6.096

Algorithms for Computational Biology

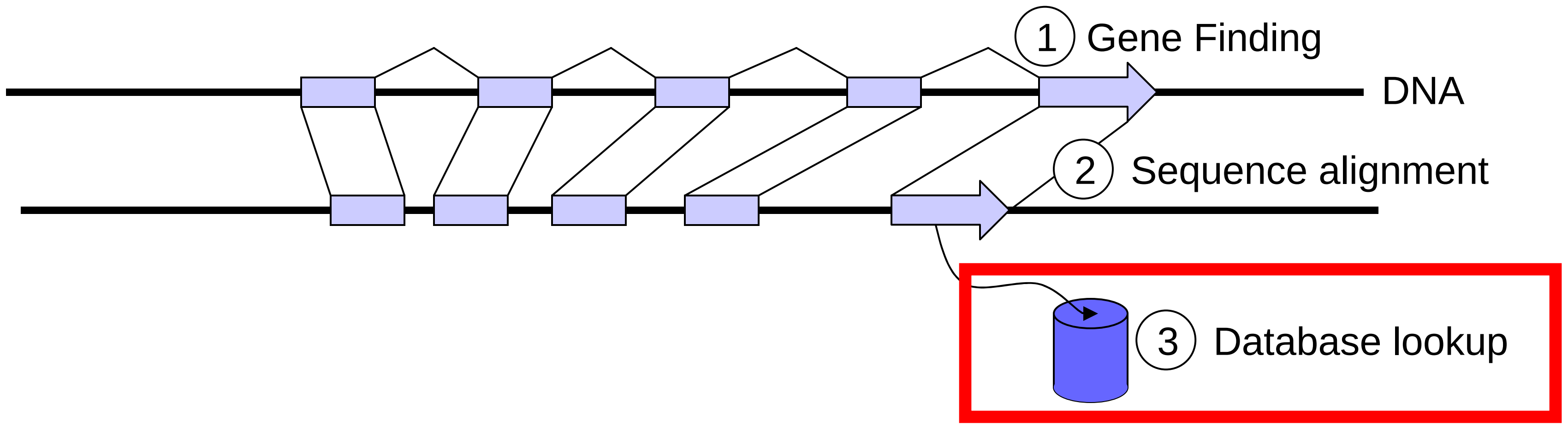
Lecture 2

BLAST & Database Search

Manolis Kellis

Piotr Indyk

In Previous Lecture



BLAST and Database Search

Setup

The BLAST algorithm

BLAST extensions

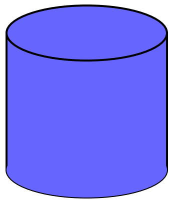
Substitutions matrices

Why K-mers work

Applications

Setup

- Sequences of symbols:
  - Bases: A,G,T,C
  - Amino-acids (a.a.): A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W, Y,V,B,Y,X
- Database search:
  - Database.



AIKWQPRSTW….
IKMQRHIKW….
HDLFWHLWH….
……………………

  - Query:

RGIKW

  - Output: sequences similar to query

What does “similar” mean ?

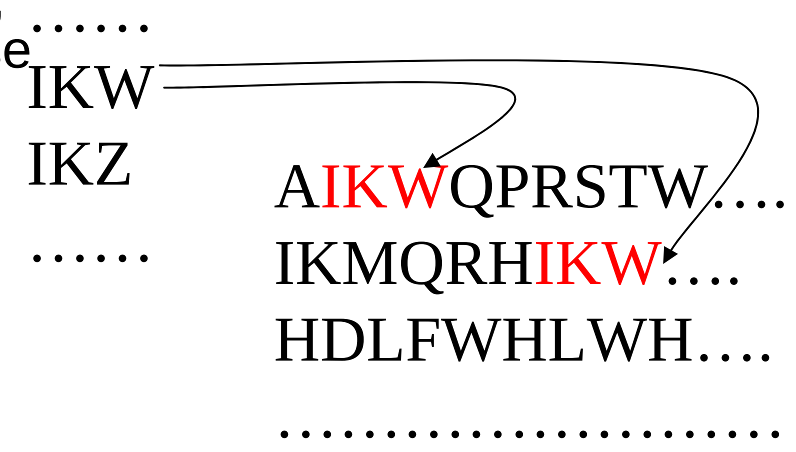
- Simplest idea: just count the number of common amino-acids
  - E.g., RGRKW matches RGIKW with idperc = 80%
- Not all matches are created equal - scoring matrix
- In general, insertions and deletions can also happen

How to answer the query

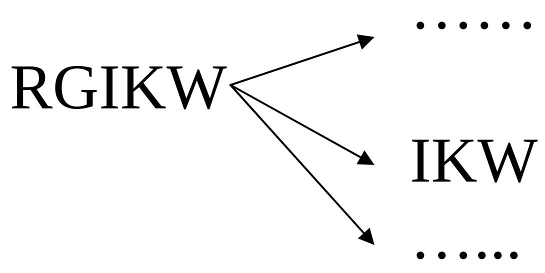
- We could just scan the whole database
- But:
  - Query must be very fast
  - Most sequences will be completely unrelated to query
  - Individual alignment needs not be perfect. Can fine-tune
- Exploit nature of the problem
  - If you're going to reject any match with idperc < 90%, then why bother even looking at sequences which don't have a fairly long stretch of matching a.a. in a row.
  - Pre-screen sequences for common long stretches, and reject vast majority of them

W-mer indexing

- W-mer: a string of length W
  - Preprocessing: For every W-mer (e.g., W=3), list every location in the database where it occurs



  - Query:
    - Generate W-mers and look them up in the database.
    - Process the results



- Benefit:
  - For W=3, roughly one W-mer in 23^3 will match, i.e., one in a ten thousand

6.046 Digression

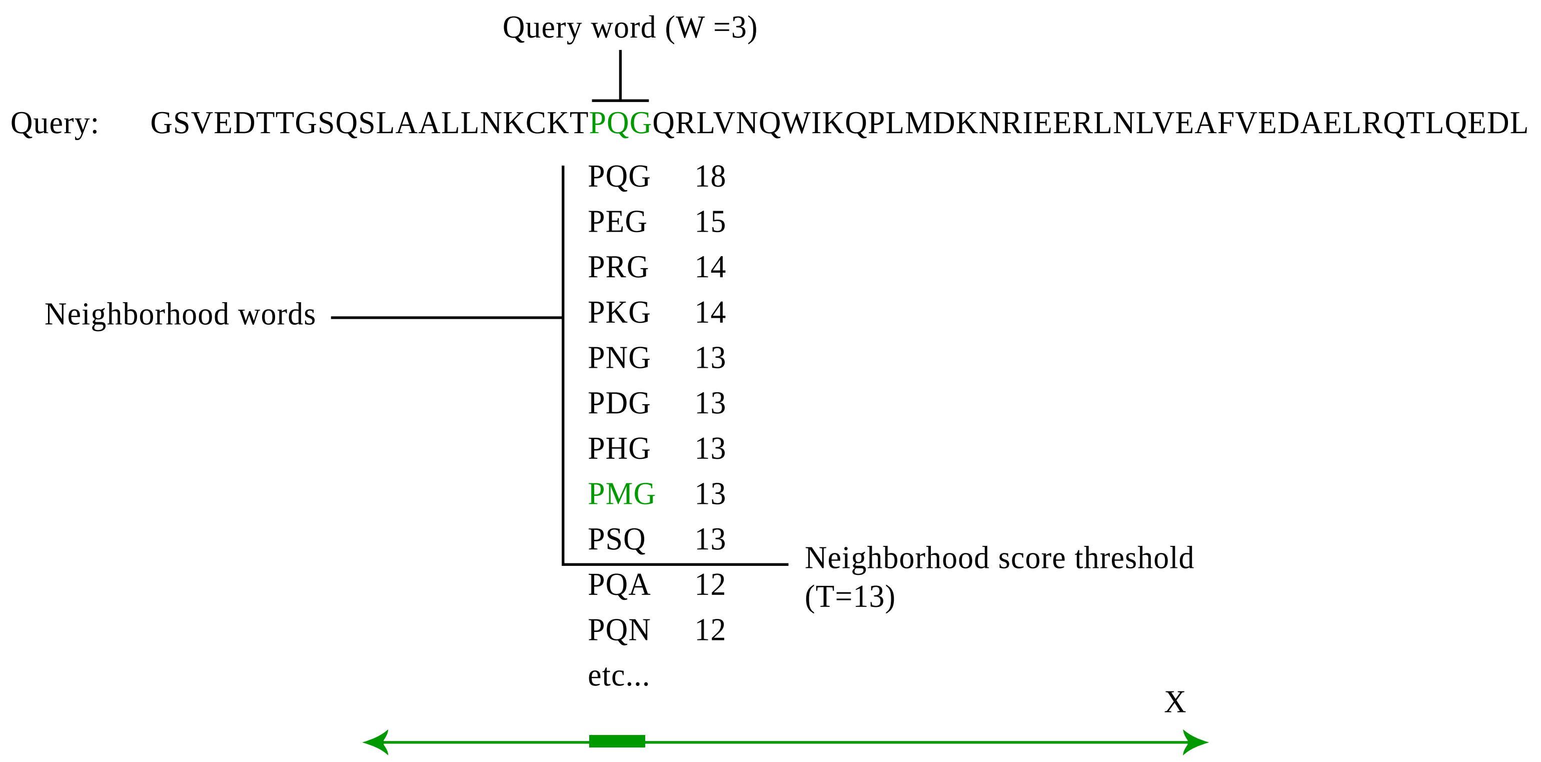
- This “lookup” technique is quite fundamental
- Will see more in 6.046, lecture 7, on hashing

BLAST and Database Search

BLAST

- Specific (and very efficient) implementation of the W-mer indexing idea
  - How to generate W-mers from the query
  - How to process the matches

THE BLAST SEARCH ALGORITHM


Query word (W =3)
Query: GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
Neighborhood words
PQG 18
PEG 15
PRG 14
PKG 14
PNG 13
PDG 13
PHG 13
PMG 13
PSQ 13
PQA 12
PQN 12
etc...
Neighborhood score threshold (T=13)
X

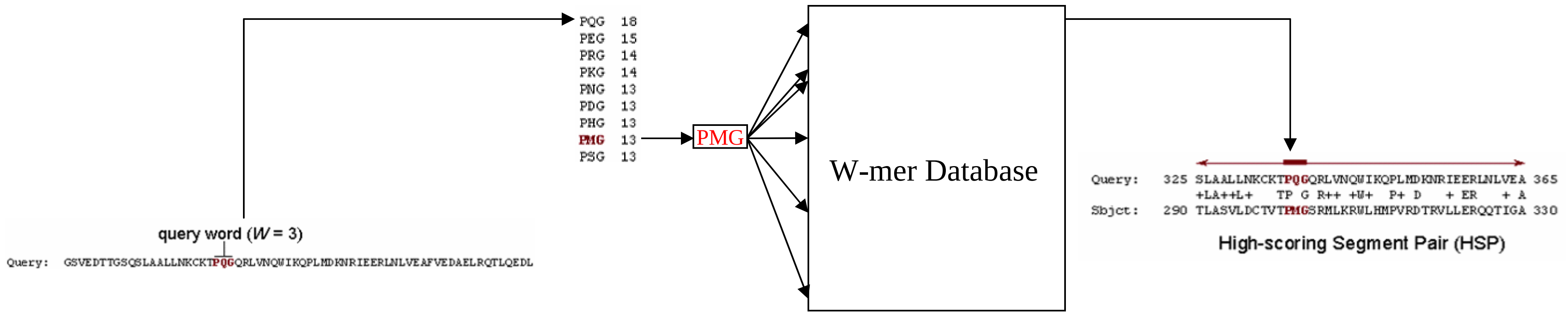
```
Query:  325  SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA  365
             +LA++L+      TP  G  R++   +W+   P+   D      +  ER      +  A
Sbjct:  290  TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA  330
```

High-scoring Segment Pair (HSP)

Adapted from: Baxevanis, Andy. "Nucleotide and Protein Sequence Analysis I." Lecture presentation, National Human Genome Research Institute Current Topics in Genome Analysis, Bethesda, MD, February 1, 2005. Figure by MIT OCW.

