.

Following this approach, two sets of single measuring sequences are sorted as two 1D numeric arrays as statistical histograms corresponding to 1D maps and the 2-tuple measuring sequence is sorted as a 2D integer array as statistic histograms being a 2D map. Under the controlling operations on the changes of shift displacement, multiple results of the three measuring sequences are transformed into 1D statistic histograms and 2D pseudo-color maps to show effective patterns from the generated sequence under various positions and conditions on a list of shift operations.

2.1 Dataset

2.1.1 USTC Resource

In the Key Laboratory of Quantum Information, USTC, CAS, and quantum random number sequences are generated [5]. This type of true random sequences supports advanced quantum communication devices of QKD systems [1].

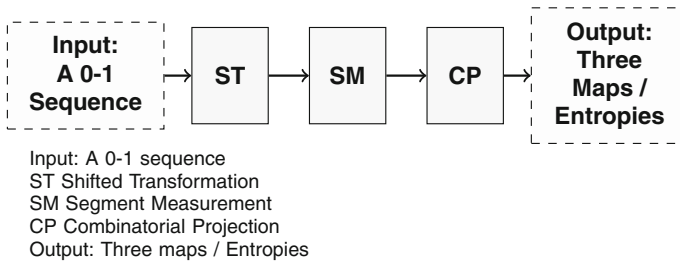


Fig. 1 Methodology for information entropy testing stationary random sequences

More than 20 GB of quantum random number sequences are provided by USTC for random streams testing. Three sequences from eight sequences are selected from three stages (1 Initial, 2 Secure, and 4 Filtered). Each random sequence has a length of about 8 MB.

3 Method

3.1 Methodology

This method consists of five steps (Fig. 1): Input, Shifted Transformation (ST), Segment Measurement (SM), Combinatorial Projection (CP), and Output.

The input of the testing system is a selected 0-1 sequence and its output is composed of three maps, two in 1D and one in 2D for visual distributions, and three maximals to be processed by ST, SM, and CP.

3.2 Description of Steps

The testing system consists of three steps: {ST, SM CP}.

Input: X $N = m * M$ bit sequence; m segment length; M total segments; r shift length;

Output: Three maps {1DP, 1DQ, 2DPQ}; Three Maximals {1DP_x, 1DQ_x, 2DPQ_x}

Process: Shifting r position from X to be $Y = X(r)$ in ST. Making segment measuring sequences in SM and then projecting three measuring sequences as three maps and extracting three maximals in CP.

Let X, Y be 0-1 sequences with N elements, ST takes the sequence X as input, then shift r position on the whole sequence to be the shifted sequence $Y = X(r)$ (i.e., a cyclic shift right + or shift left -).

$$\begin{aligned}
 Y &= X(r), Y[I] = X[I \pm r], I \pm r \pmod{N}, \\
 0 &\leq I < N; X[I], Y[I] \in \{0, 1\}
 \end{aligned}
 \tag{1}$$

SM takes the shifted vector as input and divides the vector into M segments. For the i th sub-vector $0 \leq i < M$ on the j th position $0 \leq j < m$, denoted as $Y_{i,j}$.

This sequence at the end of sub-vectors after the segmenting operation forms an $m * M$ matrix, m positions for the i th complete row vector in the sequence correspond to a pair of 2-tuple measures: (p_i, q_i) .

$$Y = \{Y_i\}_{i=0}^{M-1} \tag{2}$$

$$\begin{aligned}
 Y_i &= \{Y_{i,0}, Y_{i,1}, \dots, Y_{i,j}, \dots, Y_{i,m-1}\} \\
 0 &\leq i < M, 0 \leq j < m
 \end{aligned}
 \tag{3}$$

$$Y_i \rightarrow (p_i, q_i), 0 \leq i < M \tag{4}$$

$$\{Y_i\}_{i=0}^{M-1} \rightarrow \{(p_i, q_i)\}_{i=0}^{M-1} \tag{5}$$

The pair of 2-tuple measures (p_i, q_i) is determined by the following formula:

$$Y_{i,j} = Y[J] \in \{0, 1\}; J = i * m + j, \tag{6}$$

$$0 \leq i < M, 0 \leq j < m, 0 \leq J < m * M$$

$$p_i = \sum_{j=0}^{m-1} Y_{i,j}, Y_{i,j} \in \{0, 1\}, 0 \leq p_i \leq m; \tag{7}$$

$$q_i = \sum_{j=0}^{m-1} [(Y_{i,j-1}, Y_{i,j}) == (0, 1)], \tag{8}$$

$$j - 1 \pmod{m}, 0 \leq q_i \leq \lfloor m/2 \rfloor;$$

That is, $X = 0011010010, N = 10, M = 2, m = 5; (p_0 = 2, q_0 = 1); (p_1 = 2, q_1 = 2)$.

The output from SM are M pairs of ordered 2-tuple measures $\{(p_i, q_i)\}_{i=0}^{M-1}$.

CP consists of Split and Projection steps. Split adapts the 2-tuple measuring sequence $\{(p_i, q_i)\}_{i=0}^{M-1}$, splitting it into two independent measuring sequences: $\{p_i\}_{i=0}^{M-1}, \{q_i\}_{i=0}^{M-1}$ to keep the original order invariant.

The Three measure sequences are $\{p_i\}_{i=0}^{M-1}, \{q_i\}_{i=0}^{M-1}, \{(p_i, q_i)\}_{i=0}^{M-1}$.

The Projection step turns the sequence into histograms: Project Array (PA), Color Map (CM), and Get Entropy (GE). For three measuring sequences, two types of 1D and 2D measures will be processed separately.

The PA processes measuring sequences to transform them into integer arrays and the CM will organize them on either normalized histograms (1D measures) or color maps (2D measures), respectively.

The 1D measures involve two measuring sequences: $\{p_i\}_{i=0}^{M-1}$, $\{q_i\}_{i=0}^{M-1}$. Let $P[m+1]$, $Q[\lfloor m/2 \rfloor + 1]$ and $NP[m+1]$, $NQ[\lfloor m/2 \rfloor + 1]$ be two 1D (integer, float) arrays to represent the corresponding elements.

The 1DP statistic histogram is generated from a sequence $\{p_i\}_{i=0}^{M-1}$, NP , P two arrays (floating point, integer) with $(m+1)$ elements. For the j th element $NP[j]$, $P[j]$, $0 \leq j \leq m$, and $1DP_e$ the entropy element, the output can be obtained by the following procedure:

Initialization: $\forall NP[j] = 0.0$,
 $P[j] = 0, 0 \leq j \leq m$;
 Calculation: *for* ($i = 0; i < M; i++$)
 $\{P[p_i]++;\}$
 Normalization: *for* ($j = 0; j \leq m; j++$)
 $\{NP[j] = P[j]/M;\}$
 Get Entropy: $1DP_e = -\sum_{i=0}^m NP[j] * \log_2(NP[j])$

In the 1DP map, the PA corresponds to Initialization and Calculation; the MA handles Normalization and the GE determines the entropy element of the map.

The 1DQ statistic histogram is generated from a sequence $\{q_i\}_{i=0}^{M-1}$, NQ , Q two arrays (floating point, integer) with $(\lfloor m/2 \rfloor + 1)$ elements; For the j th element $NQ[j]$, $Q[j]$, $0 \leq j \leq \lfloor m/2 \rfloor$, and $1DQ_e$ the entropy element, the output can be obtained from the following procedure:

Initialization: $\forall NQ[j] = 0.0$,
 $Q[j] = 0, 0 \leq j \leq \lfloor m/2 \rfloor$;
 Calculation: *for* ($i = 0; i < M; i++$)
 $\{Q[q_i]++;\}$
 Normalization: *for* ($j = 0; j \leq \lfloor m/2 \rfloor; j++$)
 $\{NQ[j] = Q[j]/M;\}$
 Get Entropy: $1DQ_e = -\sum_{j=0}^{\lfloor m/2 \rfloor} NQ[j] * \log_2(NQ[j])$

Using P , NP , Q , NQ arrays, it is possible to generate corresponding 1D statistical histograms as 1D maps.

In the 1DQ map, the PA corresponds to Initialization and Calculation; the MA handles Normalization and the GE identifies the entropy element of the map.

The 2D measures specially processes one measuring sequence: $\{(p_i, q_i)\}_{i=0}^{M-1}$. Let PQ , NPQ be two 2D (integer, float) arrays.

A 2DPQ statistic histogram is generated from a sequence $\{(p_i, q_i)\}_{i=0}^{M-1}$, PQ , NPQ 2D arrays with $(m+1) \times (\lfloor m/2 \rfloor + 1)$ elements. For the i, j th element $PQ[i, j]$, $NPQ[i, j]$, $0 \leq i \leq m$, $0 \leq j \leq \lfloor m/2 \rfloor$, and $2DPQ_e$ the entropy element, their values can be obtained by the following procedure:

Initialization: $\forall PQ[i, j] = 0,$
 $0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor;$
 Calculation: $for(i = 0; i < M; i++)$
 $\{PQ[p_i, q_i] ++;\}$
 Pseudo-color: Matching proper color for
 $\forall PQ[i, j], 0 \leq i \leq m, 0 \leq j \leq \lfloor m/2 \rfloor$
 Normalization: $for(j = 0; j \leq m; j++)\{$
 $for(j = 0; j \leq \lfloor m/2 \rfloor; j++)\{$
 $\{NPQ[i, j] = PQ[i, j]/M;\}$
 Get Entropy: $1DPQ_e = - \sum_{j=0}^{\lfloor m/2 \rfloor} \sum_{i=0}^m NPQ[i, j] * \log_2(NPQ[i, j])$

In the 2DPQ map, the PA corresponds to Initialization and Calculation; the MA handles Pseudo-color, Normalization and the GE identifies the entropy element of the map.

Through the CP module, three measuring sequences are transformed into two 1D arrays and one 2D array with $(m + 1)$, $(\lfloor m/2 \rfloor + 1)$ and $(m + 1) \times (\lfloor m/2 \rfloor + 1)$ clusters.

The output of the testing system are three maps $\{1DP, 1DQ, 2DPQ\}$ and three entropies $\{1DP_e, 1DQ_e, 2DPQ_e\}$ as expected statistic distributions and representatives of the input 0-1 sequence, respectively.

4 Results

Three quantum random sequences are selected from USTC $\{1, 2, 4\}$ streams.

Typical results of testing stationary properties for three sequences in nine maps are shown in Fig. 2. Top part contains three 2D maps of global entropy curves on $r = 0 - 128$ condition. Three 2D maps of entropy curves for $r = 0 - 128$ are shown to illustrate refined properties in stationary random curves. Three sets of variant maps in $r = 0$ and their enlarged entropy curves on $r = 0 - 128$ are shown in three columns to illustrate corresponding 1DP, 1DQ, and 2DPQ maps for three sequences. Three larger maps of three global entropy curves are shown in Fig. 3.

For a G map, let G_e be an average entropy variation, ΔG_e be a region of entropy variations, and $G_e^R = \Delta G_e / G_e$ be an entropy variation ratio. Three entropy curves on three 2D maps are compared. Three entropy measurements and $\{\text{Max}, \text{Min}, \text{Max-Min}\}$ values for three sequences are listed in Table 1. Three variation ratios and their numeric quantities are listed in Table 2.

5 Result Analysis

Three 2D maps of global entropy curves show stronger stationary randomness under shift operations on $r = 0 - 128$. Three entropy curves on each map are three stable

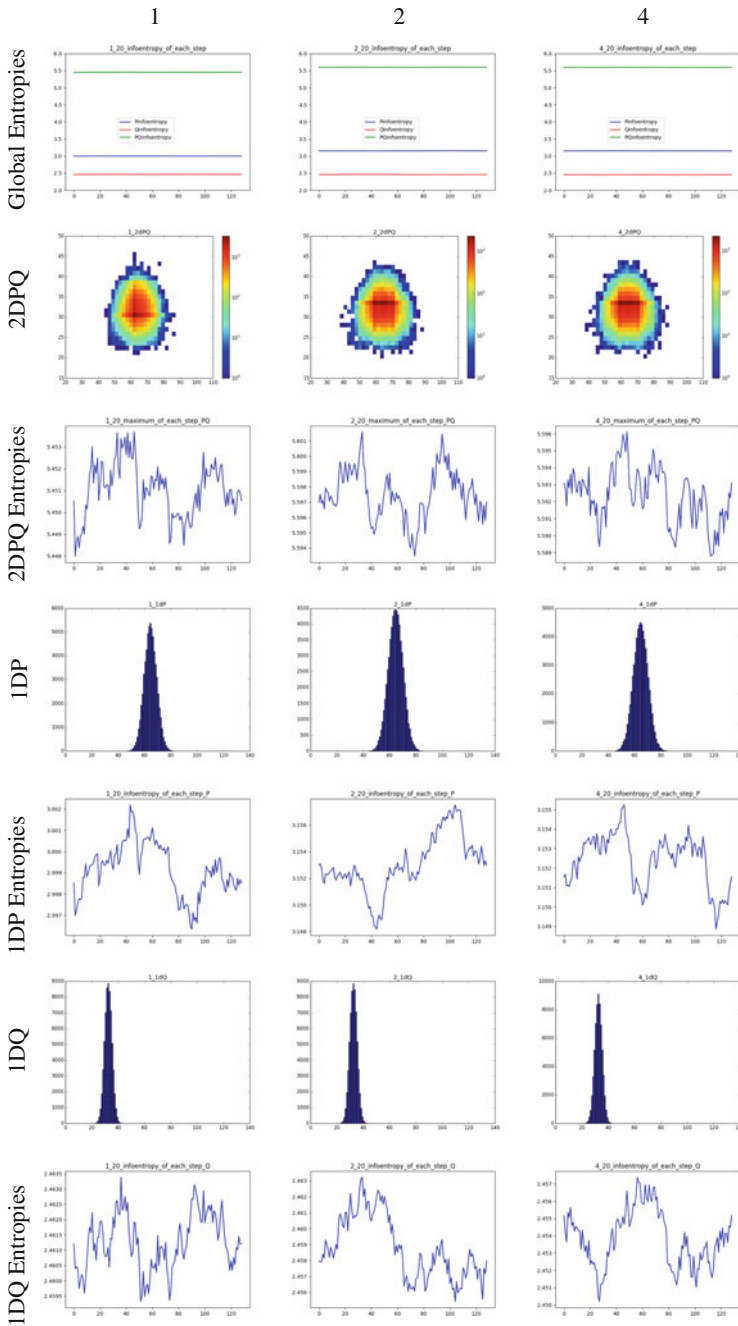


Fig. 2 Three USTC random sequences: {1, 2, 4} on 2DPQ, 1DP, and 1DQ maps and $r = 0 - 128$ entropy curves

Fig. 3 Three enlarged 2D maps of global entropy curves for three USTC random sequences

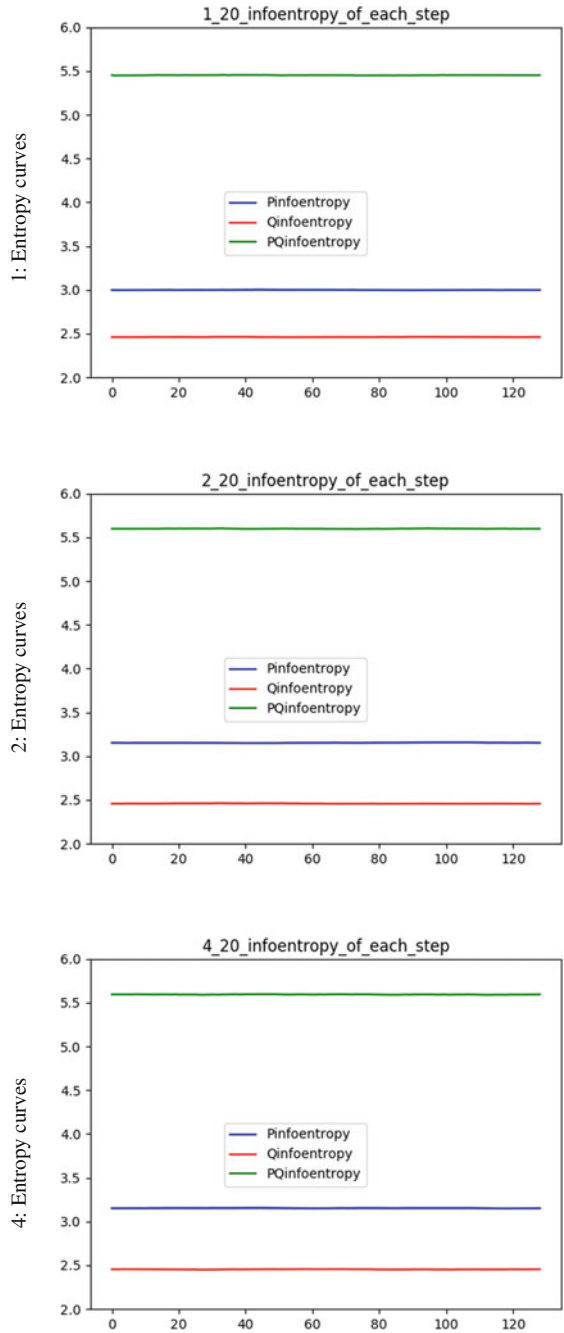


Table 1 Comparisons on three measures for three USTC samples

	Q_e	ΔQ_e	Q_e^R
1:	2.4537	0.63%	0.2559%
2:	2.4638	0.83%	0.335%
4:	2.46	0.86%	0.3481%
Min:	2.4537	0.63%	0.2559%
Max:	2.4638	0.86%	0.3481%
Max–Min:	0.0101	0.23%	0.0922%
	P_e	ΔP_e	P_e^R
1:	2.9948	0.82%	0.2742%
2:	3.1502	0.84%	0.267%
4:	3.1472	0.61%	0.1937%
Min:	2.9948%	0.61%	0.1937%
Max:	3.1502%	0.84%	0.2742%
Max–Min:	0.1554%	0.23%	0.0805%
	PQ_e	ΔPQ_e	PQ_e^R
1:	5.4397	0.81%	0.1481%
2:	5.5998	1.14%	0.2035%
4:	5.5932	0.66%	0.1183%
Min:	5.4397	0.66%	0.1183%
Max:	5.5998	1.14%	0.2035%
Max–Min:	0.1601	0.48%	0.0852%

Table 2 $Q_e + P_e : PQ_e$ measures

No.	Q_e	P_e	$Q_e + P_e$	PQ_e	$\Delta_e = Q_e + P_e - PQ_e $	PQ_e / Δ_e
1	2.4537	2.9948	5.4485	5.4397	0.0088	618
2	2.4638	3.1502	5.614	5.5998	0.0142	394
4	2.46	3.1472	5.6072	5.5932	0.014	400

horizontal lines. From a global viewpoint, there are significant differences compared with entropy curves between No. 1 (PQ and P) and No. 2 & 3 cases. Both No. 2 and 3 are in similar measures.

Nine variant maps in 2DPQ, 1DP, and 1DQ, three 2DPQ maps are 2D distributions and there are different symmetric distributions. Maximal elements in three maps show stronger vertical-oriented features. Three maps have a symmetry on left/right directions and have a broken symmetry on up/down directions. Pseudo-color pixels on three maps are shown in 3D shapes. Three 1DP maps have similar distributions in bell shapes to illustrate Poissonian distributions. Compared with three 1DP maps, three 1DQ maps have similar distributions and more narrow bell shapes to illustrate sub-Poissonian distributions.

However, nine enlarged entropy curves for each type have significantly different variations and distributions. Local curves are bounded in narrow regions with random variations.

It is difficult to tell detailed differences from entropy curves. Quantitative measurements in Table 1 are helpful to use numeric values in comparison. The difference of entropy variation ratios are on three sets, Q_e^R : [0.26, 0.35]%, P_e^R : [0.19, 0.27]%,

and PQ_e^R : [0.12, 0.20]%. Three Max-Min values of $\{Q_e^R, P_e^R, PQ_e^R\}$ are bounded in [0.08, 0.09]%. The whole structure illustrates measurable stationary properties. In Table 2, it is interesting to notice that $Q_e + P_e \sim PQ_e$.

All variation measurements are shown in distinct stationary randomness to be measured by entropy approaches.

6 Conclusion

Information entropy is a useful measurement to determine stationary randomness. Three quantum random sequences are used, distinct stationary randomness can be identified from both variant maps and numeric measurements. To explore various conditions of stationary properties, further investigations are required to explore theoretical boundaries on variant maps.

Acknowledgements Thanks to The Key project of Quantum Communication of Yunnan Province, National Science Foundation of China (61362014) and High-Level Overseas Professional Project of Yunnan Province for financial supports to this project. Thanks to the Key Laboratory of Quantum Information, USTC, and CAS for providing quantum random sequences.

References

1. W. Chen, Active phase compensation of quantum key distribution system. *Chin. Sci. Bul.* **53**(9), 1310–1314 (2008)
2. D.E. Knuth, in *The Art of Computer Programming*, vol. 4A, Combinatorial Algorithms Part 1 (Addison-Wesley, 2011)
3. NIST, in *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* (NIST, Special Publication, 2010)
4. M.B. Priestley, in *Non-linear and Non-stationary Time Series Analysis* (Academic Press, 1988)
5. X.T. Song, Phase-coding self-testing Quantum random number generator. *Chin. Phys. Lett.* **32**(8), 080302–080310 (2015)
6. Stationary process, https://en.wikipedia.org/wiki/Stationary_process
7. W.Z. Yang, J. Zheng, Variant Pseudo-random number generator. *Hakin9 Extra Timing Attack* **06**(13), 28–31 (2012)
8. J. Zheng, C. Zheng, T.L. Kunii, Interactive Maps on Variant Phase Space, in *Emerging Application of Cellular Automata* (InTech Press, 2013), pp. 113–196
9. J. Zheng, C. Zheng, Stationary randomness of quantum cryptographic sequences on variant maps, in *Proceedings on ASONAM '17*, (ACM, 2017). ISBN 987-1-4503-4993-2/17/07 <https://doi.org/10.1145/3110025.3110151>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Visual Maps of Variant Combinations on Random Sequences



Jeffrey Zheng and Jie Wan

Abstract Random sequences play the key role in network security applications. Randomness testing schemes are very important to ensure the randomness qualities for relevant sequences. This chapter proposes a visual scheme based on variant construction to measure sequences to intuitively show some combinatorial properties of key stream generated by stream ciphers. Basic models are described. This scheme provides a flexible framework for the variant measure method on the key stream of stream ciphers to describe randomness in various combinatorial maps.

Keywords Visual scheme · Variant measure · Combinatorial projection
Random sequence

1 Introduction

Random numbers play an important role in many network protocols and encryption schemas on various network security applications [1], for example, visual crypto, digital signatures, authentication protocols and stream ciphers. To determine whether a random sequence is suitable for a cryptographic application, the NIST has published a series of statistical tests as standards.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

J. Wan

The People's Bank of China, Kunming, China
e-mail: wanjiech@163.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_22

333

In network security applications, the stream ciphers play a key role that have faster throughput and easier to implement compared to block ciphers [2]. RC4, the famous stream cipher, is suitable for large packets in Wireless LANs [3]. It has been used for encrypting the internet traffic in network protocols such as Sockets Layer (SSL), Transport Layer Security (TLS), Wi-Fi Protected Access (WPA), etc. [2].

eSTREAM project collected stream ciphers from international cryptology society [4] to promote the design of efficient and compact stream ciphers suitable for widespread adoptions. After a series of tests, algorithms submitted to eSTREAM are selected into two profiles. One is more suitable for software and another one is more suitable for hardware. Non-linear structures and recursive are playing the essential roles in new development.

Different visual schemes are required to test randomness of random sequences on different stream ciphers. Along this direction, this chapter proposes a flexible framework to handle a set of mete measurements on different combinatorial projections.

2 Variant Combinatorial Visualization

Architecture of variant visualization is shown in Fig. 1.

The variant visualization architecture is separated into four core components: EAC, SCC CC and VC.

- RGC Randomness Generate Component generate a random sequence;
- VSC Variant Statistic Component handles the statistic process using the variant measure method [5];
- CC Combinatorial Component chooses combinations;
- VC Visualization Component makes visualization based on SCC measures and CC vectors.

