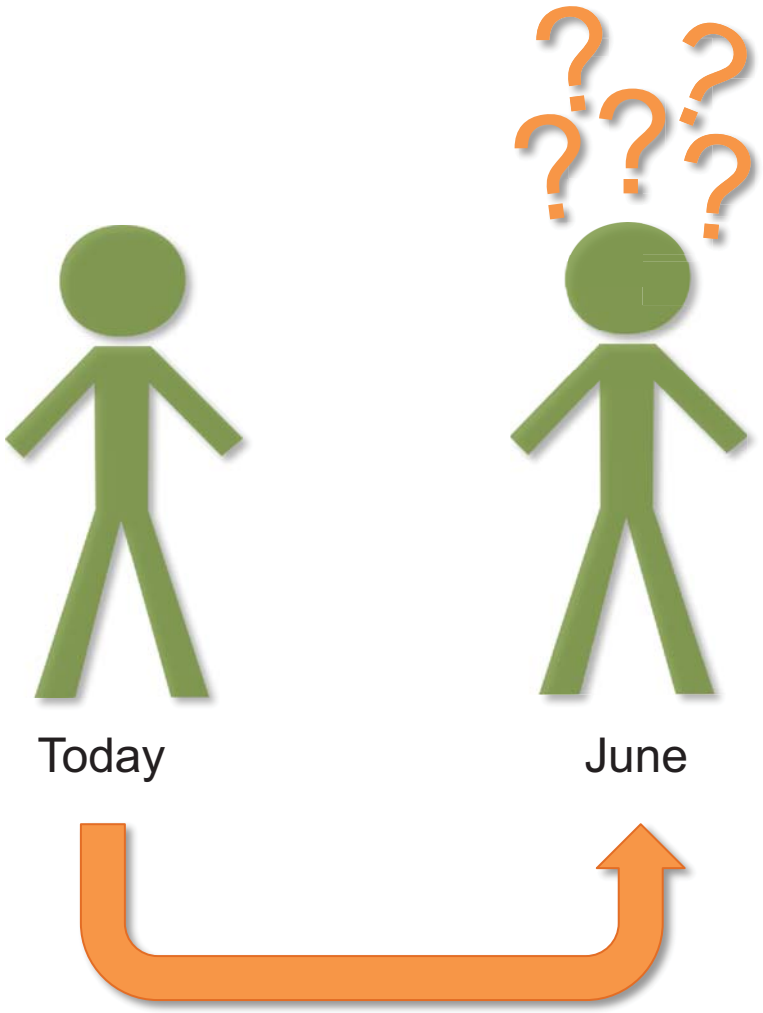
Data Management: File Organization

Data Management Services @ MIT Libraries

- Workshops
- Web guide: http://libraries.mit.edu/data-management
- Individual consultations
  - includes help with creating data management plans

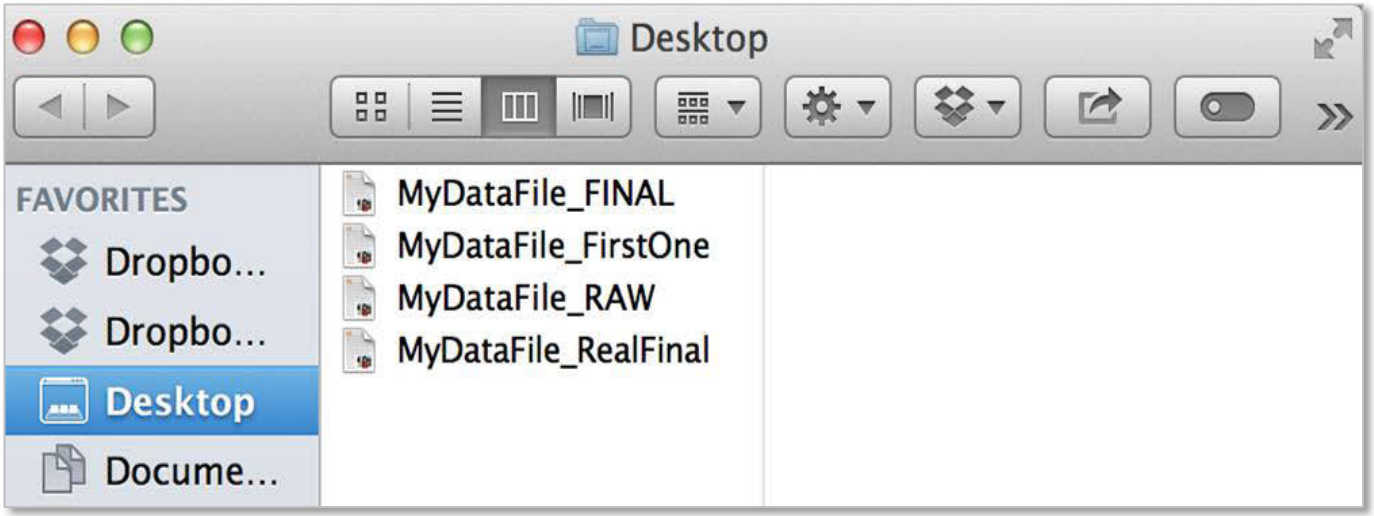
Why file organization is important



The first person with whom you will share your data is yourself.

