Large-scale data analysis in R

Research Computing Center
and the Dept. of Human Genetics
University of Chicago

Peter Carbonetto
pcarbo@uchicago.edu

November 14, 2017

Getting started

Initial setup:

- Wifi
- Power outlets
- RCC cluster access
- Yubikeys (`rcc-guest-xxxx`)
- Helper(s)
- Handout / exercises
- Pace, questions (e.g., keyboard shortcuts)

Getting started

1. Download workshop packet.

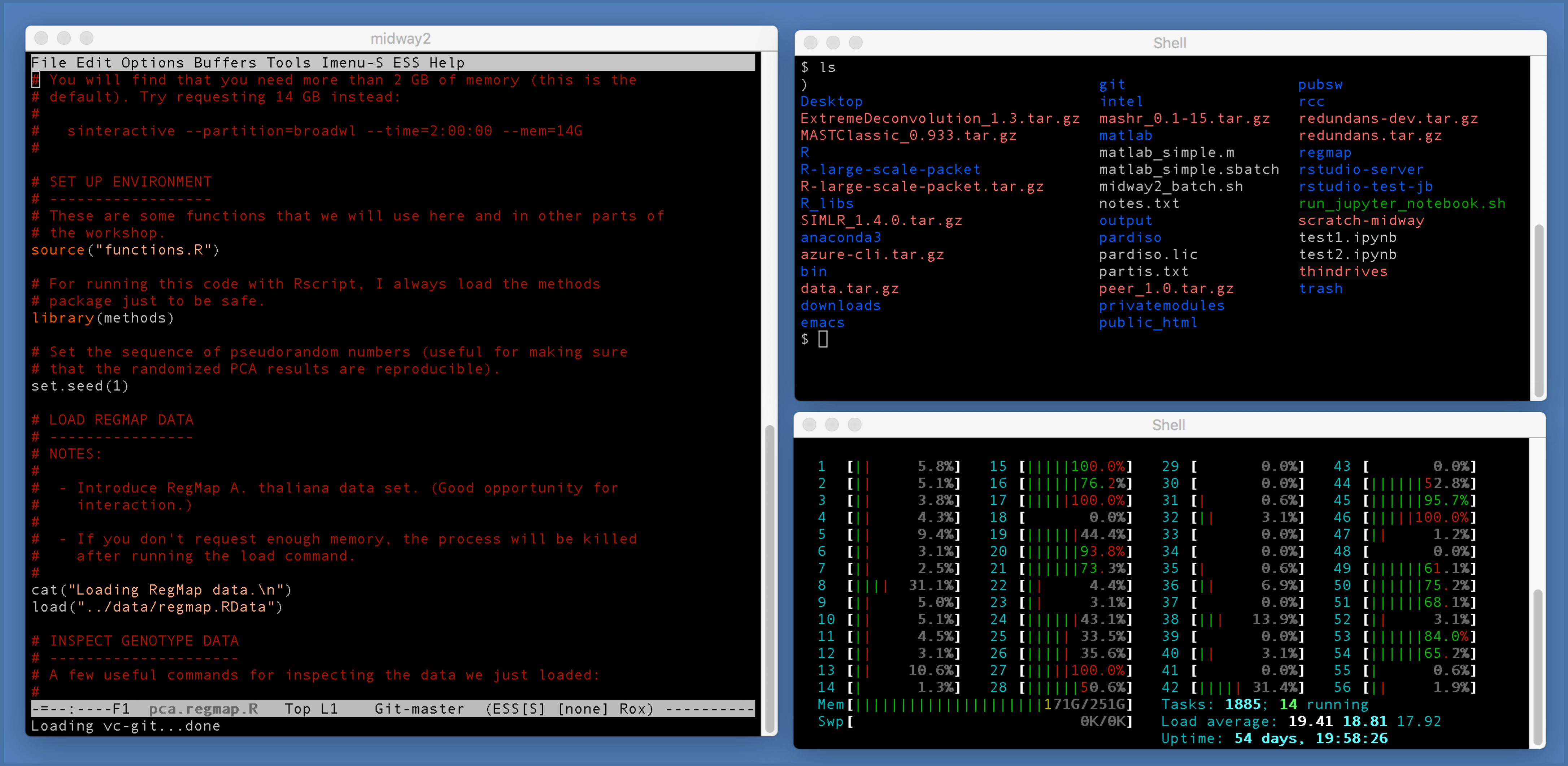
```
$ cd ~
$ wget https://tinyurl.com/yad63z9c \
    -O temp.tar.gz
$ tar zxvf temp.tar.gz
$ cd R-large-scale
$ cd code
```

