Organizing collaborative projects; capturing exploratory data analysis

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

kbroman.org
github.com/kbroman
@kwbroman

Course web: kbroman.org/AdvData

File organization and naming
are powerful weapons against chaos.

– Jenny Bryan

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.

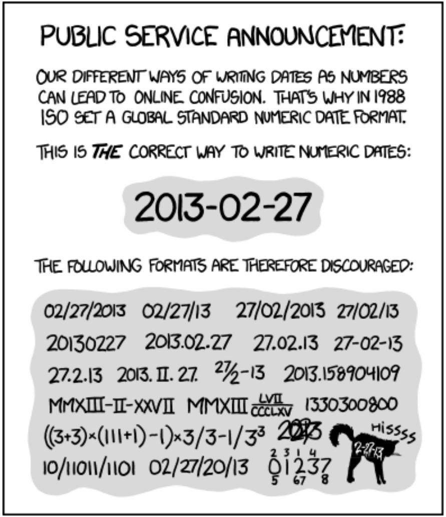
Painful bits

- 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

→ Modularity


PUBLIC SERVICE ANNOUNCEMENT:
OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.
THIS IS THE CORRECT WAY TO WRITE NUMERIC DATES:
2013-02-27
THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:
02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013. II. 27. 27/2-13 2013.158904109
MMXIII-II-XXVII MMXIII LVII/CCCLXV 1330300800
((3+3)×(111+1)−1)×3/3−1/3³ 2013
10/11011/1101 02/27/20/13 01237 2 3 1 4 5 67 8
HISSsss
2-27-13

xkcd.com/1179

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.

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.

Collaboration

- Do more, by working in parallel
- Do more, through diversity of ideas and skills
- Reproducible pipelines have immediate advantages
- Tests of reproducibility
- Code review

Genetics of metabolic disease in mice

Alan Attie, UW-Madison, Biochemistry

Karl Broman, UW-Madison, Biostat & Med Info

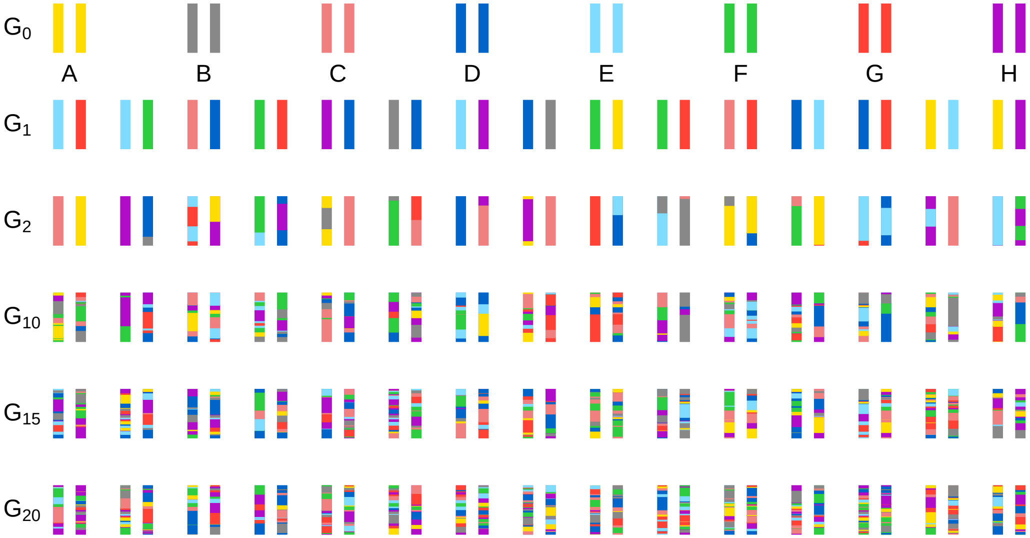
Gary Churchill, Jackson Lab

Josh Coon, UW-Madison, Chemistry

Federico Rey, UW-Madison, Microbiology

Brian Yandell, UW-Madison, Statistics

Diversity outbred mice



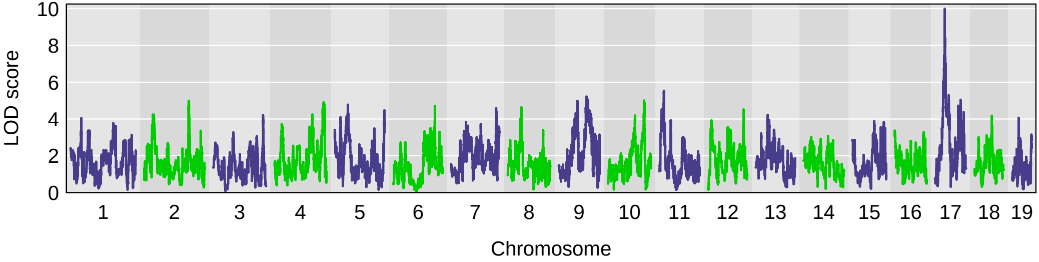
Data

- 500 DO mice
  - generations 17–23
  - high fat, high sugar diet
- GigaMUGA SNP arrays
  - 140k SNPs
- Clinical traits
  - Weekly body weight
  - Glucose tolerance test
  - Longitudinal serum samples
  - ex vivo islet insulin secretion