MITLibraries

Why file organization is important

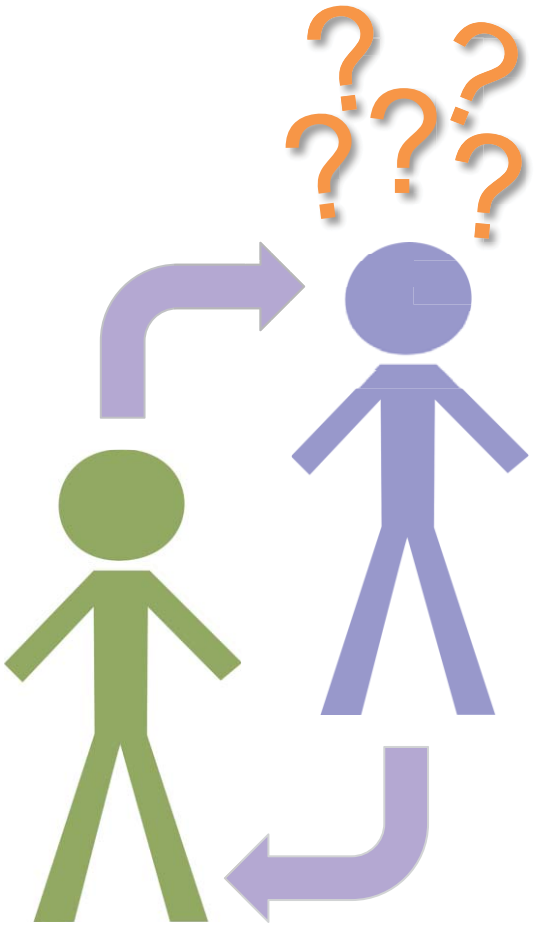


And once your research gets underway, there may be multiple files in various formats, multiple versions, methodologies, etc., all relating to your research.

Why file organization is important

Can someone else understand/use your data files?

Now?
Tomorrow?
In 5 years?



Key principles of file organization

Spending a little time upfront, can save a lot of time later on.

Be realistic: strike a balance between doing too much and too little.

There's no single right way to do it; establish a system that works for you.

Think about who your system needs to work for: Just you? You and your lab group? Collaborators?

Key principles of file organization

Clear

Concise

Consistent

Correct

Conformant

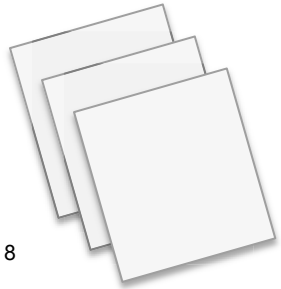
The 5 C's

What do we mean by file organization?

File structures

File naming

File versioning



File structures

where to put data so you can find it

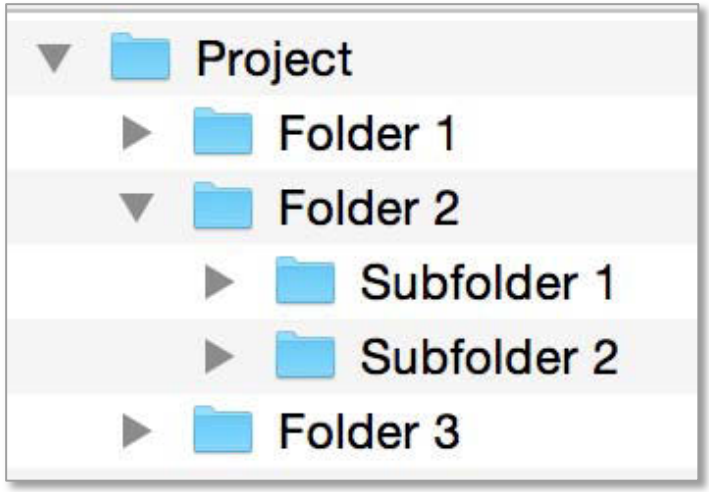
MITLibraries

Method 1: Hierarchical

Items organized in folders and subfolders

Benefits:

- Familiar & widely used
- Good at representing the structure of information
- Similar items are stored together
- Subfolders can function as task lists

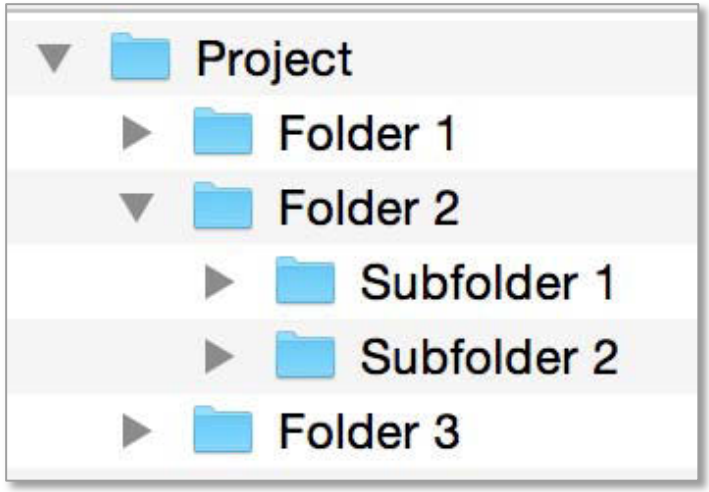


Method 1: Hierarchical

Items organized in folders and subfolders

Drawbacks:

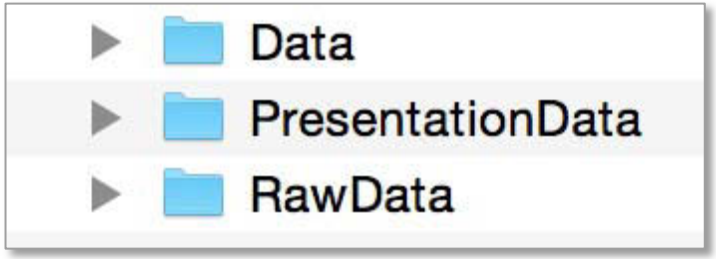
- Surprisingly hard to set up
- Challenging to get the right balance between breadth & depth
- Items can only go in one place
- Time consuming to reorganize if the hierarchy becomes out of date