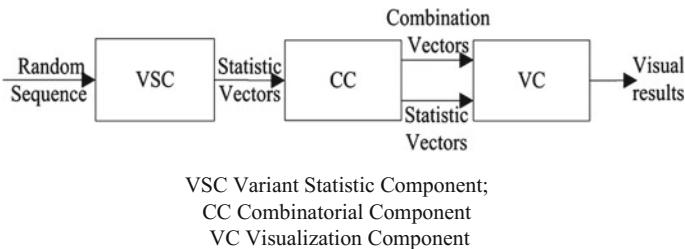


Fig. 1 Visualization architecture

The input n is the length of the binary sequence. The stream ciphers could be changed to any stream cipher that can generate binary sequence. This section focuses on the variant measure module and the visual method module.

A visual example of RC4 will be described in Sect. 2.5.

2.1 Variant Logic Framework

The variant logic framework has been proposed in [6]. Li [7] used the variant measure method to generate different symmetry results [5] based on cellular automata schemes [8]. Under such construction, even some random sequences show symmetry properties in distributions.

Under variant construction, the variant conversion operator can be defined as follows:

$$C(x, y) = \begin{cases} \perp, & x = 0, y = 0 \\ +, & x = 0, y = 1 \\ -, & x = 1, y = 0 \\ \top, & x = 1, y = 1 \end{cases} \tag{1}$$

It is convenient to list relevant variant logic variables shown in Table 1.

In the variant measure method, each sequence is converting from binary sequence to probability which generated by counting the number of each variable in $\{\perp, +, -, \top\}$ and computes the probability of each variable. The measurement method is shown in Table 1.

Table 1 The variant measure method

(a) Counting method			(b) Probability computing	
Variant variable	Number of type	Total number	Measure parameters	Number of type
\perp	N_{\perp}	$N = N_{\perp} + N_{\top} + N_{+} + N_{-}$	P_{\perp}	N_{\perp}/N
\top	N_{\top}		P_{\top}	N_{\top}/N
$+$	N_{+}		P_{+}	N_{+}/N
$-$	N_{-}		P_{-}	N_{-}/N

The variant measure method provides a set of results in measures of different 0–1 sequences. The following mechanism can transfer stream cipher sequences as relevant measures.

The essential models of variant scheme are described as follows.

2.2 VSC Variant Statistic Component

The VSC component converts the binary sequence to variant sequence in VCM module, and to compute probabilities and entropies in PECM module, respectively. The component is shown in Fig. 2.

VCM Variant Conversion Module

VCM module transfers input binary sequences by following steps:

- Step 1. Generate an n bit binary sequence $S = S_1 S_2 S_3 \dots S_n$ by a stream cipher.
- Step 2. Shift X to left by M bit (M is the length of shifting) and generate a new binary sequence $S' = S'_1 S'_2 S'_3 \dots S'_{n-M} = S_{1+M} S_{2+M} \dots S_n$.
- Step 3. Convert two sequences: S and S' to a variant sequence $V = V_i = C(S_i, S'_i), i = 1, 2, 3 \dots (n - M)$.
- Step 4. Separate V into n/N parts. N is the length of each part and $M \leq N \leq n$ to form a set of variant sequence groups

$$G = \{G_1, G_2, \dots, G_{n/N}\}$$

$$= \{\{V_1, V_2, \dots, V_N\}, \dots, \{V_{n-N}, V_{n-N+1}, \dots, V_n\}\}$$

- Step 5. Separate each item in G into N/M parts to establish a sequence group

$$G = \{\{\{V_1, \dots, V_M\}, \dots, \{V_{N-M+1}, \dots, V_N\}\}, \dots, \{\{V_{n-N}, \dots, V_{n-N+M}\}, \dots, \{V_{n-M}, \dots, V_n\}\}\}$$

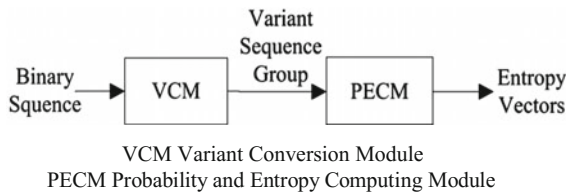


Fig. 2 Variant statistic component

PECM Probability and Entropy Computing Module

PECM converts a variant sequences group to separate it into several parts to compute probability and entropies. The equations computing the parameters have been described in Table 1. The main steps are performed as follows:

- Step 6. Compute the probability vector $P = \{P_{\perp}, P_{+}, P_{-}, P_{\top}\}$ of each part in G' ;
- Step 7. Calculate the distribute probability vector $D = \{D_{\perp}, D_{+}, D_{-}, D_{\top}\}$ of each part in G based on P vector;
- Step 8. Evaluate the entropy vector $\{E_{\perp}, E_{+}, E_{-}, E_{\top}\}$ from the D vector.

2.3 CC Combinatorial Component

In the CC component, it can be separated into two modules. One is SM module to form the vector selecting and another one is VDM module to perform the visualization.

Visual data is a set of E vectors as input for VC. For E vector, choose a projection as a visual vector to compute the visual result from E vectors. So there will be 16 visual results.

Base on the same number of variables in a combination, the combination set can be integrated into 5 parts. i.e. The selected number of variables in the combination is in 0-4.

Let the classification be $EC = \{EC_0, EC_1, EC_2, EC_3, EC_4\}$. Since the EC_0 is empty, it can be ignored. Only four distributions are of concern in Sect. 2.4.

2.4 Visualization Component

According to the variant measure method, in the rectangular axis, let E_{\perp} be the positive axis of X , E_{\top} be the negative axis of X , E_{+} the positive axis of Y , E_{-} be the negative axis of Y . The axis is shown in Fig. 3.

For $EC_1 = \{\{E_{\perp}\}, \{E_{+}\}, \{E_{-}\}, \{E_{\top}\}\}$, points are distributed to the axis.

For $EC_2 = \{\{E_{\perp}, E_{+}\}, \{E_{\perp}, E_{-}\}, \{E_{\perp}, E_{\top}\}, \{E_{+}, E_{-}\}, \{E_{+}, E_{\top}\}, \{E_{-}, E_{\top}\}\}$, points are distributed in the shadow area in Fig. 4.

For $EC_3 = \{\{E_{\perp}, E_{+}, E_{-}\}, \{E_{\perp}, E_{+}, E_{\top}\}, \{E_{\perp}, E_{-}, E_{\top}\}, \{E_{+}, E_{-}, E_{\top}\}\}$, points are distributed in the area of EC_1 and the area of EC_2 .

For $EC_4 = \{\{E_{\perp}, E_{+}, E_{-}, E_{\top}\}\}$, points are distributed in Fig. 5.

Fig. 3 Visualization axis

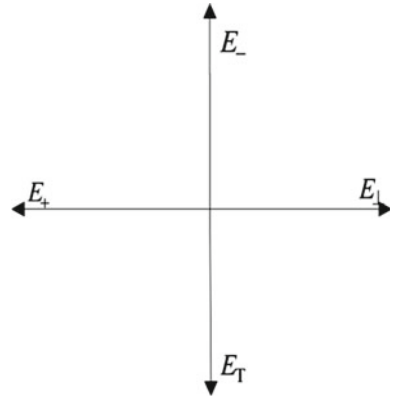


Fig. 4 Distribution areas of EC_2

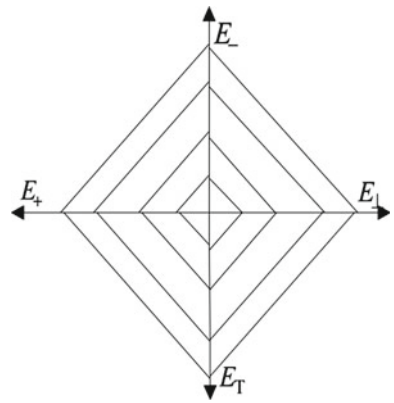
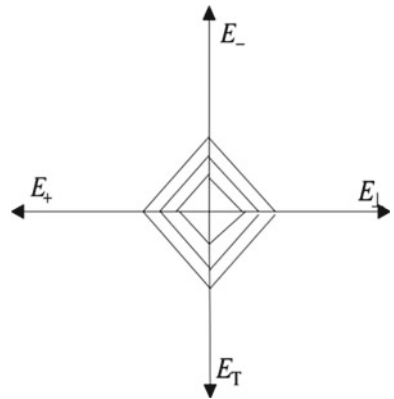


Fig. 5 Distribution areas of EC_4



2.5 Example

An example is given step by step to show how the algorithm runs. In the example, n , N and M are, respectively, assigned to 40, 16 and 8.

- Step 1. Input a 35 bit binary sequence, {010100101110101100101101011101101010101}
- Step 2. Generates S' , {11101011001011010111101101010101}.
- Step 3. Generates V , {+T+-+⊥T+--T⊥T+-T⊥+T+T+-T⊥T-T-+-T}.
- Step 4. Separate V into a G vector. The G vector is {{+T+-+⊥T+--T⊥T+-T}, {⊥+T+T+-T⊥T-T-+-T}}.
- Step 5. Separate the G into the G' vector. The G' vector in the example is {{+T+-+⊥T+, --T⊥T+-T}, {⊥+T+T+-T, ⊥T-T-+-T}}.
- Step 6. Generate probability vector P of each sequence in G' . The P vector of {+T+-+⊥T+} is { $P_{\perp} = 0.125$, $P_{+} = 0.5$, $P_{-} = 0.125$, $P_{T} = 0.25$ }.
- Step 7. Compute the distribute probability vector D of each sequence in G from P . The D vector of {+T+-+⊥T+, --T⊥T+-T} is shown in Fig. 6.
- Step 8. Compute the entropy vector E of each sequence in G from D . The E vector of {+T+-+⊥T+-T} is shown in Fig. 7.

$$\left\{ \begin{array}{l} D_{\perp} = \{P_{0.125} = 1 \\ \vdots \\ \vdots\} \\ D_{T} = \{P_{0.25} = 0.5, P_{0.725} = 0.5\} \end{array} \right\}$$

Fig. 6 D vectors of {+T+-+⊥T+, --T⊥T+-T}

$$\left\{ \begin{array}{l} E_{\perp} = (P_{0.125} \log P_{0.125} \quad \vdots \quad \vdots) = 0.0 \\ E_{T} = -(P_{0.25} \log P_{0.25} + P_{0.725} \log P_{0.725}) = 0.693147 \end{array} \right.$$

Fig. 7 E vectors of {+T+-+⊥T+-T}

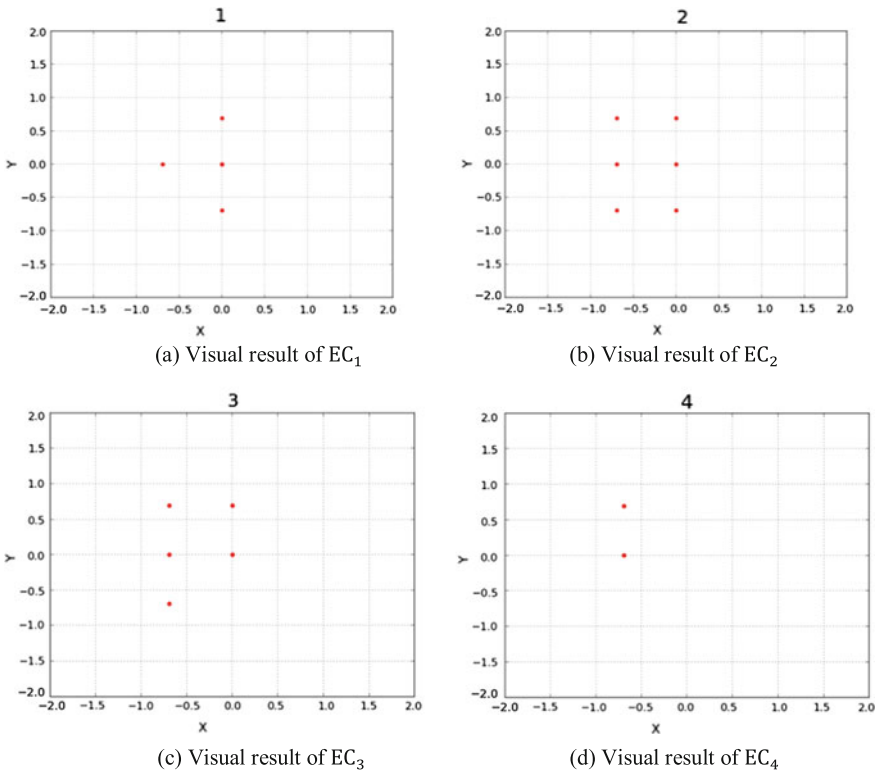


Fig. 8 Visual result of the example

- Step 9. Compute visual results from E vectors. In the E vectors of $\{+T + - + \perp T + - - \top \perp T + - \top\}$. If the selection is $\{E_{\perp}\}$, points will be $(0.0, 0.0)$. If the selection is $\{E_{\perp}, E_{\top}\}$, points will be $(0.0, -0.693147)$ and $(0.0, 0.0)$. If the selection is $\{E_{\top}, E_{-}\}$, points will be $\{E_{-} - |E_{\top}|\} = (0.0, 0.0)$ and $(0.0, 0.693147)$. If the selection is $\{E_{\perp}, E_{\top}, E_{-}\}$, points will be $\{E_{\perp}, E_{-} - |E_{\top}|\} = (0.0, 0.0)$ and $(0.0, 0.693147)$.
- Step 10. Separate visual results to EC classification. Visual results of the G in the example are shown in Fig. 8.

3 Result

3.1 Visual Result of RC4

The initial: {**n** : 128,000, **N** : 128, **M** : 16}

The visual result (Fig. 9).

The initial: {**n** : 128,000, **N** : 128, **M** : 24}

The visual result (Fig. 10).

The initial: {**n** : 128,000, **N** : 1000, **M** : 8}

The visual result (Fig. 11).

The initial: {**n** : 100,000, **N** : 100, **M** : 24}

The visual result (Fig. 12).

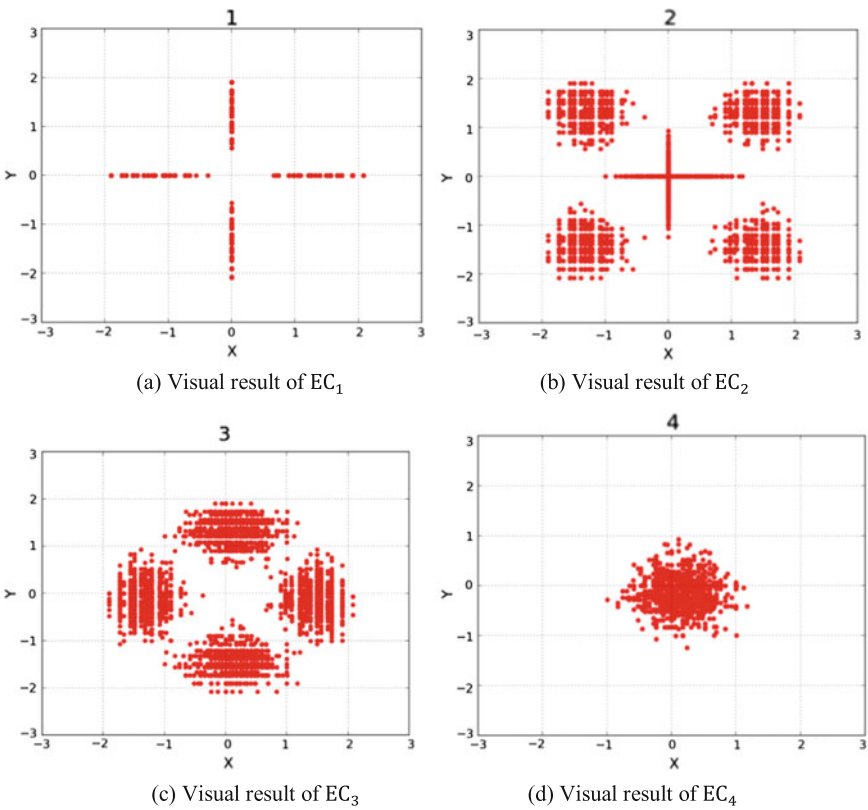


Fig. 9 Visual result of RC4 {**n** : 128000, **N** : 128, **M** : 16}

3.2 Visual Result of HC256

The initial: { n : 128,000, N : 128, M : 16}

The visual result (Fig. 13).

The initial: { n : 128,000, N : 128, M : 24}

The visual result (Fig. 14).

The initial: { n : 100,000, N : 100, M : 8}

The visual result (Fig. 15).

The initial: { n : 100,000, N : 100, M : 16}

The visual result: (Fig. 16).