Blast Algorithm Overview

- Receive query
  - Split query into overlapping words of length W
  - Find neighborhood words for each word until threshold T
  - Look into the table where these neighbor words occur: seeds
  - Extend seeds until score drops off under X
- Evaluate statistical significance of score
- Report scores and alignments

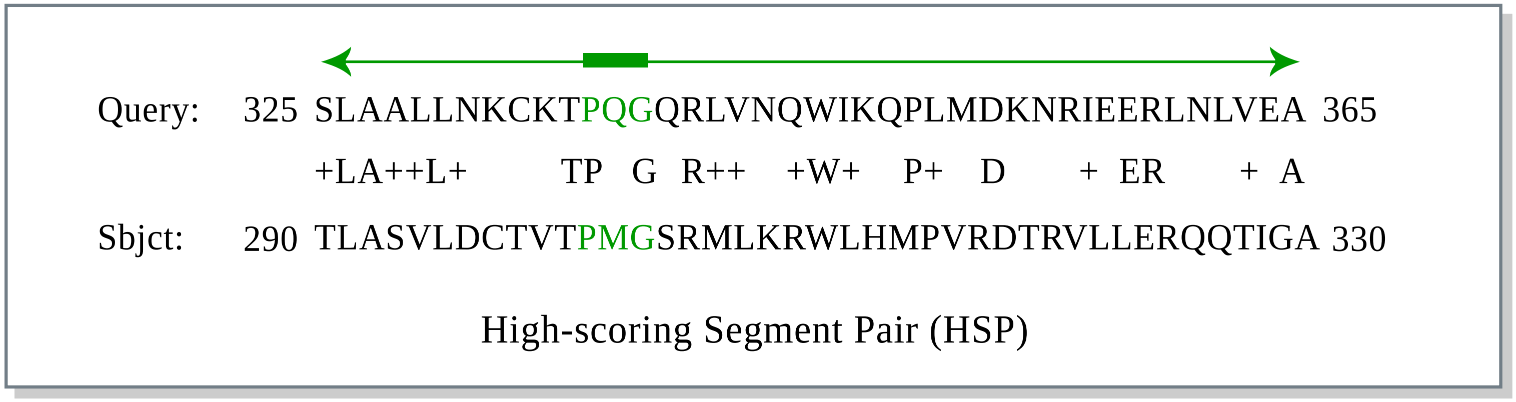


```
Query:   GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
```

PQG	18
PEG	15
PRG	14
PKG	14
PNG	13
PDG	13
PHG	13
PMG	13
PSG	13

```
Query:   325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
             +LA++L+    TP G R++ +W+  P+ D   + ER   + A
Sbjct:   290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
```

Extending the seeds

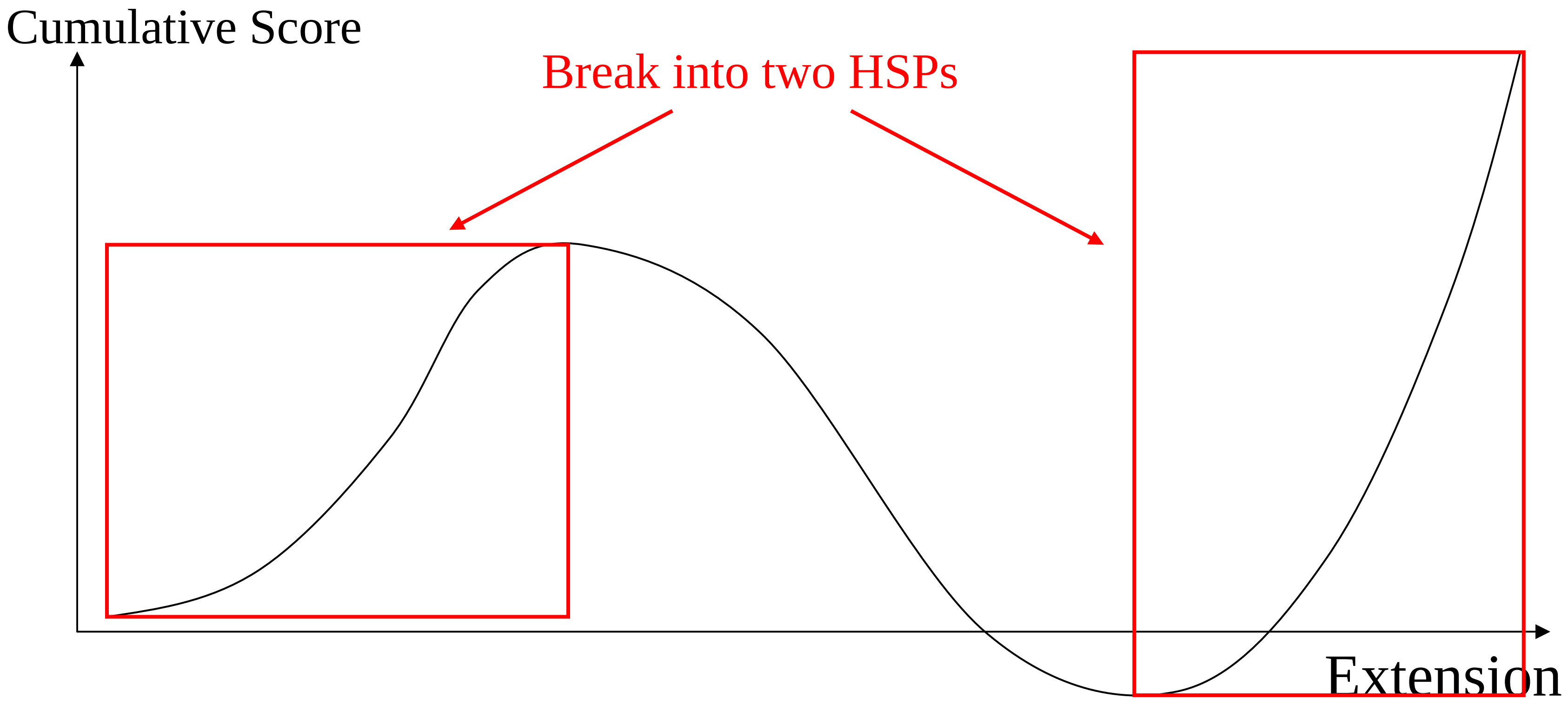


```
Query:   325  SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA  365
              +LA++L+      TP  G  R++   +W+   P+   D      +  ER      +  A
Sbjct:   290  TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA  330
```

High-scoring Segment Pair (HSP)

Figure by MIT OCW.

- Extend until the cumulative score drops



Statistical Significance

- Karlin-Altschul statistics
  - P-value: Probability that the HSP was generated as a chance alignment.
  - Score: -log of the probability
  - E: expected number of such alignments given database

BLAST and Database Search

Motivation

The BLAST algorithm

BLAST extensions

Substitutions matrices

Why K-mers work

Applications

Extensions: Filtering

- Low complexity regions can cause spurious hits
  - Filter out low complexity in your query
  - Filter most over-represented items in your database

Extensions: Two-hit blast

- Improves sensitivity for any speed
  - Two smaller W-mers are more likely than one longer one
  - Therefore it’s a more sensitive searching method to look for two hits instead of one, with the same speed.
- Improves speed for any sensitivity
  - No need to extend a lot of the W-mers, when isolated

Extensions: beyond W-mers

- W-mers (without neighborhoods):

  RGIKW → RGI , GIK, IKW
- No reason to use only consecutive symbols
- Instead, we could use combs, e.g.,

  RGIKW → R*IK* , RG**W, …
- Indexing same as for W-mers:
  - For each comb, store the list of positions in the database where it occurs
  - Perform lookups to answer the query
- Randomized projection: Buhler’01, based on Indyk-Motwani’98
  - Choose the positions of * at random
  - Example of a randomized algorithm

BLAST and Database Search

Motivation

The BLAST algorithm

BLAST extensions

Substitutions matrices

Why K-mers work

Applications

Substitution Matrices

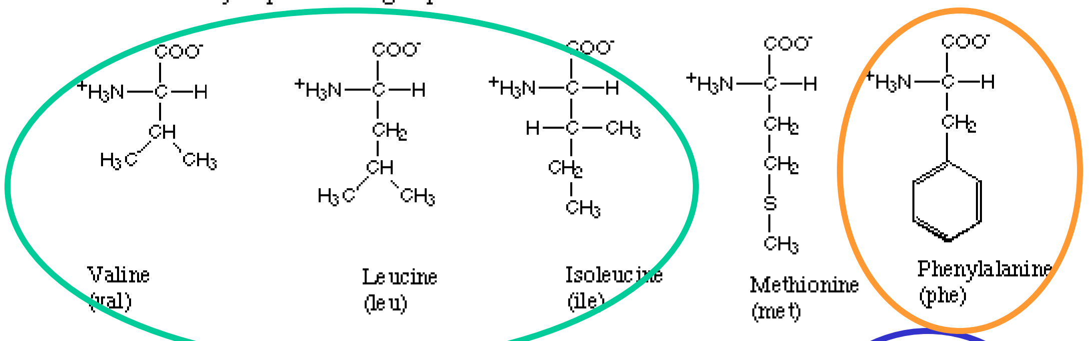
- Not all amino acids are created equal
  - Some are more easily substituted than others
  - Some mutations occur more often
  - Some substitutions are kept more often
- Mutations tend to favor some substitutions
  - Some amino acids have similar codons
  - They are more likely to be changed from DNA mutation
- Selection tends to favor some substitutions
  - Some amino acids have similar properties / structure
  - They are more likely to be kept when randomly changed
- The two forces together yield substitution matrices