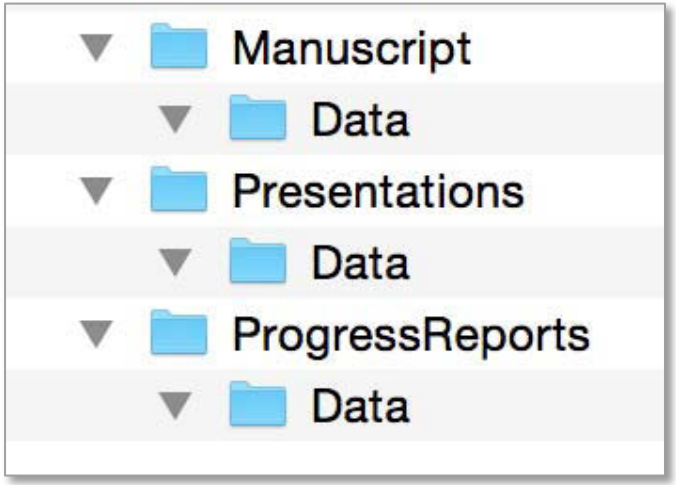


Method 1: Hierarchical

Best practices

- Avoid overlapping categories

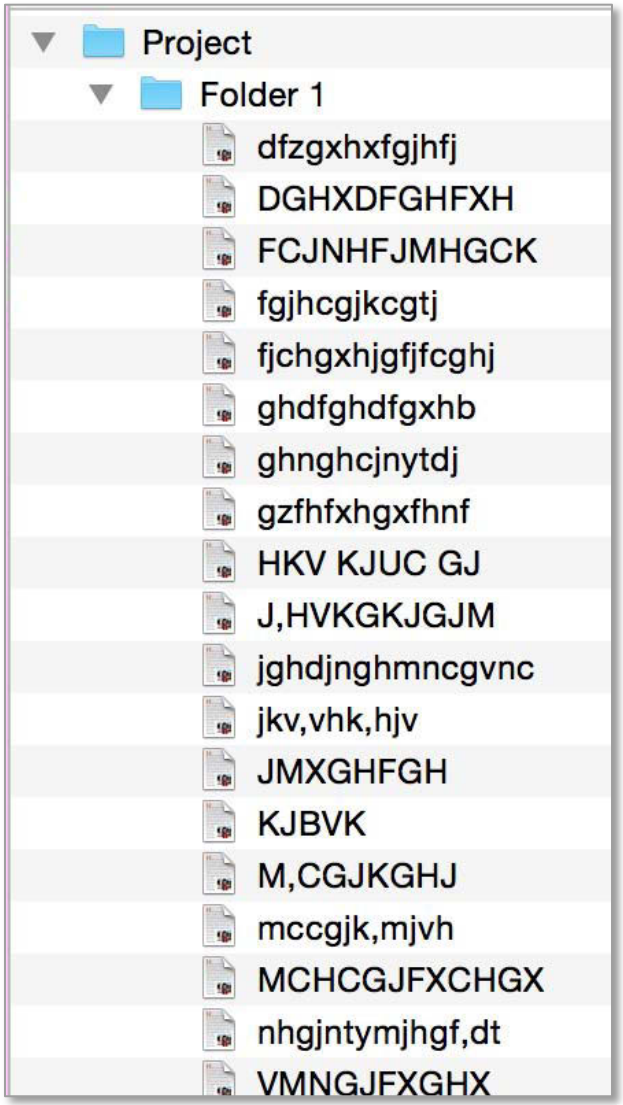




Method 1: Hierarchical

Best practices

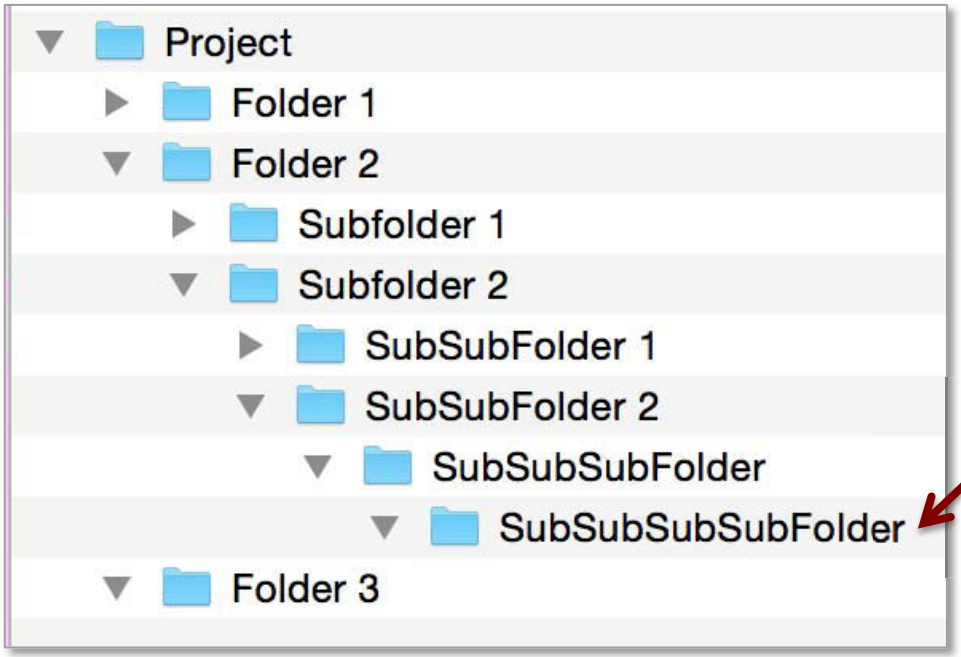
- Avoid overlapping categories
- Don’t let your folders get too big



Method 1: Hierarchical

Best practices

- Avoid overlapping categories
- Don’t let your folders get too big
- Don’t let your structure get too deep



How many clicks does it take to get there?