2. Add these three lines to your `.bashrc` file.

```
# SLURM.
export SACCT_FORMAT="jobid,partition,user,account%12,alloccpus,node%12,elapsed,totalcpu,maxRSS,ReqM"
export SQUEUE_FORMAT="%13i %12j %10P %10u %12a %8T %9r %10l %.11L %5D %4C %8m %N"
```

Getting started

3. Start up three midway2 sessions.



```
# You will find that you need more than 2 GB of memory (this is the
# default). Try requesting 14 GB instead:
#
#   sinteractive --partition=broadwl --time=2:00:00 --mem=14G
#

# SET UP ENVIRONMENT
# -----------------
# These are some functions that we will use here and in other parts of
# the workshop.
source("functions.R")

# For running this code with Rscript, I always load the methods
# package just to be safe.
library(methods)

# Set the sequence of pseudorandom numbers (useful for making sure
# that the randomized PCA results are reproducible).
set.seed(1)

# LOAD REGMAP DATA
# ----------------
# NOTES:
#
#  - Introduce RegMap A. thaliana data set. (Good opportunity for
#    interaction.)
#
#  - If you don't request enough memory, the process will be killed
#    after running the load command.
#
cat("Loading RegMap data.\n")
load("../data/regmap.RData")

# INSPECT GENOTYPE DATA
# ---------------------
# A few useful commands for inspecting the data we just loaded:
#
-=--:----F1  pca.regmap.R    Top L1     Git-master   (ESS[S] [none] Rox) ----------
Loading vc-git...done
```

```
$ ls
)                                  git                    pubsw
Desktop                            intel                  rcc
ExtremeDeconvolution_1.3.tar.gz    mashr_0.1-15.tar.gz    redundans-dev.tar.gz
MASTClassic_0.933.tar.gz           matlab                 redundans.tar.gz
R                                  matlab_simple.m        regmap
R-large-scale-packet               matlab_simple.sbatch   rstudio-server
R-large-scale-packet.tar.gz        midway2_batch.sh       rstudio-test-jb
R_libs                             notes.txt              run_jupyter_notebook.sh
SIMLR_1.4.0.tar.gz                 output                 scratch-midway
anaconda3                          pardiso                test1.ipynb
azure-cli.tar.gz                   pardiso.lic            test2.ipynb
bin                                partis.txt             thindrives
data.tar.gz                        peer_1.0.tar.gz        trash
downloads                          privatemodules
emacs                              public_html
$ 
```

```
  1  [||       5.8%]   15 [|||||100.0%]   29 [         0.0%]   43 [         0.0%]
  2  [||       5.1%]   16 [||||||76.2%]   30 [         0.0%]   44 [||||||52.8%]
  3  [||       3.8%]   17 [|||||100.0%]   31 [|        0.6%]   45 [||||||95.7%]
  4  [||       4.3%]   18 [         0.0%]   32 [||       3.1%]   46 [|||||100.0%]
  5  [||       9.4%]   19 [||||||44.4%]   33 [         0.0%]   47 [||       1.2%]
  6  [||       3.1%]   20 [||||||93.8%]   34 [         0.0%]   48 [         0.0%]
  7  [||       2.5%]   21 [||||||73.3%]   35 [|        0.6%]   49 [||||||61.1%]
  8  [||||    31.1%]   22 [||       4.4%]   36 [||       6.9%]   50 [||||||75.2%]
  9  [||       5.0%]   23 [||       3.1%]   37 [         0.0%]   51 [||||||68.1%]
 10  [||       5.1%]   24 [||||||43.1%]   38 [|||     13.9%]   52 [||       3.1%]
 11  [||       4.5%]   25 [|||||  33.5%]   39 [         0.0%]   53 [||||||84.0%]
 12  [||       3.1%]   26 [|||||  35.6%]   40 [||       3.1%]   54 [||||||65.2%]
 13  [||      10.6%]   27 [|||||100.0%]   41 [         0.0%]   55 [|        0.6%]
 14  [|        1.3%]   28 [||||||50.6%]   42 [|||||   31.4%]   56 [||       1.9%]
 Mem[||||||||||||||||||||171G/251G]   Tasks: 1885; 14 running
 Swp[                      0K/0K]   Load average: 19.41 18.81 17.92
                                    Uptime: 54 days, 19:58:26
```

Getting started

4. Request a compute node.

```
$ screen -S rcc_workshop
$ sinteractive --partition=broadwl \
    --time=2:00:00 --account=rcc-guest \
    --reservation=rworkshop2
$ echo $HOSTNAME
```