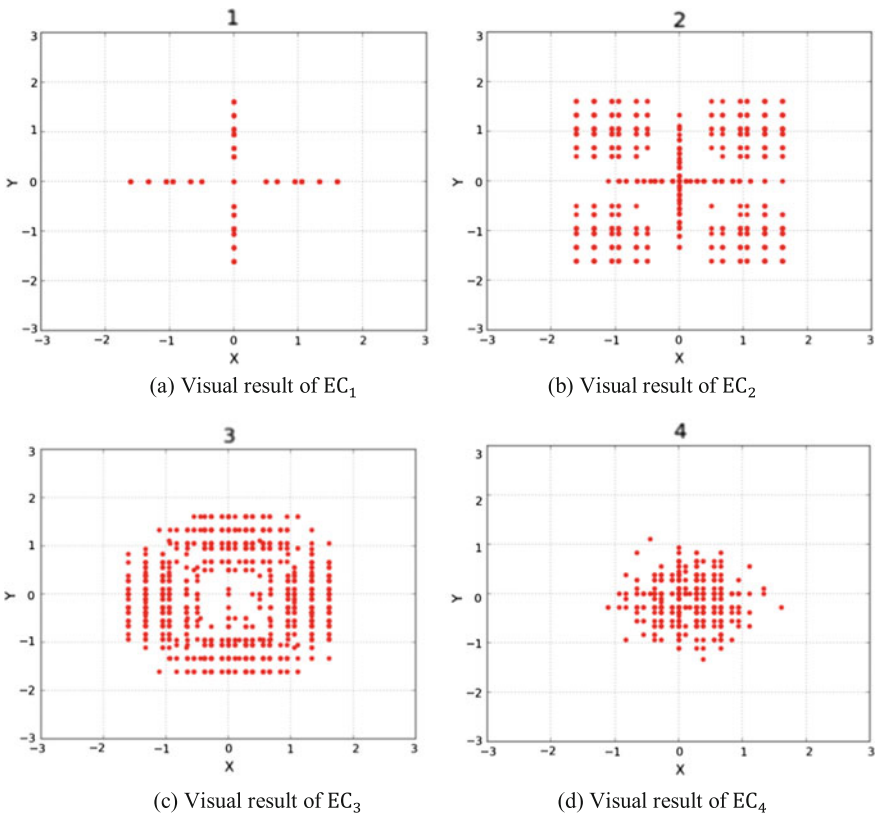


Fig. 10 Visual result of RC4 { n : 128000, N : 128, M : 24}

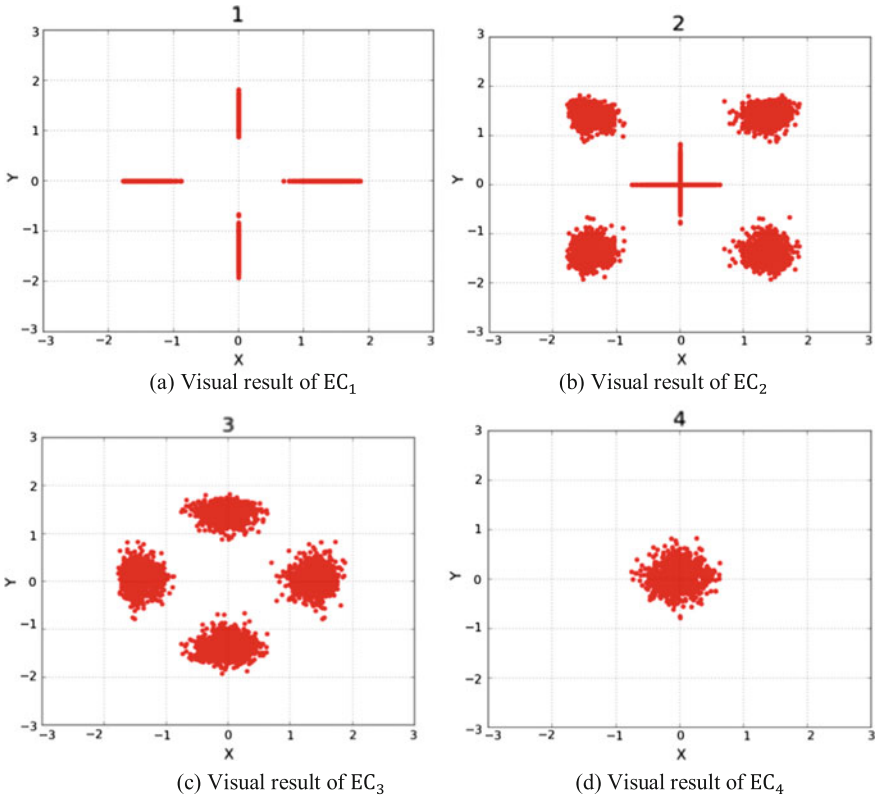


Fig. 11 Visual result of RC4 { n : 128000, N : 1000, M : 8}

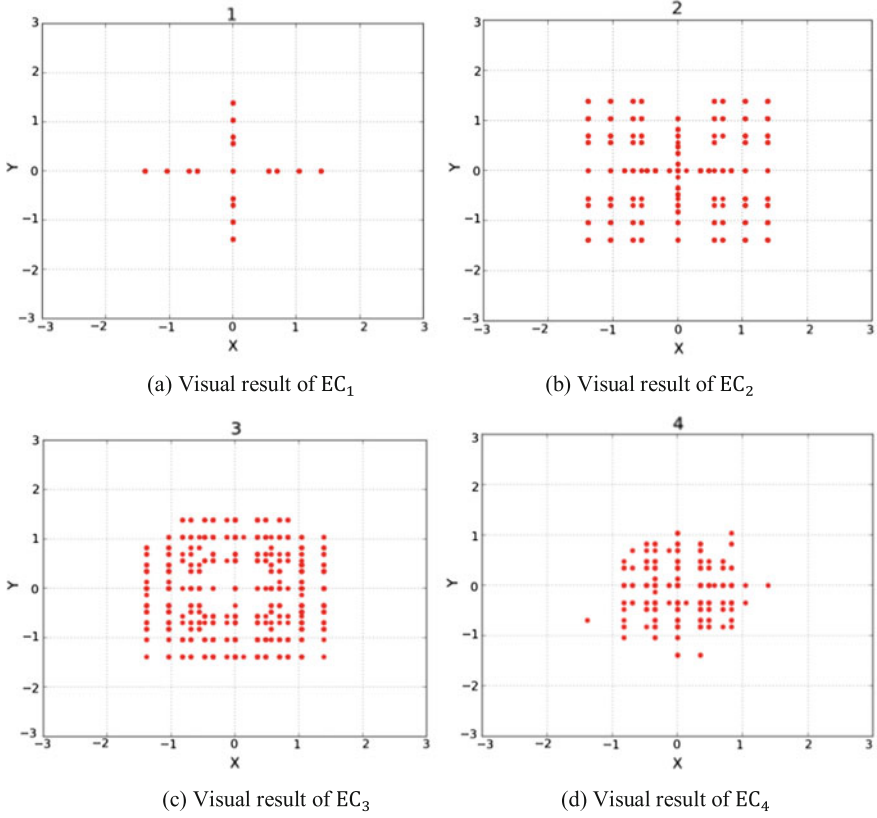


Fig. 12 Visual result of RC4 { $n : 100000, N : 100, M : 24$ }

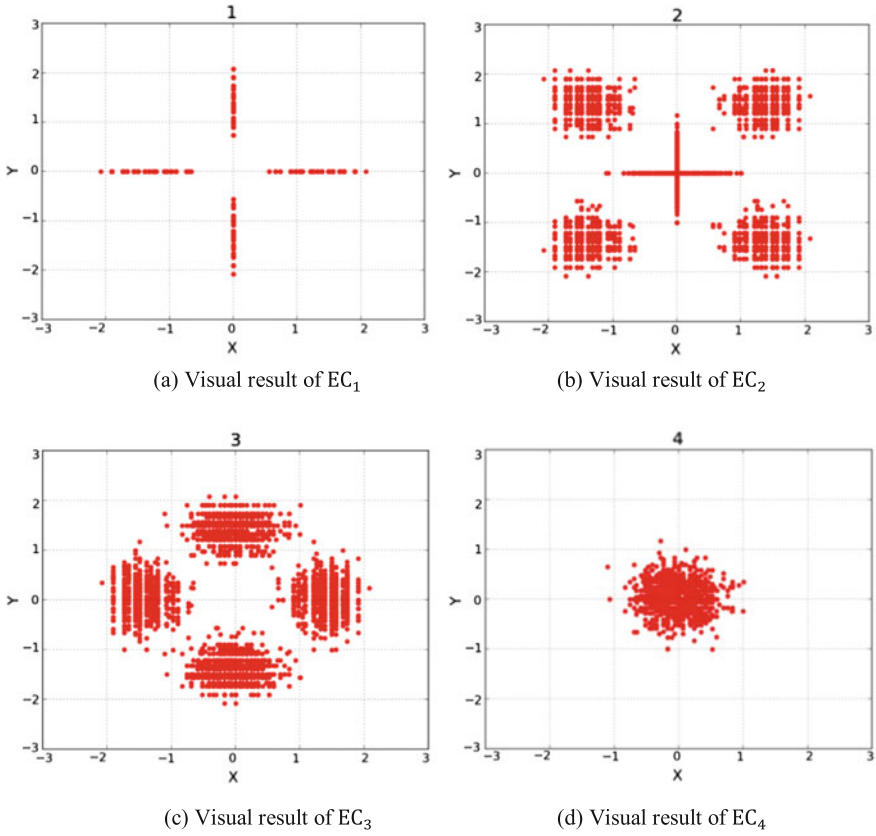


Fig. 13 Visual result of HC256 { n : 128000, N : 128, M : 16}

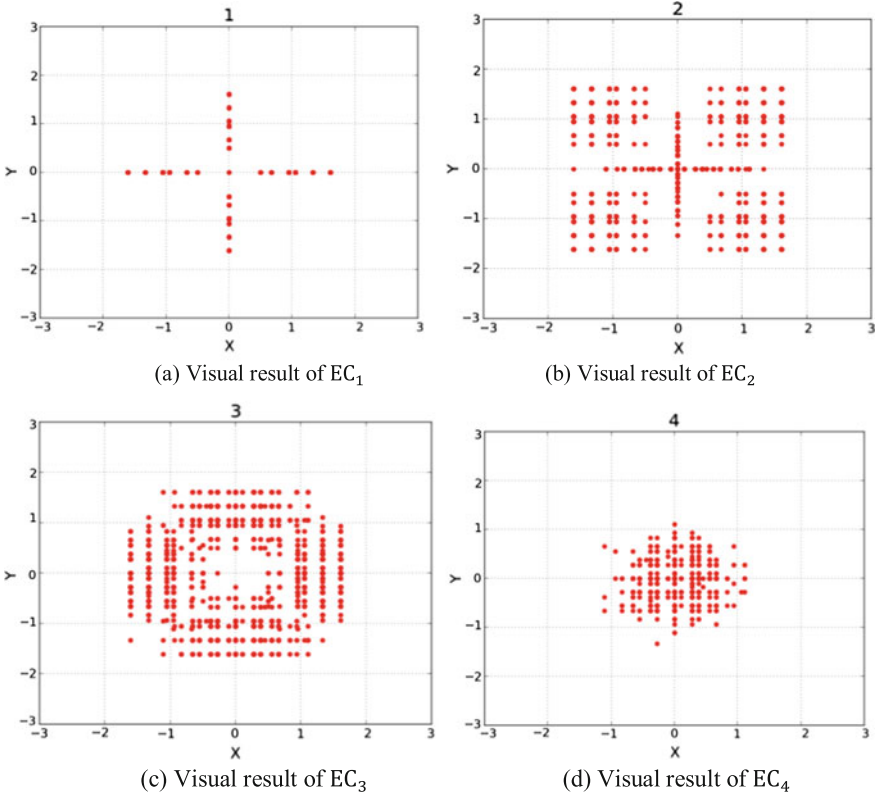


Fig. 14 Visual result of HC256 { n : 128000, N : 128, M : 24}

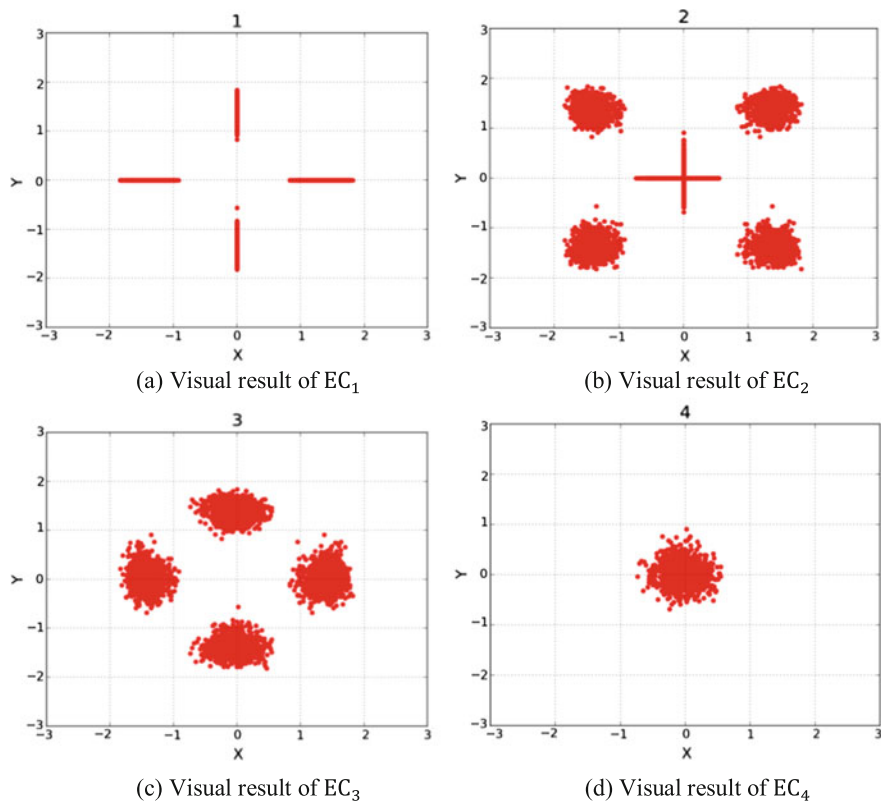


Fig. 15 Visual result of HC256 {**n** : 100000, **N** : 100, **M** : 8}

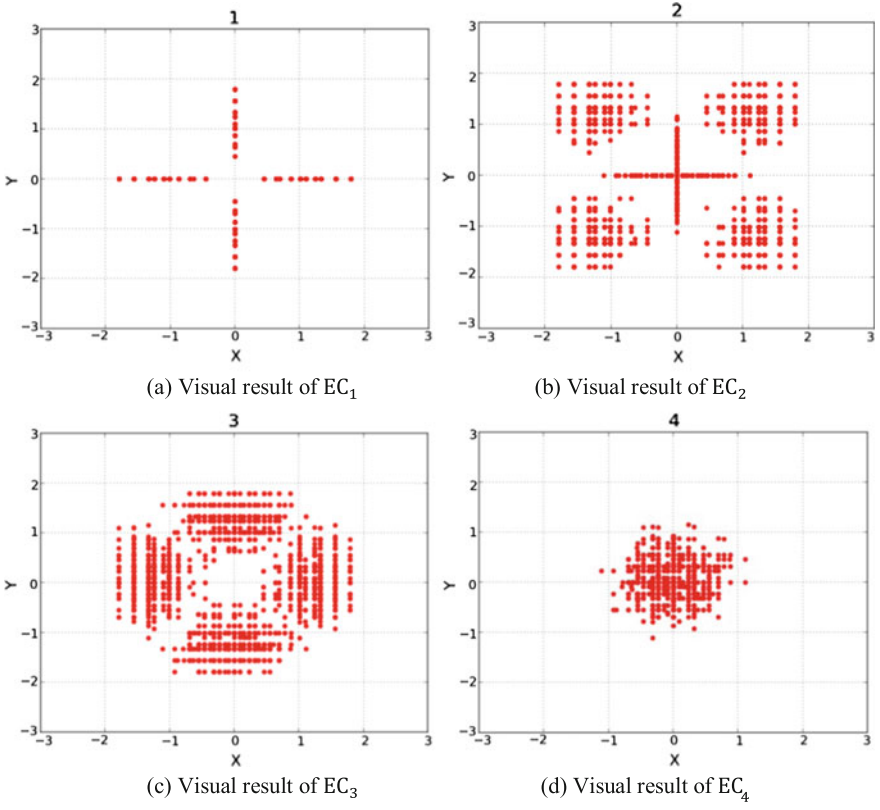


Fig. 16 Visual result of HC256 ($n : 100000, N : 100, M : 16$)

4 Conclusion

The visual results show the similar symmetry property of sequences generated by RC4 and HC256. They are showing interesting distributions and can be significantly distinguished from their combinatorial maps. From our models and illustrations, various maps can be integrated by their combinatorial projections to show different spatial distributions on random sequences. Under this configuration, the variant measure method provides a new analysis tool for stream cipher applications in further explorations.

Acknowledgements This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014) and Yunnan Advanced Overseas Scholar Project.

References

1. S. William, *Cryptography and Network Security: Principles and Practice* (Pearson Education, 2006)
2. G. Paul, S. Maitra, *RC4 Stream Cipher and Its Variants* (CRC Press, 2012)
3. P. Prasithsangaree, P. Krishnamurthy, Analysis of energy consumption of RC4 and AES algorithms in wireless LANs, in *Global Telecommunications Conference, 2003*. GLOBECOM '03. IEEE, vol. 3, pp. 1445–1449 (2003)
4. eSTREAM project, <http://www.ecrypt.eu.org/stream/>
5. J. Zheng, C. Zheng, T. Kunii, Interactive maps on variant phase spaces- from measurements—micro ensembles to ensemble matrices on statistical mechanics of particle models, in *Emerging Applications of Cellular Automata*, pp. 113–196, CC BY (2013)
6. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Election*. **5**(2), 163–172 (2010)
7. Q. Li, Z. Zheng. Spacial distributions for measures of random sequences using 2D conjugate maps, in *Proceedings of Asia-Pacific Youth Conference on Communication (APYCC)* (2010)
8. S. Wolfram, *Theory and applications of cellular automata*. Scientific (1986)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part VIII

Applications—DNA Sequences

Random numbers should not be generated with a method chosen at random.

—Donald Knuth

Natural selection is anything but random.

—Richard Dawkins

Biology is the most powerful technology ever created.

DNA is software, proteins are hardware, cells are factories.

—Arvind Gupta

Initial approaches of variant construction on DNA sequences were developed from 2012. For example, Randomness Measurement of Pseudorandom Sequence Using different Generation Mechanisms and DNA Sequence. *Journal of Chengdu University of Information Technology*. 27(6): 548–555, 2012; 2D Conjugate Maps of DNA Sequences, *Journal of Information Security* Vol. 4 No. 4 (2013), <https://doi.org/10.4236/jis.2013.44021>; Pseudo DNA Sequence Generation of Non Coding Distributions Using Variant Maps on Cellular Automata. *Applied Mathematics* 5: 153–174, 2014; Variant Map Construction to Detect Symmetric Properties of Genomes on 2D Distributions. *J Data Mining Genomics Proteomics* 5:150, 2014; Variant Maps to Identify Coding and Non-coding DNA Sequences of Genomes Selected from Multiple Species, *Biol Syst Open Access* 2016, 5:1. <https://doi.org/10.4172/2329-6577.1000153> and Mapping Whole DNA Sequence on Variant Maps, *Asunam* 2017: 1037–1040. <https://doi.org/10.1145/3110025.3110140>.

This direction contains extensive results among various applications.

This part of DNA sequences is composed of two chapters (23 and 24).

Chapter “[Variant Map System to Simulate Complex Properties of DNA Interactions Using Binary Sequences](#)” describes to use binary sequences to simulate DNA interactions under four meta basis. Different stream ciphers and real DNA sequences are applied in comparison. Their maps are illustrated similarity and differences among selected sequences.

Chapter “[Whole DNA Sequences of *Cebus capucinus* on Variant Maps](#)” applies whole DNA sequences of *Cebus Capucinus* (White Face Monkey) on variant maps. This set of maps has shown in various distributions of complex characteristics. Further researches are required.

Variant Map System to Simulate Complex Properties of DNA Interactions Using Binary Sequences



Jeffrey Zheng, Weiqiong Zhang, Jin Luo, Wei Zhou and Ruoyu Shen

Abstract Stream cipher, DNA cryptography and DNA analysis are the most important R&D fields in both Cryptography and Bioinformatics. HC-256 is an emerged scheme as the new generation of stream ciphers for advanced network security. From a random sequencing viewpoint, both sequences of HC-256 and real DNA data may have intrinsic pseudo-random properties respectively. In a recent decade, many DNA sequencing projects are developed on cells, plants and animals over the world into huge DNA databases. Researchers notice that mammalian genomes encode thousands of large noncoding RNAs (lncRNAs), interact with chromatin regulatory complexes, and are thought to play a role in localizing these complexes to target loci across the genome. It is a challenge target using higher dimensional visualization tools to organize various complex interactive properties as visual maps. The Variant Map System VMS as an emerging scheme is systematically proposed in this chapter to apply multiple maps that uses four Meta symbols as same as DNA or RNA representations. System architecture of key components and core mechanism on the VMS are described. Key modules, equations and their I/O parameters are discussed. Applying the VM System, two sets of real DNA sequences from both sample human (noncoding DNA) and corn (coding DNA) genomes are collected in comparison with pseudo DNA sequences generated by HC-256 to show their intrinsic properties in higher levels of similar relationships among relevant DNA sequences on 2D maps. Sample 2D maps are listed and their characteristics are illustrated under controllable environment. Visual results are briefly analyzed to explore their intrinsic properties on selected genome sequences.

J. Zheng (✉)

Key Laboratory of Yunnan Software Engineering, Yunnan University, Kunming, China
e-mail: conjugatologic@yahoo.com

W. Zhang

School of Software and Microelectronics, Peking University, Beijing, China

J. Luo

School of Life Sciences, Yunnan University, Kunming, China

W. Zhou · R. Shen

School of Software, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_23

Keywords Pseudo-random number generator · Stream cipher · HC-256
Binary to DNA · Pseudo DNA sequence · Large noncoding · DNA analysis
2D map · Visual distribution · Variant map system

1 Introduction

Stream ciphers [1, 2] play a key role in modern network security [3, 4] especially in multimedia network environments; its core component—pseudo random number generation mechanism [5–7]—takes the central position in modern cryptography [8, 9]. Associated with advanced development of bioinformatics, advanced DNA sequencing and analyzing techniques [10, 11] have significantly progressed over the past decade.

1.1 DNA Cryptography

DNA cryptography makes joined research in the field of DNA computing and cryptography. Scholars over the world focused on this field and different results are published such as simulating DNA evolution [12], DNA pseudorandom number generator [13–16], DNA cryptography [9, 17, 18] and so on. However in current situation, DNA cryptography is still at an earlier stage as an emerging area of advanced cryptography.

In typical results of DNA cryptography on encryption, different coding schemes could be randomly selected. E.g. the algorithm in paper [17] applies an encoding formula to express the plaintext on DNA sequence: $\{00 \rightarrow C, 01 \rightarrow T, 10 \rightarrow A, 11 \rightarrow G\}$; however in paper [18], the same author uses the coding formula $\{00 \rightarrow A, 01 \rightarrow T, 10 \rightarrow C, 11 \rightarrow G\}$ for the plaintext on DNA sequence. In encryption environment, all $4! = 24$ possible encoding methods could be equally used in different applications.

1.2 Stream Cipher HC-256

Stream ciphers are an important class of encryption algorithms. A stream cipher is a symmetric cipher which operates with a time-varying transformation on individual plaintext digits. The ECRYPT Stream Cipher Project (eSTREAM) [1] was a multi-year effort, running from 2004 to 2008, to promote the design of efficient and compact stream ciphers suitable for widespread adoption. **HC-256** is a stream cipher designed to provide bulk encryption in software at high speeds while permitting strong confidence in its security. A 128-bit variant was submitted in 2004 as an eSTREAM cipher candidate; it has been selected as one of the four final contestants

in the software profile [2, 4] in 2008 as the most advanced scheme for stream cipher applications in advanced network environment.

1.3 Large Noncoding DNA and RNA

In relation to DNA analysis, visualization methods play a key role in the Human Genome Project (HGP) [19]. After HGP completed successfully, a public research consortium—the Encyclopedia of DNA Elements (ENCODE) were launched by the National Human Genome Research Institute (NHGRI) in 2003 to find all functional elements in the human genome as one of the most critical projects by NHGRI to explore genomes after HGP.

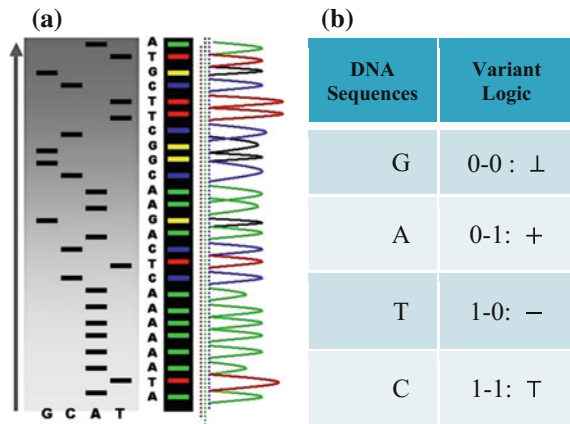
In 2012, ENCODE released a coordinated set of 30 papers published in key Journals of Nature, Genome Biology and Genome Research. These publications show that approximately 20% of noncoding DNA in the human genome is functional while an additional 60% is transcribed with no known function [20]. Much of this functional non-coding DNA is involved in the regulation of the expression of coding genes [10]. Furthermore the expression of each coding gene is controlled by multiple regulatory sites located both near and distant from the gene. These results demonstrate that gene regulation is far more complex than was previously believed [11]. Mammalian genomes encode thousands of large noncoding RNAs (lncRNAs), many of which regulate gene expression, interact with chromatin regulatory complexes, and are thought to play a role in localizing these complexes to target loci across the genome [21]. Associated with different international projects, larger numbers of Genome Databases are established and mass Genome-wide gene expression measurements are developed.

Due to huge amount of DNA sample collections and extremely difficulties to determine their variation properties in wider applications [19, 22–27], it is essential for us to extend advanced DNA analysis models, methods and tools in further extensions to explore emerging models and concepts to interpret complex interactions among complicated sets of DNA sequences in real environments.

1.4 DNA Analysis

DNA analysis plays a key role in modern genomic application [19]. The HGP is heavily relevant to advanced DNA sequencing and analysis techniques. DNA sequences are composed of four Meta symbols on {A, T, G, C} as basic structure. Classical DNA double helix structure makes the first level of pair construction of DNA sequences with A & T and G & C complementary structures as the first level of symmetric relationships. A typical DNA sequencing result is shown in Fig. 1a. Four Meta symbols could be separated as four projective sequences.

Fig. 1 Modern DNA sequencing and their correspondences on Variant Logic; **a** a sample DNA sequencing and its four projection sequences; **b** four Meta DNA symbols and linkages to variant logic



In ENCODE, recent Genomic analysis results are indicated that encoded sequences have only 20% in human genomes and around 80% genomes look like useless sequences. Under further assumptions, it seems that additional symmetric properties are required to satisfy the second, third and higher levels of structural constructions to explore complex interactive properties [10, 11, 19–29].

In current situation, it is necessary for advanced researchers to shift targets in computational cell biology from directly collecting sequential data to making higher-level interpretation and exploring efficient content-based retrieval mechanism for genomes. Using higher dimensional visualization tools, their complex interactive properties could be organized as different visual maps systematically.

1.5 Variant Construction and DNA

Variant construction is a new structure composed of logic, measurement and visualization models to analyze 0–1 sequences under variant conditions. The further details of this construction can be checked on variant logic [30, 31], 2D maps [32, 33], variant pseudo-random number generator [34], DNA maps [35] and variant phase spaces [33]. Since the variant system uses another set of four Meta symbols { \perp , +, -, \top } to describe system, a typical correspondence shown in Fig. 1b may provides a natural mapping between DNA and variant data sequences.

Since DNA sequences are played an essential role to explore different symmetric properties based on analysis approaches, in this chapter, measurement and visual models are proposed systematically to use a fixed segment structure to measure four Meta symbols distributions in their spectrum construction. Under this construction, refined symmetric features can be identified from various polarized distributions and further symmetric properties are visualized.

1.6 Target of This Chapter

The target of this chapter is to establish the Variant Map System (VMS) as a unified framework to analyze complex DNA interactions on both artificial and natural DNA sequences. The VMS has designed to use variant logic schemes [30–35] applying multiple maps on four Meta symbols as DNA or RNA representations. System architecture of key components and core mechanism on the VMS are described. Key modules, equations and their I/O parameters are discussed. Applying the VM System, two sets of real DNA sequences from both human (noncoding DNA) and corn (coding DNA) genomes are collected in comparison with pseudo DNA sequences generated artificially by HC-256 to show their intrinsic properties in higher levels of similar relationships among DNA sequences on 2D maps. Further descriptions and discussions are provided respectively.

2 System Architecture

In this section, system architecture and their core components are discussed with the use of diagrams. The refined definitions and equations of this system are described in the next section—Variant Map System.

2.1 Architecture

The four components of a variant map system are the Binary To DNA (BTD), the Binary Probability Measurement (BPM), the Mapping Position (MP), and the Visual Map (VM) as shown in Fig. 2.

The architecture is shown in Fig. 2a with the key modules of the four core components being shown in Fig. 2b–e respectively.

In the first part of the system, the t -th sequence Y^t on either $\{0, 1\}$ or $\{A, G, T, C\}$ are input data to get into the BTM module. The main function of the BTM is to output a unified sequence X^t either to transfer a 0–1 sequence or to keep a DNA sequence as a pseudo or pure DNA sequence under a set of controlled parameters.

Using this unified DNA sequence, four vectors of probability measurements are created from the t -th selected DNA sequence with N_t elements as an input. Multiple segments are partitioned by a fixed number of n elements for each segment; at least m_t segments can be identified by the BPM component. Next component uses the four vectors of probability measurements and a given k value as input data, a pair of position values are created for each Meta symbol. Four pairs of values are generated by the MP component. Then, in order to process multiple selected DNA sequences, all selected sequences are processed by the VM component and each sequence may pro-

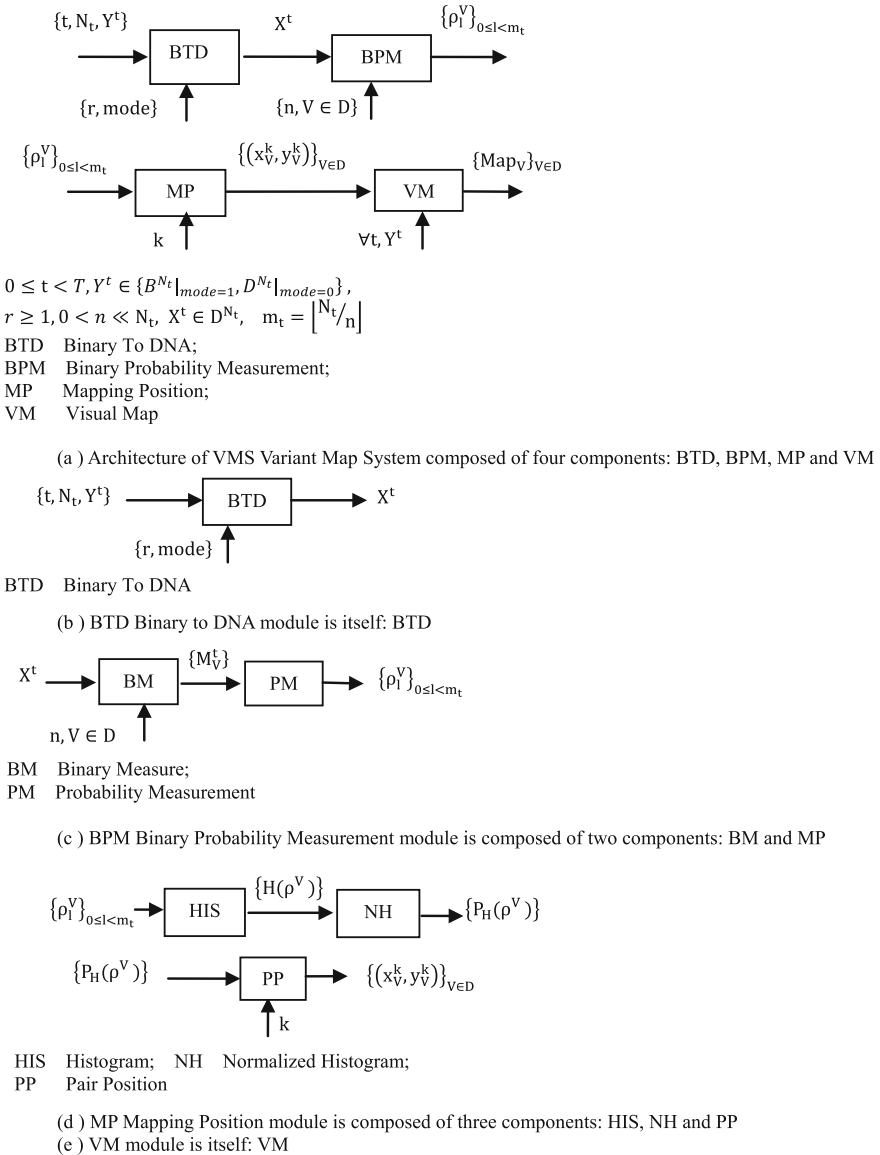


Fig. 2 Variant Map System VMS and key components **a** Architecture; **a** BTD component; **b** BPM component; **c** MP component; **d** VM component

vide a set of pair values to generate relevant variant maps to indicate their distribution properties respectively.