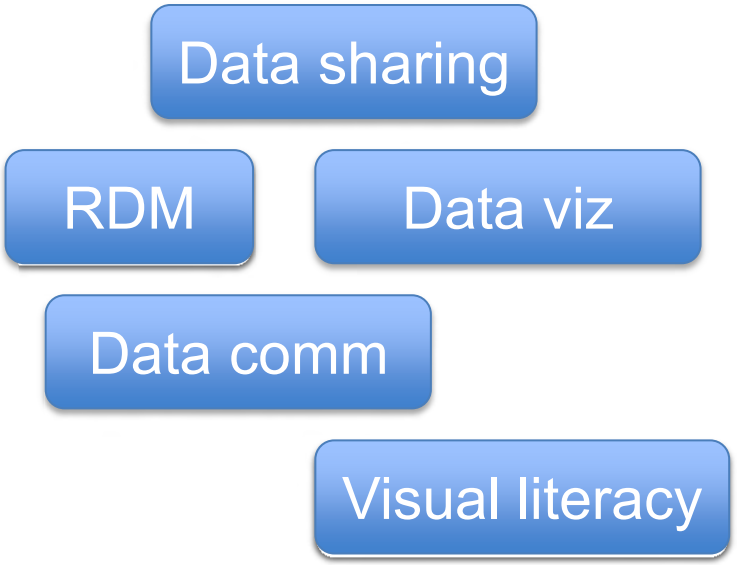
MITLibraries

Method 2: Tag-based

Each item assigned one or more tags

Benefits:

- Items can go in more than one category
- Can be quicker/easier to set up
- When collaborating, it can be easier to combine than hierarchical systems

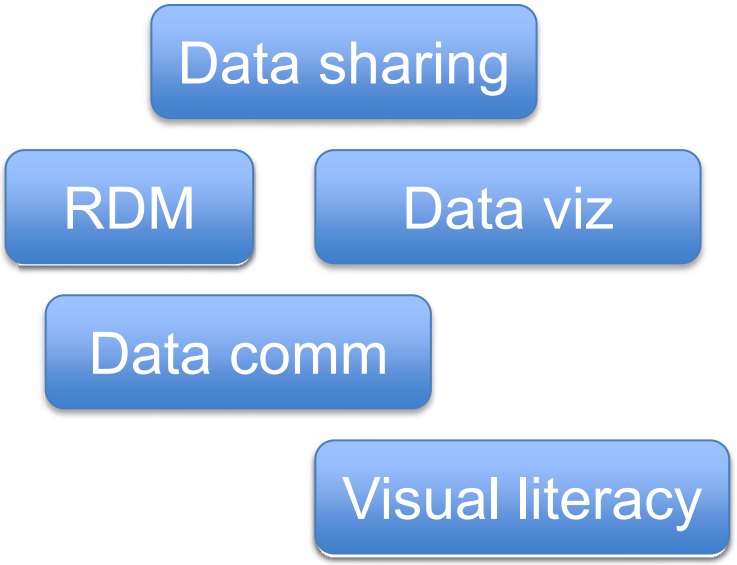


Method 2: Tag-based

Each item assigned one or more tags

Drawbacks:

- Not how operating systems store files
- If item isn't tagged properly when first acquired, it can be hard to find
- Increased risk of inconsistency
- Less good at representing the structure of information



Method 2: Tag-based

Creating a tag-based system:

In OS:

Add searchable keywords/tags to file information

In bibliographic software:

EndNote, Zotero, Mendeley...

Image management programs:

Flickr, Picasa...

Google tools

See our guide to Tagging and Finding Your Files:
http://libguides.mit.edu/metadataTools/

Your file structure

- Hierarchical
- Tag-based
- Hybrid

What sort of structure(s) do you currently use?
What's working in this system?
What's not working?

Creating a systematic file folder structure

Document your system and use it consistently

Tips for defining your system:

- Define the types of data and file formats
- Include important contextual information
- Organize folders by meaningful categories
  - primary/secondary/tertiary
  - subject/collection method/time
- Choose a directory naming convention
- Be Clear, Concise, Consistent, Correct, Conformant

A case study

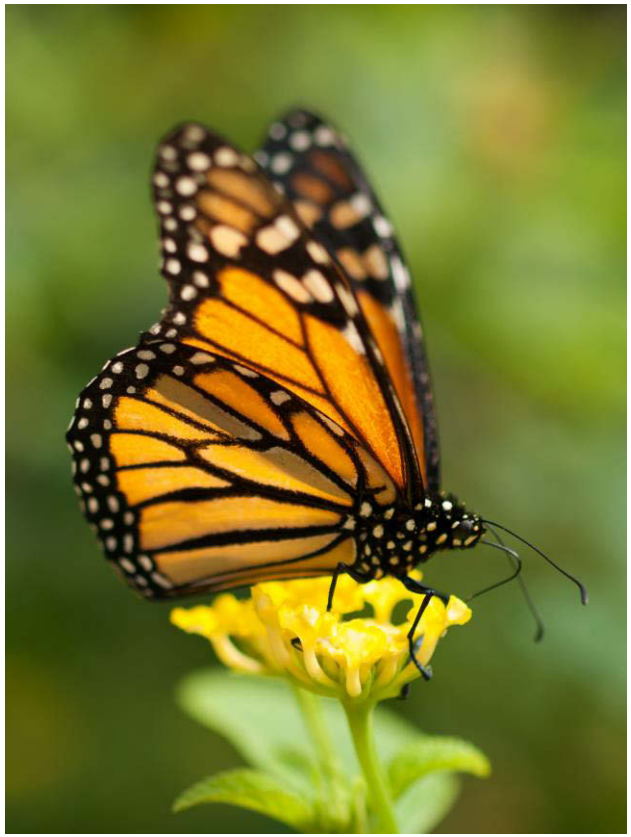


Photo Courtesy of William Warby on Flickr. License CC-BY.