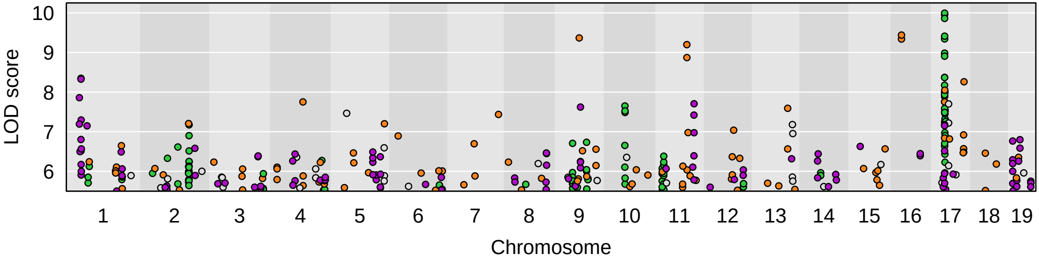
- Islet gene expression by RNA-seq
- Proteins by mass spec
- Lipids by mass spec
- Gut microbiome
  - 16S RNA
  - metagenomic data

Genome scans

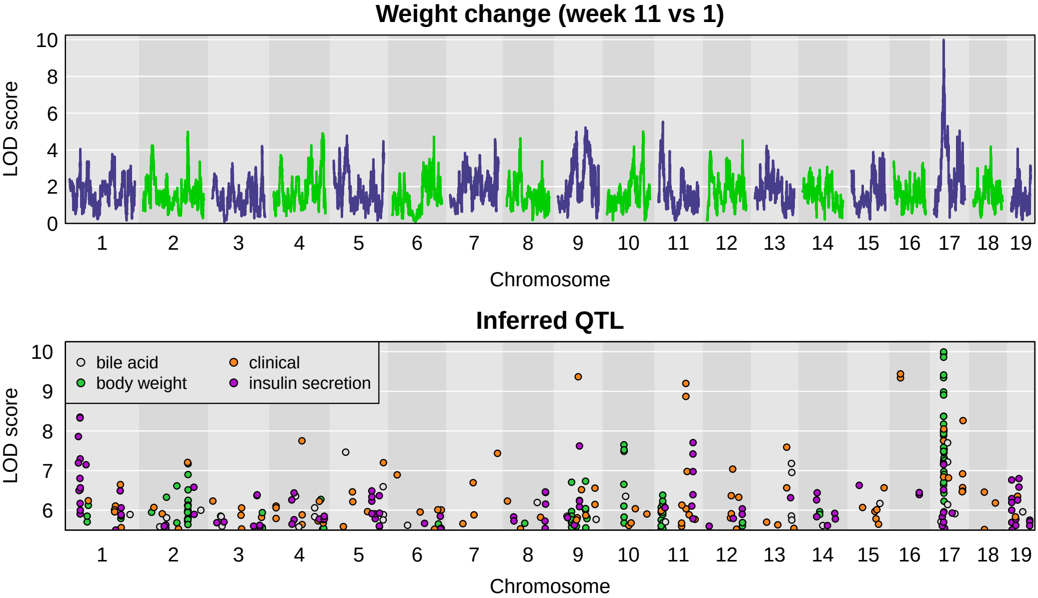
Weight change (week 11 vs 1)



Inferred QTL



Genome scans



Challenges in collaborations

- Shared vision?
- Compromise
- Coordination
- Communication
- Sharing code and data
- Synchronization

Challenges in collaborations

- Shared vision?
- Compromise
- Coordination
- Communication
- Sharing code and data
- Synchronization
- Weakest link?

Challenges

(totally hypothetical)

“Could we meet to talk about the data file structure?”

“Could we meet to talk about the data file structure?”

“No.”

“Wait, these results seem to be based on the older SNP map.”

“Could you write the methods section?”

“But I didn’t do the work,
and we don’t have the code that was used.”

“My data analyst has taken a job at Google.”

“Could you do these analyses? X said they would, but they’re not responding to my emails.”

Shared vision

- Publication
- Code & data sharing
- Who will do what
- Timeline
- Ongoing sharing of methods, results

Shared workspace

- Project structure
- Data and metadata formats
- Software environment
- Automated sync (or it won’t happen)

Technology for sharing

- Data
  - figshare
  - dropbox / box / google drive
- Code
  - github / bitbucket
- Pipeline / workflow
  - make / drake / snakemake / rake
- Full environment
  - docker containers
  - `mybinder.org` / `wholetale.org`

The most important tool is the mindset,
when starting, that the end product
will be reproducible.

– Keith Baggerly

Exploratory data analysis

- what were you 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)
```