With eight parameters in an input group, there are three sets of parameters in the intermediate group and one set of parameters in the output group.

The three groups of parameters are listed as follows.

Input Group:

- t An integer indicates the t -th DNA sequence selected, $0 \leq t < T$
- r An integer indicates a relationship distance among elements in a binary sequence, $r \geq 1$
- mode An integer indicates the mode of elements in a sequence, $\text{mode} \in \{0, 1, \dots\}$, $\text{mode} = 0$ for a DNA sequence, $\text{mode} = 1$ for a binary sequence
- N_t An integer indicates the number of elements in the t -th DNA sequence, $N_t \gg r$
- Y^t An input data vector with N_t elements, $Y^t \in \{D^{N_t} |_{\text{mod } e=0}, B^{N_t} |_{\text{mod } e=1}\}$
- n An integer indicates the number of elements in a segment, $n > 0$
- V A symbol is selected from four DNA symbols $\{A, G, T, C\} = D$, $V \in D$
- k An integer indicates the control parameter for mapping, $k > 0$.

Intermediate Group:

- X^t A unified DNA vector with N_t elements, $X^t \in D^{N_t}$
- $\{\rho_l^V\}$ Four sets of probability measurements with $0 \leq l < m_t$, $V \in D$
- $\{(x_V^k, y_V^k)\}$ Four paired values, $k > 0$, $V \in D$

Output Group:

- $\{\text{Map}_V\}$ Four 2D maps, $V \in D$

2.2 BTD Binary to DNA

The BTD component shown in Fig. 2b is composed of one module: BTD itself. Five parameters are shown as input signals and one unified vector is generated by the BTD component as the output group.

Input Group:

- t An integer indicates the t -th DNA sequence selected, $0 \leq t < T$
- r An integer indicates a relationship distance among elements in a binary sequence, $r \geq 1$
- mode An integer indicates the mode of elements in a sequence, $\text{mode} \in \{0, 1, \dots\}$, $\text{mode} = 0$ for a DNA sequence, $\text{mode} = 1$ for a binary sequence
- N_t An integer indicates the number of elements in the t -th DNA sequence, $N_t \gg r$
- Y^t An input data vector with N_t elements, $Y^t \in \{D^{N_t} |_{\text{mod } e=0}, B^{N_t} |_{\text{mod } e=1}\}$

Output Group:

- X^t A unified data vector with N_t elements, $X^t \in D^{N_t}$

The BTD component uses an input vector on either binary or DNA format as input, under a set of input parameters to process transformation. The output of the BTD component is composed of a unified vector of DNA format in a given condition.

2.3 BPM Binary Probability Measurement

The BPM component shown in Fig. 2c is composed of two modules: BM Binary Measure and PM Probability Measurement. Three parameters are listed as input signals; four vectors of binary measures are outputted from the BM component as an intermediate group and four sets of probability measurements are outputted as an output group.

Input Group:

- n An integer indicates the number of elements in a segment, $n > 0$
- V A symbol is selected from four DNA symbols $\{A, G, T, C\} = D$, $V \in D$
- X^t A DNA vector with N_t elements, $X^t \in D^{N_t}$

Intermediate Group:

- $\{M_V^t\}$ Four 0–1 vectors with N_t elements, $M_V^t(I) \in \{0, 1\} = B$, $M_V^t \in B^{N_t}$, $V \in D$

Output Group:

- $\{\rho_l^V\}$ Four sets of probability measurements with $0 \leq l < m_t$, $V \in D$

The BPM component transforms a selected DNA sequence to generate four 0–1 vectors by BM module for the input DNA sequence. Then four probability vectors are generated by the PM module as the output of the BPM under a fixed length of segment condition.

2.4 MP Mapping Position

The MP component shown in Fig. 2d is composed of three modules: HIS Histogram, NH Normalized Histogram and PP Pair Position. Two parameters are listed as input signals; four histograms and four normalized histograms are generated from the HIS component and the NH component as intermediate groups respectively. Four paired values are generated by the PP component as the output group.

Input Group:

- $\{\rho_l^V\}$ Four sets of probability measurements with $0 \leq l < m_t$, $V \in D$
- k An integer indicates the control parameter for mapping, $k > 0$

Intermediate Group:

- $\{H(\rho^V)\}$ Four histograms for relevant probability measurements, $V \in D$
 $\{P_H(\rho^V)\}$ Four normalized histograms for relevant probability measurements, $V \in D$

Output Group:

- $\{(x_V^k, y_V^k)\}$ Four paired values, $k > 0, V \in D$

The MP component uses probability measurements as input, under a given k condition to generate each relevant histogram and its normalized distribution. The output of the MP component is composed of four paired values controlled in a given condition.

2.5 VM Visual Map

The VM component shown in Fig. 2e is composed of one module: VM Visual Map. Three parameters are input signals. Collected all selected DNA sequences, four 2D maps are generated by the VM component as the output result.

Input Group:

- $\forall t$ All DNA sequences are selected, $0 \leq t < T$
 Y^t An input data vector with N_t elements, $Y^t \in \{D^{N_t} | \text{mod } e=0, B^{N_t} | \text{mod } e=1\}$
 $\{(x_V^k, y_V^k)\}^t$ Four paired values for the t -th DNA sequence, $k > 0, V \in D$

Output Group:

- $\{\text{Map}_V\}$ Four 2D maps, $V \in D$

The VM component processes all selected DNA sequences as input to generate paired values for each sequence. The output of the VM component is composed of four 2D maps to show the final visual distribution for the system.

3 Variant Map System

In this section, definitions and equations are provided to describe the VMS. In addition to the initial preparation, seven core modules are involved in the BTM, BM, PM, HIS, NH, PP and VM components respectively.

3.1 Initial Preparation

Let r an input parameter make all pairs of elements with r distance in a binary sequence to be a pseudo DNA vector, mode a controlled parameter indicate various pairs of operations performed if mode ≥ 1 . Denote $B = \{0, 1\}$ a binary base and $D = \{A, G, T, C\}$ a DNA base respectively.

3.2 BTD Module

Let Y an input sequence with N elements, $0 \leq I < N$, $Y(I) \in \{B^N |_{\text{mod } e \geq 1}, Y(I) \in D^N |_{\text{mod } e=0}\}$. This input vector could be expressed as follows.

$$\begin{aligned} Y &= (Y(0), \dots, Y(I), \dots, Y(N-1)), 0 \leq I < N \\ Y(I) &\in \{B^N |_{\text{mode} \geq 1}, Y(I) \in D^N |_{\text{mode}=0}\}. \end{aligned} \quad (1)$$

Let X denote a DNA sequence with N elements, D denote a symbol set with four elements i.e. $D = \{A, G, T, C\}$. This type of a DNA sequence can be described by a four valued vector as follows:

$$\begin{aligned} X &= (X(0), \dots, X(I), \dots, X(N-1)), \\ 0 \leq I < N, X(I) &\in D = \{A, G, T, C\}, X \in D^N \end{aligned} \quad (2)$$

From this input and associated parameters, following operations are performed. If mode = 0, for all I , $Y(I) \in D$, the output vector is equal to the input vector.

$$\forall I, X(I) = Y(I), 0 \leq I < N \quad (3)$$

If mode = 1, for all pairs of I and $I+r(\text{mod } N)$ elements of Y , $Y(I), Y(I+r) \in B$, the I -th output element $X(I)$ can be determined by the corresponding conditions shown in Fig. 1b as follows.

$$X(I) = \begin{cases} G, & \text{if } Y(I) = 0 \ \& \ Y(I+r) = 0 \\ A, & \text{if } Y(I) = 0 \ \& \ Y(I+r) = 1 \\ T, & \text{if } Y(I) = 1 \ \& \ Y(I+r) = 0 \\ C, & \text{if } Y(I) = 1 \ \& \ Y(I+r) = 1 \end{cases}, \quad (4)$$

In both conditions, X will be a unified vector with four values as the output of the BTD shown in Fig. 2b.

E.g. Let a binary sequence $Y = 100111001011$, $N = 12$, three pseudo DNA sequences ($r = 1, r = 2, r = 3$) can be represented as follows.

$$Y = 100111001011$$

$$\begin{aligned}
X_{r=1} &= TGACCTGATACC \\
X_{r=2} &= TAACTTAGCACT \\
X_{r=3} &= CAATTCGACATT \\
Y &\in B^{12}, X \in D^{12}
\end{aligned}$$

Selecting a certain r value, a relevant pseudo DNA sequence can be generated from an input binary sequence.

3.3 BM Module

For a given I -th element, four projective operators can be defined and denoted as $\{M_A(I), M_G(I), M_T(I), M_C(I)\}$.

$$\begin{aligned}
M_A(I) &= \begin{cases} 1, & \text{if } X(I) = A; \\ 0, & \text{Otherwise;} \end{cases} & M_G(I) &= \begin{cases} 1, & \text{if } X(I) = G; \\ 0, & \text{Otherwise;} \end{cases} & M_T(I) & \\
&= \begin{cases} 1, & \text{if } X(I) = T; \\ 0, & \text{Otherwise;} \end{cases} & M_C(I) &= \begin{cases} 1, & \text{if } X(I) = C; \\ 0, & \text{Otherwise;} \end{cases} & & (5)
\end{aligned}$$

Applying the four operators to all elements, the DNA sequence X can be reorganized into the four binary sequences of 0–1 values. i.e.

$$\begin{aligned}
M_V : \{X(I)\}_{I=0}^{N-1} &\rightarrow \{M_A(I), M_G(I), M_T(I), M_C(I)\}_{I=0}^{N-1}; \\
M_V(I) &\in B = \{0, 1\}, V \in D & (6)
\end{aligned}$$

E.g. Let a DNA sequence $X = CTGATTAGCCAT$, $N = 12$, its four binary sequences can be represented as follows.

$$\begin{aligned}
X &= CTGATTAGCCAT \\
M_A &= 000100100010 \\
M_G &= 001000010000 \\
M_T &= 010011000001 \\
M_C &= 100000001100
\end{aligned}$$

It is interesting to notice that the basic relationship between a DNA sequence X and its four M_V sequences are exactly same as in a modern DNA sequencing procedure to separate a selected DNA sequence into the four Meta symbol sequences shown in Fig. 1a. This correspondence could be the key feature to apply the proposed scheme naturally in simulating complex behaviors for any DNA sequence.

The projection M_V provides the essential operation in the BM component as the first module shown in Fig. 2c.

3.4 PM Module

For this set of the four binary sequences, it is convenient to partition them into m segments and each segment contained a fixed number of n elements.

For the l -th segment, let $0 \leq l < m$, $0 \leq j < n$, the I -th position will be $I = l * n + j$, four probability measurements $\{\rho_A, \rho_G, \rho_T, \rho_C, \}$ can be defined.

$$\rho_l^V = \frac{\sum_{I=l*n}^{(l+1)*n-1} M_V(I)}{n}, V \in D, 0 \leq I < N = n * m \quad (7)$$

Under this construction, four sets of probability measurements established.

$$\rho^V : \{M_A(I), M_G(I), M_T(I), M_C(I),\}_{I=0}^{N-1} \rightarrow \{\rho_l^A, \rho_l^G, \rho_l^T, \rho_l^C, \}_{l=0}^{m-1} \quad (8)$$

The probability operator ρ^V generates four probability measurement vectors in the PM component as the second module shown in Fig. 2c. After the BM and PM processes, the whole procedure of the BPM component is complete in Fig. 2c.

3.5 HIS Module

Since the BPM generates four sets of probability measurement, it is necessary to perform further operations in the MP component shown in Fig. 2d as follows.

In the HIS component as the first module in Fig. 2d, each probability sequence $\{\rho_l^V, \}_{l=0}^{m-1}, V \in D$ can be calculated from n positions, at most $n + 1$ distinguished values identified in a vector. Under this organization, a histogram distribution can be established.

Let $H(\cdot)$ be a histogram operator, for each position, it satisfies following relation,

$$H(\rho_l^V) = \begin{cases} 1, & \text{if } \rho_l^V = \frac{i}{n}, V \in D; \\ 0, & \text{Otherwise, } 0 \leq i \leq n. \end{cases} \quad (9)$$

Collecting all possible values, a histogram distribution can be established,

$$H(\rho^V) = \sum_{l=0}^{m-1} H(\rho_l^V) \quad (10)$$

The histogram $H(\rho^V)$ is the output of the HIS module. Four histograms are generated after HIS process. Further normalized process will be performed in the NH component as the second module in Fig. 2d.

3.6 NH Module

Under this construction, a normalized histogram can be defined as

$$P_H(\rho^V) = H(\rho^V)/m \tag{11}$$

After the NH component processed, its output provides the PP component for further operations as the third module in Fig. 2d.

3.7 PP Module

Relevant probability vectors have $(n + 1)$ distinguished values; four sets of normalized vectors can be organized as a linear order as follows,

$$p_i^V = \sum_{l=0}^{m-1} H\left(\rho_l^V | \rho_l^V = \frac{i}{n}\right) / m, 0 \leq i \leq n \tag{12}$$

Under this condition, four linear sets of probability vectors are established,

$$\begin{aligned} P_H(\rho^V) &= \{p_i^A, p_i^G, p_i^T, p_i^C\}_{i=0}^n, \\ p_i^V &\in [0, 1], V \in D, 0 \leq i \leq n \end{aligned} \tag{13}$$

For four vectors, their components can be normalized respectively,

$$\sum_{i=0}^n p_i^V = 1, V \in D \tag{14}$$

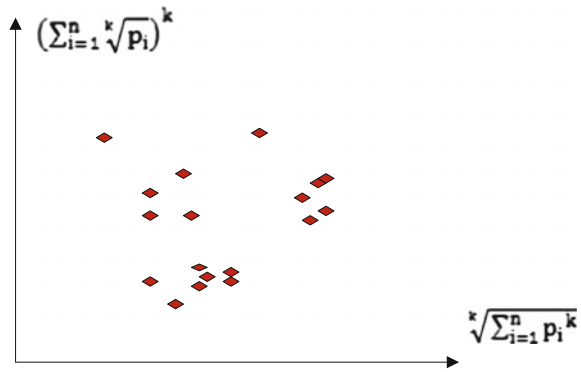
Four sets of probability vectors are composed of a complete partition on their measurements.

Using this set of measurements, two mapping functions can be established to calculate a pair of values to map analyzed DNA sequence into a 2D map as follows.

Let $y = F(P, V, k)$ and $x = F(P, V, 1/k)$ or (x_V^k, y_V^k) be a pair of values defined by following equations,

$$\begin{aligned} y_V^k &= F(P, V, k) = \left(\sum_{i=0}^n \sqrt[k]{p_i^V}\right)^k \ \& \\ x_V^k &= F(P, V, 1/k) = \sqrt[k]{\sum_{i=0}^n (p_i^V)^k}, V \in D \end{aligned} \tag{15}$$

Fig. 3 A sample 2D map of VM on multiple sequences



In the *PP* component, four paired values are generated and each pair indicates a specific position on a 2D map for the selected DNA sequence. The core operations of three key components: *BTD*, *BPM* and *MP* for a selected sequence are performed in Fig. 2b–d.

3.8 VM Module

Since only one point of a 2D map is determined for a selected DNA sequence, it is essential to apply relative larger number of DNA sequences as inputs to generate visible distributions. This type of operations will be performed in the *VM* component shown in Fig. 2e.

In a general condition, the *VM* component processes a selected data set $\{Y^t\}_{t=0}^{T-1}$ composed of *T* sequences, the *t*-th sequence with N_t elements can be expressed by $Y^t = (Y^t(0), \dots, Y^t(I), \dots, Y^t(N_t - 1))$, $Y^t \in Y(I) \in \{B^{N_t} |_{\text{mode} \geq 1}, Y(I) \in D^{N_t} |_{\text{mode}=0}\}$. Each sequence can be processed to apply the same procedures of the *BTD*, *BPM* and *MP* components. Since for each segment, its length *n* will be fixed for all selected sequences, it is essential to make number of segments be $m^t = \lfloor N_t/n \rfloor$ in convention to match each sequence. Under this expression, the last module *VM* collects all *T* pairs of positions on relevant 2D visual maps as follows,

$$VM : \{X^t\}_{t=0}^{T-1} \rightarrow \left\{ (x_V^k, y_V^k)^t \right\}_{t=0}^{T-1} \rightarrow \{MAP_V\}, V \in D \quad (16)$$

A sample 2D map of *VM* is shown in Fig. 3; this provides an assistant illustration for this type of visual maps on a case of multiple sequences.

Under this construction, a total number of *T* DNA sequences are transformed as *T* visual points on four 2D visual maps that would be help analyzers to explore their intrinsic symmetry properties among four binary sequences.

4 Sample Results on 2D Maps

Two types of data sets are selected for comparison. The first type of data sets are real DNA data sequences collected from both human and plant genomes to illustrate their differences on 2D maps. The second type of data set is collected from the Stream Cipher HC-256 to generate a pseudo random binary sequence under a certain condition.

4.1 DNA Data Resources

It is important to use some real DNA sequences to illustrate various test results of the VMS. Two sets of DNA sequences are selected and relevant resource features are described as follows.

The first data set originally comes from the human genome assembly version 37 and was taken from the reference sequences of 13 anonymous volunteers from Buffalo, New York. Hi-C technology [5] used to analyze chromatin interaction role in genome. From a genomic analysis viewpoint, this set of data may contain more complex secondary or higher level structures. A special structure nearly the GRCh37 DNA sequence has been identified to explore their spatial characteristics. After positive and negative sequencing, each data file contain 2700 DNA sequences and each sequence has around 500 elements stored in two files *left* and *right* respectively.

The second DNA data set are selected from some plant gene database for comparison. One set of DNA sequences of Corn genomes are stored in file 201–500 that contains 2700 DNA sequences and each sequence has around 200–600 elements. It may be ordinary single sequences without complex secondary structures.

4.2 Pseudo DNA Data Resources

The Stream Cipher HC-256 has being used to generate a binary sequence on a total length of 2700×500 bits in the file *hc256* that has been partitioned as 2700 subsequences and each sub-sequence in 500 bits.

Using the VMS in various parameters, three sets of pseudo DNA sequences are generated and their 2D maps are illustrated, analyzed and compared in following subsections.

4.3 Sample Results

Using the three files of DNA sequences and one pseudo binary sequence in three parameters, six sets of 2D maps are listed in Figs. 4, 5, 6, 7, 8 and 9 under different conditions to illustrate their spatial distributions using the VMS in a controllable environment.

In Fig. 4, three groups of eighteen 2D maps are shown in the range of $n = 3 \sim 50$, $k = 7$, $N \cong 200 \sim 600$, $T = 2700$ for comparison; (a1–a6) six Map_A maps for the file *Right*; (b1–b6) six Map_G maps for the file 201–500; (c1–c6) six Map_A maps for the file *hc256* respectively.

In Fig. 5, four groups of sixteen 2D maps for the file *right* are listed in the range of $n = 15$, $k = \{2, 3, 4, 7\}$, $N \cong 500$, $T = 2700$; (a) group (a1–a4) four Map_A maps; (b) group (b1–b4) four Map_T maps; (c) group (c1–c4) four Map_G maps; (d) group (d1–d4) four Map_C maps.

In Fig. 6, four groups of sixteen 2D maps for the file *hc256* are listed in the range of $n = 12$, $k = \{2, 3, 4, 7\}$, $N \cong 500$, $T = 2700$, $r = 1$, $\text{mode} = 1$; (a) group (a1–a4) four Map_A maps; (b) group (b1–b4) four Map_T maps; (c) group (c1–c4) four Map_G maps; (d) group (d1–d4) four Map_C maps.

In Fig. 7, four groups of sixteen 2D maps for the file *right* are selected in the range of $n = 15$, $k = \{2, 3, 4, 7\}$, $N \cong 500$, $T = 2700$; (a) group (a1–a4) four Map_A maps; (b) group (b1–b4) four Map_T maps; (c) group (c1–c4) four Map_G maps; (d) group (d1–d4) four Map_C maps.

In Fig. 8, three groups of twelve 2D maps for the file *hc256* are compared in the range of $n = 12$, $k = 7$, $N \cong 500$, $T = 2700$, $r = \{1, 2, 3\}$, $\text{mode} = 1$; (a) group (a1–a4) four Map_V maps $r = 1$; (b) group (b1–b4) four Map_V maps $r = 2$; (c) group (c1–c4) four Map_V maps $r = 3$.

In Fig. 9, three groups of twelve 2D maps for two files *right* and *hc256* are compared in the range of $k = 7$, $N \cong 500$, $T = 2700$; (a) the file *right* $n = 15$, $\text{mode} = 0$; (b) the file *hc256* $n = 12$, $\text{mode} = 1$, $r = 1$; (c) the file *hc256* $n = 12$, $\text{mode} = 1$, $r = 3$; (a1–c1) Map_A maps; (a2–c2) Map_T maps; (a3–c3) Map_G maps; (a4–c4) Map_C maps.

4.4 Result Analysis of 2D Maps

Six groups of 2D maps contain different information, it is necessary to make a brief discussion on their important issues as follows.

