Tools for Reproducible Research
Organizing projects; exploratory data analysis

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

biostat.wisc.edu/~kbroman
github.com/kbroman
@kwbroman
Course web: bit.ly/tools4rr

Organizing your stuff

```
Code/d3examples/
     /Others/
     /PyBroman/
     /Rbroman/
     /Rqtl/
     /Rqtlcharts/
Docs/Talks/
     /Meetings/
     /Others/
     /Papers/
     /Resume/
     /Reviews/
     /Travel/
Play/
Projects/AlanAttie/
         /BruceTempel/
         /Hassold_QTL/
         /Hassold_Age/
         /Payseur_Gough/
         /PhyloQTL/
         /Tar/
```

Organizing your projects

```
Projects/Hassold_QTL/

    Data/
    Notes/
    R/
    R/Figs/
    R/Cache/
    Rawdata/
    Refs/

    Makefile
    Readme.txt

    Python/convertGeno.py
    Python/convertPheno.py
    Python/combineData.py

    R/prepData.R
    R/analysis.R
    R/diagnostics.Rmd
    R/qtl_analysis.Rmd
```

Organizing a paper

```
Docs/Papers/PhyloQTL/

    Analysis/
    Data/
    Figs/
    Notes/
    R/
    SuppFigs/

    ReadMe.txt
    Makefile
    phyloqtl.tex
    phyloqtl.bib

    Submitted/
    Reviews/
    Revised/
    Final/
    Proofs/
```

Organizing a talk

```
Docs/Talks/SampleMixups/

    Figs/
    R/

    ReadMe.txt
    Makefile
    bmi2013.tex

    Old/
```

Basic principles

- Develop your own system
- Put everything in a common directory
- Be consistent
  - directory structure; names
- Separate raw from processed data
- Separate code from data
- It should be obvious what code created what files, and what the dependencies are.
- No hand-editing of data files
- Don't use spaces in file names
- Use relative paths, not absolute paths
  - `../blah` not `~/blah` or `/users/blah`

Your closest collaborator is you six months ago, but you don't reply to emails.

Organization takes time.

Painful bits

- Coming up with good names for things
  - Code as verbs; data as nouns
- Stages of data cleaning
- Going back and redoing stuff
- Clutter of old stuff that you no longer need
- Keeping track of the order of things
  - dependencies; what gave rise to what
- Long, messy Makefiles

Painful bits

- ▸ Coming up with good names for things
  - – Code as verbs; data as nouns

- ▸ Stages of data cleaning
- ▸ Going back and redoing stuff
- ▸ Clutter of old stuff that you no longer need
- ▸ Keeping track of the order of things
  - – dependencies; what gave rise to what

- ▸ Long, messy Makefiles

$\rightarrow$ Modularity

Problem: Variations across data files

- Different files (or parts of files!) may have different formats.
- Variables (or factor levels) may have different names in different files.
- The names of files may inconsistent.

- It's tempting to hand-edit the files. Don't!
- Create another meta-data file that explains what's what.

Tidy data

Read Hadley Wickham's paper on Tidy Data.

- ▸ Each variable forms a column.
- ▸ Each observation forms a row.
- ▸ Each type of observational unit forms a table.

Mouse	Treatment	Response
1	control	--
1	ttt	7.4
2	control	3.8
2	ttt	5.2
3	control	5.5
3	ttt	6.6

Problem: 80 million side projects

```
$ ls ~/Projects/Attie

AimeeNullSims/         Deuterium/             Ping/
AimeeResults/          ExtractData4Gary/      Ping2/
AnnotationFiles/       ForFirstPaper/         Ping3/
Brian/                 FromAimee/             Ping4/
Chr10adipose/          GoldStandard/          Play/
Chr6_extrageno/        HumanGWAS/             Proteomics/
Chr6hotspot/           Insulin/               R/
ChrisPlaisier/         Islet_2011-05/         RBM_PlasmaUrine/
Code4Aimee/            Lusis/                 R_adipose/
CompAnnot/             MappingProbes/         R_islet/
CondScans/             Microarrays/           Rawdata/
D2O_2012-02-14/        MultiProbes/           Scans/
D2O_Nrm_2012-02-29/    NewMap/                SimsRePower/
D2O_cellcycle/         Notes/                 Slco1a6/
D2Ocorr/               NullSims/              StudyLineupMethods/
Data4Aimee/            NullSims_2009-09-10/   eQTLPaper/
Data4Tram/             PepIns_2012-02-09/     transeQTL4Lude/
```

Saving intermediate results

R Markdown document with details of data cleaning.

- Within the `.Rmd` file, periodically `save` the state of things, for further exploratory analysis.
- Put those intermediate files (which might be large) in a common subdirectory.
- The subdirectory could be under separate version control.
- But you'll need to go in there and commit files.

Problem: Coordinating with collaborators

- Where to put data that multiple people will work with?
- Where to put intermediate/processed data?
- Where to indicate the code that created those processed data files?
- How to divvy up tasks and know who did what?

- Need to agree on directory structure and file naming conventions
- Consider symbolic links for shared data directories

```
ln -s /z/Proj/blah
ln -s /z/Proj/blah my_blah
```

Problem: Collaborators who don't use git

Problem: Collaborators who don't use git

Um...

Problem: Collaborators who don't use git

- ▶ Use git yourself
- ▶ Copy files to/from some shared space
  - – Ideally, in an automated way
- ▶ Commit their changes.

Exploratory data analysis

- ▶ what were trying to do?
- ▶ what you're thinking about?
- ▶ what did you observe?
- ▶ what did you conclude, and why?

Avoid

- ▶ "How did I create this plot?"
- ▶ "Why did I decide to omit those six samples?"
- ▶ "Where (on the web) did I find these data?"
- ▶ "What was that interesting gene?"

Basic principles

Step 1: slow down and document.

Step 2: have sympathy for your future self.

Step 3: have a system.

Capturing EDA

- ► copy-and-paste from an R file
- ► grab code from the `.Rhistory` file
- ► Write an informal R Markdown file
- ► Write code for use with the KnitR function `spin()`
  Comments like `#' This will become text`
  Chunk options like so: `#+ chunk_label, echo=FALSE`

A file to `spin()`

```
#' This is a simple example of an R file for use with spin().

#' We'll start by setting the seed for the RNG.
set.seed(53079239)

#' We'll first simulate some data with x ~ N(mu=10, sig=5) and
#' y = 2x + e, where e ~ N(mu=0, sig=2)
x <- rnorm(100, 10, 5)
y <- 2*x + rnorm(100, 0, 2)

#' Here's a scatterplot of the data.
plot(x, y, pch=21, bg="slateblue", las=1)
```