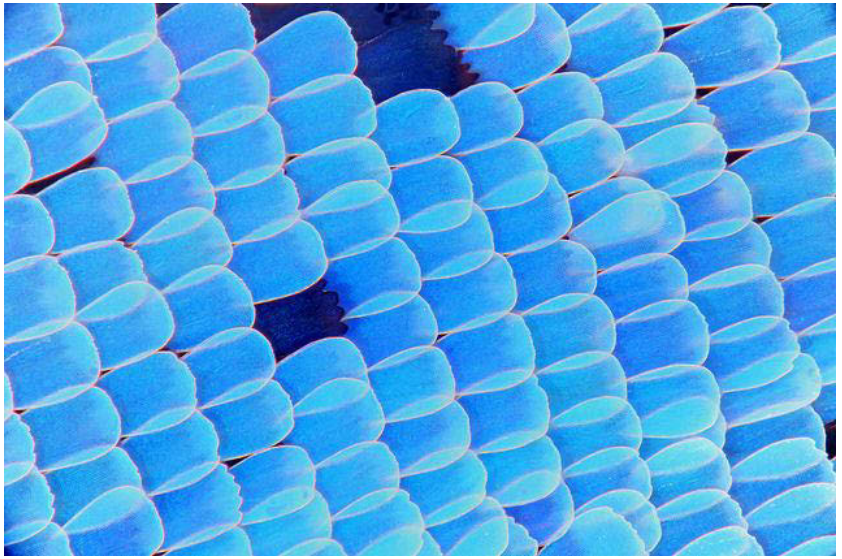


Photo Courtesy of Macroscopic Solutions on Flickr. License CC-BY.

Creating a systematic file folder structure

Type of data and file formats:

- Images (in multiple file formats)
- Data in tabular format (some captured on the fly) about each specimen collected (visual characteristics, time, location, etc.)
- Data on weather from NOAA
- Project documents (grant proposal, etc.)
- PDFs of related literature
- And more…

Creating a systematic file folder structure

Include important contextual information:

- Date
- Collection method
- Collector
- …

Creating a systematic file folder structure

Example file structure systems/directory hierarchy conventions:

/[Project]/[Sub-project]/[Experiment]/[Instrument]/[Date]
/[Research area]/[Project]/[Data vs. documentation]/[Date]

/[Project]/[Type of file]/[Data collector name]/[YYYYMMDD]

For the butterfly project:

/butterfly/images/mcneill/20160117
/butterfly/tabular/mcneill/20160117
/butterfly/projectDocs/
/butterfly/literature/

A quick word on organizing/storing articles

Would I really want to store my literature files simply in a directory? Maybe, but...

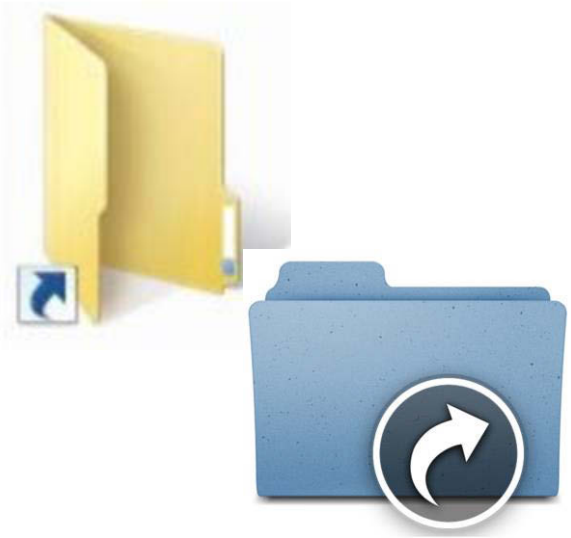
Consider using citation management tools

zotero

http://libguides.mit.edu/references

Tips for discovering your files

- Order dates beginning with the year to enable sorting by date (e.g., YYYYMMDD)
- Embed metadata in your files (if possible)
- Add shortcuts to files within other relevant folders



File naming

what to call data so you know what it is

MITLibraries

File naming conventions

Naming conventions make life easier!

Naming conventions should be:

- **Descriptive**
- Consistent

Consider including:

- Unique identifier (ie. Project Name or Grant # in folder name)
- Project or research data name
- Conditions (Lab instrument, Solvent, Temperature, etc.)
- Run of experiment (sequential)
- Date (in file properties too)
- Version #

File naming conventions

Naming conventions make life easier!

Naming conventions should be:

- Descriptive
- **Consistent**

YYYYMMDD MMDDYYYY YYMMDD MMDDYY MMDD DDMM	TimeDate DateProjectID TimeProjectID	Sample001234 Sample01234 Sample1234

Include the same information

Maintain order

File naming conventions

Best Practice	Example
Limit the file name to 32 characters (preferably less!)	32CharactersLooksExactlyLikeThis.csv
When using sequential numbering, **use leading zeros** to allow for multi-digit versions For a sequence of 1-10: 01-10 For a sequence of 1-100: 001-010-100	**NO** ProjID_1.csv ProjID_12.csv **YES** ProjID_01.csv ProjID_12.csv
Don't use special characters & , * % # ; * ( ) ! @$ ^ ~ ' { } [ ] ? < > -	**NO** name&date@location.doc
Use only one period and use it before the file extension	**NO** name.date.doc **NO** name_date..doc **YES** name_date.doc
Avoid using generic data file names that may conflict when moved from one location to another	**NO** MyData.csv **YES** ProjID_date.csv

Our case study

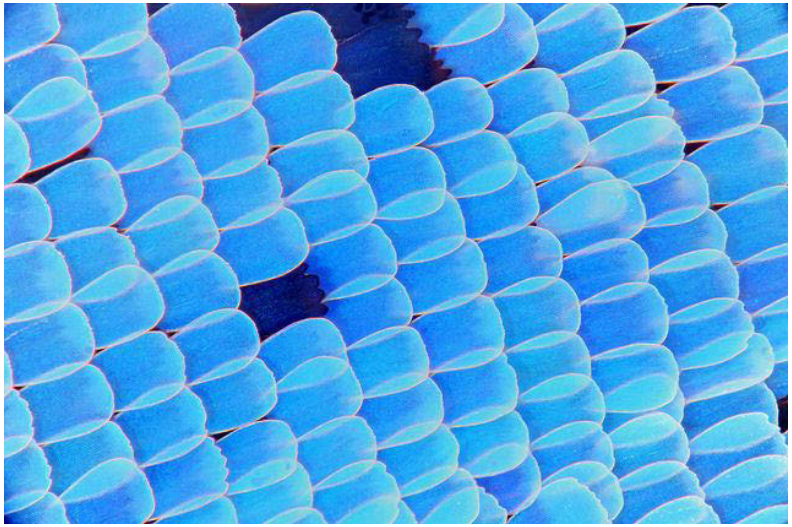


Photo Courtesy of Macroscopic Solutions on Flickr. License CC-BY.

