Barriers to Reproducible Research and a Web-Based Solution

Matthew S. Shotwell
with contributions from
JoAnn M. Álvarez

Vanderbilt University
Department of Biostatistics

May 10, 2013
Reproducible Research

- RR is the practice of presenting computational research so that a scientific community may easily reproduce and verify the results.
- Distinct from “scientific replication”
- Reproducibility *i.e.* RR verifies an experimental result
- Replication strengthens evidence about a scientific theory
Origins of RR

- Jon Claerbout; geophysical image/signal processing; Stanford, mid 1980’s:
  - “a few months after completing a project, the researchers at our laboratory were usually unable to reproduce their own work without considerable agony”.
- [Schwab et al., 2009]
- Reviewing published results were no help (no code, no data)!
- Led to reverse engineering a colleague’s, even one’s own work!
- In biomedical research, [Baggerly and Coombes, 2009] call this “forensic bioinformatics”.

VANDERBILT UNIVERSITY
Claerbout’s 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 the complete set of instructions which generate the figures.

[Buckheit and Donoho, 1995]
[de Leeuw, 2001]
Claerbout’s Principle Clarified

The scholarship does not only consist of theorems and proofs but also (and perhaps even more important) of data, computer code and a runtime environment which provides readers with the possibility to reproduce all tables and figures in an article.

[Hothorn et al., 2009]
The Beneficiaries of RR

Quoting Schwab and Claerbout:

*It takes some effort to organize your research to be reproducible. We found that although the effort seems to be directed to helping other people stand up on your shoulders, the principal beneficiary is generally the author herself. This is because time turns each one of us into another person, and by making effort to communicate with strangers, we help ourselves to communicate with our future selves.*
Timeliness of RR (and sessions like this)

- As research becomes more technical, RR is more important
  - Journal page requirements haven’t increased
  - Online supplements help
- The barriers to RR are not philosophical, but practical
  - Few incentives from journals
    - *Biostatistics* - 2009 - ”kite-mark" D,C,R
    - *Nature* - 2013 - new methods policies
  - Proprietary/expensive research tools
  - Scooping/Coattailing (reciprocity/”viral” clause)
  - RR software tools (solved?)
  - Perception of effort
  - Survey evidence
Prevalence of RR

[Hothorn et al., 2009]: Considered v.50 Biometrical Journal:
  ▶ Among 53 articles with simulations:
    ▶ 17 provide data
    ▶ 8 provide code
    ▶ 6 “contain the whole scholarship” (data + code)

[Ioannidis et al., 2008] found similar in gene microarray articles
[Hothorn and Leisch, 2011] found better in Bioinformatics
Barriers to RR

- In conversation, statisticians often admit the benefits of RR.
- So, why isn’t reproducible research more prevalent?
- What are the barriers to adopting reproducible practices?
- We polled biostatisticians of VU Dept. of Biostatistics to assess:
  - the prevalence of fully scripted data analyses
  - the prevalence of literate programming practices
  - the perceived barriers to reproducible research
Prevalence of RR

Quartiles of RR Prevalence

- Faculty – Literate Programming
- Staff – Literate Programming
- Faculty – Scripted Data Manipulation
- Staff – Scripted Data Manipulation
- Faculty – Scripted Data Analysis
- Staff – Scripted Data Analysis

Proportion of Projects
Barriers to RR

The biggest obstacle to always reproducibly scripting your work?

<table>
<thead>
<tr>
<th>Barrier</th>
<th>Staff</th>
<th>Faculty</th>
</tr>
</thead>
<tbody>
<tr>
<td>No significant obstacles.</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>I haven't learned how.</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>It takes more time.</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>It makes collaboration difficult. (e.g. file compatibility)</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>The software I use doesn't facilitate reproducibility.</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>It's not always necessary for my work to be reproducible.</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Claerbout’s lab [Schwab et al., 2009], adopted an RR framework centered around the make utility.

GNU make: http://www.gnu.org/software/make/

make synchronizes the generation of output from source files.

make is configured using a Makefile with targets, dependencies and commands.

