Literate Programming and Reproducible Research

L. Gatto and S. J. Eglen

October 24, 2012

Outline

A simple system

A first step to reproduce (as in trace, understand and repeat) a piece of analysis is to be able to trace what has been done to obtain a results.

- `S00-environment.R` load packages and defines global variables (colours, ...).
- `S01-functions.R` stores project specific functions.
- `S02-loadData.R` manages all the data input.
- `S03-analyse1.R` a first batch of analyses.
- `S04-analyse2.R` another batch of analyses.
- Figures are saves as `F01-firstFig.pdf`, ...
- Data is saved/exported as `D01-data.csv`, `D01-result.rda`, ...
- Possibly in their own directories.

Works for simple analyses, but gets quickly messy.

See other's advices

- http://www.biostars.org/post/show/821/how-do-you-manage-your-files-directories-for-your-projects/
- http://stats.stackexchange.com/questions/2910/how-to-efficiently-manage-a-statistical-analysis-project
- http://stackoverflow.com/questions/1429907/workflow-for-statistical-analysis-and-report-writing

Even better

Use specific frameworks to support the code and file management

- ProjectTemplate: http://projecttemplate.net/

`R` itself provides a solution

- Build your project package, including documented code, data, vignette, tests, ...

Outline

Literate Programming

From the web page describing his book *Literate Programming*, Donald E Knuth writes:
"Literate programming is a methodology that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language. The main idea is to treat a program as a piece of literature, addressed to human beings rather than to a computer. The program is also viewed as a hypertext document, rather like the World Wide Web. (Indeed, I used the word WEB for this purpose long before CERN grabbed it!) ..."

http://www-cs-faculty.stanford.edu/~uno/lp.html

Tangling and Weaving:

- CWEB: system for documenting C, C++, Java:

```
CTANGLE
    converts a source file foo.w to a compilable program file
CWEAVE
    converts a source file foo.w to a prettily-printable and
    cross-indexed document file foo.tex.
```

http://sunburn.stanford.edu/~knuth/cweb.html

In R , you would use Stangle and Sweave.

What is Reproducible Research (RR)?

- Gentleman et al (2004)[1] advocate RR:

  Buckheit and Donoho (35) , referring to the work and philosophy of Claerbout, state the following principle: "An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and that complete set of instructions that generated the figures."

- Bioconductor packages are good examples of reproducible research.
- This article is also good background reader for open software development.
- IMHO, Bioconductor has had a positive impact on genomic data analysis, ranging far outside of the CBB area.

[1] http://genomebiology.com/2004/5/10/R80

The case of the Duke cancer trials

- Technical details (37 mins, Cambridge 2010) http://videolectures.net/cancerbioinformatics2010_baggerly_irrh/
- Wide audience, but rather narrow-sighted: 13-minute video from 60 minutes: http://www.cbsnews.com/video/watch/?id=7398476n

Approaches to RR

1. Makefiles
2. Sweave
3. Others

Outline

Make and Makefiles

- Make is an automated build system, designed to avoid costly recomputation.
- `make` examines a **Makefile**, which contains a set of rules describing dependencies among files.
- A rule is run (i.e the recipes are executed) if the **target** is older than any of its **dependencies** (**prerequisites**).

```
target: prerequisites ...
    recipe
    ...
```

- `make` works backwards from the target to the prerequisites and compares creation time of files (`timestamp`).

Make and Makefile

- Example:

```
res.txt: param1.dat param2.dat
	simulation param1.dat param2.dat > res1.dat
	post-process res1.dat > res.txt
```

- Commands to be run should be indented with a TAB.

