Data organisation

On this page

What is the best way to name a file?

Description

Brief and descriptive file names are important in keeping your data files organised. A file name is the principal identifier for a file and a good name gives information what the file contains and helps in sorting them, but only if you have been consistent with the naming.

Considerations

- Best practice is to develop a file naming convention with elements that are important to your project already when the project starts.
- When working in collaboration with others, it is important to follow the same file naming convention.

Solutions

Tips for naming files

- Balance with the amount of elements: too many makes it difficult to understand vs too few makes it general.
- Order the elements from general to specific.
- Use meaningful abbreviations.
- Use underscore (_), hyphen (-) or capitalized letters to separate elements in the name. Don't use spaces or special characters: ?!& , * % # ; * ( ) @$ ^ ~ ' { } [ ] < >.
- Use date format ISO8601: YYYYMMDD, and time if needed HHMMSS.
- Include a unique identifier (see: Identifiers)
- Include a version number if appropriate: minimum two digits (V02) and extend it, if needed for minor corrections (V02-03). The leading zeros, will ensure the files are correctly.

- Write your file naming convention down and explain abbreviations in your data documentation.
- If you need to rename a lot of files in order to organize your project data and manage your files better, it is possible to use applications like Bulk Rename Utility (Windows, free) and Renamer4Mac (Mac).

Example elements to include in the file name

- Date of creation
- Project number / Experiment / Acronym
- Type of data (Sample ID, Analysis, Conditions, Modifications, etc.)
- Location / Coordinates
- Name / Initials of the creator
- Version number
- Reserve the last 3-letters for file format (e.g. .xls, .rtf, .mov, .tif, .doc)

Examples of good file names

- Honeybee project, experiment 2 done in Helsinki, data file created on the second of December 2020
  - File name: `20201202_HB_EXP2_HEL_DATA_V03.xls`
  - Explanation: `Time_ProjectAbbreviation_ExperimentNumber_Location_TypeOfData_VersionNumber`
- Cropped image of an ant head taken on the third of December 2020 by Meg Megson
  - File name: `20201203_MM_HEAD_CROPPED_V1.psd`
  - Explanation: `Time_CreatorData_TypeModification_Version`

How to choose the appropriate file formats?

Description

File formats play an important role when you try to open files later or another person would like to work with the data. Some file formats keep structured data and even permit metadata inclusion, hereby enabling machine-readability and promoting interoperability. Others are easy for humans to understand. Each type of format has use cases. However, as a general principle, choosing open and widely supported file formats ensures long-term compatibility and accessibility in the foreseen future.

Considerations

When making a selection for an appropriate file format, you should consider the following factors whenever feasible:

- non-proprietary;
- based on open standard;
- commonly used in your research domain;
- uncompressed;
- unencrypted.

It is important to differentiate between file formats intended for active phase (data acquisition, data reduction and primary data analysis) and those designed for long-term storage or reuse (sharing, publishing and archiving). For the latter purpose, we recommend utilizing file formats that adhere to open standards, have a broad acceptance, and are unlikely to become obsolete. In the active phase, it is fine to use proprietary device-specific file formats if needed. This is acceptable until you reach the phase of sharing the data for subsequent analysis, and for data validation or control with other team members. At this point, you need to convert (or export) the data for it be usable by members without access to proprietary software or instrumentation that generated it.

Solutions

The best file formats depend on data types, availability and common acceptance of open file formats and research domain. There is no one size fits all solution. You need to choose the best for your case.

The following table lists the recommended file formats for best practices in research data management. Acceptable and non-recommended file formats represent commonly used file formats that do not fulfill above-mentioned criteria (listed under 'Considerations').