Amino Acids

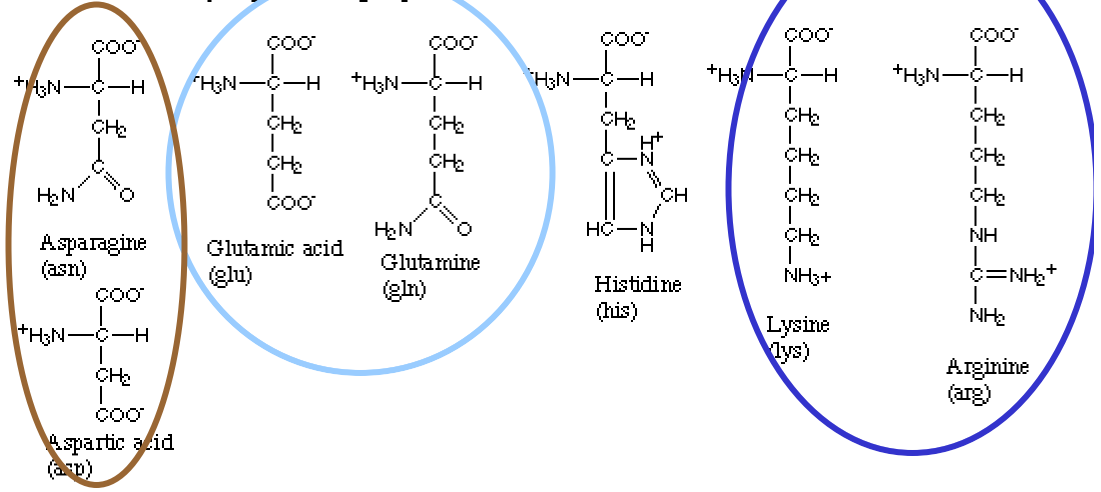
	T	C	A	G	
T	TTT: PheF	TCT: Ser	TAT: Tyr	TGT: Cys	T
	TTC: Phe	TCC: Ser	TAC: Tyr	TGC: Cys	C
	TTA: Leu	TCA: Ser	TAA: *	TGA: *	A
	TTG: Leu	TCG: Ser	TAG: *	TGG: TrpW	G
C	CTT: Leu	CCT: Pro	CAT: His	CGT: Arg	T
	CTC: Leu	CCC: Pro	CAC: His	CGC: Arg	C
	CTA: Leu	CCA: Pro	CAA: GlnQ	CGA: Arg	A
	CTG: Leu	CCG: Pro	CAG: Gln	CGG: Arg	G
A	ATT: Ile	ACT: Thr	AAT: Asn	AGT: Ser	T
	ATC: Ile	ACC: Thr	AAC: Asn	AGC: Ser	C
	ATA: Ile	ACA: Thr	AAA: LysK	AGA: Arg	A
	ATG: Met	ACG: Thr	AAG: Lys	AGG: Arg	G
G	GTT: Val	GCT: Ala	GAT: AspD	GGT: Gly	T
	GTC: Val	GCC: Ala	GAC: Asp	GGC: Gly	C
	GTA: Val	GCA: Ala	GAA: GluE	GGA: Gly	A
	GTG: Val	GCG: Ala	GAG: Glu	GGG: Gly	G

	T	C	A	G	
T	TTT: 273	TCT: 238	TAT: 186	TGT: 85	T
	TTC: 187	TCC: 144	TAC: 140	TGC: 53	C
	TTA: 263	TCA: 200	TAA: 10	TGA: 8	A
	TTG: 266	TCG: 92	TAG: 5	TGG: 105	G
C	CTT: 134	CCT: 134	CAT: 137	CGT: 62	T
	CTC: 61	CCC: 69	CAC: 76	CGC: 28	C
	CTA: 139	CCA: 178	CAA: 258	CGA: 33	A
	CTG: 111	CCG: 56	CAG: 120	CGG: 20	G
A	ATT: 298	ACT: 198	AAT: 356	AGT: 147	T
	ATC: 169	ACC: 124	AAC: 241	AGC: 103	C
	ATA: 188	ACA: 181	AAA: 418	AGA: 203	A
	ATG: 213	ACG: 84	AAG: 291	AGG: 94	G
G	GTT: 213	GCT: 195	GAT: 365	GGT: 217	T
	GTC: 112	GCC: 118	GAC: 196	GGC: 97	C
	GTA: 128	GCA: 165	GAA: 439	GGA: 114	A
	GTG: 112	GCG: 63	GAG: 191	GGG: 61	G

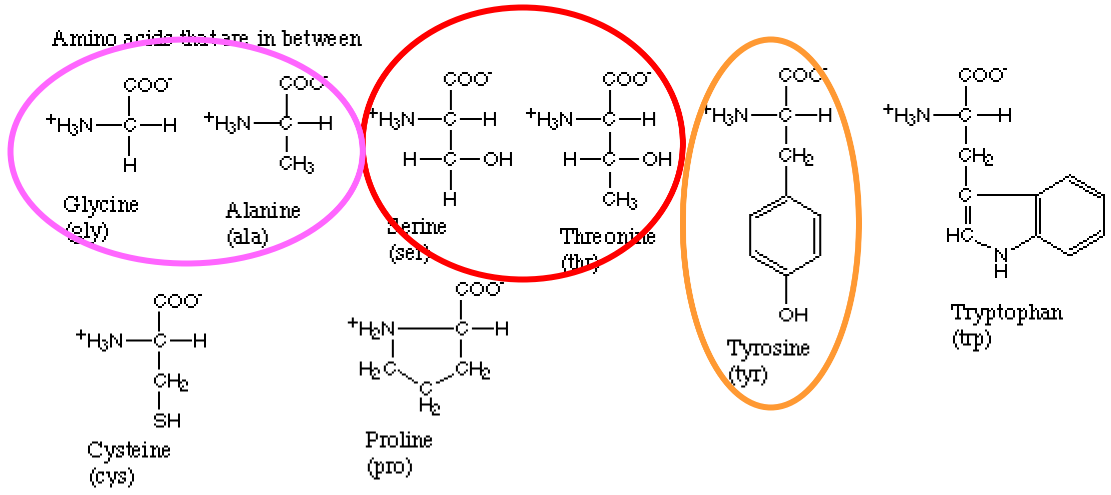
Amino acids with hydrophobic side groups



Amino acids with hydrophilic side groups



Amino acids that are in between



PAM matrices

- PAM = Point Accepted mutation

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	**2**	-2	0	0	-2	-1	0	1	-2	-1	-2	-2	-1	-3	1	1	1	-5	-3	0	0	0	0	-7
R	-2	**6**	-1	-2	-3	1	-2	-3	1	-2	-3	3	-1	-4	-1	-1	-1	1	-4	-3	-1	0	-1	-7
N	0	-1	**3**	2	-4	0	1	0	2	-2	-3	1	-2	-3	-1	1	0	-4	-2	-2	2	1	0	-7
D	0	-2	2	**4**	-5	1	3	0	0	-3	-4	0	-3	-6	-2	0	-1	-6	-4	-3	3	2	-1	-7
C	**-2**	**-3**	**-4**	**-5**	**9**	**-5**	**-5**	**-3**	**-3**	**-2**	**-6**	**-5**	**-5**	**-5**	**-3**	**0**	**-2**	**-7**	**0**	**-2**	**-4**	**-5**	**-3**	-7
Q	-1	1	0	1	-5	**5**	2	-2	2	-2	-2	0	-1	-5	0	-1	-1	-5	-4	-2	1	3	-1	-7
E	0	-2	1	3	-5	2	**4**	0	0	-2	-3	-1	-2	-5	-1	0	-1	-7	-4	-2	2	3	-1	-7
G	1	-3	0	0	-3	-2	0	**4**	-3	-3	-4	-2	-3	-4	-1	1	-1	-7	-5	-2	0	-1	-1	-7
H	-2	1	2	0	-3	2	0	-3	**6**	-3	-2	-1	-3	-2	-1	-1	-2	-3	0	-2	1	1	-1	-7
I	-1	-2	-2	-3	-2	-2	-2	-3	-3	**5**	2	-2	2	0	-2	-2	0	-5	-2	3	-2	-2	-1	-7
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	**5**	-3	3	1	-3	-3	-2	-2	-2	1	-4	-3	-2	-7
K	-2	3	1	0	-5	0	-1	-2	-1	-2	-3	**4**	0	-5	-2	-1	0	-4	-4	-3	0	0	-1	-7
M	-1	-1	-2	-3	-5	-1	-2	-3	-3	2	3	0	**7**	0	-2	-2	-1	-4	-3	1	-3	-2	-1	-7
F	-3	-4	-3	-6	-5	-5	-5	-4	-2	0	1	-5	0	**7**	-4	-3	-3	-1	5	-2	-4	-5	-3	-7
P	1	-1	-1	-2	-3	0	-1	-1	-1	-2	-3	-2	-2	-4	**5**	1	0	-5	-5	-2	-1	-1	-1	-7
S	1	-1	1	0	0	-1	0	1	-1	-2	-3	-1	-2	-3	1	**2**	1	-2	-3	-1	0	-1	0	-7
T	1	-1	0	-1	-2	-1	-1	-1	-2	0	-2	0	-1	-3	0	1	**3**	-5	-3	0	0	-1	0	-7
W	**-5**	**1**	**-4**	**-6**	**-7**	**-5**	**-7**	**-7**	**-3**	**-5**	**-2**	**-4**	**-4**	**-1**	**-5**	**-2**	**-5**	**12**	**-1**	**-6**	**-5**	**-6**	**-4**	-7
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-2	-2	-4	-3	5	-5	-3	-3	-1	**8**	-3	-3	-4	-3	-7
V	0	-3	-2	-3	-2	-2	-2	-2	-2	3	1	-3	1	-2	-2	-1	0	-6	-3	**4**	-2	-2	-1	-7
B	0	-1	2	3	-4	1	2	0	1	-2	-4	0	-3	-4	-1	0	0	-5	-3	-2	**3**	2	-1	-7
Z	0	0	1	2	-5	3	3	-1	1	-2	-3	0	-2	-5	-1	-1	-1	-6	-4	-2	2	**3**	-1	-7
X	0	-1	0	-1	-3	-1	-1	-1	-1	-1	-2	-1	-1	-3	-1	0	0	-4	-3	-1	-1	-1	**-1**	-7
*	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	**1**

BLOSUM matrices

- BloSum = BLOck SUbstitution matrices

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	**4**	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	**5**	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	**6**	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	**6**	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	**0**	**-3**	**-3**	**-3**	**9**	**-3**	**-4**	**-3**	**-3**	**-1**	**-1**	**-3**	**-1**	**-2**	**-3**	**-1**	**-1**	**-2**	**-2**	**-1**	**-3**	**-3**	**-2**	-4
Q	-1	1	0	0	-3	**5**	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	**5**	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	**6**	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	**8**	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	**4**	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	**4**	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	**5**	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	**5**	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	**6**	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	**7**	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	**4**	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	**5**	-2	-2	0	-1	-1	0	-4
W	**-3**	**-3**	**-4**	**-4**	**-2**	**-2**	**-3**	**-2**	**-2**	**-3**	**-2**	**-3**	**-1**	**1**	**-4**	**-3**	**-2**	**11**	**2**	**-3**	**-4**	**-3**	**-2**	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	**7**	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	**4**	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	**4**	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	**4**	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	**1**

Computing Substitution Matrices

- Take a list of 1000 aligned proteins
  - Every time you see a substitution between two amino acids, increment the similarity score between them.
  - Must normalize it by how often amino acids occur in general. Rare amino acids will give rare substitutions.
- BLOSUM matrices vs. PAM
  - BLOSUM were built only from the most conserved domains of the blocks database of conserved proteins.
  - BLOSUM: more tolerant of hydrophobic changes and of cysteine and tryptophan mismatches
  - PAM: more tolerant of substitutions to or from hydrophilic amino acids.