A complete Makefile – file:rr_make/

```
report.pdf: report.tex sim1.pdf sim2.pdf
	texi2pdf report.tex

sim1.dat: params.R simulator.R
	Rscript simulator.R rnorm > sim1.dat

sim2.dat: params.R simulator.R
	Rscript simulator.R runif > sim2.dat

sim1.pdf: sim1.dat plotter.R
	Rscript plotter.R sim1.dat

sim2.pdf: sim2.dat plotter.R
	Rscript plotter.R sim2.dat


.PHONY: all clean

all: report.pdf

clean:
	rm -f report.pdf report.log report.aux
	rm -f sim1.* sim2*
```

Graphical description of dependencies

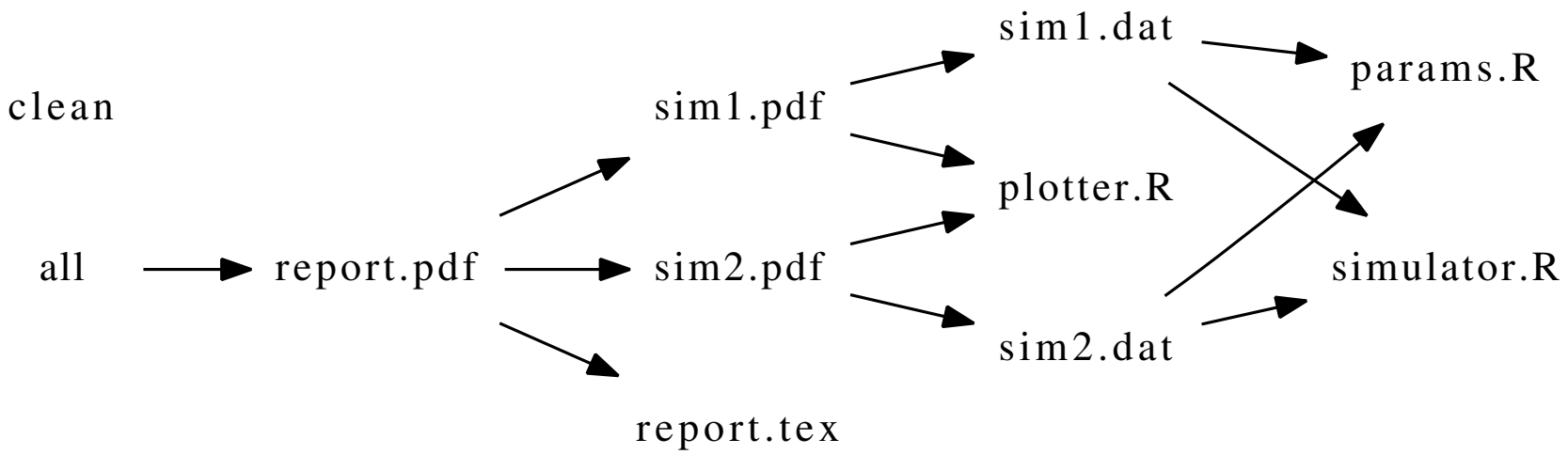

Figure: Makefile dependency graph

Makefile conventions

- PHONY targets: denote actions; ignore filenames with same name. PHONY targets are always out of date, and so always run.

```
.PHONY: all clean
all: report.pdf

clean:
        rm -f report.pdf report.log report.aux
        rm -f sim1.* sim2*
```

command	action
make	check first rule
make all	rebuild everything
make clean	remove files that can be rebuilt
touch file	update timestamp, preserving contents

Makefile: next steps

- variables
- implicit rules
- saving space:

```
sim2.dat: params.R simulator.R
	Rscript simulator.R runif > sim2.dat

sim2.dat: params.R simulation.R
	Rscript simulator.R runif > $@
```

- parallel processing `make -j2 job`

Makefile references

- Further reading:

http://linuxdevcenter.com/pub/a/linux/2002/01/31/make_intro.html

- Managing Projects with GNU Make

http://oreilly.com/catalog/make3/book/index.csp

- The GNU make manual

http://www.gnu.org/software/make/manual/make.html

Makefile: example lab work

- In the lab session, download `rr_make.zip`
- Experiment with remaking report after changing parameters.
- Add a new plot to the report, using sim3 – sampling N numbers from rgamma with new parameters (stored in params.R). You will need to edit simulator.R too.

