Reproducible Research

Sarah Marzi

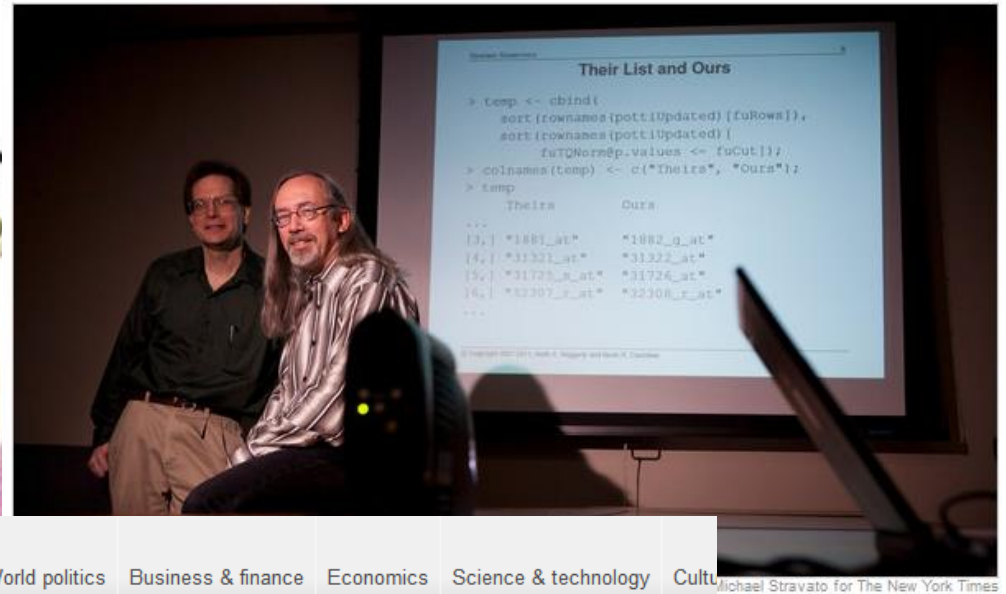
19/06/2014

Institute of Psychiatry at The Maudsley

KING'S College LONDON

The Potti scandal

How Bright Promise in Cancer Testing Fell Apart



Michael Stravato for The New York Times

ound flaws in research on tumors.

biggest medical resear

h world was
CBS report,
2 and titled
, described
ti, an oncolo-
e University,
t.
serting that
based treat-
combat can-
ulated in a
rnal Journal
e years later,
papers and
on which he
r.
raised early
ith Baggerly
st the M D
Center in
ti's team dis-
ical errors'.
stressed that
of careless-
results based on absent data and
inconsistencies in Potti's resume —
Rhodes Scholar, whether he had
and Institute of Medical Research,

"Anil Potti resigned
entific misconduct in
tinuing," according to
Sarah Avery, senior

The Economist | World politics | Business & finance | Economics | Science & technology | Cultu

Misconduct in science

An array of errors

Investigations into a case of alleged scientific misconduct have revealed numerous holes in the oversight of science and scientific publishing