The first group of results shown in Fig. 4 presents three sets of eighteen 2D maps from three data files: *right*, 201–500 and *hc256* undertaken various lengths of basic segment from 3 to 50 to illustrate their variations respectively. Six 2D maps of each group in Fig. 4 (a1–a6) show significant trace on their visual distributions; the numbers of main visible clusters identified are decreased when the length of segment has being increased e.g. (a3–a6). However lesser length of segment does not pro-

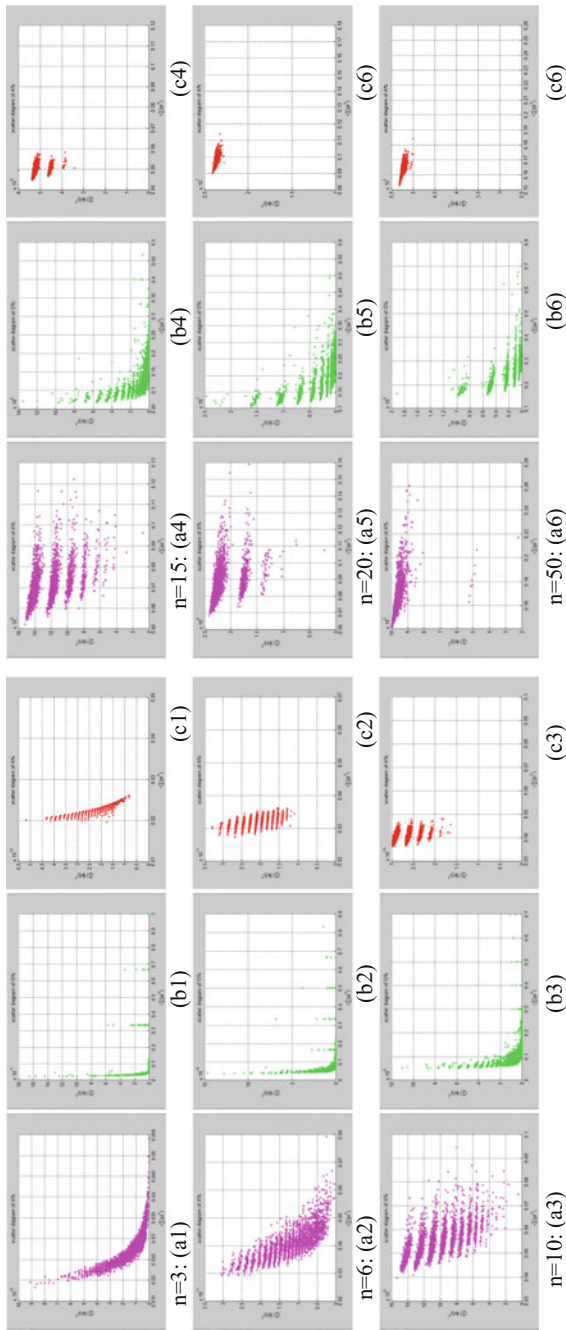


Fig. 4 Three groups of eighteen 2D maps in the range of $n=3$ –50, $k=7$, $N \cong 200$ –600, $T=2700$; (a1–a6) Map_A for the file 201–500; (c1–c6) Map_A for the file *hc256* mode = 1, $r = 1$

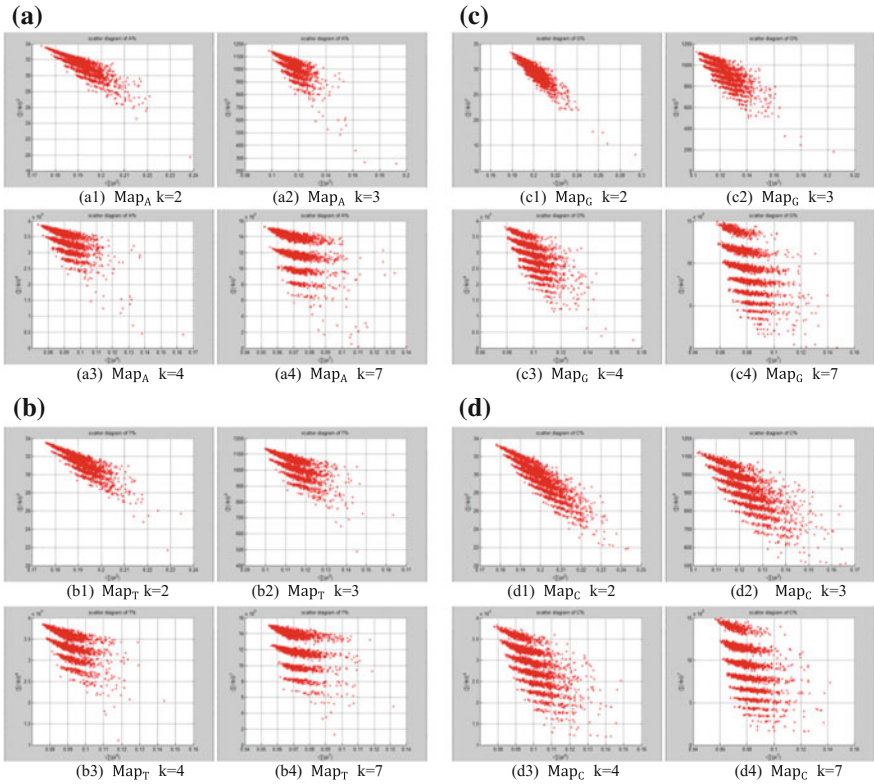


Fig. 5 Four groups of sixteen 2D maps in the range of $n = 15, k = \{2, 3, 4, 7\}, N \cong 500, T = 2700$; **a** group (a1–a4) four Map_A maps; **b** group (b1–b4) four Map_T maps; **c** (c1–c4) four Map_C maps; **d** (d1–d4) four Map_C maps for the file *right*

vide refined visual distinctions with larger region in fuzzy areas e.g. (a1–a2). From a structural viewpoint, middle ranged numbers of length provide better clustering results e.g. (a3–a5) for further analysis targets. To check another six 2D maps of Fig. 4 (b1–b6) for the file 201–500, significantly different visual distributions can be observed than (a1–a6); the numbers of main visible clusters identified are decreased when the length of segment has being increased less significantly e.g. (b4–b6). However lesser length of segment does not provide refined visual distinctions with wider regions in fuzzy areas e.g. (b1–b3). In general, middle ranged numbers of length still provide better clustering effects e.g. (b4–b6) for further analysis purpose. To check six 2D maps of Fig. 4 (c1–c6) for the file *hc256 r=1*, similar visual distributions can be observed than (a1–a6) and significantly differences are observed than (b1–b6); the numbers of main visible clusters identified are decreased when the length of segment has being increased less significantly e.g. (c3–c6). However lesser length of segment does provide refined visual distinctions with regions in fuzzy areas e.g. (b1). In general, middle ranged numbers of length still provide better clustering effects

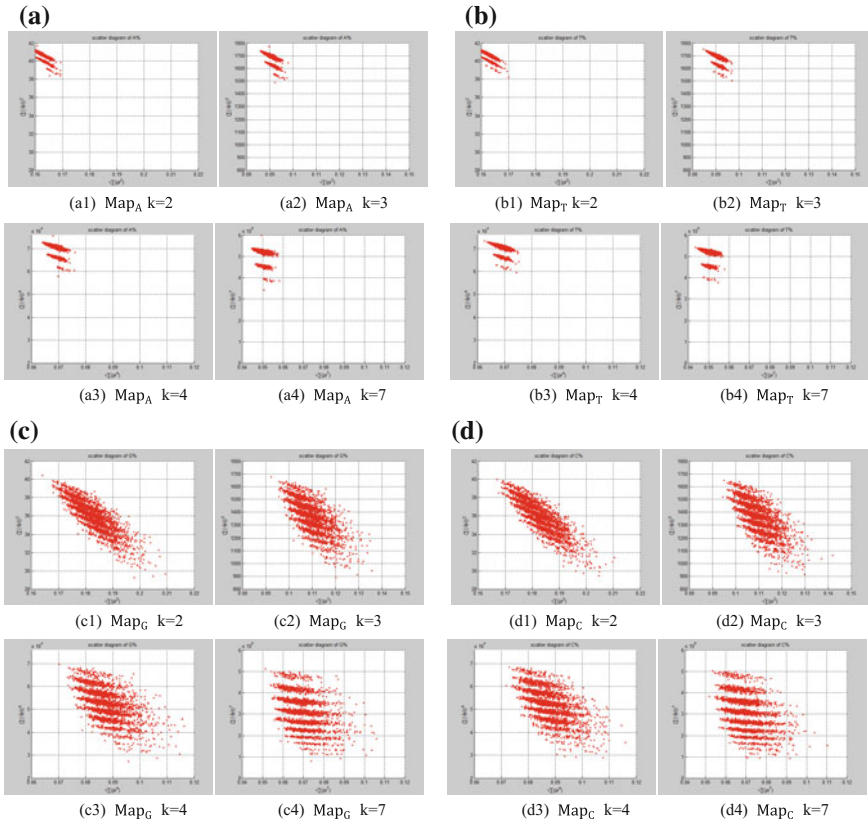


Fig. 6 Four groups of sixteen 2D maps in the range of $n = 12, k = \{2, 3, 4, 7\}, N \cong 500, T = 2700$ for the file *hc256*, $r = 1$, mode = 1; **a** group (a1–a4) four Map_A maps; **b** group (b1–b4) four Map_T maps; **c** (c1–c4) four Map_G maps; **d** (d1–d4) four Map_C maps

e.g. (c2–c4) for further analysis purpose. From their distributions, groups (a) and (c) have shared much stronger similar properties than group (b).

It is interesting to observe different maps when control parameter k changed. Four groups of sixteen 2D maps for the file *right* are shown in Fig. 5 on the range of $n = 15, k = \{2, 3, 4, 7\}, N \cong 500, T = 2700$; four groups in (a)–(d) provide four maps to share the same other parameters with different k values. Checking visible clusters in different maps, it is important to notice nearly same numbers of clusters identified in the same group, but different groups may contain significantly different numbers. Lesser k value (e.g. $k = 2$) makes a tighter distribution and larger k value (e.g. $k = 7$) takes better separation on the maps. Through $k = 7$ maps provide better separation effects, it is easy to observe their axis values already in 10^8 range.

Four groups of sixteen 2D maps for the file *hc256* are shown in Fig. 6 in the range of $n = 12, k = \{2, 3, 4, 7\}, N \cong 500, T = 2700, r = 1$. This group of 2D maps

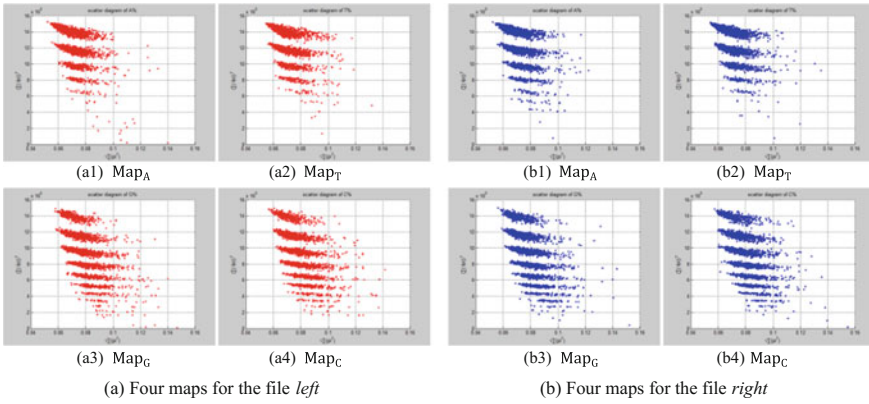


Fig. 7 Two groups of eight 2D maps in the range of $n = 15, k = 7, N \cong 200 \sim 600, T = 2700$; **a** group (a1–a4) four Map_V maps for the file *left*; **b** group (b1–b4) four Map_V maps for the file *right*

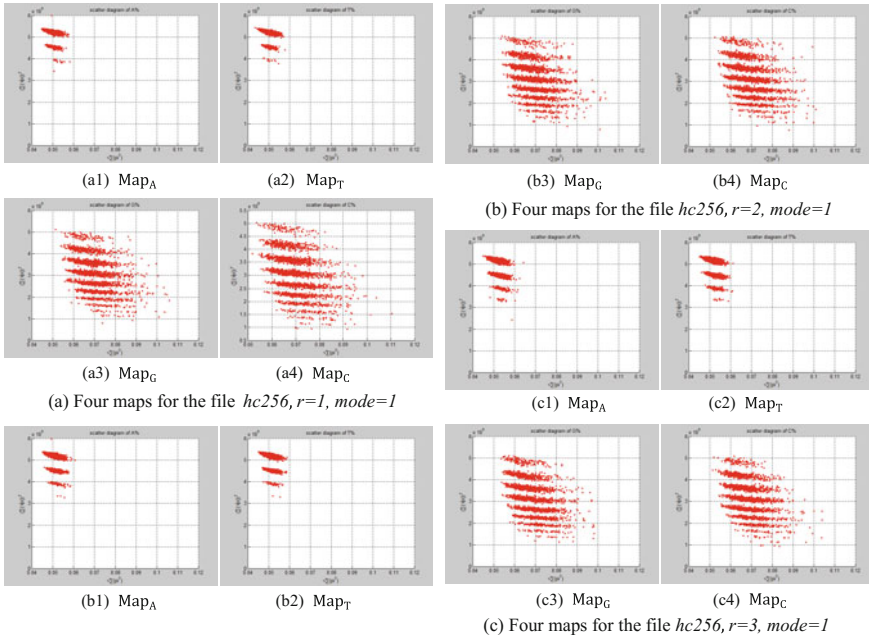


Fig. 8 Three groups of twelve 2D maps in the range of $n = 12, k = 7, N = 500, T = 2700$ for the file *hc256, r = {1,2,3}, mode = 1*; **a** group (a1–a4) four Map_V maps $r = 1$; **b** group (b1–b4) four Map_V maps $r = 2$; **c** group (c1–c4) four Map_V maps $r = 3$

can be compared with 2D maps in Fig. 5. Under the same parameters, similar visible effects and feature clustering properties could be observed if various k values are selected.

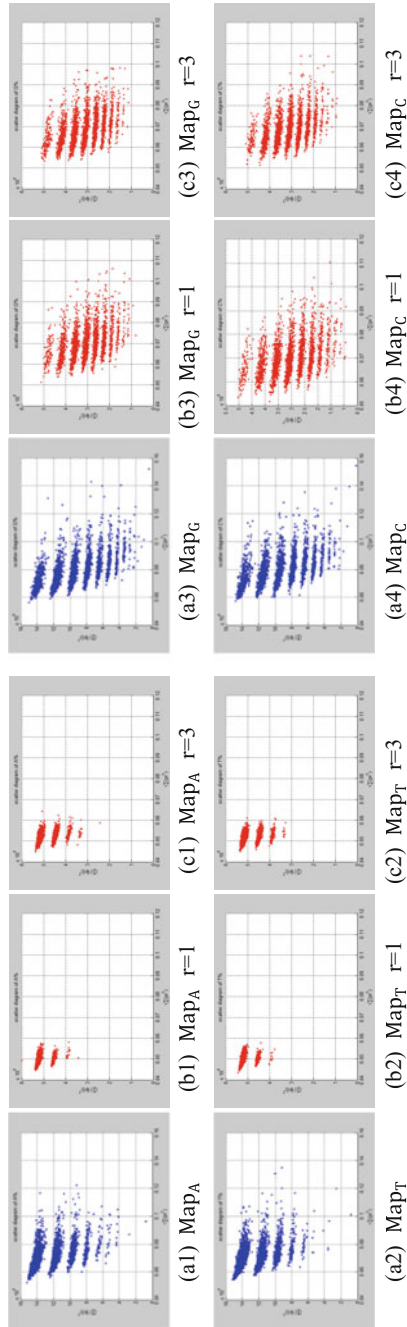


Fig. 9 Three groups of twelve maps in the ranges: $N=500$, $T=2700$, $k=7$; **a** Real DNA data; (a1–4) DNA sequences from the file *right*; **(b–c)** Simulation data; (b1–4) Binary sequences from the file *hc256*, $r=1$; (c1–4) Binary sequences from the file *hc256*, $r=3$

Using a set of selected parameters, two groups of eight 2D maps are compared in Fig. 7 for two files: *left*, *right* to explore higher levels of symmetric properties for secondary or higher levels of structures potentially contained in DNA sequences. Selected parameters are in the range of $n = 15, k = 7, N \cong 500, T = 2700$. Group (a) provides four Map_V maps (a1–a4) for the file *left*; group (b) uses four Map_V maps (b1–b4) for the file *right*.

In convenient description, let \sim be a similar operator, for groups (a) and (b), four pairs of $\{(a1)\sim(b1), (a2)\sim(b2), (a3)\sim(b3), (a4)\sim(b4)\}$ maps i.e. (*left*-A \sim *right*-A, *left*-T \sim *right*-T, *left*-G \sim *right*-G, *left*-C \sim *right*-C) have a stronger similar distribution between *left* & *right*. In addition, only two clustering classes could be significantly identified as $\{(a1)\sim(a2)\sim(b1)\sim(b2), (a3)\sim(a4)\sim(b3)\sim(b4)\}$ i.e. (*left*-A \sim *right*-A \sim *left*-T \sim *right*-T, *left*-G \sim *right*-G \sim *left*-C \sim *right*-C) respectively. This type of similar clustering distributions may strongly indicate eight maps with intrinsically higher levels of DNA sequences with extra A–T and G–C pairs of symmetric relationships between two files: *left* & *right*.

Using a set of selected parameters, three groups of twelve 2D maps are listed in Fig. 8 for the file *hc256*, $r = \{1, 2, 3\}$ to explore properties for their higher levels of structures potentially contained in pseudo DNA sequences. Selected parameters are in the range of $n = 12, k = 7, N \cong 500, T = 2700$. Group (a) provides four Map_V maps (a1–a4) for $r = 1$; group (b) uses four Map_V maps (b1–b4) for $r = 2$ (c) uses four Map_V maps (c1–c4) for $r = 3$. Using a similar operator, for groups (a–c), four pairs of $\{(a1)\sim(b1)\sim(c1), (a2)\sim(b2)\sim(c2), (a3)\sim(b3)\sim(c3), (a4)\sim(b4)\sim(c4)\}$ maps i.e. (A($r = 1$) \sim A($r = 2$) \sim A($r = 3$), ..., C($r = 1$) \sim C($r = 2$) \sim C($r = 3$)) have a stronger similar distribution among $r = \{1, 2, 3\}$. In addition, only two clustering classes could be significantly identified as $\{(a1)\sim(a2)\sim(b1)\sim(b2)\sim(c1)\sim(c2), (a3)\sim(a4)\sim(b3)\sim(b4)\sim(c3)\sim(c4)\}$ i.e. three maps are shown in (A \sim T, G \sim C) respectively.

In a convenient comparison, using a set of selected parameters, three groups of twelve 2D maps are compared in Fig. 9 for the files: *right* and *hc256*, $r = \{1, 3\}$ to check their distribution properties contained in both DNA and created pseudo DNA sequences. Group (a) provides four Map_V maps (a1–a4) for the file *right*; groups (b) and (c) provide four Map_V maps (b1–b4) for *hc256*, $r = 1$ (c) and (c1–c4) for *hc256*, $r = 3$.

Using a weak similar operator \simeq , for groups (a–c), four pairs of $\{(a1)\simeq(b1)\sim(c1), (a2)\simeq(b2)\sim(c2), (a3)\sim(b3)\sim(c3), (a4)\sim(b4)\sim(c4)\}$ maps have a stronger similar distribution between $r = \{1, 3\}$ and a weak similar distribution on A and T cases. In addition, only two clustering classes could be significantly identified as $\{(a1)\sim(a2)\simeq(b1)\sim(b2)\sim(c1)\sim(c2), (a3)\sim(a4)\sim(b3)\sim(b4)\sim(c3)\sim(c4)\}$ i.e. three maps are strongly shown in relationships among (A \sim \simeq T, G \sim C) for different cases respectively.

In addition, this set of results illustrates directly visual comparisons with stronger similarity between DNA and pseudo DNA on VMS maps, their similarly clustering distributions may indicate those maps with comparable mechanism to express real DNA sequences with extra A–T and G–C pairs of symmetric relationships in their higher levels of relationships applying the Stream Cipher mechanism.

5 Conclusion

This chapter proposes architecture to support the Variant Map System. Using a binary random sequence as input, a set of special pseudo DNA sequences can be generated. Under variant measures, probability measurement and normalized histogram, a pair of values can be determined by a series of controlled parameters. Collecting relevant pairs on multiple DNA sequences, four 2D maps can be generated.

The main results of this chapter provide the VMS architecture description in diagrams, main components, modules, expressions and important equations for the VMS. Core models and diagrams, sample results are illustrated to apply two types of data sets selected from real DNA sequences and generated from the pseudo random sequences from the Stream Cipher HC-256 for comparison under the VMS testing. After proper set of parameters selected, suitable visual distributions could be observed using the VMS. Results in Figs. 4, 5, 6, 7, 8 and 9 provide useful evidences systematically to support proposed VMS useful in checking higher levels of symmetric/similar properties among complex DNA sequences in both natural and artificial environment.

This construction could provide useful insights to spatial information on complex DNA expressions especially on large encoding RNA/DNA construction via 2D maps to explore higher levels of complex interactive environments in near future.

Acknowledgements Thanks to the school of software Yunnan University, to the key laboratory of Yunnan software engineering and the key laboratory for Conservation and Utilization of Bio-resource for excellent working environment, to the Yunnan Advanced Overseas Scholar Project (W8110305), the Key R&D project of Yunnan Higher Education Bureau (K1059178) and National Science Foundation of China (61362014) for financial supports to this project. This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018Z1002), NSF of China (61362014) and Yunnan Advanced Overseas Scholar Project.

References

1. ESTREAM project, <http://en.wikipedia.org/wiki/ESTREAM>
2. H.J. Wu, Stream Cipher HC-256, ESTREAM 2004, http://www.ecrypt.eu.org/stream/p3ciphers/hc/hc256_p3.pdf
3. M. Santha, U.V. Vazirani, Generating Quasi-Random Sequences from slightly random sources. *J. Comput. Syst. Sci.* **33**, 75–87 (1986)
4. G. Paul, S. Maitra. *RC4 Stream Cipher and Its Variants* (CRC Press, 2011)
5. M. Gude, Concept for a high-performance random number generator based on physical random noise. *Frequenz* **39**, 187–190 (1985)
6. D. Eastlake, S.D. Crocker, J.I. Schiller, Randomness requirements for security, in *RFC 1750* (Internet Engineering Task Force, 1994)
7. C. Plumb, Truly random numbers. *Dr. Dobbs J.* **19**(13), 113–115 (1994)
8. G.B. Agnew, Random source for cryptographic systems, in *Advanced in Cryptology—EURO-CRYPT'87 Proceedings* (Springer, 1988), pp. 77–81
9. A. Gehani, T. LaBean, J. Reif, DNA-based Cryptography, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 54, (2000) 233–249. <http://www.cs.duke.edu/~reif/paper/DNAcrypt/DNA5.DNAcrypt.pdf>

10. B.E. Bernstein, E. Birney, I. Dunham et al., An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**(7414), 57–74 (2012). <https://doi.org/10.1038/nature11247>. PMID22955616
11. E. Pennisi, Genomics. ENCODE project writes eulogy for junk DNA. *Science* **337** (6099), 1159, 1161, (2012). <https://doi.org/10.1126/science.337.6099.1159>. PMID22955811
12. Michael Schoöniger, Arndt von Haeseler, Simulating efficiently the evolution of DNA sequences. *Comput. Appl. Biosci.* **11**(1), 111–115 (1995). <https://doi.org/10.1093/bioinformatics/11.1.111>
13. F. Piva, G. Principato, RANDNA: a random DNA sequence generator, *Silico Biol.* **6**, 0024 (2006). <http://www.bioinfo.de/isb/2006060024/>
14. C.M. Gearheart, B. Arazi, E.C. Rouchka, DNA-based random number generation in security circuitry. *Biosystems* **100**(3), 208–214 (2010)
15. O. O. Babatunde, On pseudorandom number generation from programmable and computable biomolecules: deoxyribonucleic (DNA) as a novel pseudorandom number generator. *World Applied Programming*, **1**(3), 215–227 (2011)
16. G.C. Sirakoulis, Hybrid DNA cellular automata for pseudorandom number generation, in *International Conference on High Performance Computing and Simulation (HPCS)* (2012)
17. Y. Zhang, Y. Zhu, Z. Wang, R.O. Sinnott, Index-based symmetric DNA encryption algorithm, in *4-th International Congress on Image and Signal Processing (CSIP) 2011*. <http://dtl.unimelb.edu.au/researchfile287042.pdf>
18. Y. Zhang, L.H.B. Fu, in *Research on DNA Cryptography, Applied Cryptography and Network Security*, ed by J. Sen (InTech Press, 2012), pp. 357–376. <http://www.intechopen.com/books/applied-cryptography-and-network-security/research-on-dna-cryptography>
19. Erez Lieberman-Aiden et al., Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009). <https://doi.org/10.1126/science.1181369>
20. M.B. Gerstein, A. Kundaje, M. Hariharan et al. (2012) Architecture of the human regulatory network derived from ENCODE data, *Nature* 489 91–100, 2012. <https://doi.org/10.1038/nature11245>
21. J.M. Engreitz, A. Pandya-Jones, P. McDonel et al., Large noncoding RNAs can localize to regulatory DNA targets by exploring the 3D architecture of the genome, in *Proceedings of The Biology of Genomes* (Cold Spring Harbor Laboratory Press, 2013), p. 122
22. K. Sakamoto, Molecular computation by DNA hairpin formation. *Science* **283**, 1223–1227 (2000)
23. A. Arneodo, C. Vaillant, et al., Multi-scale coding of genomic information: From DNA sequence to genome structure and function. *Phys. Rep.* **498**(2), 45–188 (2011)
24. S. Engela, A. Alemany, NuriaForns, folding and unfolding of a triple-branch DNA molecule with four conformational states. *Phil. Mag.* **91**(13), 2049–2065 (2011)
25. J.M. Urquiza, I. Rojas, et al., Method for prediction of protein–protein interactions in yeast using genomics/proteomics information and feature selection, *Neurocomputing* **74**(16), 2683–2690 (2011)
26. H. Zhang, X. Liu, A CLIQUE algorithm using DNA computing techniques based on closed-circle DNA sequences. *Biosystems* **105**(1), 73–82 (2011)
27. B. Banfai, H. Jia, J. Khatun et al., Long noncoding RNAs are rarely translated in two human cell lines. *Genome Research*, Cold Spring Harbor Laboratory Press **22**, 1646–1657 (2012). <https://doi.org/10.1101/gr.134767.111>
28. W. Ford, Doolittle, is junk DNA bunk? A critique of ENCODE, in *Proceedings of the National Academy of Sciences* (2013)
29. J. Wang, M. Yan. *Numerical Methods in Bioinformatics* (Science Press, 2013)
30. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China.* **5**(2), 163–172 (2010). <http://www.springerlink.com/content/91474403127n446u/>.

