Reproducible Research.. Using Sweave, Knitr and Pandoc

Aedín Culhane

aedin@jimmy.harvard.edu

Nov 20th 2012

My R Course Website http://bcb.dfci.harvard.edu/~aedin/

My HSPH homepage http://www.hsph.harvard.edu/research/aedin-culhane/

When issues of reproducibility arise

- ``Remember that microarray analysis you did six months ago? We ran a few more arrays. Can you add them to the project and repeat the same analysis?''
- ``The statistical analyst who looked at the data I generated previously is no longer available. Can you get someone else to analyze my new data set using the same methods (and thus producing a report I can expect to understand)?''
- ``Please write/edit the methods sections for the abstract/paper/grant proposal I am submitting based on the analysis you did several months ago.''

From Keith Baggerly

EDITORIAL

nature genetics

Mostly, your results matter to others

High-thr
by journ
data use

Invest
stanc
Edito
when aut
even witl

ANALYSIS

nature genetics

Repeatability of published microarray gene expression analyses

John P A Ioannidis[1–3], David B Allison[4], Catherine A Ball[5], Issa Coulibaly[4], Xiangqin Cui[4], Aedín C Culhane[6,7], Mario Falchi[8,9], Cesare Furlanello[10], Laurence Game[11], Giuseppe Jurman[10], Jon Mangion[11], Tapan Mehta[4], Michael Nitzberg[5], Grier P Page[4,12], Enrico Petretto[11,13] & Vera van Noort[14]

NATURE GENETICS | VOLUME 41 | NUMBER 2 | FEBRUARY 2009

Repeatability of published microarray gene expression analyses

- Selected articles published in *Nature Genetics between* January 2005 and December 2006 that had used profiling with microarrays
- Of the 56 items retrieved electronically, 20 articles were considered potentially eligible for the project
- The four teams were from
  - University of Alabama at Birmingham (UAB)
  - Stanford/Dana-Farber (SD)
  - London (L) and Ioannina/Trento (IT)
- Each team was comprised of 3-6 scientists who worked together to evaluate each article.

Results

- Result could be reproduced n=2
- Reproduced with discrepancy n=6
- Could not be reproduced n=10
  - No data n=4 (no data n=2, subset n=1, no reporter data n=1)
  - Confusion over matching of data to analysis (n=2)
  - Specialized software required and not available (n=1)m
  - Raw data available but could not be processed n=2