Maybe started with:
abcdefghijklmnopqrstuvwxyz.sam

Sashimi Microscope format

Date as YYYYMMDD

Ascension # because part of a series

sam_monarch_wing_20150415_CM_001.tif

File format

Descriptive element

Initials because working in a group

File naming & instrumentation

Check to see if your instrument, software, or other equipment that outputs your data files can be set with a file naming system

Less work than retrospectively changing filenames

But if you still have to change many file names downstream…

File naming & batch/bulk renaming

Can use tools that retrospectively align file/folder names with naming conventions

Caveats:

- Ideally you want to be able to map the original to new names
- Make sure it doesn't change the file extension

Some File Renaming Tools:

Bulk Rename Utility
Renamer
PSRenamer
WildRename

File naming & discipline standards

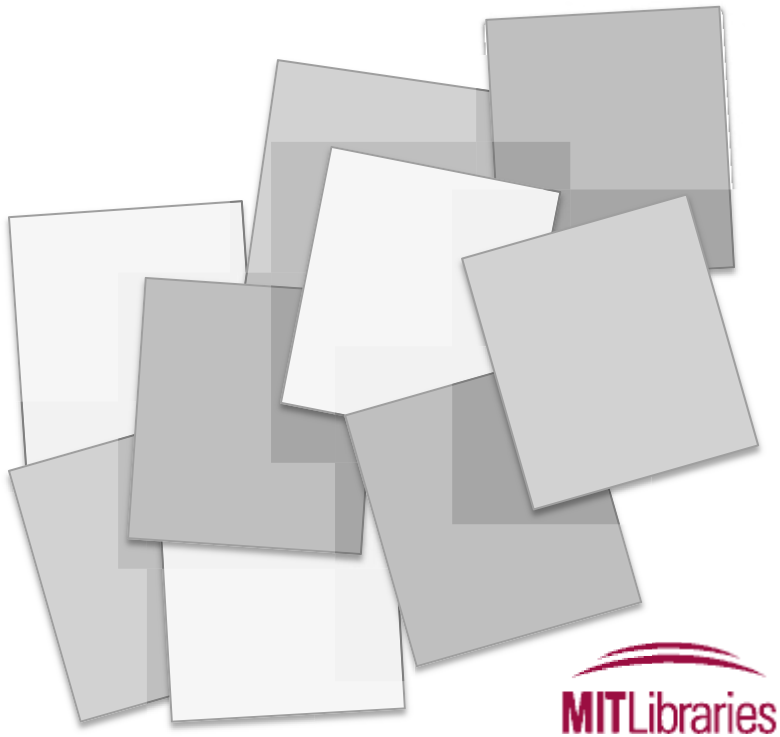
Check for established file naming conventions in your discipline

Some examples:

DOE's Atmospheric Radiation Measurement (ARM) program
GIS datasets from Massachusetts
The Open Biological and Biomedical Ontologies

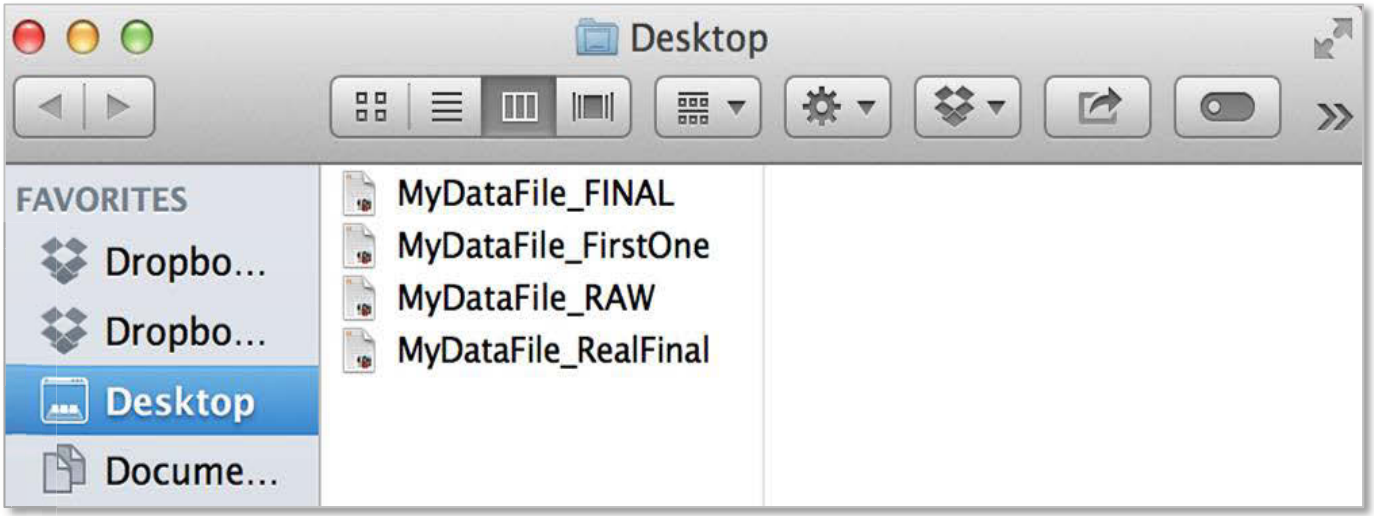
File versioning

keeping track of data



MITLibraries

Versioning: *the why*



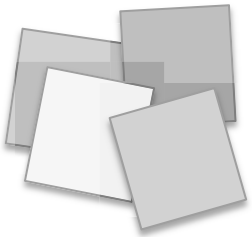
Versioning: *the when*

Depending upon practices in your field, version either:

