

# Steps toward reproducible research

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

[kbroman.org](http://kbroman.org)

[github.com/kbroman](https://github.com/kbroman)

@kbroman

Slides: [bit.ly/Chi2016](http://bit.ly/Chi2016)



Karl -- this is very interesting,  
however you used an old version of  
the data (n=143 rather than n=226).

I'm really sorry you did all that  
work on the incomplete dataset.

Bruce

The results in Table 1 don't seem to correspond to those in Figure 2.

In what order do I run these scripts?

Where did we get this data file?

Why did I omit those samples?

How did I make that figure?

"Your script is now giving an error."

”The attached is similar to the code we used.”

Reproducible

Reproducible

vs.

Replicable

Reproducible

vs.

Correct

# Levels of quality

- ▶ Are the tables and figures reproducible from the code and data?
- ▶ Does the code actually do what you think it does?
- ▶ In addition to **what** was done, is it clear **why** it was done?  
(e.g., why did you omit those six subjects?)
- ▶ Can the code be used for other data?
- ▶ Can you extend the code to do other things?

# Steps toward reproducible research

[kbroman.org/steps2rr](http://kbroman.org/steps2rr)

# 1. Everything with a script

If you do something once,  
you'll do it 1000 times.

## 2. Organize your data & code

File organization and naming  
are powerful weapons against chaos.

– Jenny Bryan

## 2. Organize your data & code

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

(paraphrasing [Mark Holder](#))

## 2. Organize your data & code

```
RawData/           Notes/  
DerivedData/      Refs/  
  
Python/           ReadMe.txt  
R/                ToDo.txt  
Ruby/             Makefile
```

### 3. Automate the process (GNU Make)

```
R/analysis.html: R/analysis.Rmd Data/cleandata.csv
  cd R;R -e "rmarkdown::render('analysis.Rmd')"

Data/cleandata.csv: R/prepData.R RawData/rawdata.csv
  cd R;R CMD BATCH prepData.R

RawData/rawdata.csv: Python/xls2csv.py RawData/rawdata.xls
  Python/xls2csv.py RawData/rawdata.xls > RawData/rawdata.csv
```

## 4. Turn scripts into reproducible reports

### Gough project diagnostics

Karl Broman, 3 March 2014

#### Combine genotypes and phenotypes

I've combined the initial genotypes (using the re-clustered genotypes for plates 14-16) with the well-behaved portion of the re-run genotypes. I'm focusing on 36813 markers that are informative (though, as we'll see, there are still a lot of badly behaved and basically non-informative markers that need to be removed). I've combined data on replicate samples, to give one set of genotype calls for each sample.

There are 1497 genotyped mice and 1464 phenotyped mice. All of the mice in the phenotype data have genotypes, but there are 33 genotyped mice with no phenotypes, including 3 Gough mice and 30 F2 progeny.

## 4. Turn scripts into reproducible reports

### Gough project diagnostics

Karl Broman, 3 March 2014

Comb

I've comb  
the well-  
informat  
informat  
give one

There are  
data have  
mice and

```
25 I've combined the initial genotypes (using the re-clustered genotypes
26 for plates 14-16) with the well-behaved portion of the re-run
27 genotypes. I'm focusing on `r totmar(g)` markers that are informative
28 (though, as we'll see, there are still a lot of badly behaved and
29 basically non-informative markers that need to be removed).
30 I've combined data on replicate samples, to give one set of genotype
31 calls for each sample.
32
33 There are `r nind(g)` genotyped mice and `r nrow(phe)` phenotyped
34 mice. All of the mice in the phenotype data have genotypes, but there
35 are `r sum(is.na(match(gid, pid)))` genotyped mice with no phenotypes,
36 including `r sum(g$pheno$gen[which(is.na(match(gid, pid)))]==0)`
37 Gough mice and `r sum(g$pheno$gen[which(is.na(match(gid, pid)))]==2)`
38 F2 progeny.
```

