Managing your data

Data (mis)management in practice.

	Data acquisition	Analysis	First submission	Review	Second submission	Publication
Raw data	Data arrives in cumbersome and proprietary format.	Gets converted to format of choice. Original files (and conversion settings) are lost.		Leads a quiet life on the HPC cluster, until the project expires and the data has to be urgently retrieved.	Ends its days on an external hard drive on the researcher's desk.	"Data available upon request".
Metadata	In researcher's lab journal.	Hard-coded in various analysis scripts.	Mailed back and forth between collaborators in ever-changing (but nicely colored) Excel sheets.		Reformatted and included as PDF in the supplementary.	

FAIR data

Strive to make your data FAIR:

- Findable
- Accessible
- Interoperable
- Reusable

for both machines and humans.

Wilkinson, Mark et al. “The FAIR Guiding Principles for scientific data management and stewardship”. Scientific Data (2016)

Data management plan

- Check requirements of funding agency and field of research.
- Determine required storage space for short and long term.
- Provide helpful metadata.
- Consider legal/ethical restrictions if working with sensitive data.
- Find suitable data repositories.
- Strive towards uploading data to its final destination already at the beginning of a project.

VR Data management plan requirements

Data sharing

Why Open Access?

- Publicly funded research should be unrestricted.
- Published results should be verifiable by others.
- Enables other to build upon previous work.

Organizing your projects

Which sample file represents the latest version?

```
$ ls -l data/
-rw-r--r--  user  staff  samples.mat
-rw-r--r--  user  staff  samplesFinal.mat
-rw-r--r--  user  staff  samplesFinalV2.mat
-rw-r--r--  user  staff  samplesUSE_THIS_ONE.mat
-rw-r--r--  user  staff  samplesV2.mat
```

The project directory

The first step towards working reproducibly: Get organized!

- Divide your work into distinct projects.
- Keep all files needed to go from raw data to final results in a dedicated directory.
- Use relevant subdirectories.

There are many ways to organize a project

One example: NBISweden/project_template

code/	code needed to go from input files to final results
data/	raw and primary data (never edit!)
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
scratch/	temporary files that can be safely deleted or lost
config.yml	configuration of the project workflow
Dockerfile	recipe to create a project container
environment.yml	project dependencies list used to create software environment
README.md	project description and instructions
Snakefile	workflow file used by snakemake

There are many ways to organize a project

Another example: snakemake-workflows/template

```
config/
  - config.yaml
  - samples.tsv
resources/
results/
  - logs/
  - plots/
  - tables/
workflow/
  - envs/
  - report/
  - rules/
  - schemas/
  - scripts/
  Snakefile
LICENSE
README.md
```

Helpful tools

syntax highlighting, autocomplete, git integration etc

- Atom
- RStudio
- PyCharm

Questions?

Topics for discussion in breakout rooms

- Do you organize your work in distinct projects?
- How do you organize your files in this context?
- Are you happy with the way you work today?
- Does your group have a data management plan in place?
- Do you know "your" repositories and how to submit data to them?