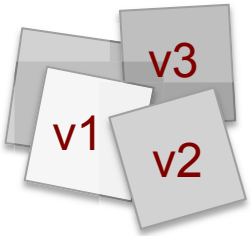
- Analysis/program/script files
- Data files themselves

Also important for project documentation and files

Versioning: *the how*



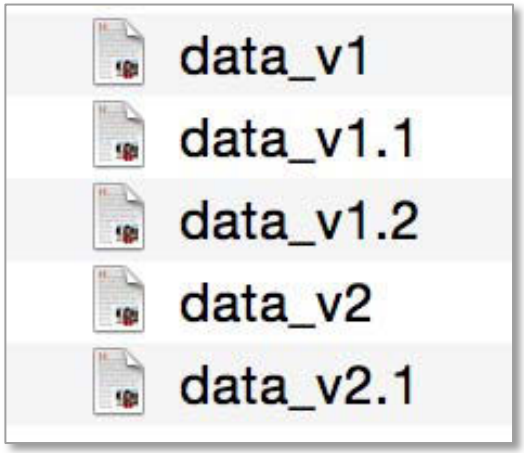
Save new versions



Establish a consistent convention

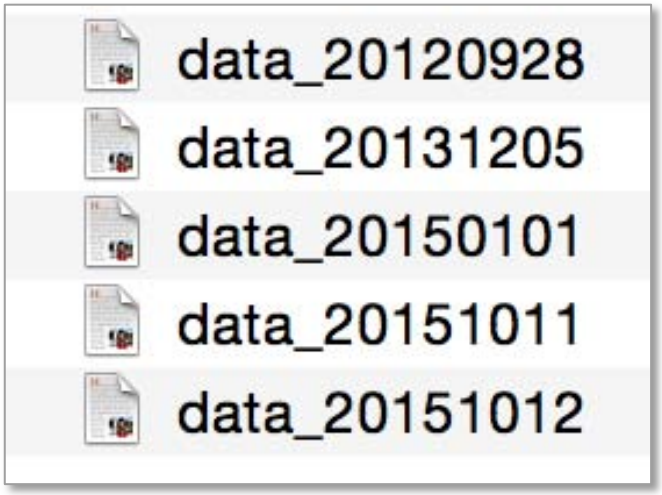
Versioning: *the how*

Use ordinal numbers (1,2,3,etc) for major version changes and a decimal for minor changes



Versioning: *the how*

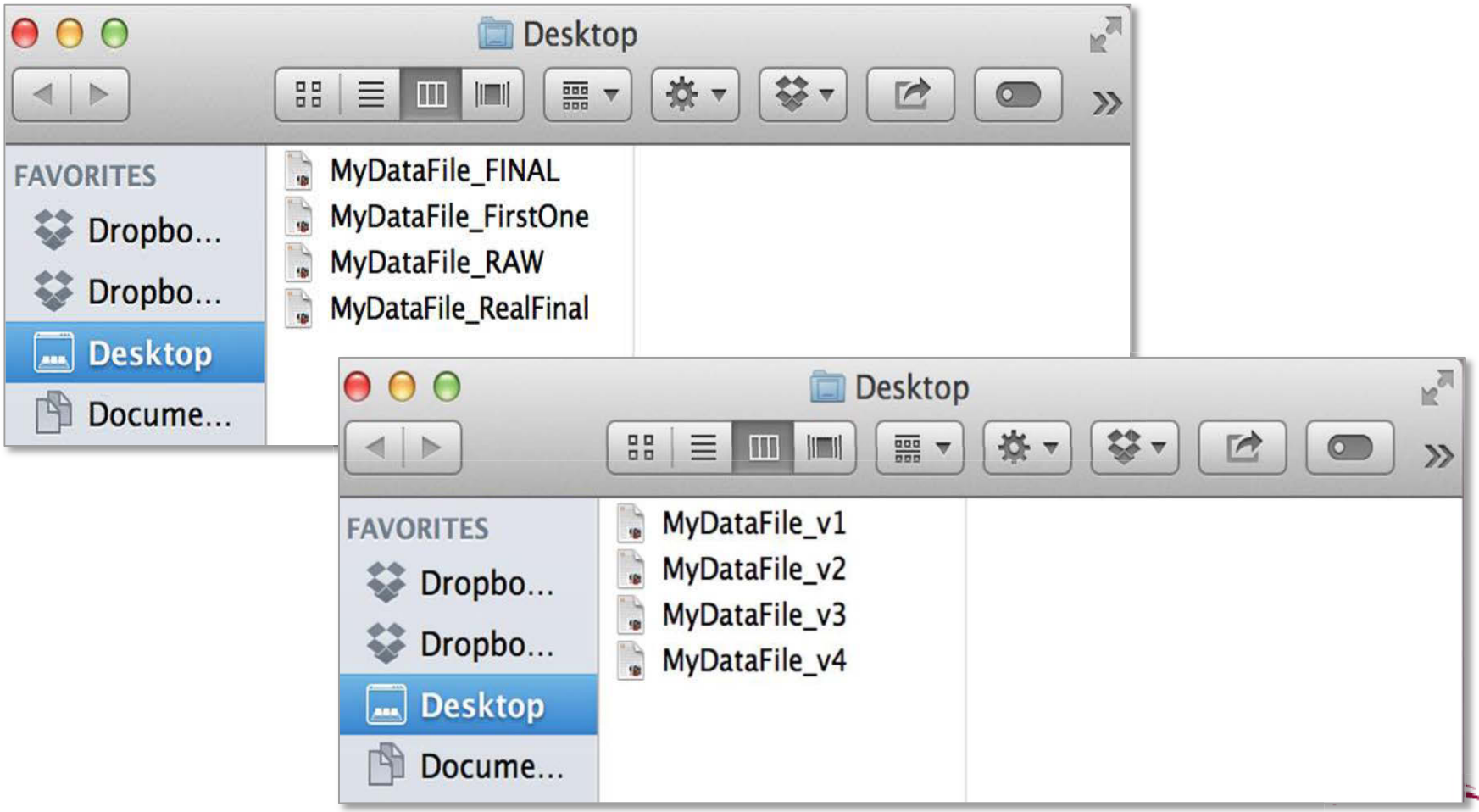
Use dates to distinguish between successive versions



Not ideal when you can potentially have multiple versions in a day.