Targets are generated from their dependencies using the associated commands.

Example

target: dependency
  command1
  command2

ReDoc make targets: build, clean, burn, view

File naming conventions.
Example Makefile

build: results.pdf

results.pdf: results.tex results.bib
      pdflatex results.tex
      bibtex results
      pdflatex results.tex
      pdflatex results.tex

clean:
      rm -f results.aux results.log results.bbl

burn:
      rm -f results.pdf

view: results.pdf
      xpdf results.pdf
The RR Compendium

[Gentleman and Temple Lang, 2007]

- RR Compendium: a dynamic document containing text, code, and data.
- The complete scholarship, a la Claerbout’s principle.
- *Transformations* are applied to the compendium to view its various aspects (*e.g.* convert raw data into a graphic).
- Recommend using `make` to synchronize transformations.
Sweave

- Sweave [Leisch, 2002] is a tool for working with RR compendia in R
- Compendium: a \texttt{\LaTeX} file mixed with R code + data
- Transformations: weaving + tangling [Knuth, 1984]
Simple Sweave Example

example.Rnw:

\documentclass{article}
\begin{document}

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse elementum magna et velit molestie lobortis eget eget magna. In quis tincidunt risus. Mauris congue lacinia augue non varius. Vestibulum posuere nisi vel turpis consequat at bibendum libero convallis. \[ F(b) - F(a) = \int_a^b f(x)dx \]

<<fig=TRUE, keep.source=TRUE>>=
# From ?persp
y <- x <- seq(-10, 10, length= 30)
f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
z <- outer(x, y, f)
z[is.na(z)] <- 1
persp(x, y, z, theta = 30, phi = 30, expand = 0.5)
@

Integer eu purus non mi sagittis venenatis. Integer venenatis, nulla ac scelerisque volutpat, ante felis consectetur enim, vitae fringilla purus lorem et elit. Curabitur congue facilisis ipsum, non cursus tortor dignissim eu. Donec convallis feugiat eros et vestibulum.
\end{document}
Simple Sweave Example

R CMD Sweave example.Rnw → example.tex:

\documentclass{article}
\usepackage{Sweave}
\begin{document}

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse elementum magna et velit molestie lobortis eget eget magna. In quis tincidunt risus. Mauris congue lacinia augue non varius. Vestibulum posuere nisi vel turpis consequat at bibendum libero convallis. \[ F(b) - F(a) = \int_a^b f(x)dx \]

\begin{Schunk}
\begin{Sinput}
> # From ?persp
> y <- x <- seq(-10, 10, length= 30)
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5)
\end{Sinput}
\end{Schunk}

Integer eu purus non mi sagittis venenatis. Integer venenatis, nulla ac scelerisque volutpat, ante felis consectetur enim, vitae fringilla purus lorem et elit. Curabitur congue facilisis ipsum, non cursus tortor dignissim eu. Donec convallis feugiat eros et vestibulum.

\end{document}

\[ F(b) - F(a) = \int_{a}^{b} f(x)dx \]

```r
> # From ?persp
> y <- x <- seq(-10, 10, length= 30)
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5)
```

Integer eu purus non mi sagittis venenatis. Integer venenatis, nulla ac scelerisque volutpat, ante felis consectetur enim, vitae fringilla purus lorem et elit. Curabitur congue facilisis ipsum, non cursus tortor dignissim eu. Donec convallis feugiat eros et vestibulum.
Web-based RR Compendia

Why web-based?