## 5. Turn repeated code into functions

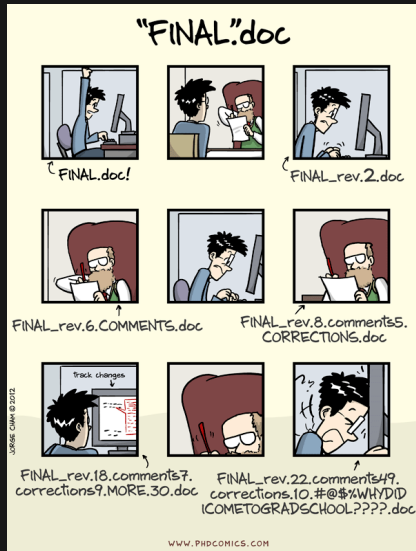
```
# Python
def read_genotypes (filename):
    "Read matrix of genotype data"
```

```
# R
plot_genotypes <-
function(genotypes, ...)
{
}
```


## 6. Create a package/module

Don't repeat yourself

# 7. Use version control (git/GitHub)




# 7. Use version control (git/GitHub)



PUBLIC  kbroman / Talk\_MAGIC Unwatch 1 Star 0 Fork 0








Talk for MAGIC workshop in Cambridge, UK, 12 June 2013 — Edit


97 commits | 1 branch | 0 releases | 1 contributor

branch: master - Talk\_MAGIC / 

Greatly simplify the public domain stuff in the ReadMe

 kbroman authored 15 days ago latest commit f1777ef192 

 Figs	Add crazy table from preCC paper	4 months ago
 Perl	Add lines_of_code_by_version.csv to repository	4 months ago
 R	Another fix regarding map expansion in 8-way RIL by setting at k=0	4 months ago
 g@ignore	Add lines_of_code_by_version.csv to repository	4 months ago
 Makefile	Revise README to link to version for web	4 months ago
 ReadMe.md	Greatly simplify the public domain stuff in the ReadMe	15 days ago
 magic.tex	Fix two slight bugs in slides:	4 months ago

 ReadMe.md


## Talk for **MAGIC Workshop** in Cambridge, UK

These are slides for a talk I will give at the [Workshop on MAGIC-type populations](#) in Cambridge, UK, on 12 June 2013.

The PDF is [here](#).

---

To the extent possible under law, [Karl Broman](#) has waived all copyright and related or neighboring rights to "MAGIC design and other topics". This work is published from: United States.



# 7. Use version control (git/GitHub)

The screenshot shows the GitHub interface for the repository 'kbroman / Talk\_MAGIC'. At the top, it indicates the repository is 'PUBLIC'. There are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). The repository description is 'Talk for MAGIC workshop in Cambridge, UK, 12 June 2013 — Edit'. Below this, there are statistics: 97 commits, 1 branch, 0 releases, and 1 contributor. On the right side, there are links for 'Code', 'Issues', and 'Pull requests'.

## Greatly simplify the public domain stuff in the ReadMe



**kbroman** authored 15 days ago

latest commit [f1777ef192](#)

<a href="#">Figs</a>	Add crazy table from preCC paper	4 months ago
<a href="#">Perl</a>	Add lines_of_code_by_version.csv to repository	4 months ago
<a href="#">R</a>	Another fix regarding map expansion in 8-way RIL by selfing at k=0	4 months ago
<a href="#">.gitignore</a>	Add lines_of_code_by_version.csv to repository	4 months ago
<a href="#">Makefile</a>	Revise Readme to link to version for web	4 months ago
<a href="#">ReadMe.md</a>	Greatly simplify the public domain stuff in the ReadMe	15 days ago
<a href="#">magic.tex</a>	Fix two slight bugs in slides:	4 months ago

rights to "MAGIC design and other topics". This work is published from: United States.