5. Start up R programming environment.

```
$ module load R/3.4.1
$ R
R> getwd()
```

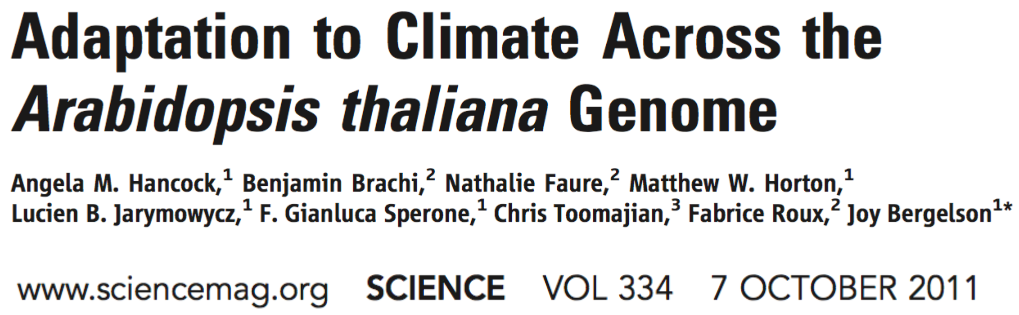
Aims of workshop

- ~~Do an analysis of a large-scale data set in R.~~
- Develop some useful skills for large-scale data analysis in R *within a high-performance computing environment*, and apply these skills to a medium-scale data set.

Outline

- Initial setup.
- Brief introduction.
- Part 1: Warm-up with PCA of RegMap data.
- Part 2: Implementing multithreaded computation in R for analysis of genetic adaptation to climate.
- Brief recap.

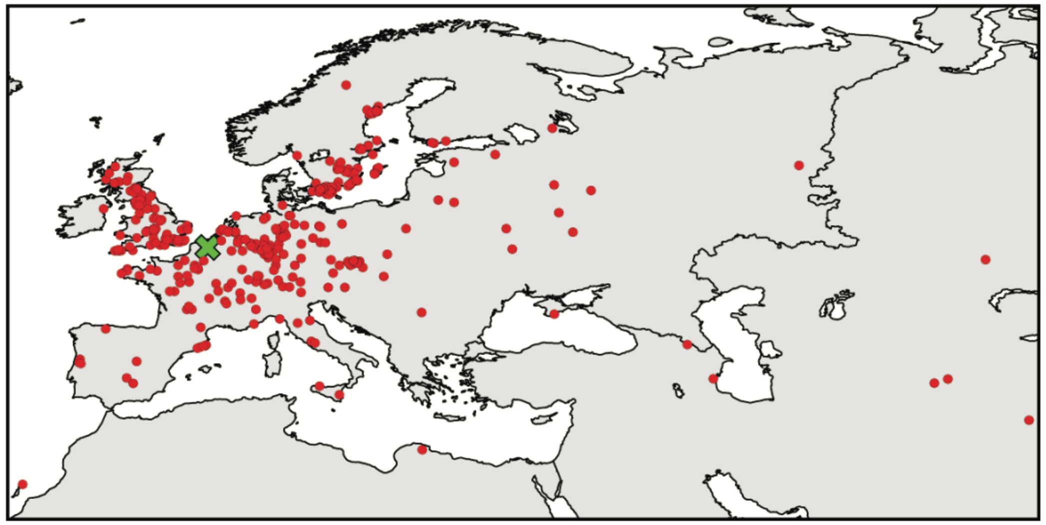
Part 1: PCA of the RegMap data



Adaptation to Climate Across the *Arabidopsis thaliana* Genome

Angela M. Hancock,[1] Benjamin Brachi,[2] Nathalie Faure,[2] Matthew W. Horton,[1] Lucien B. Jarymowycz,[1] F. Gianluca Sperone,[1] Chris Toomajian,[3] Fabrice Roux,[2] Joy Bergelson[1]*

www.sciencemag.org **SCIENCE** VOL 334 7 OCTOBER 2011



1. Inspect RegMap data.
2. Run PCA interactively and assess time & memory needs.
3. Examine PCA results.
4. Run PCA using Rscript.
5. Run PCA using SLURM job engine, and inspect output.