► Pros:
  ► software requirements are a minimal barrier
  ► software compatibility and abstraction
  ► security (restricted access, encrypted transmission)
  ► centralized online storage
  ► persistent (keep a record of your work!)
  ► images/videos handled more naturally
  ► can be interactive (e.g. nomogram)
  ► mathematical typesetting

► Cons:
  ► centralized online storage
  ► mathematical typesetting
  ► browser variability
Web-based RR Compendia

- Use HTML rather than \LaTeX\ markup
- Mix text with code:
  - Sweave (HTML driver)
  - brew
  - yarr
- Can use ReDoc strategy with additional `make` target: `make publish`
- Publish just HTML or entire compendium
yarr

- yarr is an R package with facilities for mixing special text (e.g., R code) with plain text (or markup).
- In the yarr framework, special text must be delimited by “<<“ and “>>” (but this is customizable).
- Additional characters that follow the opening delimiter “<<“ tell yarr what to do with the delimited text. For example, special text that follows the opening delimiter “<</@” is treated like R code, and the output is inserted within the plain text, as if it had been entered into the R console.
- yarr is customizable so that special text may be treated in any number of ways (e.g., R code evaluated)
Simple yarr Example

`R -e 'yarr::yarr(''example.html.R'')' → example.html:`

```html
<html>
<head>
  <script type="text/javascript"
    src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML">
  </script>
</head>
<body>
  << source("R/common.R") >>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse elementum magna et velit molestie lobortis eget eget magna. In quis tincidunt risus. Mauris congue lacinia augue non varius. Vestibulum posuere nisi vel turpis consequat at bibendum libero convallis. \[ F(b) - F(a) = \int_a^b f(x)dx \]</p>
  <pre style="border: 1px solid #aaa;">
    perspplot <- expression(
      # From ?persp
      y <- x <- seq(-10, 10, length= 30)
      f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
      z <- outer(x, y, f)
      z[is.na(z)] <- 1
      persp(x, y, z, theta = 30, phi = 30, expand = 0.5)
    )
    >>
  </pre>
  << html.basegr(perspplot) >>
  <p>Ieteger eu purus non mi sagittis venenatis. Integer venenatis, nulla ac scelerisque volutpat, ante felis consectetur enim, vitae fringilla purus lorem et elit. Curabitur congue facilisis ipsum, non cursus tortor dignissim eu. Donec convallis feugiat eros et vestibulum.</p>
</body>
</html>
```
yarr Code Handlers

[[1]]
[[1]]$regex
[1] ""

[[1]]$handler
function (code, envir) {
  capture_handler(code, envir, output = FALSE)
}

[[2]]
[[2]]$regex
[1] "~="

[[2]]$handler
function (code, envir) {
  code <- sub("~=" , "", code)
  capture_handler(code, envir, source = FALSE)
}

[[3]]
[[3]]$regex
[1] "~&"

[[3]]$handler
function (code, envir) {
  code <- sub("~&", "", code)
  capture_handler(code, envir, prompt = FALSE)
}
- code handlers
  - may be modified on the fly
  - may handle special text that is not R code
  - add significant extensibility to the “literate programming” paradigm in R

- yarr at github: https://github.com/biostatmatt/yarr

- Example: VU Collaboration Example
Revision Control Systems

Have you ever:

▶ Made a change to code, realised it was a mistake and wanted to go back?
▶ Lost code or had a backup that was too old?
▶ Had to maintain multiple versions of a product?
▶ Wanted to see the difference between two (or more) versions of your code?
▶ Wanted to prove that a particular change broke or fixed some piece of code?
▶ Wanted to submit a change (patch) to someone else’s code?
▶ Wanted to see how much work is being done (where/when/who)?
▶ Wanted to experiment with a new feature without interfering with working code?

In all of the above cases, a version control system should make your life easier.

▶ Useful for modeling decisions (which aren’t often documented)
▶ Modern RCSs: Git, Subversion, Mercurial
User Friendly Web-Based Alternative

- RStudio/knitr/RMarkdown
Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology.

WaveLab and reproducible research.

Reproducible research: the bottom line.

Statistical Analyses and Reproducible Research.

Biometrical Journal and reproducible research.

Case studies in reproducibility.
*Briefings in Bioinformatics, 12*(3):288–300.

Repeatability of published microarray gene expression analyses.
Literate programming.

**Sweave**: Dynamic generation of statistical reports using literate data analysis.

Making scientific computations reproducible.