Type	Preferred	Acceptable	Non-recommended
Rich text documents	ODT (.odt) Markdown (.md) LaTeX (.tex) for read-only documents: PDF/A (.pdf)	Office Open XML (.docx)	Microsoft Word (.doc) PDF other than PDF/A (.pdf)
Plain text documents	Unicode text (.txt)		Non-Unicode text (.txt)

Type	Preferred	Acceptable	Non-recommended
Tabular data	CSV (.csv, .tsv)	Office Open XML Workbook (.xlsx)	Microsoft Excel (.xls)
Containers and compression	ZIP (.zip)	tar (.tar) gzip (.gz) bzip2 (.bz2)	RAR (.rar) 7-Zip (.7z)
Raster images	TIFF (.tif, .tiff) DICOM (.dcm)	proprietary microscopy formats (CZI, LIF, NEF) PNG (.png)	JPEG (.jpg, .jpeg) PS (.ps) EPS (.eps) BMP (.bmp)
Vector images	SVG (.svg)	PS (.ps) EPS (.eps)	Adobe Illustrator (.ai) WMF/EMF (.wmf, .emf) CDR (.cdr)
Audio	Matroska (.mka) FLAC (.flac)	WAVE (.wav) MP3 (.mp3)	
Video	Matroska (.mkv)	MPEG/MPG animation (.mpg, .mp4, .mjpeg)	AVI (.avi) QuickTime (.mov, .qt)
Machine-readable metadata	JSON (.json) XML (.xml)		

For domain-specific file formats, please check the appropriate domain page.

How do you manage file versioning?

Description

File versioning is a way to keep track of changes made to files and datasets. While the implementation of a good file naming convention will indicate that different versions exist, this will not explain the difference between two (or more) versions. File versioning will provide transparency about which actions and changes were made and when. This makes it e

backtrack and find something that was present in a previous version, but was later deleted or changed.

Considerations

- Do you need to collaborate on files, perhaps at the same time?
- Is there a need to be able to backtrack and restore a previous version?
- Will there be many changes made?

Solutions

- Smaller demands of versioning can be managed manually e.g. by keeping a log where the changes for each respective file is documented, version by version.
- For automatic management of versioning, conflict resolution and back-tracing capabilities, use a proper version control software such as Git, hosted by e.g. GitHub, GitLab and Bitbucket.
- Use a Cloud Storage service (see Data storage page) that provides automatic file versioning. It can be very handy for spreadsheets, text files and slides.

How do you organise files in a folder structure?

Description

A carefully planned folder structure, with intelligible folder names and an intuitive design, is the foundation for good data organisation. The folder structure gives an overview of which information can be found where, enabling present as well as future stakeholders to understand what files have been produced in the project.

Considerations

- The decisions on how to organise the files should be made during planning and design of the project, so that the strategy can be implemented from the start.
- Consider to consistently apply the same strategy in every project within the research group.

Solutions

Folders should:

- follow a structure with folders and subfolders that correspond to the project design and workflow
- have a self-explanatory name that is only as long as is necessary
- have a unique name – avoid assigning the same name to a folder and a subfolder

The top folder should have a README.txt file describing the folder structure and what files are contained within the folders. For information on the content of a README file see corresponding section on Documentation and metadata page. See also A Quick Guide to Organizing Computational Biology Projects.

An example:

```
project/
  code/                code needed to go from input files to final results
  data/                raw and primary data (never edit!)
     raw_external/
     raw_internal/
     meta/
  doc/                 documentation of the study
  intermediate/        output files from intermediate analysis steps
  logs/                logs from the different analysis steps
  notebooks/           notebooks that document your day-to-day work
  results/             output from workflows and analyses
     figures/
     reports/
     tables/
  scratch/             temporary files that can safely be deleted or lost
  README.txt           file and folder description
```

- Structured directories can be made by using Cookiecutter ,a command-line utility that creates projects from cookiecutters (project templates), e.g. creating a Python package project from a Python package project template.

Related pages

Tool assembly

OMERO

OMERO is a software platform for managing, sharing and analysing images data.

Tool assembly

TransMed

TransMed tool assembly from ELIXIR Luxembourg supports projects in clinical and translational biomedicine.