Reproducibility of Analysis

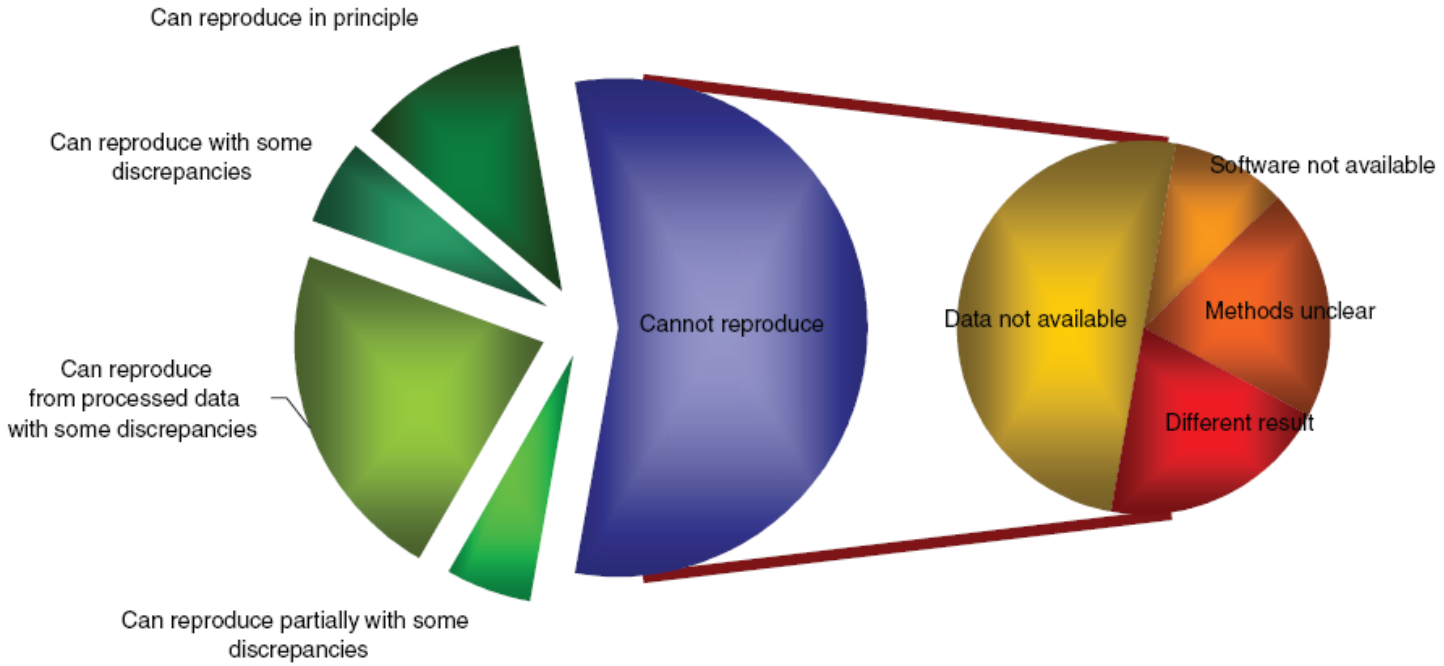


Figure 1 Summary of the efforts to replicate the published analyses.

Ioannidis JP, Allison DB, Ball CA, Coulibaly I, Cui X, Culhane AC, et al, (2009) **Repeatability of published microarray gene expression Analyses**. *Nature Genetics* 41(2):149

Reproducible Research in R

- Sweave
- Knitr
- Knitr + pandoc

Typical LaTeX

```
\documentclass{article}
\usepackage{times}

\begin{document}

% Article top matter
\title{How to Structure a \LaTeX{} Document}
```

Blah blah blah blah…..

```
\end{document}   %End of document.
```

http://en.wikibooks.org/wiki/LaTeX/simple.tex

Sweave

- R embedded in Latex
- Produce pdf or html files
- R code is run each time, so you are sure the code works
- Document includes results of the code

```
Sweave (filename.rnw)
Stangle (filename.rnw)
```

Quick Start to Sweave

- Insert an R code chunk starting with << >>=
- Terminate the R code chunk with an @ sign

```
<<easySweave>>=
x <- mean(1:10)
print(x)
@
```

- Save LaTeX with extension ``Rnw''

Embedding code in text

- To embed a simple R calculation within a document \Sexpr

```
The sum is \Sexpr{1+2}

\Sexpr{paste(“result is”, 2^x)}
```

Sweave works in a html document

Create a basic html document and process with Sweave

```
Sweave("filename.rnw",
  driver=RweaveHTML)
```

```
<html>
<head>
<title>Sweave and html</title>
</head>
<body>

Blah blah

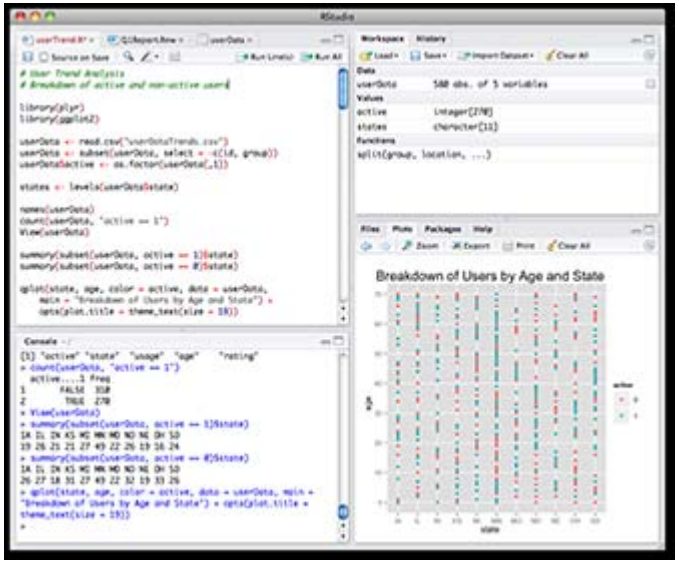
<<SweaveCode>>=
1+2
sum(1:10)
@

blah blah
</body>
</html>
```