BLAST and Database Search

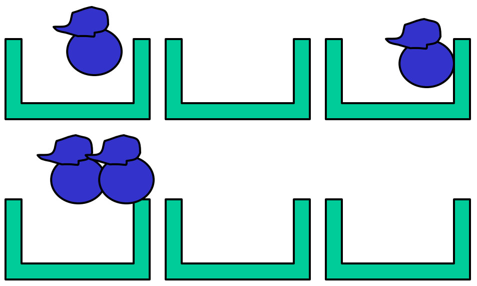
Overview: Why this works

- In worst case:
  - W-mer: W=3
  - Combs/random projection
- In average case
- Simulations
- Biological case: counting W-mers in real alignments
  - Long conserved W-mers do happen in actual alignments
  - There's something biological about long W-mers

Query: RKIWGDPRS
Datab.: RKIVGDRRS
7 identical a.a

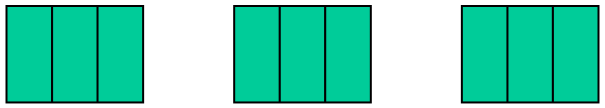
Pigeonhole principle

- Pigeonhole principle
  - If you have 2 pigeons and 3 holes, there must be at least one hole with no pigeon



Pigeonhole and W-mers

- Pigeonholing mis-matches
  - Two sequences, each 9 amino-acids, with 7 identities
  - There is a stretch of 3 amino-acids perfectly conserved



In general:

Sequence length: n

Identities: t

Can use W-mers for W= [n/(n-t+1)]

Combs and Random Pojections

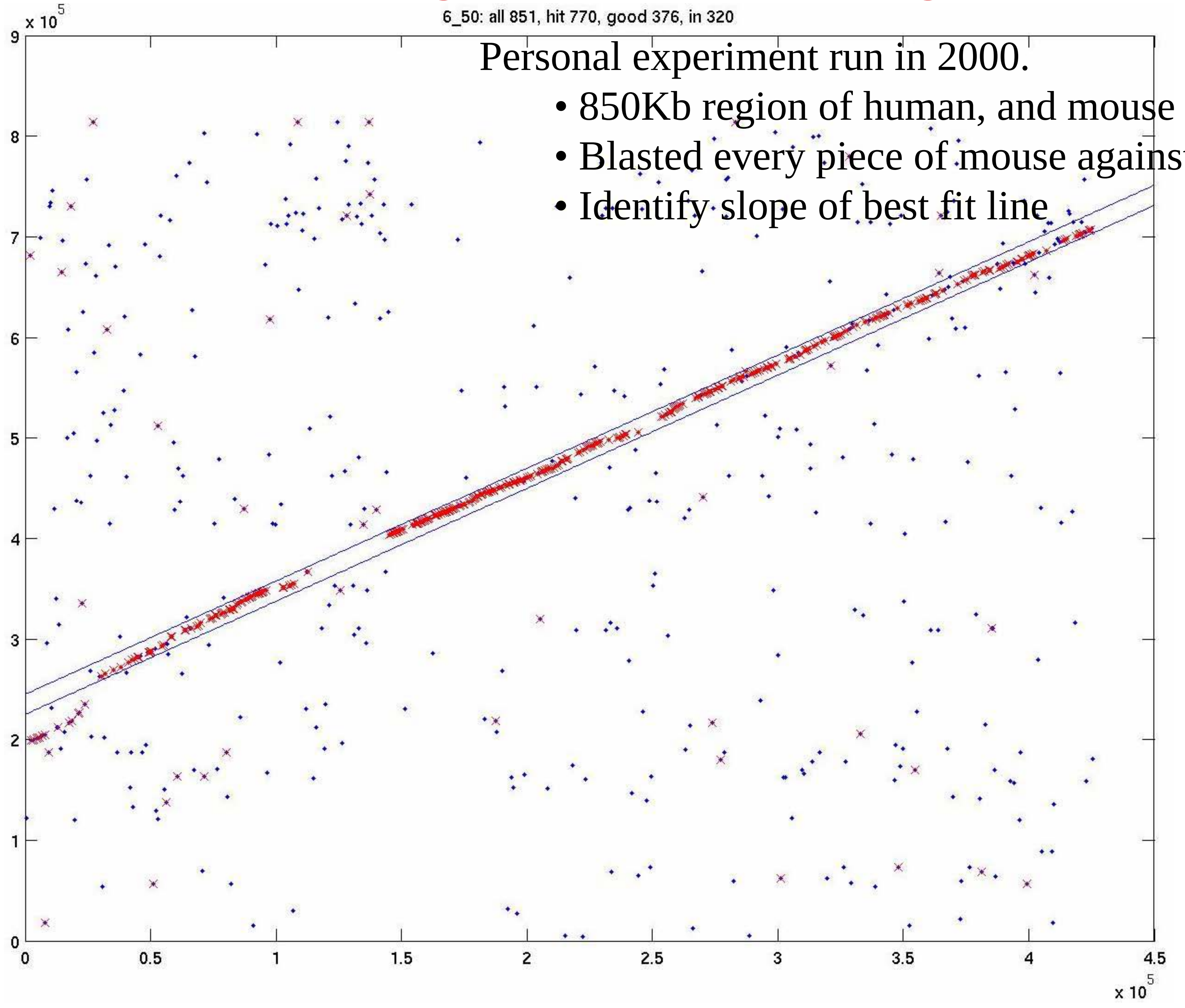
- Assume we select k positions, which do not contain *, at random with replacement
- What is the probability we miss a sequence match ?
  - At most: 1-idperc^k
  - In our case: 1-(7/9)^4 =0.63...
- What if we repeat the process l times, independently ?
  - Miss prob. = 0.63^l
  - For l=5, it is less than 10%

Query: RKIWGDPRS
Datab: RKIVGDRRS

$\Big\downarrow$ k=4

Query: *KI*G***S
Datab.: *KI*G***S

True alignments: Looking for K-mers



Personal experiment run in 2000.

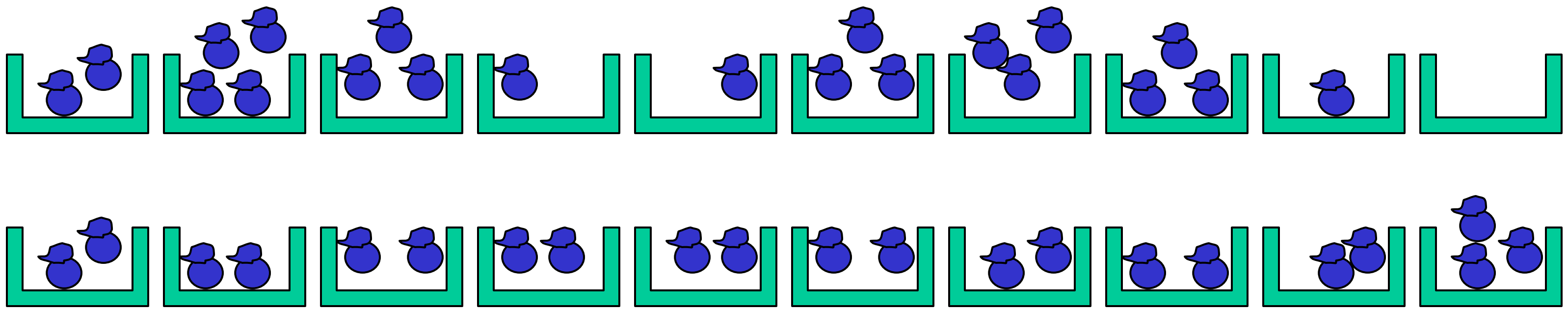
- 850Kb region of human, and mouse 450Kb ortholog.
- Blasted every piece of mouse against human (6,50)
- Identify slope of best fit line

Conclusions

- Table lookup – very powerful technique
- Deterministic, randomized
- More (on hashing) in 6.046

Extending pigeonhole principle

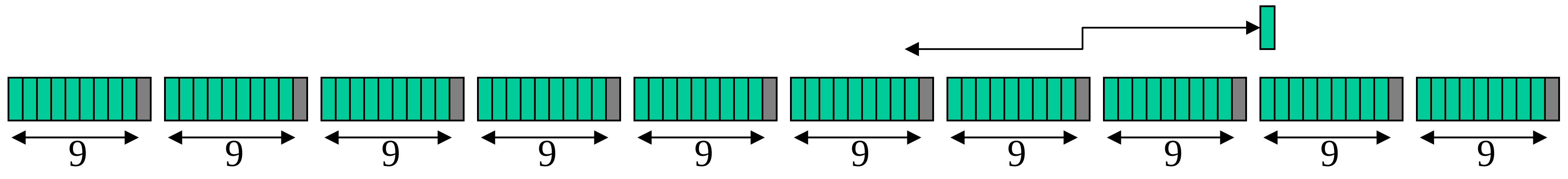
- Pigeonhole principle
  - If you have 21 pigeons and only 10 holes, there must be at least one hole with more than two pigeons.



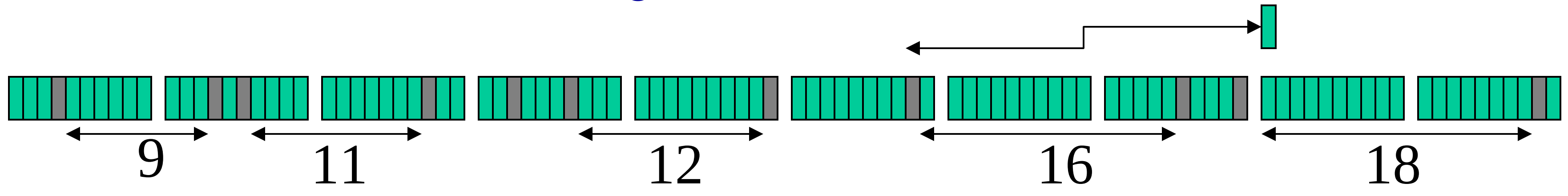
Proof by contradiction

Assume each hole has <= 2 pigeon. 10 holes together must have <= 10*2 pigeons, hence <=20. We have 21.