31. W.Z. Yang, J. Zheng, Pseudo-random number generator based on variant logic model, in *ChinaCom 2012 Conference Proceedings* (2012)
32. J. Zheng, C. Zheng, T. Kunii, A framework of variant logic construction for cellular automata, in *Cellular Automata—Innovative Modelling for Science and Engineering*, ed by A. Salcido (InTech Press, 2011), pp. 325–352. <http://www.intechopen.com/chapters/20706>
33. J. Zheng, C. Zheng, T. Kunii, Interactive maps on variant phase spaces—from measurements—micro ensembles to ensemble matrices on statistical mechanics of particle models, in *Emerging Application of Cellular Automata*, ed by A. Salcido (InTech Press, 2013), pp. 113–196. <http://dx.doi.org/10.5772/51635>
34. W.Z. Yang, J. Zheng, Variant pseudo-random number generator, *Hakin9 Extra*, **6**(13), 28–31 (2012). <http://hakin9.org/hakin9-extra-62012/>
35. W.Q. Zhang, J. Zheng, Randomness Measurement of Pseudorandom Sequence Using different Generation Mechanisms and DNA Sequence. *J. Chengdu Univ. Inf. Technol.* **27**(6), 548–555 (2012)
36. N.A. Tchurikov, O.V. Kretova, D.M. Fedoseeva et al., DNA double-strand breaks coupled with parp1 and hnrnpa2b1 binding sites flank coordinately expressed domains in human chromosomes. *PLoS Genet.* **9**(4), e1003429 (2013). <https://doi.org/10.1371/journal.pgen.1003429>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Whole DNA Sequences of *Cebus capucinus* on Variant Maps



Yuyuan Mao, Jeffrey Zheng and Wenjia Liu

Abstract DNA sequences as a big data stream have been researched for years. However, researches on whole DNA sequences have various limitations to use existing research methods. A new scheme is proposed to map whole DNA sequences as 2D maps in this chapter, the whole DNA sequence of Capuchin monkey (*Cebus capucinus*) in apes was used as an example to demonstrate the mapping results.

Keywords Gene sequence · *Cebus capucinus* · Mapping method
Sequential model · Variant map

1 Introduction

In modern biologics, DNA sequences are being sequenced from wider species from human to simple cells in DNA data banks as big data streams. It is difficult to process various DNA streams for classification and identification on various species from whole sequences. The main task of present genomic research [1, 2] is to obtain

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014) and Yunnan Advanced Overseas Scholar Project.

Y. Mao
School of Software, Yunnan University, Kunming, China
e-mail: m805792943@foxmail.com

J. Zheng (✉)
Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

W. Liu
Yunnan University, Kunming, China
e-mail: 8avalon8@gmail.com

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_24

more biological information by processing and analyzing of the DNA sequence from multi-angles and multilevels [4–7]. In recent years, the processing and utilization of biological gene data are being carried out in a variety of ways, such as gene feature extraction, gene sequence location [7–9], and so on.

Variant map is an emerging technology to handle four symbols as meta-structure to process random sequences from cryptographic sequences, DNA sequences [3, 10] to ECG signals. Multiple statistical probability distributions are generated from selected sequences to form 2D–3D visual maps in representation. This scheme makes whole data sequences more compact and effectively visualized, and mapping results may be useful to explore nonlinear complex behaviors of whole genomics. A whole DNA sequence of a night monkey has mapped [11] on variant maps.

In this chapter, a special scheme is proposed to show a series of mapping results from a selected gene sequence of a capuchin monkey.

2 Process Model

A. Architecture

The architecture of the process model is shown in Fig. 1a. The process model consists of five parts: input, processing, measurement, projection, and output. There are three modules: Processing, Measurement, and Projection.

Input: A DNA sequence

Output: A 2D map

Modules: Processing, Measurement, and Projection

Process: From a selected DNA sequence, multiple segments are divided by a fixed length m on the whole sequence sequentially in the Processing module. Each segment needs to count four symbols: $\{A, C, G, T\}$ in the segment to transfer all segments into a measuring sequence of four measures in Measurement module. A special combination on $X: \{AT\}$ and $Y: \{AG\}$ is selected to determine four measures in a projection position and the whole measuring sequence projected to be a 2D map in Projection module.

B. Processing Module

From an input DNA sequence, multiple segments can be separated by a fixed length m to generate a sequence of segments.

Input: a DNA sequence

Output: a sequence of segments

C. Measurement Module

In this module, shown in Fig. 1b, each segment counts four numbers of $\{A, G, C, T\}$ in each proportions, respectively. As the result, each count is an integer number between 0 and m to transfer a segment sequence into a measuring sequence of four measures.

Input: a sequence of segments
 Output: a sequence of four measures

D. *Projection Module*

The projection module is shown in Fig. 1c as two units: Position and Projecting. For each four measures, two axis positions are determined by $X(AT)$ and $Y(AG)$, respectively. When all measures are processed, a 2D histogram is established as a statistical distribution as a 2D map.

Input: a sequence of four measures
 Output: a 2D map

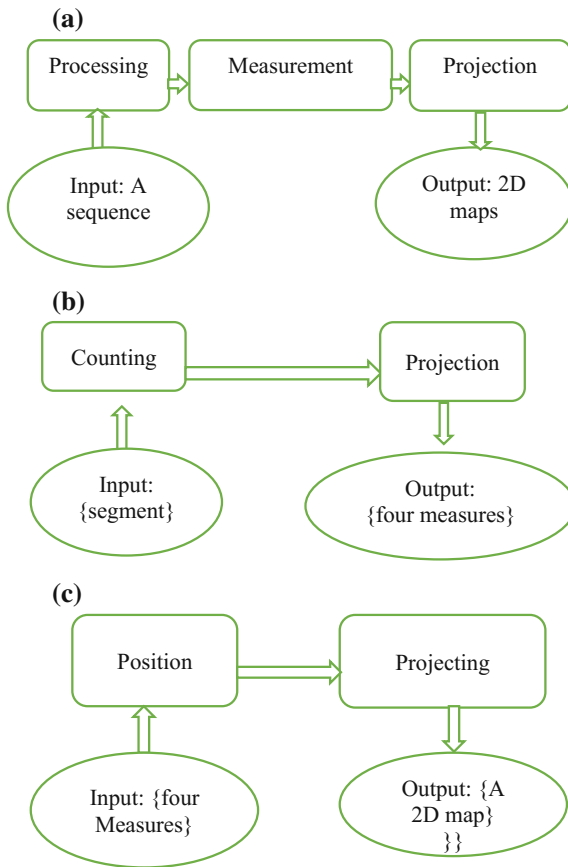


Fig. 1 Architecture of mapping scheme (a)–(c). **a** Architecture; **b** Measurement module; **c** Projection module

3 Details

A. Relevant Parameters

m : segment length

V : Two bases of combination: $\{AT, AG\}$

$$\text{num}(AT) = \text{num}(A) + \text{num}(T);$$

$$\text{num}(AG) = \text{num}(A) + \text{num}(G);$$

$$P_v = \text{num}(V)$$

P_v : The proportion of a base or combinatorial base

$(X_{P_{AT}}, Y_{P_{AG}})$: a pair of XY mapping positions.

B. Parameter in Module

Since the output quality of generating maps is dependent on the number of projection points, it is necessary for a refined map to include a larger number of coordinate points. The mapping projection forms the superposition to add up a larger number of coordinate points in 2D histogram representing a color map.

C. Measurement module.

- m : subsection length of a DNA sequence
- $\text{num}(AT) = \text{num}(A) + \text{num}(T)$
- V : AT or AG , $\{AT, AG\} \in D$.
- P_v : The proportion of AT or AG on the length of the sequence M .
- $P_v = \text{num}(V)/m$
- P : The proportion of AT
- P_{AG} : The proportion of AG
- $(X_{P_{AT}}^i, Y_{P_{AG}}^j)$: a pair of XY mapping coordinates. i, j are different subsections.

D. Parameter in Module

Calculating the proportion of AT and AG in the subsection according to the basic rules of mathematics. Two proportions can form a coordinate $(X_{P_{AT}}^i, Y_{P_{AG}}^j)$, which map a point on the two-dimensional graph.

The mapping relation between x and y :

$$X : P_{AT}$$

$$Y : P_{AG}$$

It is necessary for a distinct graph that includes a large number of coordinate points. Only a large number of DNA sequences can get a large number of coordinates points and pretty projection results. The graphics projection module completes the superposition of a large number of coordinate points.

4 Results Display

4.1 Maps on Various Segmented Length

Different parameters are shown in Fig. 2a–l for $m = \{20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 150, 200\}$, Fig. 3a–f for $m = \{54, 56, 58, 60, 62, 64\}$, Fig. 4a–d for $m = \{59, 60, 61, 62\}$ and Fig. 5 for $m = 60$, respectively.

In the map, similar color of pixels indicates the similar number of segments in the cluster.

4.2 Brief Analysis

From Fig. 2, it is interesting to notice that when $m < 50$, maps have more symmetric properties than larger numbers. Changing segmented lengths, significant patterns appear in $m = 54–64$ region shown in Fig. 3 and refined lengths are shown in Fig. 4.

From a visual observation, when $m = 60$, the map has shown the better effects.

5 Conclusion

Using the proposed mapping scheme, it is feasible to transfer a whole DNA sequence as a color map with significant visual features. In addition to mapping method and selected functions, a set of sample sequences in various segmented lengths illustrate colorful distributions as variant maps.

Checking symmetric information among different maps, it is possible to identify specific spatial features under different configurations.

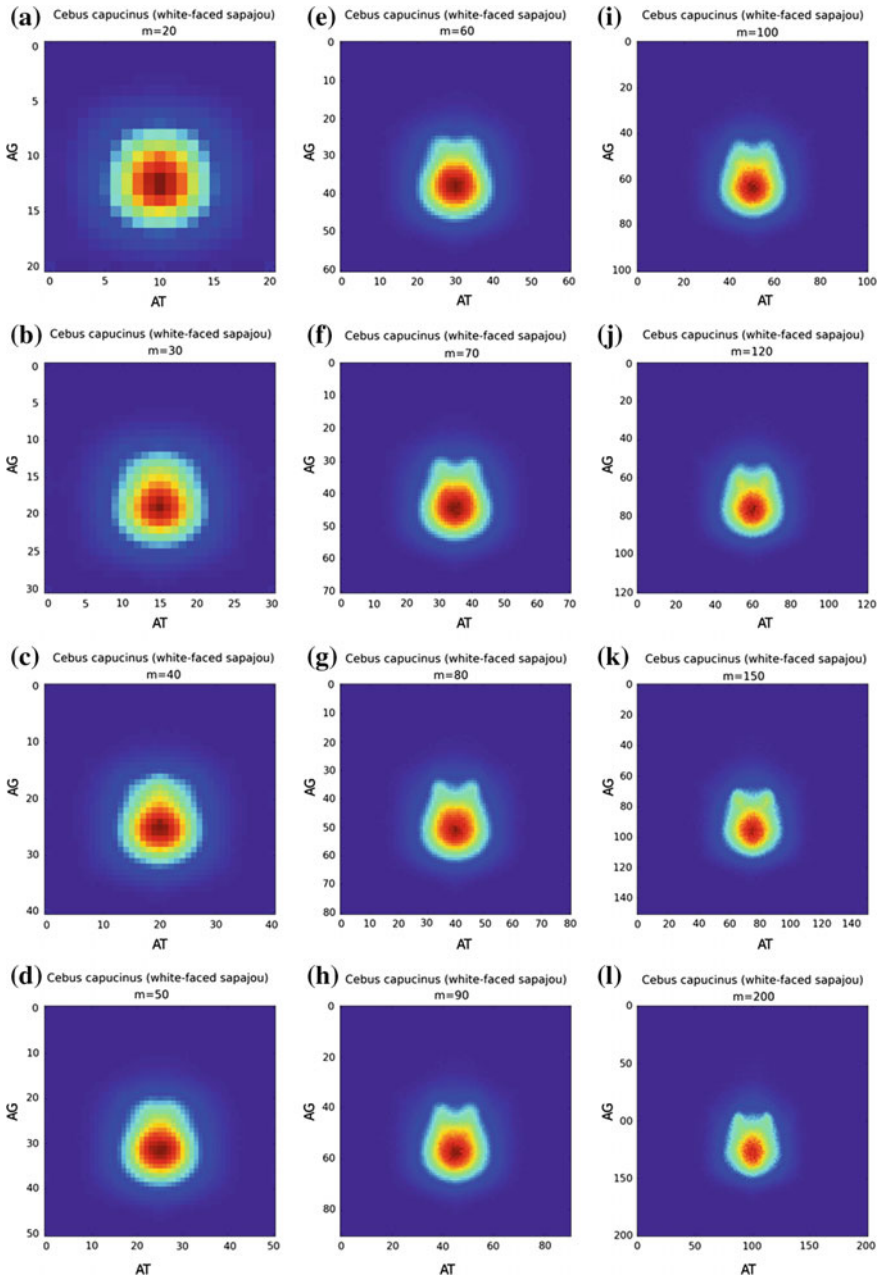


Fig. 2 Variant maps of *Cebus capucinus* on various segmented lengths (a)–(l) $m = \{20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 150, 200\}$. **a** $m = 20$; **b** $m = 30$; **c** $m = 40$; **d** $m = 50$; **e** $m = 60$; **f** $m = 70$; **g** $m = 80$; **h** $m = 90$; **i** $m = 100$; **j** $m = 120$; **k** $m = 250$; **l** $m = 200$

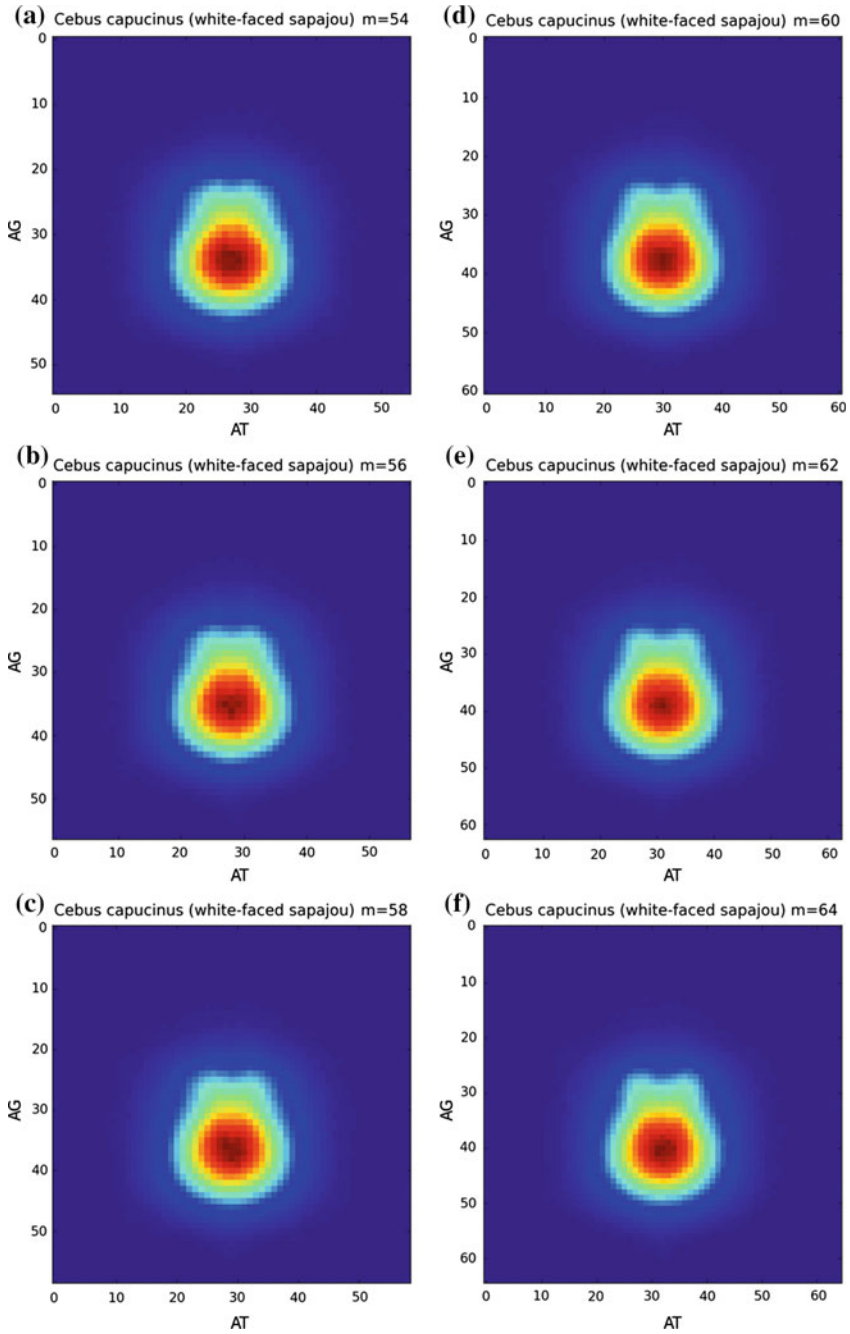


Fig. 3 Variant maps of *Cebus capucinus* on various segmented lengths (a)–(f) $m = \{54, 56, 58, 60, 62, 64\}$; **a** $m = 54$; **b** $m = 56$; **c** $m = 58$; **d** $m = 60$; **e** $m = 62$; **f** $m = 64$

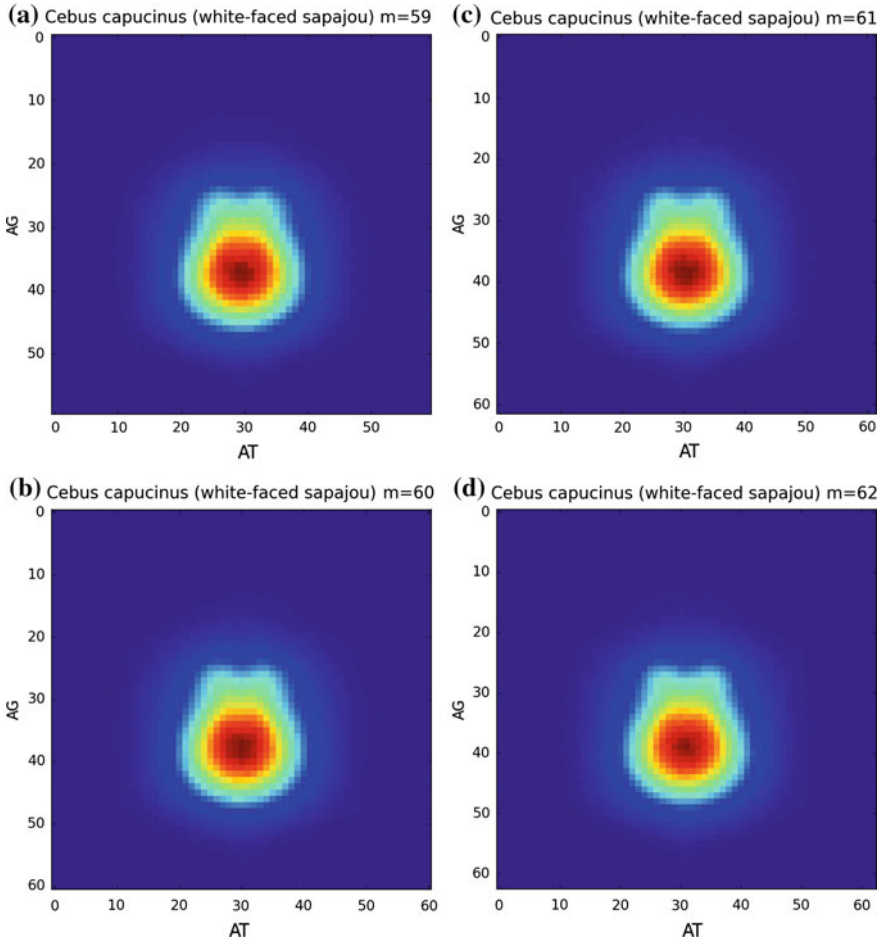


Fig. 4 Variant maps of *Cebus capucinus* on various segmented lengths (a)–(d) $m = \{59, 60, 61, 62\}$. **a** $m = 59$; **b** $m = 60$; **c** $m = 61$; **d** $m = 62$

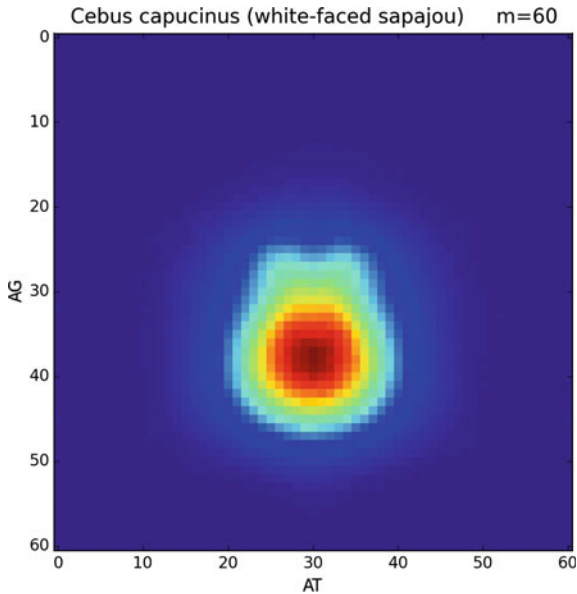


Fig. 5 Variant maps of *Cebus capucinus* on segmented lengths $m = 60$

Since this is an initial step to make a whole DNA sequence in mapping operation, further researches and explorations are required.

References

1. J.A. Berger, S.K. Mitra, M. Carli, A. Neri, Visualization and analysis of DNA sequences using DNA walks. *J. Franklin Inst.* **341**(1/2) (2004)
2. J.N. Pitt, I. Rajapakse, A.R. Ferré-D'Amaré, SEWAL: an open-source platform for next-generation sequence analysis and visualization. *PMC* **38**(22), 7908–7915 (2010)
3. L. Yuqian, Z. Zhijie, The Visual Analysis of Coding and Non-Coding DNA Sequences. *Hans J. Comput. Biol.* **4**, 20–31 (2014)
4. J. Hellman, S. Drucker, N.R. Riche, B. Lee, A deeper understanding of sequence in narrative visualization. *IEEE Trans. Vis. Comput. Graph.* **19**(12) (2013)
5. G.-D. Sun, Y.-C. Wu, R.-H. Liang, S.-X. Liu, A survey of visual analytics techniques and applications: state-of-the-art research and future challenges (2013). <https://doi.org/10.1007/s11390-013-1383-8>
6. J. Batley, D. Edwards, Genome sequence data: management, storage, and visualization. *BioTechniques* **46**(5) (2009)
7. Y. Nakamura, T. Gojobori, T. Ikemura, Codon usage tabulated from international DNA sequence databases: status for the year 2000. *Nucl. Acids. Res.* **28**, 292 (2000)
8. N. Rusk, Focus on next-generation sequencing data analysis. *Nat. Methods* **6**, S1 (2009)
9. R. Durrett, *Probability Models for DNA Sequence Evolution* (Springer, 2008)

10. J. Zheng, W. Zhang, J. Luo, W. Zhou, R. Shen, Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Math.* **3**(7A), 5–24 (2013)
11. Y. Mao, J. Zheng, W. Liu, Mapping Whole DNA Sequence on Variant Maps, *ASONAM '17 Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pp. 1037–1040

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Part IX

Applications—Multiple Valued Sequences

Experience without theory is blind,
but theory without experience is mere intellectual play.

—*Immanuel Kant*

Make everything as simple as possible, but not simpler.

—*Albert Einstein*

Science cannot progress without reliable and accurate measurement
of what it is you are trying to study.

The key is measurement, simple as that.

—*Robert D. Hare*

Processing multiple valued sequences, it is necessary to use more complex structures in transformation. Various signals such as ECG, EEG, and BEC (Bat Echolocation Calls) were tested. From 2016, various papers were published on ECG processing. For example, Variant Maps on Normal and Abnormal ECG Data Sequences, *Biol Med (Aligarh)* 8:336. <https://doi.org/10.4172/0974-8369.1000336>; Mapping ECG Signals on Variant Maps, <https://doi.org/10.1145/3110025.3110134>; Visualization of P wave characteristics in ECG, <https://doi.org/10.1109/CISP-BMEI.2017.8302247>.