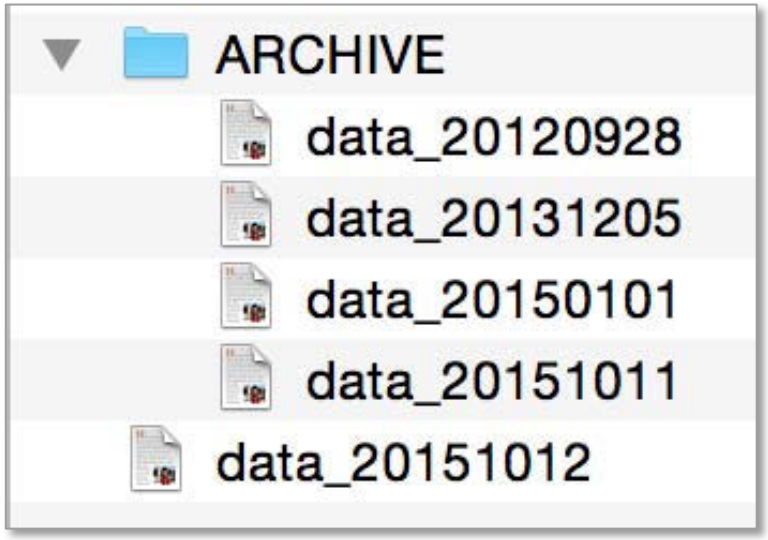
Versioning: *the how*

Avoid imprecise “final” labels



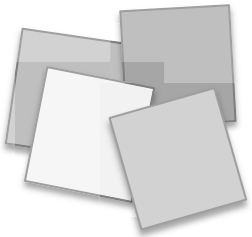
Versioning: *the how*

Put older versions in a separate folder

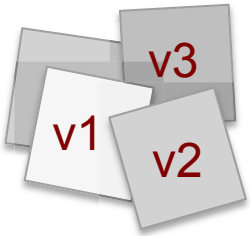


Do you really need to keep obsolete versions?

Versioning: *the how*



Save new versions



Establish a consistent convention

Document your convention

Versioning: *document it!*

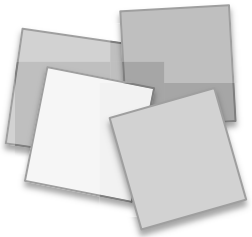
Some options:

- Create a version table or file history w/in or alongside your data files

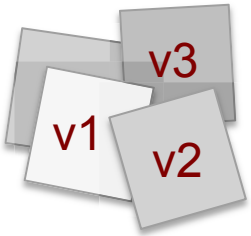
- Use built-in capabilities of software (when available)
  - Wikis, Google docs, etc. that track changes
  - Platforms that allow for checking in/out files
  - Setting permissions

- Use version control software
  - Git, GNU RCS, Mercurial (Hg), etc.

Versioning: *the how*



Save new versions



Establish a consistent convention

Document your convention

Consider your version control needs

Version control: *general tip*

Be careful when syncing across platforms & simultaneously editing!

Appendix: detailed tips

Tip 1: Embedding metadata

- If feasible, try to enter basic information about the data file within its contents (e.g., author, date created/modified, project, grant, version)
  - May be able to <comment> information in a file
  - May help to identify files using your system's full-text searching capabilities
- Embed metadata in header
- May also be able to assign this information as tags (external to your files); see our guide to Tagging and Finding Your Files: http://libguides.mit.edu/metadataTools/
  - Caveat: some programs strip tags during file transfer or transformation, so don't rely solely upon these

Tip 2: adding searchable keywords to files in Windows

- Open up the Windows folder view and highlight (don't click to open) your file of interest
- In the pane at the bottom of the folder window, you'll see metadata about your file
- Click the property that you want to change/add (you'll see the box for tags all the way on the right), type the new property, and then click Save.
- To add >1 tag, separate each with a semicolon.
- Terms entered here will be found by the Windows search function

Tip 3: Adding tags on a Mac

- When you save a file, from the document menu, or in Finder
- Spotlight Comments (and use Spotlight to search)
- http://support.apple.com/kb/HT5839
- http://www.maclife.com/article/howtos/mavericks_howto_organizing_files_and_folders_tags
- http://computers.tutsplus.com/tutorials/how-to-tag-files-and-create-spotlight-comments-on-a-mac--mac-46431

Tip 4: Shortcuts in Windows

- Shortcuts allow you to open a file from multiple places
- Functions to place a file in >1 category
- Use for frequently accessed items
- Use to create project folders

Tip 5: Shortcuts on a Mac

- On OS X you can create "symbolic links" using the terminal and the 'ln -s' command
- Use Automator (http://support.apple.com/kb/ht2488), alone or in conjunction with AppleScript (http://www.macosxautomation.com/applescript/)

Appendix 2: Batch renaming tools

- Ant Renamer (Windows)
- Bulk Rename Utility (Windows)
- ImageMagick (Windows, Mac, or Linux)
- GNOME Commander (Linux)
- GPRename (Linux)
- Name Mangler (Mac)
- PSRenamer (Windows, Mac, or Linux)
- Renamer4Mac (Mac)
- WildRename (Windows)

In **Unix**: Use the **grep** command to search for regular expressions

RES.STR-002 Data Management

Spring 2016