Outline

Sweave: literate programming for R

- Sweave is the system for mixing LaTeXand R code in the same document.
- Used within R often to create "vignettes" which can be dynamically run.
- Allows you to write reports where results (tables, graphs) are automatically generated by your R code.

Sweave: including code chunks

- An example code chunk: by default we are in 'LaTeX mode'.

```
We can then test the procedure a few
times, using the default number
of darts, 1000:

<<>>=
replicate(9, estimate.pi())
@
```

Sweave: including graphs

- Automatically creates filenames, e.g. `estimate-001.pdf`
- By default will generate .ps and .pdf; so change options:

```
\SweaveOpts{echo=TRUE,pdf=TRUE,eps=FALSE,eval=TRUE,keep.source

\setkeys{Gin}{width=0.6\textwidth}
\begin{center}
<<fig=TRUE>>=
r <- 1; n <- 50; par(las=1)
plot(NA, xlim=c(-r,r), ylim=c(-r,r), asp=1, bty='n',
     xaxt='n', yaxt='n', xlab='', ylab='')
axis(1, at=c(-r,0,r)); axis(2, at=c(-r,0,r))
symbols(x=0, y=0, circles=r, inch=F, add=T)
...
rect(-r, -r, r, r, border='blue', lwd=2)
@
\end{center}
```

Sweave: including tables

- Use the *xtable* package from CRAN.
- Example from that package:

```
<<echo=FALSE>>=
library(xtable)
data(tli)
@

<<label=tab1,echo=FALSE,results=tex>>=
     ## Demonstrate data.frame
     tli.table <- xtable(tli[1:20,])
     digits(tli.table)[c(2,6)] <- 0
     print(tli.table)
@
```

Sweave: including inline computation

```
In this case the number of darts within
the circle is \Sexpr{d}, and so the estimated
value is $\pi \approx \Sexpr{4*d/n}$.
```

Sweave: a full example

- Example application: estimate the value of π using the dartboard method.
- estimate.Rnw
- See handout of estimate.Rnw and estimate.pdf
- For nice ways to customize Sweave output

http://proteome.sysbiol.cam.ac.uk/lgatto/teaching/files/Sweave-customisation.pdf

- Compiling the document with make:

```
estimate.pdf: estimate.Rnw
        R CMD Sweave estimate.Rnw
        pdflatex estimate.tex
```

Sweave: issues and next steps.

- If you edit `.tex`, Sweave code is re-run. Compare with Makefiles, which offer finer-level control.
- Tedious to keep running with long calculations. `cacheSweave` package will help to cache results.
- FAQ available:

`http://www.stat.uni-muenchen.de/~leisch/Sweave/FAQ.htm`

- *odfWeave* and *RHTML* packages allow for output to OpenOffice and HTML.
- matrices and data frames can be export, e.g. using `xtable` package.

Outline

Other approaches to RR

- `R` packages: truly reproducible research. `R` packages allow you to include code, data, documentation, vignettes.
- The `R` package `ascii`[2] allows you to embed `R` code into your documents.
- **Org mode**[3] and **Org babel**[4]: Only Emacs users need apply. Key advantage: allows many different languages to be included in one document, with textual communication between those programs. Org mode exports to multiple formats.
  (show source of these slides)
- `knitr` is an alternative to `Sweave`, that uses caching, syntax highlighting, code tidy-up, ... by default. Also weaves to html. Good integration with `rstudio`.

---

[2] http://eusebe.github.com/ascii
[3] http://orgmode.org/
[4] http://orgmode.org/worg/org-contrib/babel/

Extra handouts

1. Makefile: report.pdf
2. Sweave: `estimate.Rnw` and `estimate.pdf`
3. Using `kntir`: `estimatek.Rnw` and `estimatek.pdf`

Available at

- http://proteome.sysbiol.cam.ac.uk/lgatto/teaching/files/estimate.zip
- http://proteome.sysbiol.cam.ac.uk/lgatto/teaching/files/rr_make.zip

and `pwf:~/COMPBIO/spr/practicals/`