This part of multiple valued sequences is composed of two chapters (25 and 26).

Chapter “[Successful Creation of Regular Patterns in Variant Maps from Bat Echolocation Calls](#)” processes BEC signals on variant maps to identify variant maps into two distinct groups.

Chapter “[Visual Analysis of ECG Sequences on Variant Maps](#)” uses visual analysis of ECG sequences on variant maps; various normal and abnormal ECG sequences are selected in comparison. Significant characteristics of various distributions are observed.

Successful Creation of Regular Patterns in Variant Maps from Bat Echolocation Calls



D. M. Heim, O. Heim, P. A. Zeng and Jeffrey Zheng

Abstract We created variant maps based on bat echolocation call recordings and outline here the transformation process and describe the resulting visual features. The maps show regular patterns while characteristic features change when bat call recording properties change. By focusing on specific visual features, we found a set of projection parameters which allowed us to classify the variant maps into two distinct groups. These results are promising indicators that variant maps can be used as basis for new echolocation call classification algorithms.

Keywords Echolocation · Algorithms · Morphometry · Fourier · Analysis
Quaternions

This work was supported by NSF of China (61362014), Yunnan Advanced Overseas Scholar Project and the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002).

D. M. Heim

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: dennis.heim@gmx.net

O. Heim

Leibniz Institute for Zoo and Wildlife Research, 10315 Berlin, Germany

O. Heim

Animal Ecology, Institute of Biochemistry and Biology, University of Potsdam, 14469 Potsdam, Germany
e-mail: bats@o-heim.de

P. A. Zeng

Yunnan University, Kunming, China
e-mail: 895158562@qq.com

J. Zheng (✉)

Key Laboratory of Yunnan Software Engineering, Yunnan University,
Yunnan 650091, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_25

1 Introduction

The identification of echolocation calls is essential to the research and conservation of bat species [1]. However, automatic classification algorithms have not yet been proven capable of providing 100% correct classifications or getting close enough to this ideal performance [2]. Since our approach of using variant maps [3] shows already promising results, we are confident that it will continue adding valuable contributions to the field of automatic bat call identification.

Automated bat echolocation call identification algorithms were developed since the late 1990s [4–7]. At that time, multivariate discriminant function analysis or neural networks were used for the classification of the calls. Since then, other methods have been applied, e.g., algorithms of pattern recognition [8], support vector machines [9], hierarchical ensembles of neural networks [9, 10], geometric morphometry [11], machine learning [12], CART [13], and random forest classification [14]. For a critical analysis of the performance of the applied methods, we refer to [2] and the references therein.

Using variant maps for the classification of bat echolocation calls differ completely from these conventional techniques. The main difference is the preprocessing step, where the recordings are transformed into variant maps. This step offers the possibility to analyze the bat call recordings from a completely different point of view. It provides additional degrees of freedom which allow a further optimization of the identification process, e.g., by supplementing the information obtained from a Fourier analysis of the bat calls.

Our method to transform the bat call recordings is based on measures proposed by Zheng [15] in the 1990s to partition special phase spaces in binary image analysis. These methods were extended in the 2010s [3, 16] and successfully used to classify quantum interactions [17, 18], differently encrypted messages [19], and noncoding DNA [20, 21].

Similar to these works, we transform the bat call recordings using variant measures to obtain variant maps. Each recording contains several calls of one bat species. We used calls of four aerial-hawking bat species in this study. Recordings were made at three types of crop fields far away from woody vegetation. The created variant maps have a regular structure, but characteristic features vary strongly with each recording. These results show that variant maps can be used to extract usable information from bat echolocation recordings.

2 Transformation

The processed bat echolocation calls were recorded with a sampling rate of 500 kHz and saved as “raw” 16-bit audio files. In the following, we describe in four steps (A–D) how we transformed these files into variant maps.

Step A: From analogue to digital audio

In a recording of data length N , the amplitude of the bat echolocation calls is stored in N samples. Each sample corresponds to a floating-point number of 16 bits. For simplicity, we transformed the floating-point numbers to integer numbers of 16 bits.

Step B: From digital audio to quaternions

Next, we transform the integer sequence into a sequence of four metastates $\{\perp, +, -, \top\}$ which resemble the quaternions {Bottom, Plus, Minus, Top}. For this step, we select the i -th sample A_i and its next neighbor A_{i+1} and define the difference $\Delta A = A_{i+1} - A_i$ and local average $L = (A_i + A_{i+1})/2$. Additionally, we require the maximum A_{\max} and minimum A_{\min} of the current sequence to define a middle value $V = (A_{\min} + A_{\max})/2$ and we define a tolerance T . Using these values, we transform the integer sequence $A_1 \cdots A_N$ into a sequence of quaternions $B_1 \cdots B_N$ using the rules

$$\begin{aligned}
 &\text{if } \Delta A < T \text{ and } L > V : B_i = \top \\
 &\text{if } \Delta A < T \text{ and } L \leq V : B_i = \perp \\
 &\text{if } \Delta A \geq T \text{ and } A_i > A_{i+1} : B_i = - \\
 &\text{if } \Delta A \geq T \text{ and } A_i < A_{i+1} : B_i = + \text{ .}
 \end{aligned}$$

As an example, the values $T = 4$ and $V = 10$ lead to the sequence

A_i	0	3	3	2	0	8	20	20	11
A_{i+1}	3	0	8	6	4	3	15	18	13
B_i	\perp	\perp	+	+	+	-	-	\top	\top

Step C: From quaternions to meta-measures

We subdivide the quaternion sequence into segments of length M and obtain, in this way, $S = N/M$ segments. For each segment, we define four meta-measures $\{M_{\perp}, M_+, M_-, M_{\top}\}$. One measure represents the number of associated quaternions in one segment. These meta-measures satisfy the relations $0 \leq M_{\perp}, M_+, M_-, M_{\top} \leq M$ and $M_{\perp} + M_+ + M_- + M_{\top} = M$. The quaternion sequence with N units is now represented by S segments where each segment contains four meta-measures.

Step D: From meta-measures to variant maps

There are many possibilities to combine meta-measures for the creation of variant maps [3, 15–21]. To transform the bat echolocation calls into 2D color maps, we defined for each segment of meta-measures the axis values $X = M_+ + M_{\perp}$ and $Y = M_{\perp} + M_- + M_{\top}$. One Z value is obtained by counting the number of segments where one specific X - Y combination was found. Each Z value is represented by a color in an $(M + 1) \times (M + 1)$ matrix.

As an example, we depicted in Fig. 1 the variant map of an echolocation call recording from the bat species *Nyctalus noctula*. It has a data length $N = 967,139$ and we chose a segment length $M = 237$. At the position $X = 80$ and $Y = 200$ marked by a white circle, the color indicates a value $Z = 10$. That is, we found 10 segments where the conditions $M_+ + M_{\perp} = 80$ and $M_{\perp} + M_- + M_{\top} = 200$ apply.

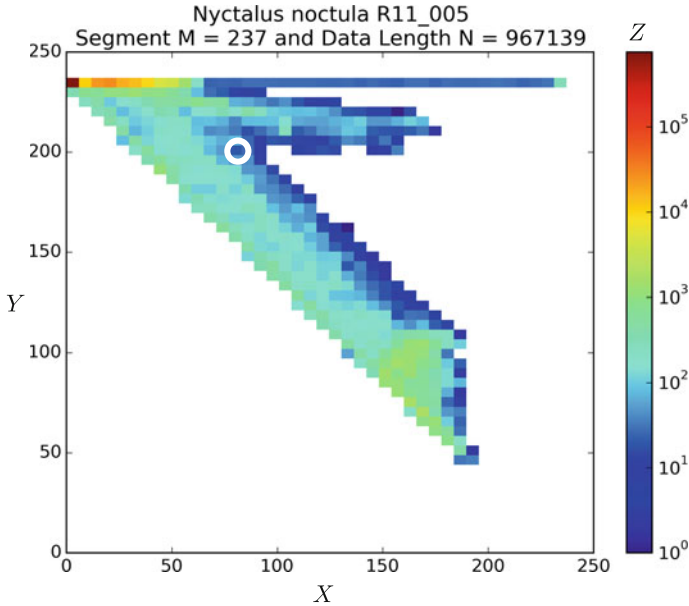


Fig. 1 The variant map of an echolocation call recording from the species *Nyctalus noctula* created by following the processing steps A–D described in Sect. 2. We highlighted the position $X = 80$ and $Y = 200$ by a white circle to illustrate the processing step D. At this position, the conditions $M_+ + M_\perp = 80$ and $M_\perp + M_- + M_\top = 200$ apply. Further visual features are discussed in Sect. 3 in more detail

White areas indicate regions without any projection point on this sequence. For a discussion of further visual features which appear in this figure we refer to Sect. 3.

These types of maps offer the possibility to visualize long data sequences with $>10^6$ samples on compact matrices. We use this scheme to transform each bat call recording into a 2D color figure. It can be optimized for the identification of bat species, recording locations or times.

3 Variant Maps

Our main result is that all variant maps created from bat echolocation calls show regular patterns while characteristic visual features vary with each recording. In the following, we describe the data we processed in detail and discuss the visual features we observed.

3.1 Data Description

We processed 44 files which were recorded in August 2012 in the Uckermark region (Brandenburg, Germany) [22]. Each recording contains only calls of one of the four European bat species *Nyctalus noctula*, *Pipistrellus nathusii*, *Pipistrellus pipistrellus*, or *Pipistrellus pygmaeus*. These files were recorded on arable fields cultivated with three different crop types: corn (C), rapeseed (R), or wheat (W). The record length varies between 30 s and 2 min.

3.2 Visual Features

We transformed all 44 files of bat calls into variant maps by steps A to D described in Sect. 2. That is, we used the axis values $X = M_+ + M_\perp$ and $Y = M_\perp + M_- + M_\top$ and a segment length $M = 237$. By focusing on the visual features, we clustered the resulting maps into two groups. A typical member of each group is shown in Fig. 2.

One group consists only of maps showing patterns which have two significant maxima with values $> 10^5$. We call members of this group **double-maxima** maps. The example shown in Fig. 2a has maxima at the positions $X=0, Y=237$ and $X=120, Y=200$. Besides these two maxima, there are distinct positions on diagonal areas with values of the orders $1-10^3$.

All other maps belong to the group of **non-double-maxima** maps. As an example, the map in Fig. 2b has its significant maximum at the position $X=0, Y=237$ while other projection regions have values of the orders $1-10^3$. In addition, most values of interest are located around a diagonal region and form a slat band on the map.

All 44 resulting maps are shown in Figs. 3 and 4. They are separated into **double-maxima** maps (Fig. 3) and **non-double-maxima** maps (Fig. 4). In principle, it is possible to further subdivide the variant maps by identifying additional visual features. However, since we did not yet find a direct connection between visual features and bat call properties, a further subdivision goes beyond the scope of this manuscript and will be the topic of a future publication.

3.3 Discussion

On all generated maps, the positions on the left-down triangle area are empty. This is because our choice of axis obeys $X + Y \geq M$. Empty positions in the right-upper area appear because the bat call recordings consist of discrete short pulses with a longer time period of silence in between.

Similarly, other visual characteristics in the colored areas can be directly related to properties of the bat call recordings. As an example, a signal of constant frequency can be transformed into a single position on a variant map by choosing suitable

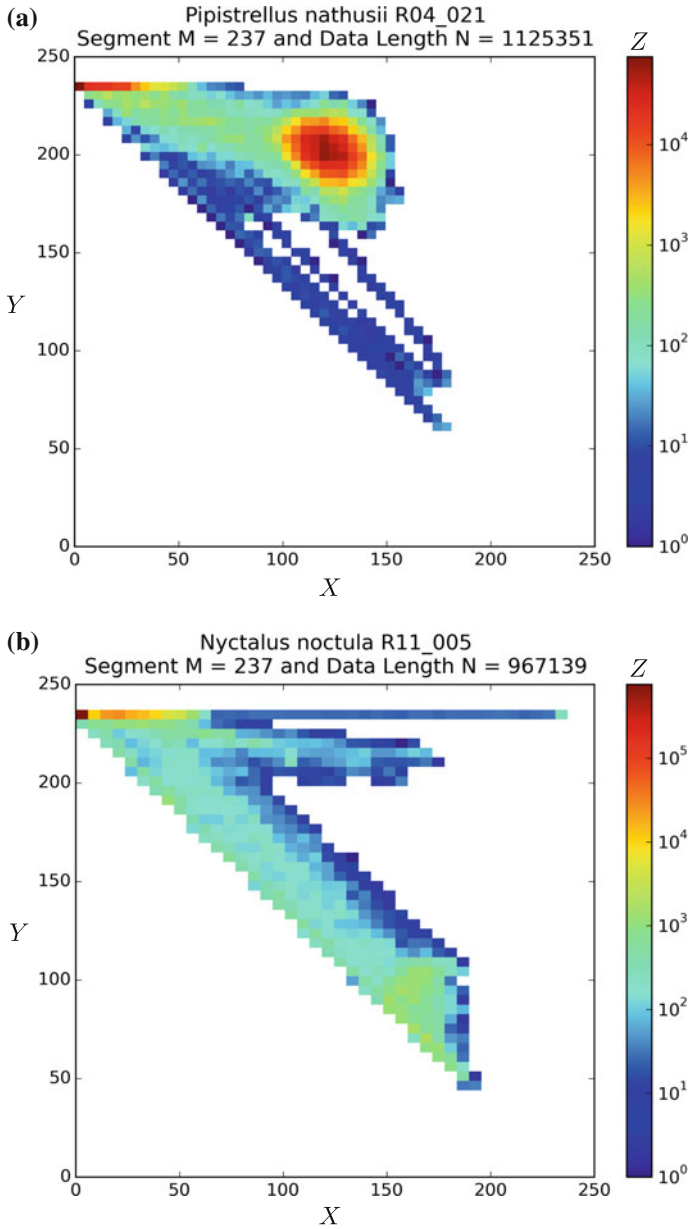


Fig. 2 Variant maps of **a** *Pipistrellus nathusii* and **b** *Nyctalus noctula*, both recorded on a rapeseed field. The figures were created by applying the transformation process described in Sect. 2. **a** which shows a typical **double-maxima** map with two significant maxima, while **b** belongs to the group of **non-double-maxima** maps

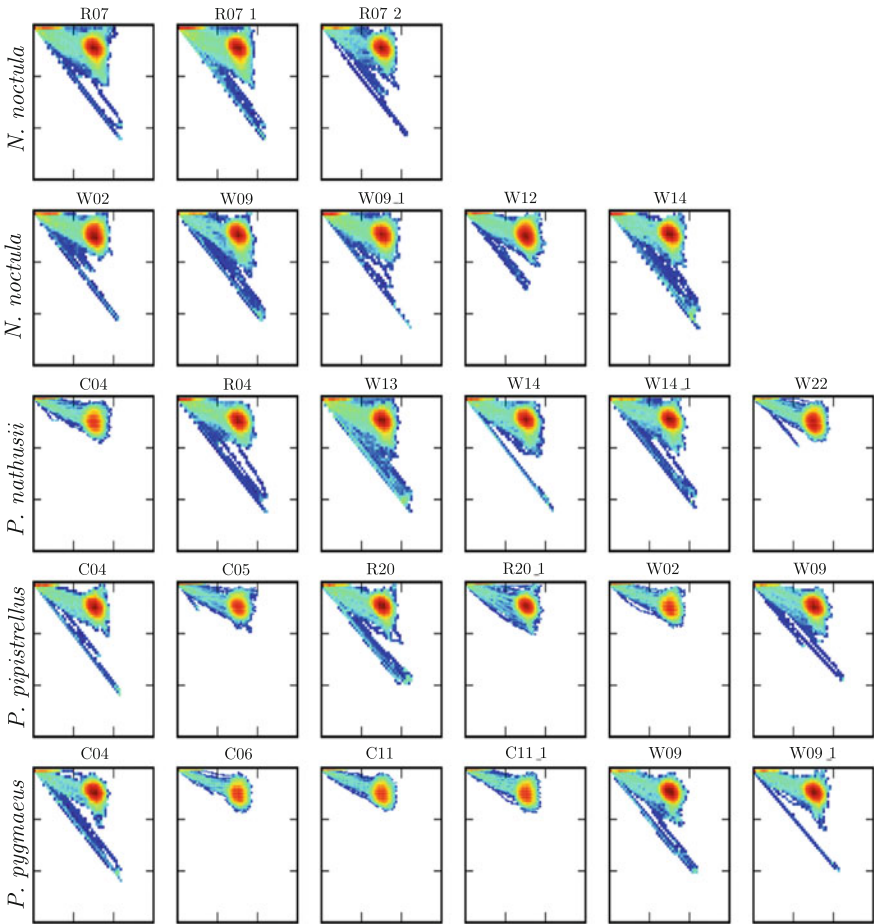


Fig. 3 These variant maps show **double-maxima** patterns. They have two significant maxima with values $>10^5$. The axis ranges are the same as in Fig. 2. Each map originates from a bat echolocation recording on a corn (C), rapeseed (R), or wheat (W) field

parameters. This means that by optimizing the variant map transformation, it is possible to focus on features of the initial bat echolocation call for the creation of variant maps.

This is the first time to our knowledge that quaternion structures have been used to transform bat calls. Our transformation process could be used to add optimizing parameters to current bat call identification schemes and in this way form the basis for a new identification algorithm.

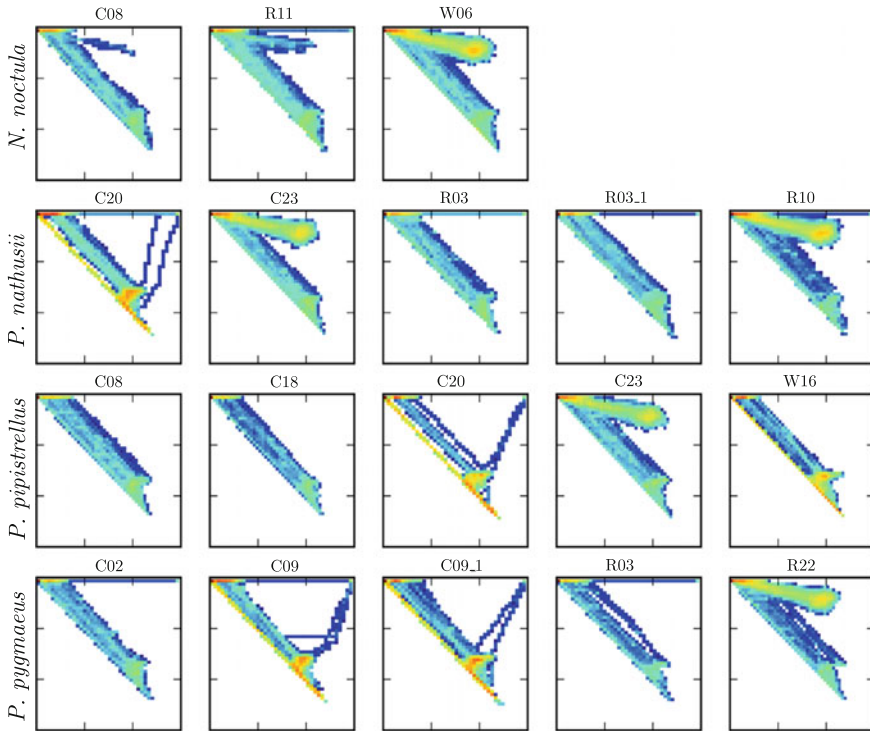


Fig. 4 These variant maps show **non-double-maxima** patterns. That is, they explicitly do not have two distinct maxima with values $>10^5$ in contrast to the **double-maxima** maps shown in Fig. 3

4 Summary and Outlook

We transformed 44 bat echolocation files into variant maps. All created variant maps have a similar structure and can be classified by focusing on specific visual features. As an example, we found a set of projection parameters which allowed us to classify the recordings into **double-maxima** and **non-double-maxima** maps.

Features like this can be traced back to the signal nature of the recordings. In this way, variant maps offer the possibility to focus on individual features of bat echolocation calls. Since there are multiple numbers of possible combinations to create variant maps, we are very positive that a suitable projection combination can be found to fulfill our ultimate goal of identifying single bat species.

In order to meet this target, it is necessary to process a much higher number of bat calls to create a sufficiently large database for the effective determination of possible projections and associated maps. This would form the perfect basis for the development of a new echolocation call identification algorithm.

Acknowledgements We thank C. C. Voigt for providing the processed bat echolocation data and S. A. Troxell for revising the manuscript. Financial support by the National Science Foundation of China NSFC (No. 61362014) and the Overseas Higher-level Scholar Project of Yunnan Province, China, is gratefully acknowledged. Moreover, we appreciate the financial support by the Federal Ministry of Science, Research and Culture in Brandenburg, the University of Potsdam, the Leibniz Institute for Zoo and Wildlife Research and the Deutsche Forschungsgemeinschaft (Vo 890/22).

References

1. H.-U. Schnitzler, E.K.V. Kalko, Echolocation by insect-eating bats. *Bioscience* **51**(7), 557 (2001). [https://doi.org/10.1641/0006-3568\(2001\)051\[0557:EBIEB\]2.0.CO;2](https://doi.org/10.1641/0006-3568(2001)051[0557:EBIEB]2.0.CO;2)
2. D. Russo, C.C. Voigt, The use of automated identification of bat echolocation calls in acoustic monitoring: a cautionary note for a sound analysis. *Ecol. Indic.* **66**, 598 (2016). <https://doi.org/10.1016/j.ecolind.2016.02.036>
3. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* **5**(2), 163–172 (2010). <https://doi.org/10.1007/s11460-010-0011-4>
4. P.E. Zingg, Akustische Artidentifikation von Fledermäusen (Mammalia: Chiroptera) in der Schweiz, *Revue suisse de zoologie*, vol 97 (1990). <http://www.biodiversitylibrary.org/part/92388>, Genve, Kundig [etc.], 263
5. N. Vaughan, G. Jones, S. Harris, Identification of British bat species by multivariate analysis of echolocation call parameters. *Bioacoustics* **7**(3), 189 (1997). <https://doi.org/10.1080/09524622.1997.9753331>
6. S. Parsons, G. Jones, Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. *J. Exp. Biol.* **203**(17), 2641 (2000). <http://jeb.biologists.org/content/203/17/2641>
7. D. Russo, G. Jones, Identification of twenty-two bat species (Mammalia: Chiroptera) from Italy by analysis of time-expanded recordings of echolocation calls. *J. Zool.* **258**(1), 91 (2002). <https://doi.org/10.1017/S0952836902001231>
8. M.K. Obrist, R. Boesch, P.F. Flückiger, Variability in echolocation call design of 26 Swiss bat species: consequences, limits and options for automated field identification with a synergetic pattern recognition approach. *Mammalia mamm* **68**(4), 307 (2007). <https://doi.org/10.1515/mamm.2004.030>
9. R.D. Redgwell, J.M. Szweczek, G. Jones, S. Parsons, Classification of echolocation calls from 14 species of bat by support vector machines and ensembles of neural networks. *Algorithms* **2**(3), 907 (2009) *Molecular Diversity Preservation International*. <https://doi.org/10.3390/a2030907>
10. C.L. alters, R. reeman, A. Collen, C. Dietz, M. Brock Fenton, G. Jones, M.K. Obrist, S.J. Puechmaille, T. Sattler, B.M. Siemers, S. Parsons, K.E. Jones, A continental-scale tool for acoustic identification of European bats. *J. Appl. Ecol.* **49**(5), 1064 (2012). <https://doi.org/10.1111/j.1365-2664.2012.02182.x>
11. N. MacLeod, J. Krieger, K.E. Jones, Geometric morphometric approaches to acoustic signal analysis in mammalian biology. *Hystrix. Ital. J. Mammal.* **24**(1), 110 (2013). <https://doi.org/10.4404/hystrix-24.1-6299>
12. M.D. Skowronski, J.G. Harris, Acoustic detection and classification of microchiroptera using machine learning: lessons learned from automatic speech recognition. *J. Acoust. Soc. Am.* **119**(3), 1817 (2006). <https://doi.org/10.1121/1.2166948>
13. D.G. Preatoni, M. Nodari, R. Chirichella, G. Tosi, L.A. Wauters, A. Martinoli, Identifying bats from time-expanded recordings of search calls: comparing classification methods. *J. Wildlife Manage.* **69**(4), 1601 (2005). [https://doi.org/10.2193/0022-541X\(2005\)69\[1601:IBFTRO\]2.0.CO;2](https://doi.org/10.2193/0022-541X(2005)69[1601:IBFTRO]2.0.CO;2)

14. U. Marckmann, V. Runkel, Die automatische Rufanalyse mit dem batcorder-System. Version 1.01. ecoObs GmbH, (2010). <http://www.ecoobs.de/downloads/Automatische-Rufanalyse-1-0.pdf>
15. Z.J. Zheng, in *Conjugate Transformation of Regular Plan Lattices for Binary Images* (Monash University, 1994)
16. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, in *A Framework of Variant Logic Construction for Cellular Automata, Cellular Automata - Innovative Modelling for Science and Engineering*, ed. by A. Salcido (InTech, 2011), pp. 326–352. <https://doi.org/10.5772/15400>, <http://www.intechopen.com/books/cellular-automata-innovative-modelling-for-science-and-engineering/a-framework-of-variant-logic-construction-for-cellular-automata>
17. J. Zheng, C. Zheng, Variant measures and visualized statistical distributions. *Acta Photonica Sinica* **40**(9), 1397–1404 (2011). http://www.photon.ac.cn/EN/abstract/article_15668.shtml, <https://doi.org/10.3788/gzxb20114009.1397>
18. J. Zheng, C. Zheng, Variant simulation system using quaternion structures. *J. Mod. Opt.* **59**(5), 484–490 (2012). <https://doi.org/10.1080/09500340.2011.636152>
19. H. Wang, J. Zheng, in *Proceedings of the 14th Australian Information Warfare and Security Conference*, vol. 14, Perth (2013). <http://ro.ecu.edu.au/isw/53>
20. J. Zheng, J. Luo, W. Zhou, Pseudo DNA sequence generation of non-coding distributions using variant maps on cellular automata. *Appl. Math.* **5**(1), 153 (2014). <https://doi.org/10.4236/am.2014.51018>
21. J. Zheng, W. Zhang, J. Luo, W. Zhou, R. Shen, Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Math.* **3**, 5 (2013) Scientific Research Publishing. <https://doi.org/10.4236/apm.2013.37A002>
22. O. Heim, A. Schröder, J. Eccard, K. Jung, C.C. Voigt, Seasonal activity patterns of European bats above intensively used farmland. *Agric. Ecosyst. Environ.* **130**, 233 (2016). <http://www.media/dheim/Bond/Desktop/physics/paper/olga/2016-heim-seasonality.pdf>, <https://doi.org/10.1016/j.agee.2016.09.002>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



Visual Analysis of ECG Sequences on Variant Maps



Zhihui Hou and Jeffery Zheng

Abstract This chapter presents the variant measurement based on the variant logic, which uses the ECG sequence as the signal source, and outputs the variant maps of ECG sequences. It provides a supplementary study for ECG detection. Samples of ECG signal are collected from the First People's Hospital of Yunnan Province. Under variant maps, main parameters of various interval values are checked and corresponding maps are illustrated.