# 7. Use version control (git/GitHub)

The screenshot shows a GitHub repository page for 'kbroman / Talk\_MAGIC'. At the top, it indicates the repository is 'PUBLIC' and shows 'Unwatch 1', 'Star 0', and 'Fork 0'. Below this, the current branch is 'master' and the view is 'Talk\_MAGIC / Commits'. The commits are grouped by date:

- Sep 27, 2013**
  - Greatly simplify the public domain stuff in the ReadMe** (commit: f1777ef192) - kbroman authored 15 days ago. [Browse code]
  - Fix url in ReadMe.md file** (commit: a6515823f9) - kbroman authored 15 days ago. [Browse code]
- Jun 17, 2013**
  - Another fix regarding map expansion in 8-way RIL by selfing at k=0** (commit: 208a482f2c) - kbroman authored 4 months ago. [Browse code]
  - Fix two slight bugs in slides:** (commit: 51d4aa9ceb) - kbroman authored 4 months ago. [Browse code]
    - 8-way RIL by selfing: map expansion = 1 at k=0
    - Slight repair to definition of 3-pt coincidence
- Jun 10, 2013**
  - Change one page number** (commit: e8e0688615) - kbroman authored 4 months ago. [Browse code]
  - Add missing paren** (commit: f4975dee6e) - kbroman authored 4 months ago. [Browse code]
  - who's -> who is** (commit: 886f20f098) - kbroman authored 4 months ago. [Browse code]
  - rublish -> bad** (commit: e6fbf2f647) - kbroman authored 4 months ago. [Browse code]
  - Add link to R/qtl page** (commit: 4edf3e8676) - kbroman authored 4 months ago. [Browse code]
  - Revise slide re analysis issues** (commit: 14ebb1eeb5) - kbroman authored 4 months ago. [Browse code]
  - italicize 'de novo'** (commit: 45dda4b4c7) - kbroman authored 4 months ago. [Browse code]
  - replace plain right arrow with fat arrow** (commit: 8bbe385d6c) - kbroman authored 4 months ago. [Browse code]

# 7. Use version control (git/GitHub)

PUBLIC kbroman / Talk\_MAGIC Unwatch 1 Star 0 Fork 0

**Fix two slight bugs in slides:** [Browse code](#)

- 8-way RIL by selfing: map expansion = 1 at k=0
- Slight repair to definition of 3-pt coincidence

master

kbroman authored 4 months ago 1 parent e0e8608 commit 51d4aa9ceb104bbf26e8cbe185a5c7f8dc02a832

Showing 2 changed files with 5 additions and 3 deletions. [Show Diff Stats](#)

**6** R/map\_expansion\_func.R [View file @ 51d4aa9](#)

```
... .. @@ -25,8 +25,10 @@ mesibA4 <- function(k)
25 25 #####
26 26 # Eight-way
27 27 #####
28 -mesif8 <- function(k)
29 - 4 - (((1)/(2)))^(k-2)
+mesif8 <- function(k) {
29 + if(k==0) return(1)
30 + 4 - (((1)/(2)))^(k-2)
31 +}
30 32
31 33 mesibX8 <- function(k)
32 34 ((14)/(3)) - (((30 + 14*sqrt(5))/(15))) * (((1+sqrt(5))/(4)))^k - (((30 - 14*sqrt(5))/(15))) * (((1-s
```

**2** magic.tex [View file @ 51d4aa9](#)

```
... .. @@ -636,7 +636,7 @@
636 636
637 637 \hspace{20mm} {\color{myblue} = $\mathsf{Pr}(\text{rec'n in 23})$ |
638 638 \ \text{rec'n in 12})$ /
639 - Pr(\text{rec'n in 12})}$)
639 + Pr(\text{rec'n in 23})}$)
640 640
641 641 \item
642 642 No interference { \color{myblue} = 1 }
```

## 8. License your software

Pick a license, any license

– Jeff Atwood

# Other considerations

- ▶ **Testing**

are you getting the right answers?

- ▶ **Software versions**

will your stuff work when dependencies change?

- ▶ **Large-scale computations**

computation time + dependence on cluster environment

- ▶ **Collaborations**

coordinating who does what and where things live

- ▶ **Distribution**

where and how to distribute data and code?

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

– Keith Baggerly

# Summary

1. Everything with a script
2. Organize your data & code
3. Automate the process (GNU Make)
4. Turn scripts into reproducible reports
5. Turn repeated code into functions
6. Create a package/module
7. Use version control (git/GitHub)
8. Pick a license, any license

Slides: [bit.ly/Chi2016](http://bit.ly/Chi2016)



[kbroman.org](http://kbroman.org)

[github.com/kbroman](https://github.com/kbroman)

@kwbroman