Reproducible Research
Dag Ahrén
Best Practices in Handling Genomic Data
Interests outside of academia
My Background

- Biologist turned Bioinformatician
- Genomics research since before NGS
My Research

Single cell population genomics
*Gonyostomum semen*

Avian malaria host parasite interactions

Toxin gene clusters in *Microcystis*

Adaptation to radiation in black yeasts

Methanogens and methanotrophs in SubArctic and Arctic Ecosystems
National Bioinformatics Infrastructure Sweden

~100 staff at six different sites across Sweden with expertise in many different omics-related areas
Best practices in Handling Genomic Data
Lets start cooking!
Ingredients

- Data management
- Reproducible research
- Lab
What is data management anyway?
Research data management concerns the:

- organization
- storage
- preservation
- sharing of data that is collected or analysed during a research project.

Data should be FAIR (Findable, Accessible, Interoperable & Reusable) [Wilkinson et al 2016]
Good Data Management practices

Important at all stages of the project and beyond
Drives Open Science!

Funding agencies such as European Research Council (ERC) requires DM So do many US agencies as well (e.g. NSF & NIH)

- Keep data secure & safe
- Deposit data
- Follow community practices
- Maintain a Data Management Plan
Data management plan

Get help from NBIS, (National Bioinformatics Infrastructure Sweden)

Data management plan wizards can help: Data management plan

Also DMPonline can be helpful
Reproducibility research
What does research reproducibility mean?

NSF definition from Goodman et al (Science 2016):

Reproducibility refers to the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator.

It may still not be correct! Reproducing the same bias/mistake = same (wrong) results
Why important?

- To be able to rerun analyses again with new parameters/data
- Assist in keeping track of what was done when publishing
- To fulfill the requirements of grant agencies, journals
- Increase the usability of the data and results for the community (and yourself!)

Reproducible research should be called research -Niclas Jareborg (NBIS)
If so important why are we not doing it?

Anonymous question on reproducible research 29 PhD students participated

Make it easier & less time consuming!
My thoughts...

- Set realistic goals but do set them
- A small step forward is progress!
- Set deadline
- Share and help each other & give positive feedback (e.g. github repo)

Comments?
What I aim to cover

- Backups
- File names
- Project and File structure
- Version control (Git)
- Package and Environment manager (Conda)
Not covered

- Containers
- Workflows and pipelines
- Markdown
- Jupyter
The technical bits!
Backup backup backup

- Get an off-site backup for your raw data as soon as it arrives
- Make sure metadata is backed up with the raw data
- Once initial QC is complete, submit raw data to a data repository (with embargo)
- Get frequent backups of scripts
- Backup intermediate results

```bash
rsync -Pa
```
Map your environment

- What are the backup options where you work? e.g. Cloud storage, NAS
- Who is responsible for the data storage and backup?
TMUX Terminal Multiplexer

On request...

- Split views in the same terminal window
- Reattach to a previous tmux session
TMUX

- tmux new -s ArcticMetagenome
- tmux ls
- tmux attach -t 0.

Basic commands: 

- \textit{ctrl-b} \% Split into two vertical panes
- \textit{ctrl-b} " Split into horizontal panes
- \textit{ctrl-b d} Detach from tmux session
File names

- Use extensions to guide you (.txt .csv .fastq)
- Name files so that it is easy to understand and describe where it comes from (AT1_R1_trimmed.fq)
- Avoid any label that implies order relative to other files (Final1.txt UltraFinal.txt This_is_my_Final_Final_version2.txt)
File names

**Protip:** Never look in someone else's documents folder.
My take on a strategy (but with support from literature)

- Totally fine if you have another strategy...
My take on a strategy (but with support from literature)

- Totally fine if you have another strategy... ... but remember that chaos does not count as a strategy!!
Project

- Good descriptive name of project, e.g. ArcticMetatranscriptome2023
- Include information about the goal and reasoning for the project *README*

- Data
- Analysis
- Docs
- Scripts
- Pogs
Data

Read-only, raw data and meta data

```
chmod -R Data
```

This is an exact **COPY** of the data at the start of the project

**Note:** Keep a backup at a separate location

Submit raw data to public repository early, with embargo
Docs

Put documentation (e.g. R markdown, Notes etc)
Scripts

Scripts, such as sbatch, bash, R scripts etc
Progs

Store software installed manually Keep a record of software & versions
Analysis

Make a separate folder for each of the steps in the analysis I like to number them to get a nice order:

1. raw_data is a symbolic link:

```bash
ln -s Data/SRRZ123447_R1.fastq sampleA_R1.fastq
```
Work reproducibly

- Ten simple rules for Reproducible Computational Research

  1. Track how results were produced
  2. Avoid manual data manipulation
  3. Archive/document all external software used. Versions!!
  4. Version control custom scripts
  5. Make it all available!
So you have a Project and File structure

Now what?
Version control

Git & Github

*What is Git?*
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

*What is GitHub?*
GitHub is a web page were git repositories can be shared. It is a essentially social platform for code. Good for most things that fit with Git.
Git is distributed

**Centralized version control system**

1. Server
2. Repository
3. Working copy
4. Workstation/PC #1
5. Workstation/PC #2
6. Workstation/PC #3

**Distributed version control system**

1. Server
2. Repository
3. Working copy
4. Workstation/PC #1
5. Workstation/PC #2
6. Workstation/PC #3
Basic git workflow
Shortlist of the most useful terms in git

status
stage (add)
commit
push
pull
clon
branch
Recommendations when committing to the repository

- Commit on a regular basis, ideally when one set of work has been performed and tested.
- Write short descriptive comments to each commit
Conda

Package and environment manager

- Install software with dependencies
- Avoid dependency issues
- Save the software versions and dependencies in a file
Conda commands

conda create -n project_A
conda env list.
conda activate project_A.
conda info --envs.
conda install -c bioconda sra-tools.

Save the environment software and dependencies to a file

conda env export > project_A_condaenv.yml
Other tools for reproducible science

- Workflows such as Snakemake & Nextflow
- Containers Docker & Singularity
Take home messages

Do not try to do all at once.

Start with file structure and backup.
then consider more advanced steps such at git and conda Set goals that are realistic
Summary

1. Create a new git repository for the project (e.g., GitHub)
2. Add a README file which should contain the required information on how to run the project
3. Create a Conda environment.yml file with the required dependencies
4. Create a R Markdown or Jupyter notebook to run your code
5. Alternatively, create a Snakefile to run your code as a workflow and use a config.yml file to add settings to the workflow
6. Use git to continuously commit changes to the repository
7. Possibly make a Docker or Singularity image for your project
Lab on Git and Conda

NBIS Data management & Reproducibility courses
Setup

git clone https://github.com/NBISweden/workshop-reproducible-research.git

Avoid creating a repo inside another repo
Git tutorial

```bash
git config --global init.defaultBranch "main"
```

Conda tutorial

To cover lab book Electronic Version control of scripts. Git & Github
Thanks

Look forward to talk to you about:

- Reproducible research
- Different career paths
- Work-Life balance
- Life in Sweden/UK/Greece