Random model: Average case



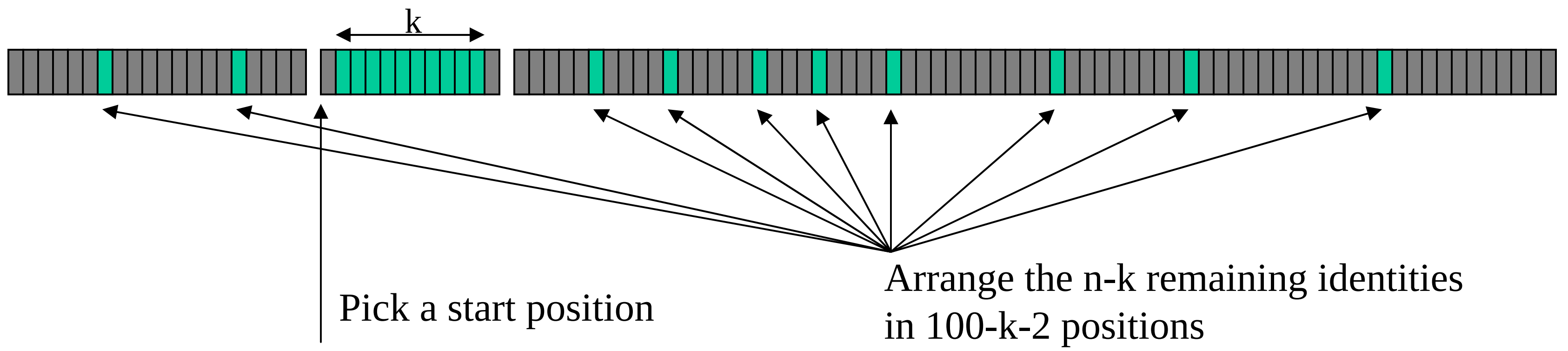
- In random model, things work better for us



In entirely random model, mismatches will often fall near each other, making a longer conserved k-mer more likely
mismatches also fall near each other but that doesn’t hurt
Birthday paradox: if we have 32 birthdays and 365 days they could fall in, 2 of them will coincide with P=.753
Similarly, counting random occurrences yields the following

Random Model: Counting

100 positions
n identities
k must be contiguous



Try this equation out, and get k-mers more often

$$(100-k-2)\cdot\frac{\left(\frac{(100-k-2)!}{(n-k)!(100-n-2)!}\right)}{\left(\frac{100!}{n!(100-n)!}\right)}$$

Increasing k-mer size →

↑ Increasing percent id

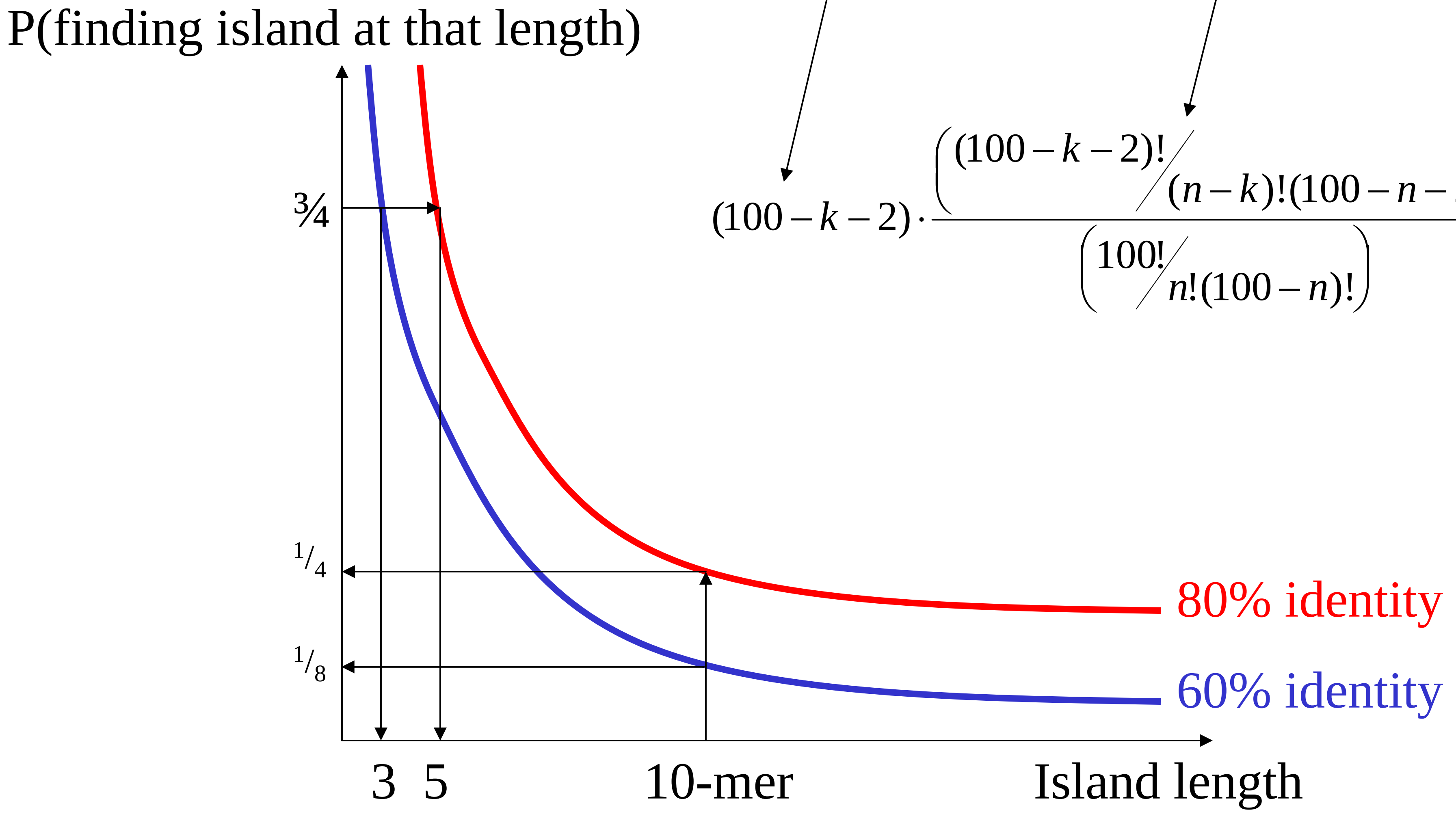
n \ k	6	7	8	9	10	11
80%	1.00	0.80	0.63	0.50	0.39	0.31
70%	1.01	0.69	0.47	0.32	0.22	0.14
60%	0.69	0.40	0.23	0.13	0.07	0.04

Random Model: simulation

Possible starts of the conserved k-mer

Possible arrangements of the remaining 100-k

$$(100-k-2)\cdot\frac{\left(\frac{(100-k-2)!}{(n-k)!(100-n-2)!}\right)}{\left(\frac{100!}{n!(100-n)!}\right)}$$



Random Model: simulation

Conservation **60%** over **1000 bp**

2 species

```
   |      L =  6    |      L =  7    |      L =  8    |      L =  9    |      L = 10    |      L = 11    |      L = 12    |
---+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+
 65| 84.966+/-10.234| 75.458+/-10.399| 66.440+/-10.590| 83.582+/-10.434| 74.822+/-10.463| 66.484+/-10.514| 81.592+/-10.438|
 80| 53.170+/-10.459| 42.060+/-10.469| 32.032+/-10.427| 26.432+/-10.419| 47.300+/-10.523| 37.084+/-10.581| 31.248+/-10.567|
 90| 16.598+/-10.358| 10.424+/-10.295|  6.870+/-10.254|  4.594+/-10.197| 17.754+/-10.394| 13.326+/-10.369|  9.736+/-10.330|
 95| 14.868+/-10.318| 10.896+/-10.334|  7.578+/-10.201|  4.740+/-10.207|  3.842+/-10.203|  1.854+/-10.157|  1.280+/-10.122|
100| 15.386+/-10.239| 11.078+/-10.297|  7.838+/-10.228|  5.114+/-10.198|  3.412+/-10.186|  2.094+/-10.176|  1.422+/-10.157|
```

3 species

```
   |      L =  6    |      L =  7    |      L =  8    |      L =  9    |      L = 10    |      L = 11    |      L = 12    |
---+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+
 65| 53.804+/-10.398| 39.560+/-10.503| 27.238+/-10.429| 45.112+/-10.474| 31.856+/-10.475| 22.966+/-10.568| 37.318+/-10.496|
 80| 21.618+/-10.341| 12.790+/-10.278|  7.416+/-10.250|  4.918+/-10.249| 12.488+/-10.425|  8.212+/-10.346|  4.544+/-10.214|
 90|  4.000+/-10.152|  1.844+/-10.096|  0.978+/-10.111|  0.426+/-10.056|  2.254+/-10.154|  1.272+/-10.143|  0.946+/-10.105|
 95|  3.436+/-10.147|  1.934+/-10.121|  0.840+/-10.093|  0.664+/-10.081|  0.232+/-10.044|  0.044+/-10.022|  0.048+/-10.024|
100|  3.360+/-10.146|  2.288+/-10.129|  0.746+/-10.081|  0.618+/-10.073|  0.124+/-10.034|  0.070+/-10.028|  0.050+/-10.025|
```

4 species

```
   |      L =  6    |      L =  7    |      L =  8    |      L =  9    |      L = 10    |      L = 11    |      L = 12    |
---+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+
 65| 29.614+/-10.379| 17.102+/-10.326|  9.492+/-10.262| 18.508+/-10.378| 10.376+/-10.264|  5.502+/-10.239| 11.190+/-10.346|
 80|  8.738+/-10.193|  3.536+/-10.167|  1.636+/-10.117|  0.934+/-10.095|  2.128+/-10.126|  1.250+/-10.111|  0.554+/-10.099|
 90|  0.820+/-10.066|  0.396+/-10.055|  0.138+/-10.041|  0.026+/-10.018|  0.128+/-10.035|  0.198+/-10.053|  0.026+/-10.018|
 95|  0.740+/-10.073|  0.394+/-10.049|  0.136+/-10.032|  0.028+/-10.020|  0.040+/-10.020|  0.000+/-10.000|  0.000+/-10.000|
100|  0.794+/-10.070|  0.476+/-10.051|  0.184+/-10.046|  0.074+/-10.025|  0.020+/-10.014|  0.044+/-10.022|  0.000+/-10.000|
```