Keywords Arrhythmia · Visualization · ECG sequences · Variant map

1 Introduction

The world is concerned about the cardiovascular disease [1]. Mainly relying on the detection of ECG signals to promote research on related issues of cardiovascular diseases. The electrocardiogram represents cardiac function and graphic signals [2], which is an important means of diagnosing abnormal cardiac activity.

ECG signals are the product of a wide range of clinical ECG techniques. In recent years, research methods for ECG signals have made significant progress, such as using machine learning [3], neural network, clustering [4], partial fractal dimension [5], wavelet transform [6], and other methods to classify the detection of arrhythmia. The most typical representative of the emerging ECG research method is ECG scatter gram [7–9].

Project supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZJ002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

Z. Hou
Yunnan University, Kunming, China
e-mail: 1660919714@qq.com

J. Zheng (✉)
Key Laboratory of Yunnan Software Engineering, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

© The Author(s) 2019
J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_26

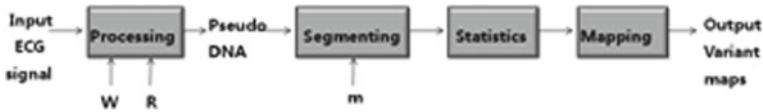


Fig. 1 The overall structure of the variant map for ECG

The variant method is an emerging technique for dealing with spatial changes in signal phase. Since the 1990s, the application of the variant method in processing binary image classification and transformation [10, 11] had been proposed, and the variant method has been perfected until now [12, 13]. Variant method is applied to different data samples: quantum sequences [14, 15], random sequences [16], non-coding DNA [17–19], bat echo signals [20], and electrocardiographic signals [21, 22], and effective research results have been obtained in these samples.

This chapter is a further study of the use of variant measurements in the detection of ECG sequences. The sample ECG signals are provided by the First People’s Hospital of Yunnan Province. In this chapter, two groups of signals are used: normal ECG signal and abnormal ECG signal groups. In the second part of this chapter, we describe variant map for ECG. Showing sample results and making a brief analysis in the third part, the last part is the summary of the chapter.

2 Variant Map for ECG

Variant map for ECG is composed of six parts: Input, Processing, Segmenting, Statistics, Mapping, and Output. Figure 1 is the overall structure of the variant map for ECG, which specific content about each part in the following description:

A. Input Part

Testing ECG signals are provided by the hospital as a data source. Let ECG signals be p with N elements.

$$p = \{p_0, \dots, p_{N-1}\}$$

B. Processing Part

In processing part, a multivalve ECG signal sequence will be transformed into a four-valued pseudo-DNA sequence.

Input: the ECG sequence

$$p = \{p_0, \dots, p_{N-1}\}$$

Parameters: W sliding window value; R interval value.

Output: a four-valued pseudo-DNA sequence

$$q = \{q_0, \dots, q_{N-1}\}$$

Processing:

Let \bar{p}_i be an average value; r be a range value; t_i be a conversion value. Three values are calculated in the equations:

$$\bar{p}_i = \sum_{i=0}^{N-1} \frac{p_i}{W}$$

$$p_{\max} = \max\{p_i\}, 0 \leq i < N - 1$$

$$p_{\min} = \min\{p_i\}, 0 \leq i < N - 1$$

$$r = (p_{\max} - p_{\min}) * \frac{R}{2}$$

$$t_i = \frac{2(p_i - \bar{p}_i)}{r * R}$$

Transforming rules: $0 \leq i < N - 1$

if $t_i > R > 0 : q_i = A$; if $0 < t_i < R : q_i = G$;

if $0 > t_i > -R : q_i = C$; if $0 > -R > t_i : q_i = T$;

C. Segmenting Part

Input: $q = \{q_0, \dots, q_{N-1}\}$.

Parameters: m is a segment value.

Output: $Q = \{Q_0, \dots, Q_j, \dots, Q_{M-1}\}, 0 \leq j < M$; M is segments and $N = m * M$.

Processing: the j -th element in $Q = \{Q_0, \dots, Q_j, \dots, Q_{M-1}\}$;

$$Q_j = \{q_{j*m}, \dots, q_{j*m+i}, \dots, q_{j*m+m-1}\}, 0 \leq i < m, 0 \leq j < M.$$

D. Statistics Part

Input: $Q = \{Q_0, \dots, Q_j, \dots, Q_{M-1}\}, 0 \leq j < M$

Output: $S = \{S_j^A, S_j^C, S_j^G, S_j^T\}, 0 \leq j < M$

S_j^A is value of the number of A element in Q_j

S_j^C is value of the number of C element in Q_j

S_j^G is value of the number of G element in Q_j

S_j^T is value of the number of T element in Q_j

E. Mapping Part

Selecting a pair of two elements in $S = \{S_j^A, S_j^C, S_j^G, S_j^T\}$, $0 \leq j < M$, as a mapping object. This chapter selects (S_j^C, S_j^G) . S_j^C is corresponding to the X -axis and S_j^G is corresponding to the Y -axis. All M pairs are mapping to the 2D map as output.

F. Output

The results of the mapping are output in the form of 2D variant maps.

3 Sample Results and Brief Analysis

Visualization results of ECG signal obtained by variant map for ECG show that the morphological features of ECG signals have regular changes. Sample results are illustrated and a brief analysis is described.

A. Data Source Description

The ECG signals in this chapter are provided by the First People Hospital of Yunnan Province. The ECG signals contain a total of 202,626 cases. There are 104,742 normal cases and 97,884 abnormal cases of records. For this experiment, 97,884 normal cases and 97,884 abnormal cases were selected.

Since ECG signals have multiple attributes, this chapter chooses the attributes of the P wave samples to be processed. Figure 2 is the sample of part of abnormal ECG data source.

B. Visualization Features

Using the variant map for ECG, multiple maps can be generated.

The interesting finding is that the changes of the parameters affect the spatial characteristics and phase changes of the maps.

Overall in Fig. 3, two 2D maps are illustrated for two normal/abnormal maps, parameters are $W = 24$, $R = 0.95$, $m = 50$. X and Y are (S_j^C, S_j^G) $0 \leq j < M$, the ECG variant map shows the regular characteristics. In Fig. 3a, a normal map for P wave is an oval. In Fig. 3b, an abnormal map for P wave is a stick.

In Fig. 4, a list of normal maps for P wave on parameters $R = \{0.6, 0.72, 0.84, 0.96, 65, 1.08, 1.2\}$. When the parameter R increases, the feature of relevant maps has a nonlinear displacement along the top right corner of the image.

STUDYINSTANCEUID	CONCLUSION	QRSZRS	WIDTP	WIDTHR	WIDHT	WIDTH	QTC	P
GRKS2014042900001	窦性心动过缓、ST-T改变	89	80	-43	89	0	465	110
JZNK2S2015052100019	窦性心动过缓、ST-T改变	33	86	68	33	0	430	96
JZNK2S2015042400005	窦性心动过缓、ST-T改变	-9	84	31	-9	0	459	94
JZNK2S2015060100008	窦性心动过缓、ST-T改变	17	84	-14	17	0	423	118
JZNK2S2015061300006	窦性心动过缓、ST-T改变	48	76	-61	48	0	480	92
JZNK3S2015083000004	窦性心动过缓、ST-T改变	16	88	-47	16	0	454	90
JZNK3S2015072900012	窦性心动过缓、ST-T改变	56	94	-4	56	0	472	120
JZNK3S2015072700025	窦性心动过缓、ST-T改变	32	82	31	32	0	430	98
JZNK3S2015091500013	窦性心动过缓、ST-T改变	67	82	90	67	0	460	134
JZNK3S2015092600013	窦性心动过缓、ST-T改变	8	92	4	8	0	451	82
JZNK3S2015100600001	窦性心动过缓、ST-T改变	-35	128	-58	-35	0	514	102
JZNK2014091500001	窦性心动过缓、ST-T改变	90	88	-13	90	0	419	74
JZNK2014101000001	窦性心动过缓、ST-T改变	36	92	30	36	0	498	80

Fig. 2 The sample of part of abnormal ECG data source

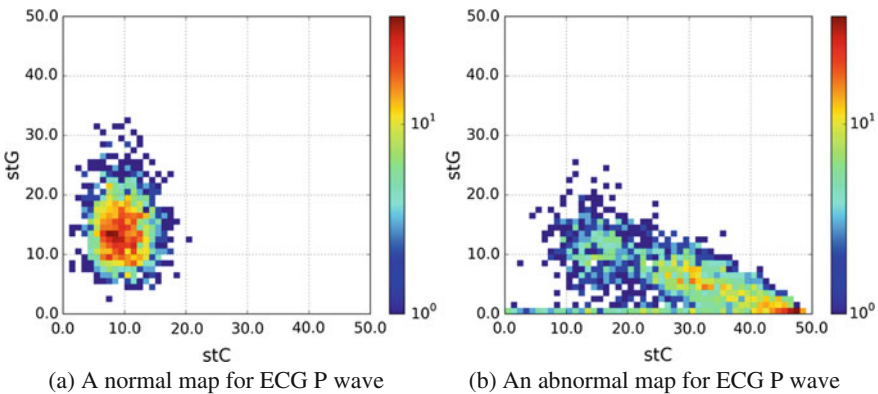


Fig. 3 The example of normal and not ECG variant map

In Fig. 5, a list of abnormal maps for *P* wave on parameters $R = \{0.6, 0.72, 0.84, 0.96, 65, 1.08, 1.2\}$. When the parameter R increases, the feature of relevant maps has a nonlinear displacement along the top right corner of the image.

Comparing with Figs. 4 and 5, differences between normal and abnormal map features.

4 Summary and Prospect

Electrocardiogram (ECG) detection is the key to clinical diagnosis of heart disease and has important clinical value. At present, the automatic analysis function of dynamic ECG detection is not satisfactory. There are also problems that the features of waveform lesions are small and cannot be marked, and even the characteristics of lesions are neglected. Therefore, excavating the effective information existing in the massive ECG signal can avoid the blind area of ECG analysis to some extent, which has certain application value.

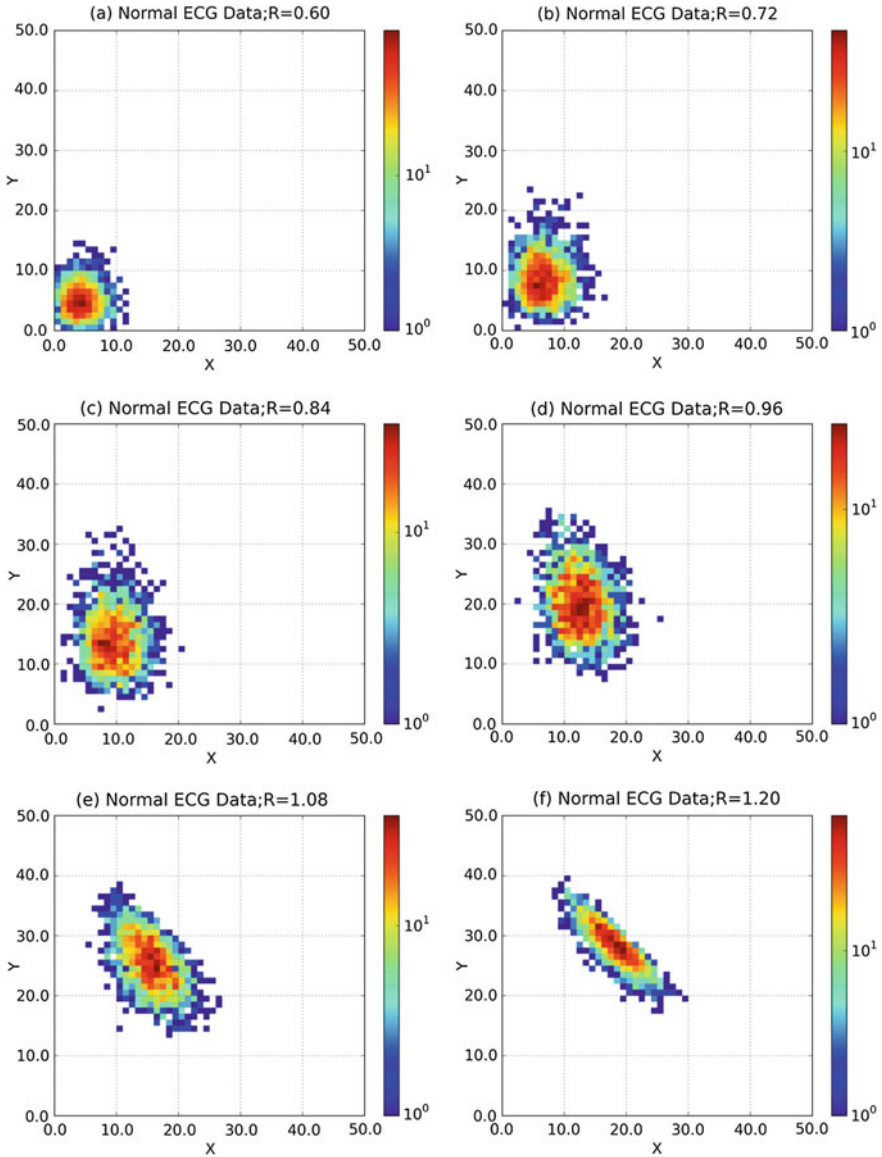


Fig. 4 A list of normal maps for P wave on parameters $R = \{0.6, 0.72, 0.84, 0.96, 1.08, 1.2\}$; a–f maps on $R = \{0.6, 0.72, 0.84, 0.96, 1.08, 1.2\}$

This chapter presents a new scheme of statistical distribution, variant map for ECG. This method can process massive ECG data sequences as 2D maps with visual characteristics. The sample results show classification of arrhythmia characteristics

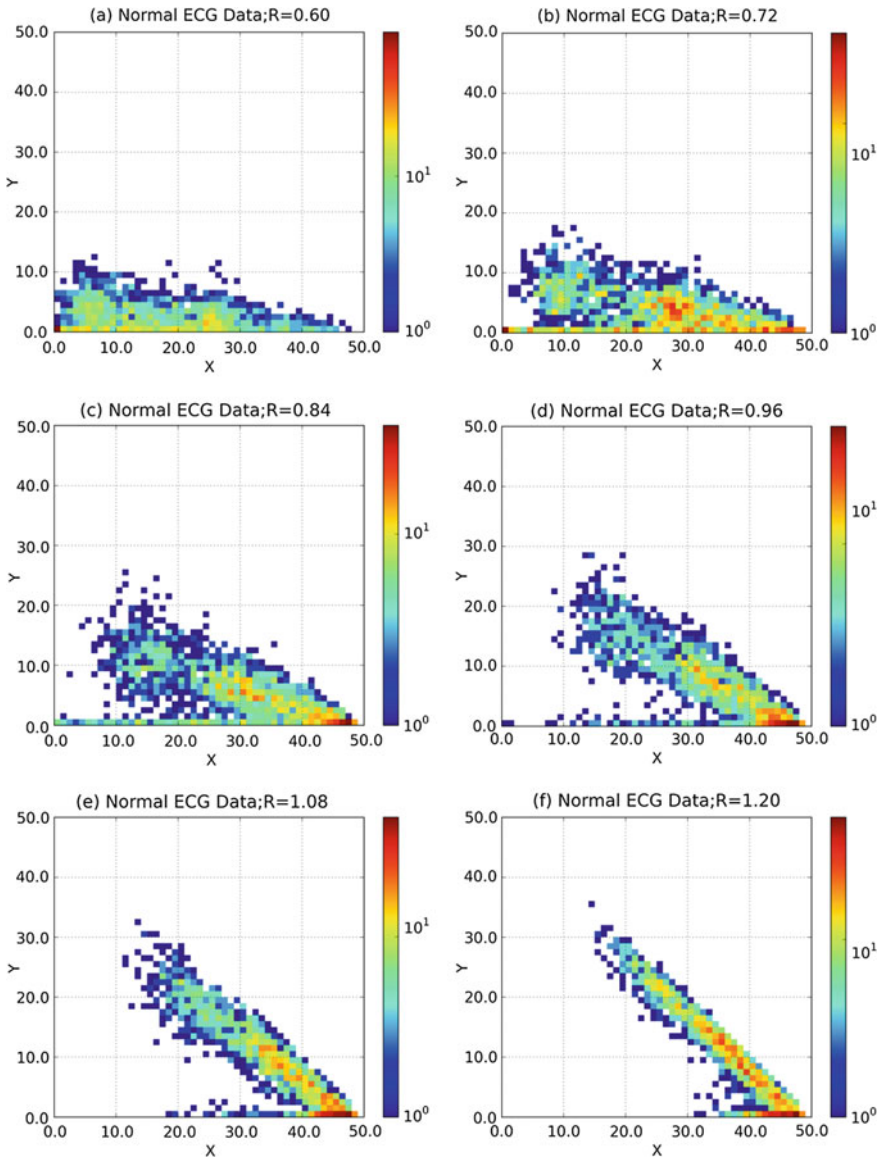


Fig. 5 A list of abnormal maps for *P* wave on parameters $R = \{0.6, 0.72, 0.84, 0.96, 1.08, 1.2\}$; a–f maps on $R = \{0.6, 0.72, 0.84, 0.96, 1.08, 1.2\}$

to identify the normal ECG signals and abnormal ECG signals significantly different. Further explorations and more experiments are required.

Acknowledgements Thanks to the First People's Hospital of Yunnan Province for ECG data sequences, to National Science Foundation of China NSFC (No. 61362014) and the Overseas Higher level Scholar Project of Yunnan Province, China (No. W8110305) for financial support to the project.

References

1. S.A. Sabab, Md. A.R. Munshi, A.I. Pritom, S. Shihabuzzaman, Cardiovascular disease prognosis using effective classification and feature selection technique, in *The International Conference on Medical Engineering, Health Informatics and Technology (MediTec)* (Dhaka, Bangladesh, 2016), pp. 1–6
2. B. Subramanian, K. Ramya, R. Asokan, Anatomizing electro cardiogram using fractal features and GUI based detection of P and T waves, in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (India, 2016), pp. 469–473
3. M.D. Skowronski, J.G. Harris, Acoustic detection and classification of microchiroptera using machine learning: Lessons learned from automatic speech recognition. *J. Acoust. Soc. Am.* **119**, 1817 (2006)
4. Y. Özbay, R. Ceylan, B. Karlik, A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Comput. Biol. Med.* **36**(4), 376–388 (2006)
5. A.K. Mishra, S. Raghav, Local fractal dimension based ECG arrhythmia classification. *Biomed. Sig. Process. Control* **5**(2), 114–123 (2010)
6. Y. Kutlu, D. Kuntalp, Feature extraction for ECG heartbeats using higher order statistics of WPD coefficients. *Comput. Methods Programs Biomed.* **105**(3), 257–267 (2012)
7. W. Wang, Y. Wei, N. Guan, Y. Wang, The automatic detection and analysis of electrocardiogram based on Lorenz Plot, in *IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Zhuhai, China, 2015), pp. 644–649
8. M.P.S. Chawla, Segment classification of ECG data and construction of scatter plots using principal component analysis. *J. Mech. Med. Biol.* **08**(03), 421–458 (2008)
9. M. Brennan, M. Palaniswami, P. Kamen, Comparison of normal and arrhythmic electrocardiograms using scatter plots. *IEEE Trans. Biomed Eng.* **48**(11), 1342–1347 (2001)
10. Z. Zheng, A.J. Maeder, *The conjugate classification of the kernel form of the hexagonal grid* (Springer, Tokyo, Japan, 1992)
11. Z.H. Zheng, Conjugate transformation of regular plan lattices for binary images, Ph.D. thesis, Monash University, 1994
12. J.Z.J. Zheng, C.H. Zheng, A framework to express variant and invariant functional spaces for binary logic. *Front. Electr. Electron. Eng. China* **5**, 163 (2010)
13. J.Z.J. Zheng, C.H.H. Zheng, T.L. Kunii, *A Framework of Variant Logic Construction for Cellular Automata* (InTech, Shanghai, 2011)
14. J. Zheng, C. Zheng, Variant measures and visualized statistical distributions. *Acta Photonica Sinica* **40**, 1397 (2011)
15. J. Zheng, C. Zheng, Variant simulation system using quaternion structures. *J. Mod. Opt.* **59**, 484 (2012)
16. C. Wang, J. Zheng, in *Proceedings of the 14th Australian Information Warfare and Security Conference* (Perth, 2013)
17. J. Zheng, W. Zhang, J. Luo, W. Zhou, R. Shen, Variant map system to simulate complex properties of DNA interactions using binary sequences. *Adv. Pure Math.* **3**, 5 (2013)
18. J. Zheng, J. Luo, J. Zhou, Pseudo DNA sequence generation of non-coding distributions using variant maps on cellular automata. *Appl. Math.* **5**, 153 (2014)

19. J. Zheng, W. Zhang, J. Luo, W. Zhou, V. Liesaputra, Variant map construction to detect symmetric properties of genomes on 2D distributions. *J. Data Mining Genomics Proteomics* **5**, 1 (2014)
20. D.M. Heim, O. Heim, P.A. Zeng, J. Zheng, Successful creation of regular patterns in variant maps from bat echolocation calls. *Biol. Syst. Open Access* **5**, 166 (2016)
21. Y. Ji, J. Zheng, Y. Xie, T. Shou, Variant maps on normal and abnormal ECG data sequences. *Biol. Med. Biol. Med. (Aligarh)* **8**(336) (2016)
22. Z. Hou, J. Zheng, Mapping ECG signals on variant maps, in *ASONAM'17: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, July 2017

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

