Rproject design and organization

BIGslu 2022-02-22

General principles by RStudio

- Different project designs for different project goals
- Flexibility so structure is applicable to diverse projects
- Structure is meant to help, not cause undue burden
- Rules are good but tools are better
- Structures should evolve and change

General principles by others

- Data should be separate from results
- Raw data should be separate from processed/clean data
- README or another form of project introduction is helpful
- .Rproj help you avoid hard file paths and promote reproducibility
- Version control is a bonus!

Kim’s Rproject design

Directory structure

- data_clean/
- data_raw/
- figs/
    - gene_level/
    - module_level/
- publication/
- results/
    - enrichment/
    - gene_level/
    - model_selection/
- scripts/
- .Rmd
- Knit .html / .pdf

.Rmd

- Data cleaning
- Model selection
- Statistics
- Enrichment…

.Rmd structure

- Load packages
- Load data
- Data cleaning
- Analysis
- Figures
- Summary

Pros

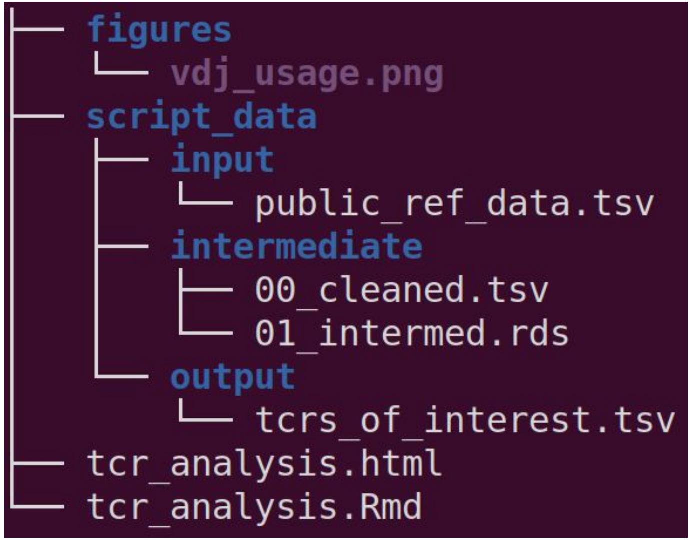
- Consistent structure across all projects
- Reproducible scripts to setup this structure
- Version control
- Rmd usage promotes commenting and interpretation right along with code

Cons

- Rmd can get very long; commenting not always so necessary
- Difficult to balance code visibility vs readability for diverse audience when Rmd is the main scripting file
- Rigid design not application to truly exploratory projects

Emma’s Rproject design

Directory structure in google drive:



```
├── figures
│   └── vdj_usage.png
├── script_data
│   ├── input
│   │   └── public_ref_data.tsv
│   ├── intermediate
│   │   ├── 00_cleaned.tsv
│   │   └── 01_intermed.rds
│   └── output
│       └── tcrs_of_interest.tsv
├── tcr_analysis.html
└── tcr_analysis.Rmd
```

Raw data is in central gdrive folder (hard paths)

Everything else local for faster I/O, then uploaded to gdrive project folder:

- Rmd from local github repo
- Folders from local working directory

Rmd structure:

- Intro
- Load libraries
- Load data
- Data cleaning and exploration
- Running tools and using their output
- Analysis + figures

Pros

- Single, centralized copy of raw data
- Fast reading/writing of files
- Version control of Rmd

Cons

- Raw data paths are unique to each user’s mounted gdrive
- Isn’t obvious what raw data is used without opening script (README?)
- Need to manually upload/re-upload project elements as they change
- Hard to use Rproj files b/c project directory not working directory

Elisabeth’s project design

All in the cloud. Cannot use RStudio.

- project/
  - ANALYSIS/
    - analysis_00/
      - 001/
      - 002/
      - 100/
      - 102/
    - analysis_01/
      - …
  - DATA
  - README_FIND

← Folders starting with the same number have related outputs

← README has in-depth descriptions of numbered directories

Take-homes

- Everything is one place is best when possible. Avoids file path issues and ensures versions within a project are consistent with each other.
  - However, large data may need to be stored elsewhere. Options include cloud services (AWS, GCP, Terra, OneDrive…) or in-house servers.
  - Consider processing data stored in the cloud with the cloud. Avoids time-intensive and expensive upload/download.
- Clear documentation is a must.
  - README for the overall project with directory structure explanations.
  - Commented code throughout.
- Custom functions and packages may be useful if you find yourself copying the same code across multiple projects.
  - The activation energy for a function is almost always worth it; R packages are a lot more work.
  - `source( )` to load a custom function from GitHub. Make sure it's the "raw" URL such as `source("https://raw.githubusercontent.com/kdillmcfarland/R_bioinformatic_scripts/master/RNAseq_rare_gene_filter.R")`
- Version control!
- Flexibility is needed and the process should help you, not be a burden.