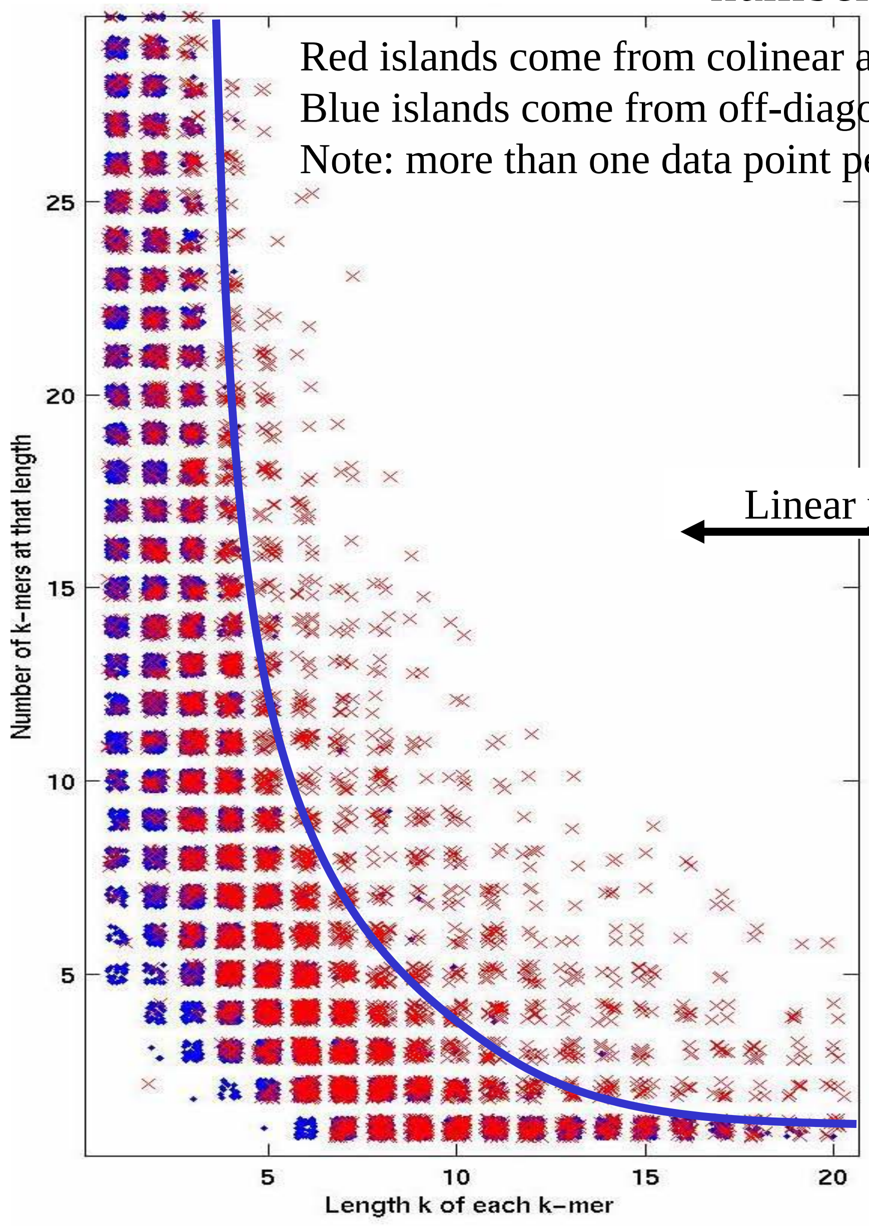
5 species

```
   |      L =  6    |      L =  7    |      L =  8    |      L =  9    |      L = 10    |      L = 11    |      L = 12    |
---+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+ ---------------+
 65| 15.144+/-10.290|  6.784+/-10.259|  2.934+/-10.182|  5.782+/-10.262|  2.802+/-10.167|  1.314+/-10.129|  2.354+/-10.160|
 80|  2.650+/-10.129|  0.902+/-10.086|  0.170+/-10.034|  0.248+/-10.046|  0.384+/-10.068|  0.164+/-10.041|  0.000+/-10.000|
 90|  0.250+/-10.044|  0.058+/-10.020|  0.032+/-10.016|  0.000+/-10.000|  0.024+/-10.017|  0.000+/-10.000|  0.000+/-10.000|
 95|  0.148+/-10.027|  0.018+/-10.013|  0.016+/-10.011|  0.000+/-10.000|  0.000+/-10.000|  0.000+/-10.000|  0.000+/-10.000|
100|  0.244+/-10.038|  0.100+/-10.025|  0.034+/-10.017|  0.018+/-10.013|  0.000+/-10.000|  0.022+/-10.016|  0.000+/-10.000|
```

True alignments: Looking for K-mers

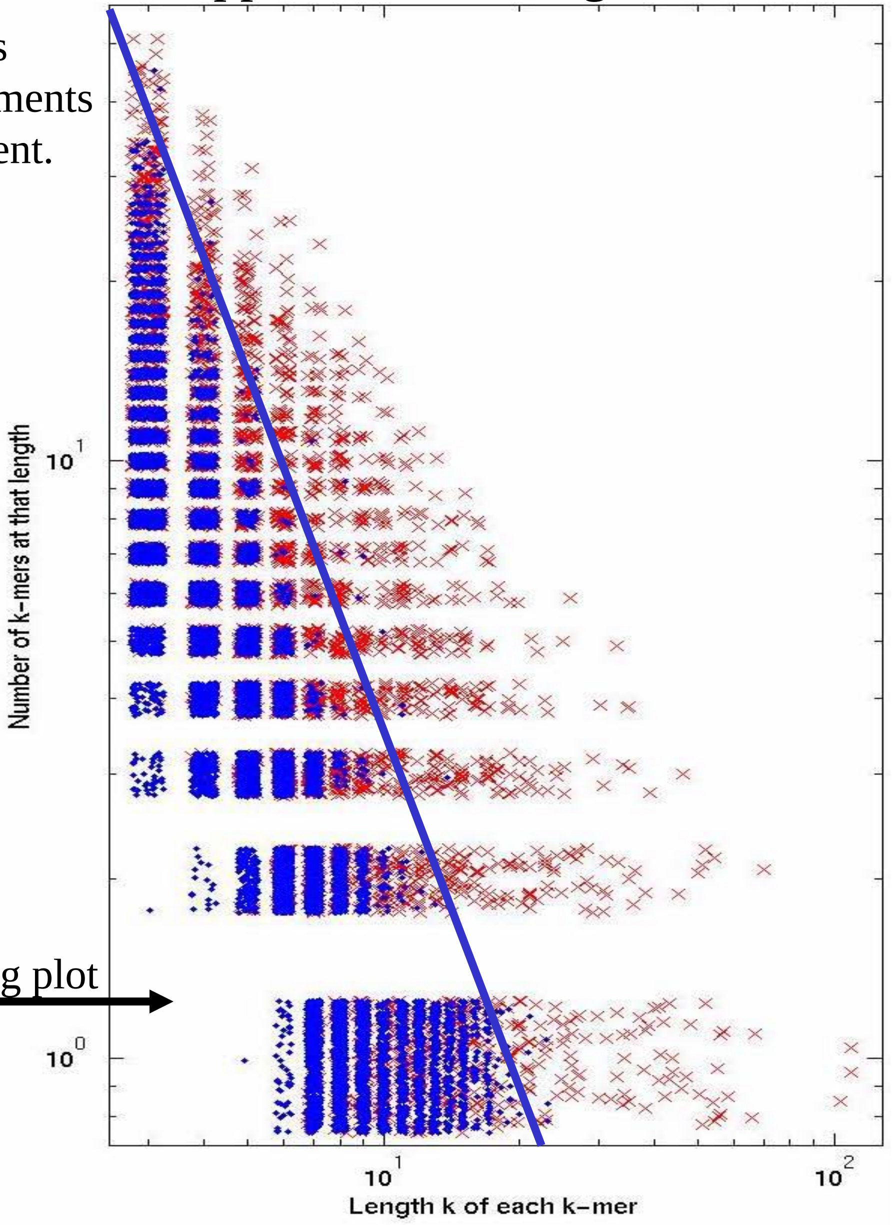
number of k-mers that happen for each length of k-mer.

Red islands come from colinear alignments
Blue islands come from off-diagonal alignments
Note: more than one data point per alignment.



Linear plot

Log Log plot



Summary: Why k-mers work

- In worst case: Pigeonhole principle
  - Have too many matches to place on your sequence length
  - Bound to place at least k matches consecutively
- In average case: Birthday paradox / Simulations
  - Matches tend to cluster in the same bin. Mismatches too.
  - Looking for stretches of consecutive matches is feasible
- Biological case: Counting k-mers in real alignments
  - From the number of conserved k-mers alone, one can distinguish genuine alignments from chance alignments
  - Something biologically meaningful can be directly carried over to the algorithm.