Exercise: Run PCA using SLURM job engine, and inspect output.

```
sbatch pca.sbatch
```

- Useful commands while job is running:

```
squeue --user=<cneitd> | less -S
ssh midway2-xxxx
```

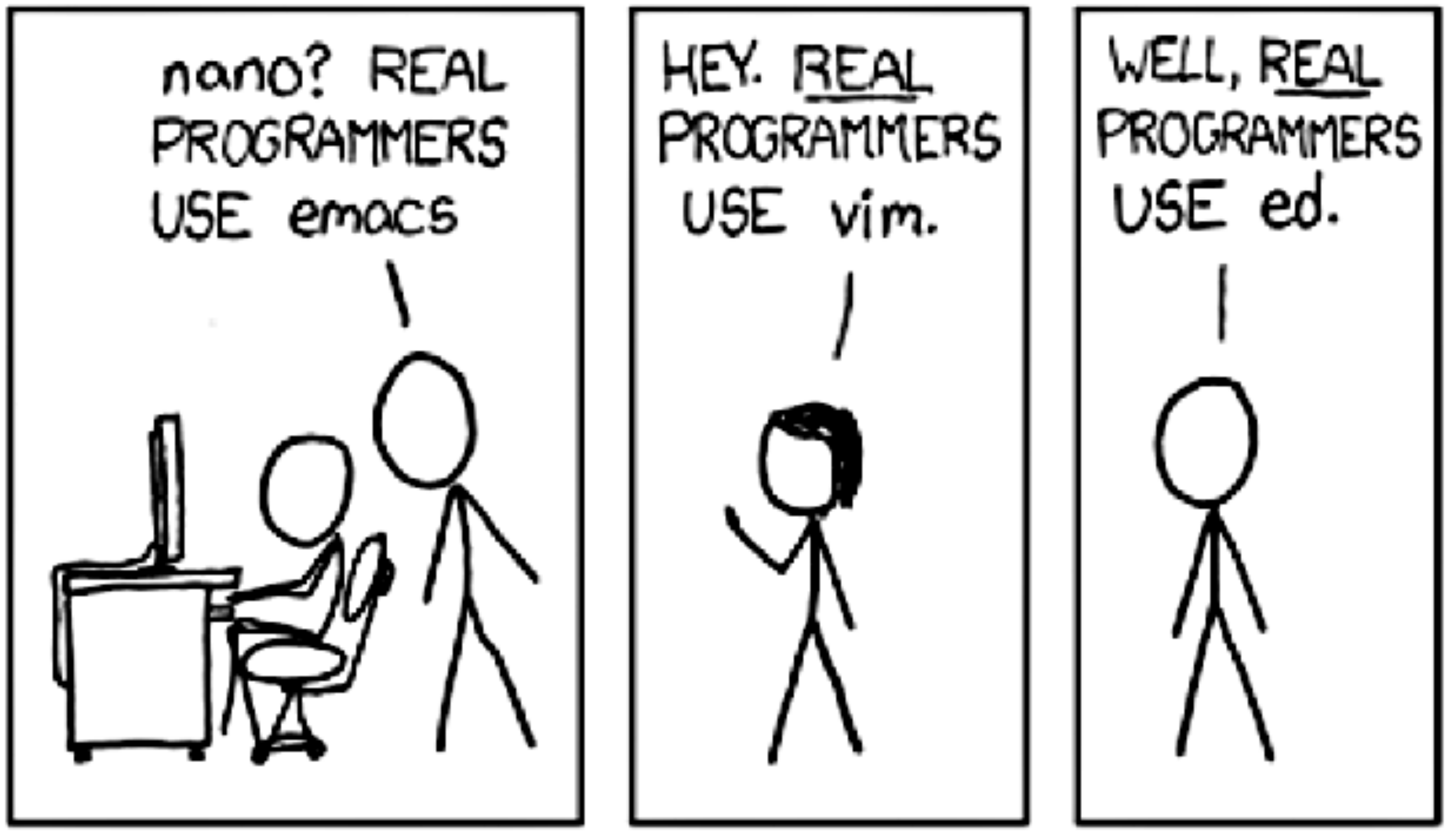
- Useful commands after job has completed:

```
htop --user=<cnetid>
sacct --user=<cnetid> --units=G | less -S
less ../output/pca_err.txt
less ../output/pca_out.txt
```

Part 2: Genetic analysis of climate variables in RegMap data

1. Inspect climate ("phenotype") data.
2. Run analysis interactively and assess time & memory needs.
3. Experiment with multithreaded matrix operations to improve computation time.
4. Experiment with parallel computation of the weights using `mclapply`.
5. Automate `climate.R` analysis using command-line arguments.
6. Run analysis using SLURM job engine, and inspect output.
7. Use SLURM job engine to automate analysis of all 48 climate variables.

There is no best tool—use *whatever works for you*.


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.

https://xkcd.com/378/

Some general advice

1. `help(package = cool_package)`.
2. Use **midway2**, not midway1.
3. Email help@rcc.uchicago.edu R help on the RCC cluster.
4. Learn to avoid loops as much as possible; e.g., use `apply()`, `lapply()`, `tapply()`, `do.call()`.
5. Document your setup—start with `sessionInfo()`.

After the workshop

- Feedback.
- Suggestions for future R workshop topics (e.g., Rcpp).