Sweave.sty

- Style sheet for R code

RStudio



- 4 windows
  -Editor, Console, History, Files/plots
- Code completion
- Easy access to help (F1)
- One step Sweave pdf generation
- Searchable history
- Keyboard Shortcuts
  - http://www.rstudio.org/docs/using/keyboard_shortcuts

- Nice short cut button to build Sweave docs

KnitR

- **knitr** ≈ Sweave + cacheSweave + pgfSweave + weaver + R2HTML + more
- The design of **knitr** allows any input languages (e.g. R, Python and Awk) and any output markup languages (e.g. LaTeX, HTML, Markdown and reStructuredText)
- The name knitr was coined with weave in mind, and it also aims to be *neater*

Features of knitR

- Faithful
  - knitr writes everything that you see in an R terminal by default (results, plots and warnings)
- Built-in cache
- Formatting R code.
  - Colors. Uses **format R** package to “fix code” wrap long lines, add spaces and indent, etc
- Graphics
  - over 20 graphics devices, can set size etc
- Can use custom regular expressions to parse R

Converting Sweave Rnw to KnitR Rnw

- Very simple
- No spaces Chunk names
- results='hide' (need quotes)
- More chunk options (will review on Rstudio)

Format	Source file ending	Output	R Code Chunk	R expression
Rnw	Rnw (.Rnw)	**Tex, pdf**	<<R example>>= x <- 1+1rnorm(5) @	\Sexpr{pi}
Github format markdown	Markdown (.Rmd or .md)	**md, html**	``` {r example} x <- 1+1rnorm(5) ```	`r pi`.
HTML	Rhtml	**.html**	<!--R example x <- 1+1 rnorm(5) end.rcode-->	<!--rinline pi -->
reStructured Text	.Rst	**.rst**	.. {R example} .. x <- 1+1.. rnorm(5) .. .. NOTE:include space after the ..	:r:`pi`

Commands

```
knit(“tmp.Rnw”)
 purl("tmp.Rnw")

knit("example.Rmd")
knit2html("example.Rmd")
 knit2pdf("example.Rmd")
```

Markdown using knitR

- Markdown is not latex
- Very simple language

eg Emphasis

italic

bold

```
``` {r example}
x <- 1+1rnorm(5)
```
```

```
```{r}
plot(1:10)
hist(rnorm(1000))
```
```

Versatile – Converting MD with Pandoc

- Pandoc a universal document converter
  - http://johnmacfarlane.net/pandoc/index.html
  - Easy to convert markdown file to many formats

pdf file

```
system("pandoc -s example.md -t latex -o
   example.pdf")
```

html file

```
system("pandoc -s example.md -o example.html")
```

OpenOffice File

```
system("pandoc example.md -o example.odt")
```

Microsoft Word

```
system("pandoc example.md -o example.docx")
```

HTML5 Slides

```
system("pandoc -s -S -i -t dzslides --mathjax slides.md -o slides.html")
```

http://bcb.dfci.harvard.edu/~aedin/courses/ReproducibleResearch/slides.html

If nothing else…. 1. Organize

- Create new folder for each Project
  - Can even use Project -> new project in Rstudio
- Store scripts with incremental names
  - S001project.R, S002project.R etc
- In the top of the folder create a readme text file will list the scripts and what they do

2. Backup

- Use a document versioning system
  - eg SVN, CVS or GIT. Rstudio has simple support for SVN and GIT
- GIT
  - load packages directly from GIT into R using the devtools library function install_github()
- If nothing else store scripts on dropbox or other auto-backup system
  - So you can revert to previous version if it goes terribly wrong

3. Make a package

- Easier than you think

  package.skeleton()

- Tutorial on my website

Online publishing - Rpubs

- Free, from Rstudio
- Create a new R Markdown Doc
  - File -> New -> R Markdown.
- Click the **Knit HTML** button
- Preview click **Publish**

- **http://rpubs.com/**

Online Publishing – Shiny

- R package shiny
  - Shiny allows R developers to build simple interactive Web-based interfaces for R scripts, using only R code (no JavaScript required!)
  - http://www.rstudio.com/shiny/

Please feel free to contact me

aedin@jimmy.harvard.edu