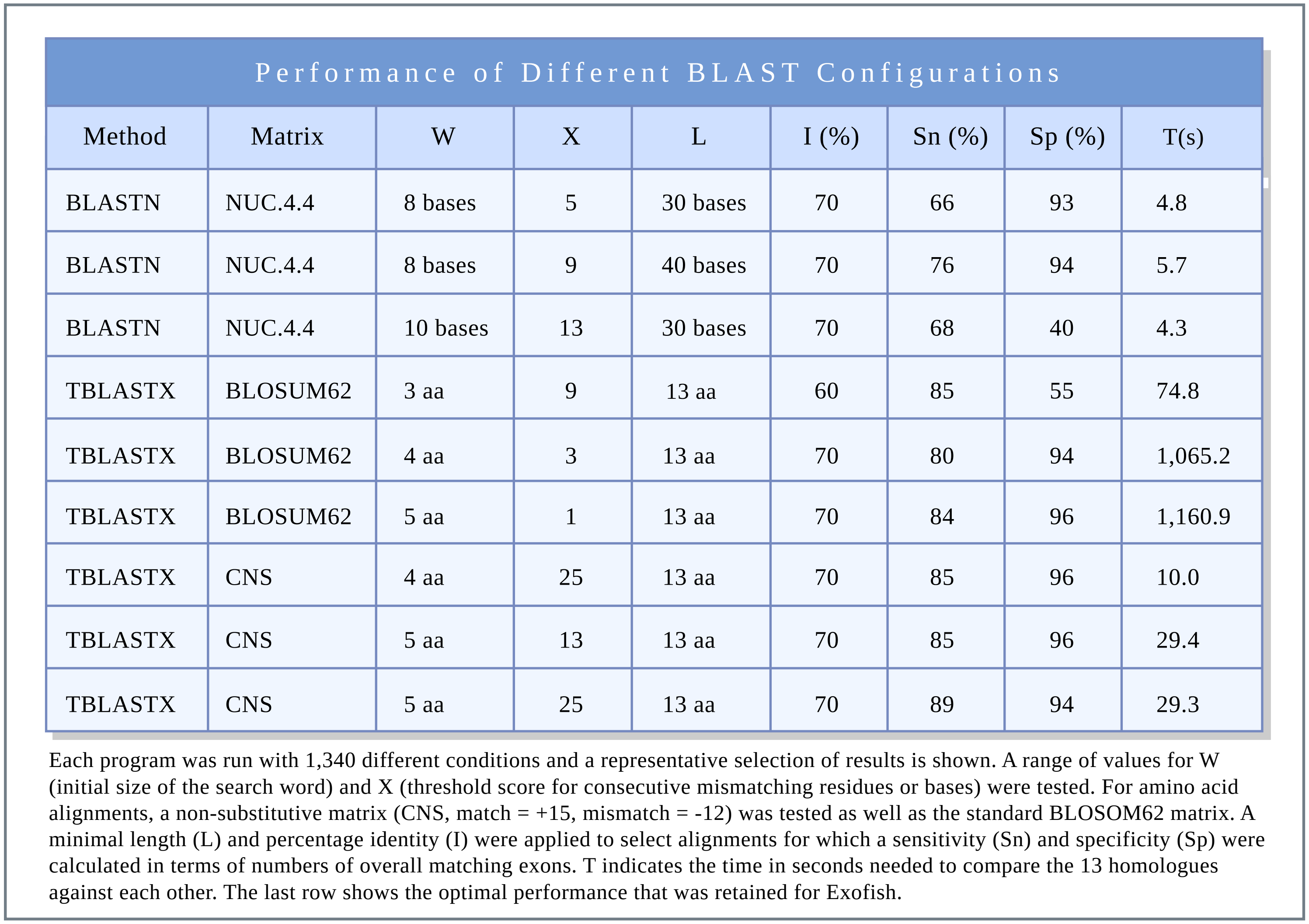
BLAST and Database Search

Identifying exons

- Direct application of BLAST
  - Compare Tetraodon to Human using BLAST
  - Best alignments happen only on exons
  - Translate a biological property into an alignment property
    - Exon = high alignment
  - Reversing this equivalence, look for high alignments and predict exons
- Estimate human gene number
  - Method is not reliable for complete annotation, and does not find all genes, or even all exons in a gene
  - Can be used however, to estimate human gene number

Part I - Parameter tuning

- Try a lot of parameters and find combination with
  - fastest running time
  - highest specificity
  - highest sensitivity



Performance of Different BLAST Configurations

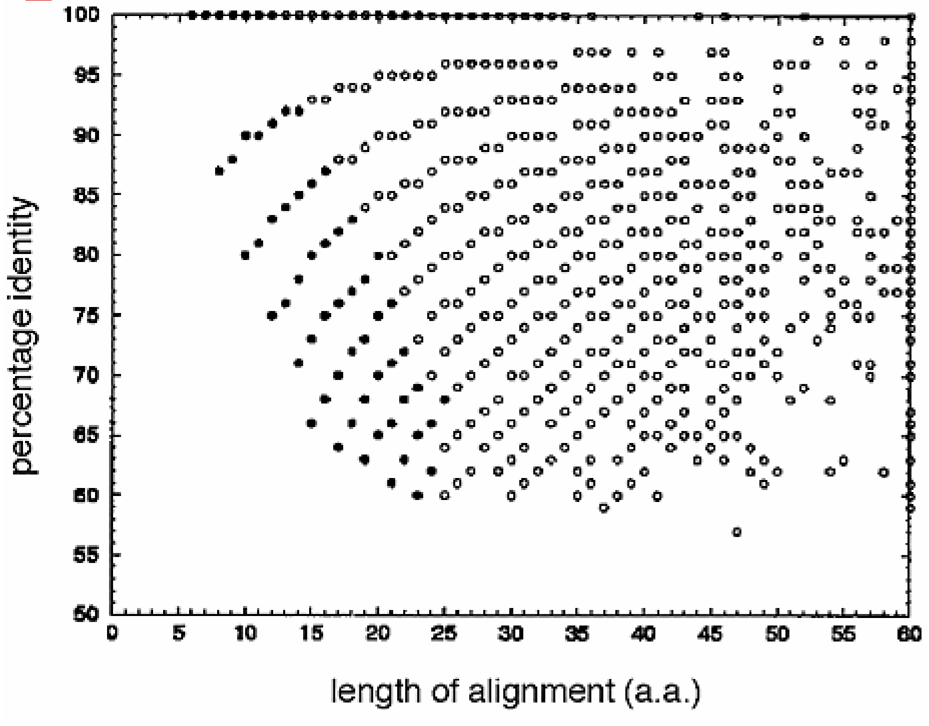
Method	Matrix	W	X	L	I (%)	Sn (%)	Sp (%)	T(s)
BLASTN	NUC.4.4	8 bases	5	30 bases	70	66	93	4.8
BLASTN	NUC.4.4	8 bases	9	40 bases	70	76	94	5.7
BLASTN	NUC.4.4	10 bases	13	30 bases	70	68	40	4.3
TBLASTX	BLOSUM62	3 aa	9	13 aa	60	85	55	74.8
TBLASTX	BLOSUM62	4 aa	3	13 aa	70	80	94	1,065.2
TBLASTX	BLOSUM62	5 aa	1	13 aa	70	84	96	1,160.9
TBLASTX	CNS	4 aa	25	13 aa	70	85	96	10.0
TBLASTX	CNS	5 aa	13	13 aa	70	85	96	29.4
TBLASTX	CNS	5 aa	25	13 aa	70	89	94	29.3

Each program was run with 1,340 different conditions and a representative selection of results is shown. A range of values for W (initial size of the search word) and X (threshold score for consecutive mismatching residues or bases) were tested. For amino acid alignments, a non-substitutive matrix (CNS, match = +15, mismatch = -12) was tested as well as the standard BLOSUM62 matrix. A minimal length (L) and percentage identity (I) were applied to select alignments for which a sensitivity (Sn) and specificity (Sp) were calculated in terms of numbers of overall matching exons. T indicates the time in seconds needed to compare the 13 homologues against each other. The last row shows the optimal performance that was retained for Exofish.

Figure by MIT OCW.

Part II - choosing a threshold

- For best parameters
  - Find threshold by observing alignments
  - Anything higher than threshold will be treated as a predicted exon



Part III - Gene identification

- Matches correspond to exons
  - Not all genes hit
    - A fish doesn’t need or have all functions present in human
    - Even those common are sometimes not perfectly conserved
  - Not all exons in each gene are hit
    - On average, three hits per gene. Three exons found.
    - Only most needed domains of a protein will be best conserved
  - All hits correspond to genuine exons
    - Specificity is 100% although sensitivity not guaranteed

Please see: Crollius, Hugues R., Olivier Jaillon, Alain Bernot, Corinne Dasilva, et al. "Estimate of human gene number provided by genome-wide analysis using Tetraodon nigroviridis DNA sequence." Nature Genetics 25(2000): 235-238. Figure by MIT OCW

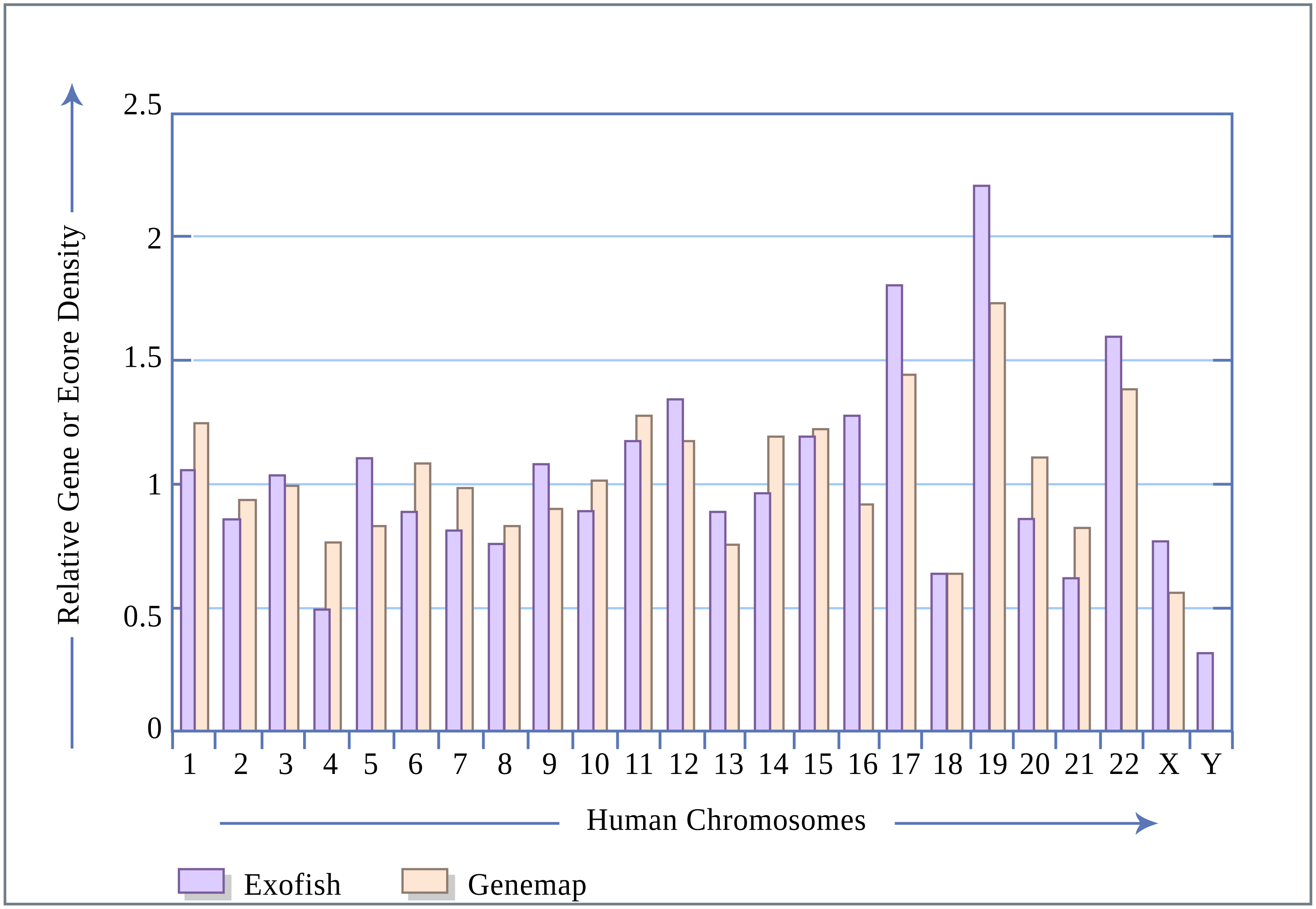
Estimating human gene number

- Extrapolate experimental results
  - Incomplete coverage
    - Model how number would increase with increasing coverage
  - Not perfect sensitivity
    - Estimate how many we're missing on well-annotated sequence
    - Assume ratio is uniform
    - Estimate gene number

Please see: Crollius, Hugues R., Olivier Jaillon, Alain Bernot, Corinne Dasilva, et al. "Estimate of human gene number provided by genome-wide analysis using Tetraodon nigroviridis DNA sequence." Nature Genetics 25(2000): 235-238. Figure by MIT OCW

Gene content by Chromosome

- G ene density varies throughout human genome
  - –ExoFish predicted density corresponds to GeneMap annotation density



Adapted from: Crollius, Hugues R., Olivier Jaillon, Alain Bernot, Corinne Dasilva, et al. "Estimate of human gene number provided by genome-wide analysis using Tetraodon nigroviridis DNA sequence." Nature Genetics 25(2000): 235-238. Figure by MIT OCW

What is hashing

- Content-based indexing
  - Instead of referencing elements by index
  - Reference elements by the elements themselves,
  - by their content
- A hash function
  - Transforms an object into a pointer to an array
  - All objects will map in a flat distribution on array space
  - Otherwise, some entries get too crowded
- What about a database
  - List every location where a particular n-mer occurs
  - Retrieve in constant time all the places where you can find it

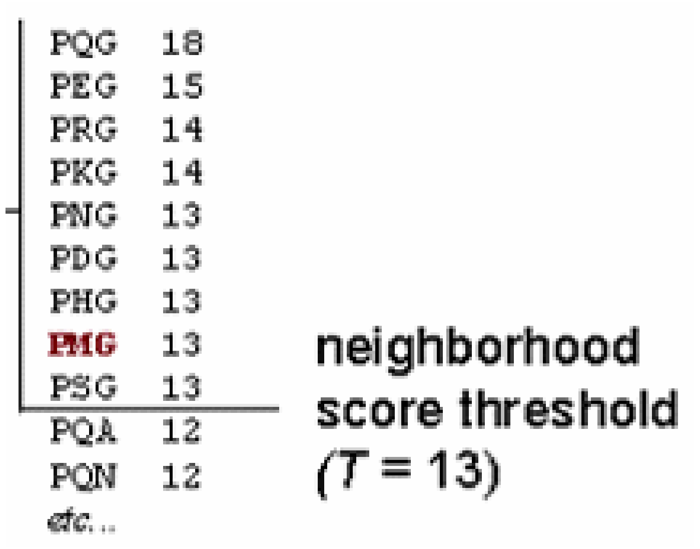
Breaking up the query

query word (W = 3)

Query: GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL

- List them all
  - every word in the query
  - overlapping w-mers

Generating the neighborhood



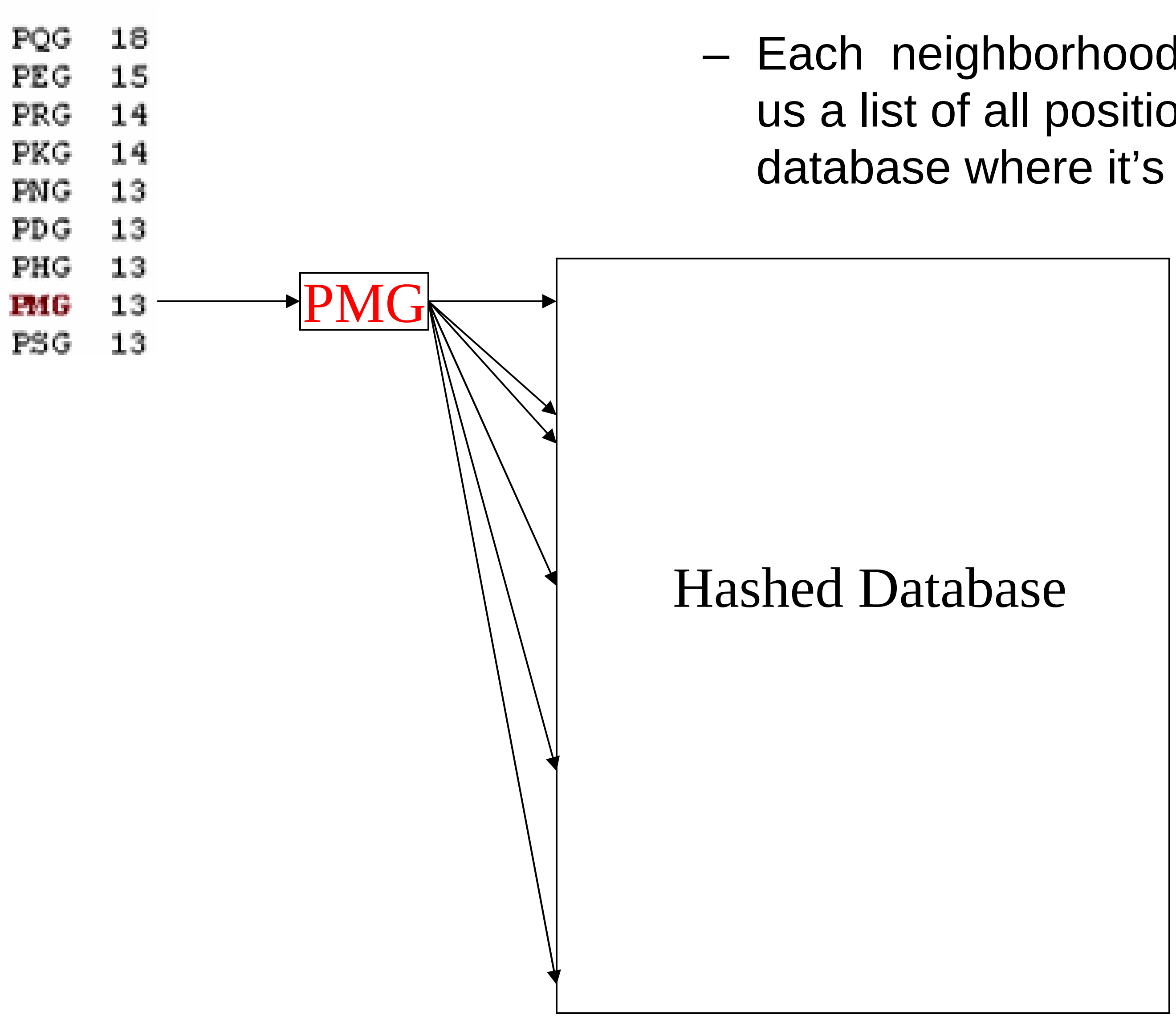
PQG	18
PEG	15
PRG	14
PKG	14
PNG	13
PDG	13
PHG	13
PMG	13
PSG	13
PQA	12
PQN	12
etc...	

neighborhood score threshold (T = 13)

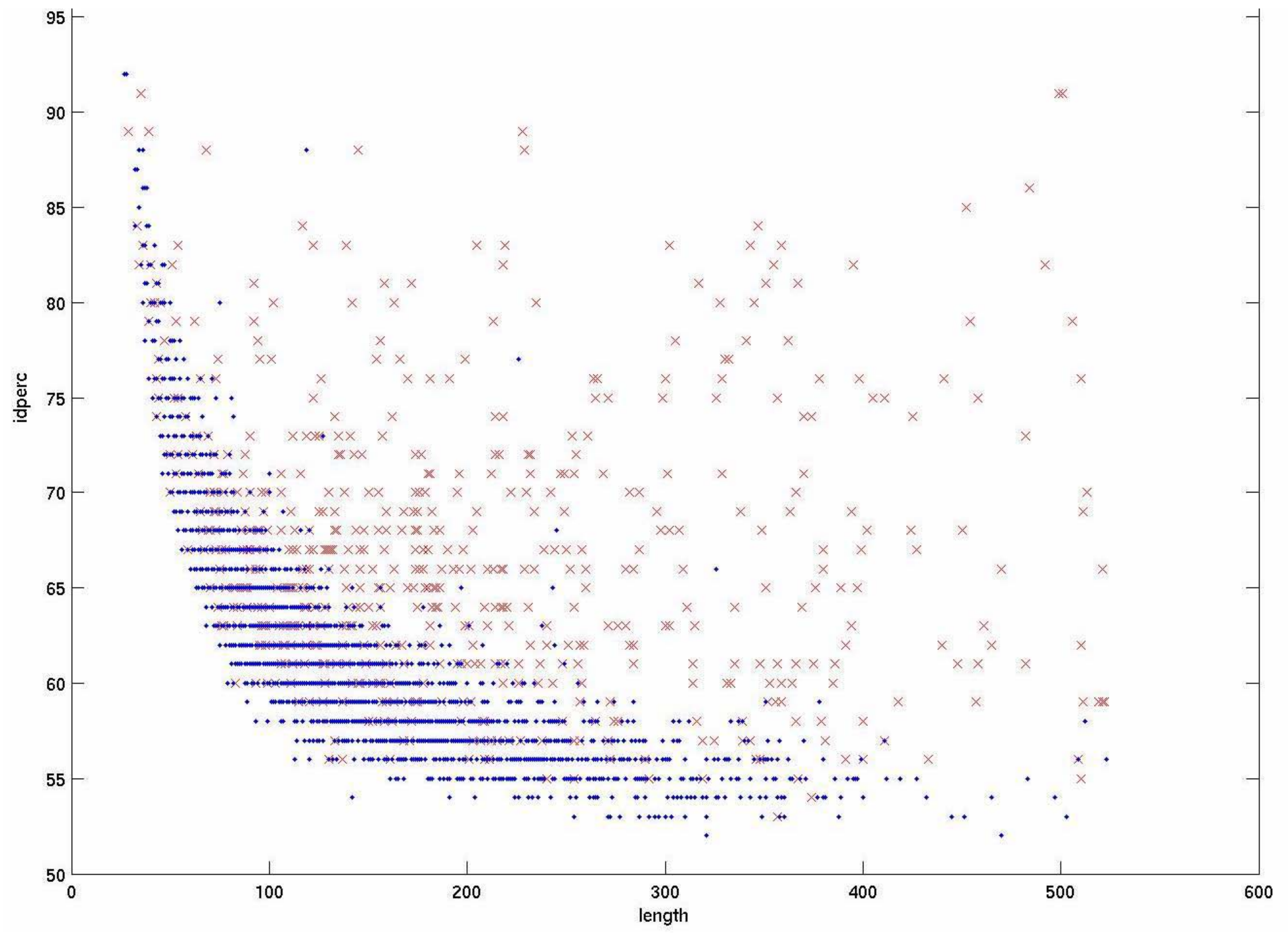
- Enumerate
  - For every amino acid in the word, try all possibilities
  - Score each triplet obtained
  - Only keep those within your threshold

Looking into database

- Follow Pointers
  - Each neighborhood word gives us a list of all positions in the database where it's found



Length and Percent Identity


idperc
length