Sep 10th 2011 | From the print edition

Timekeeper | Like 974 | Tweet 218

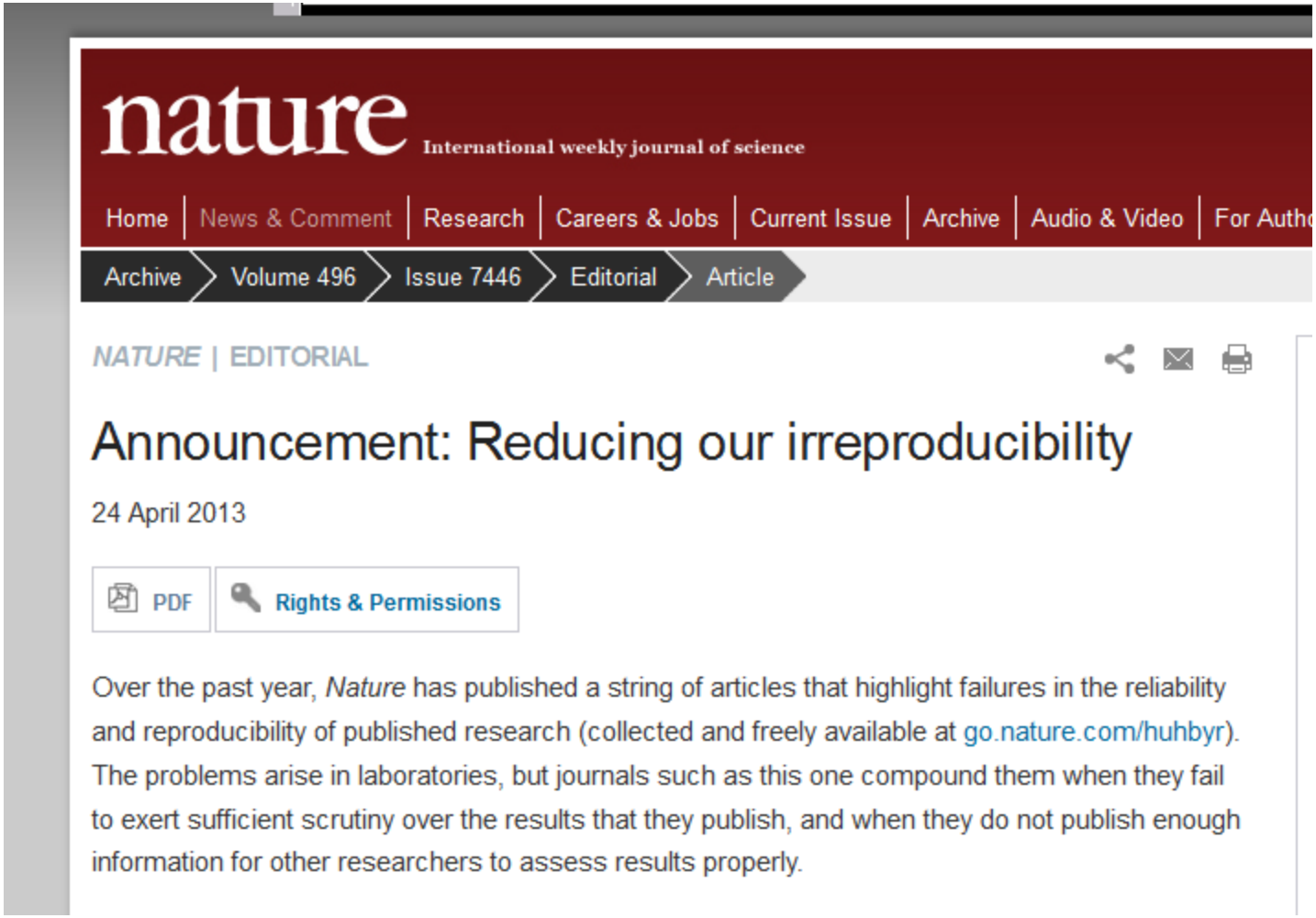
Institute of Psychiatry at The Maudsley

King's College London

Why reproducibility matters

- Replication as ultimate standard – but not always feasible

- For yourself:
  - Coming back to results, analyses, code after some time
- For collaborators:
  - Working with other bioinformaticians/statisticians
  - Communicating with scientists not specialised in statistics
- For the field of research as a whole:
  - Verification of analyses
  - Errors found more easily



nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Autho

Archive > Volume 496 > Issue 7446 > Editorial > Article

NATURE | EDITORIAL

Announcement: Reducing our irreproducibility

24 April 2013

PDF Rights & Permissions

Over the past year, *Nature* has published a string of articles that highlight failures in the reliability and reproducibility of published research (collected and freely available at go.nature.com/huhbyr). The problems arise in laboratories, but journals such as this one compound them when they fail to exert sufficient scrutiny over the results that they publish, and when they do not publish enough information for other researchers to assess results properly.

. . . results incorporate several simple errors that may be putting patients at risk. One theme that emerges is that the most common errors are simple (e.g., row or column offsets); conversely, it is our experience that the most simple errors are common -- Baggerly & Coombes, 2009

Dimensions of reproducible research

1. **Data**: Made available

   (Where? Permission obtained?)

2. **Code** which generated results published

   (Reproduce **all** published results)

3. **Documentation** and annotation of code and data

   (How easy is it to reproduce all results?)

Peng, Roger D. "Reproducible research and Biostatistics." *Biostatistics* 10.3 (2009): 405-408.

Data – what's wrong?

The poor availability of psychological research data for reanalysis.

By Wicherts, Jelte M.; Borsboom, Denny; Kats, Judith; Molenaar, Dylan
American Psychologist, Vol 61(7), Oct 2006, 726-728.

- Interested in 141 empirical articles published by the American Psychological Association (APA) fro reanalysis
- Contacted the corresponding author of every article that appeared in the last two 2004 issues of four major APA journals
- APA Certification of Compliance with APA Ethical Principles, which includes the principle on sharing data for reanalysis
- 6 months later and more than 400 e-mails later…

73% of the authors did not share their data.

Data – promising developments

- Several top tier journals including *Nature*, require authors to make data available as a condition of publishing
- Funding bodies (e.g. Wellcome Trust, CRUK) require researchers to make data publicly available
- Growing trend to link publications to the datasets which underpin the findings
- Big Data repositories are emerging
  - UCSC genome Browser
  - TCGA: http://cancergenome.nih.gov/
  - Ensembl
- …Metadatabases: http://biomart.org/

Data – Limitations

- Repositories often not ideally structured
- Data download for secondary research can be complicated and require very manual work
- Direct connections to some repositories can be achieved through specific R packages BUT
  - You have to know where to find the data you need
  - Names of data sets are often complicated
- Not all repositories are well curated
- Of all studies published, only some groups publish their data, even fewer make it available to compound repositories

SIAM NEWS >

Top Ten Reasons To Not Share Your Code (and why you should anyway)

April 1, 2013

Randall J. LeVeque

- It's just a research code, not software designed for others to use
- It's forbidden to publish proprietary code
- The code may run only on certain systems today, and nowhere tomorrow
- Code is valuable intellectual property

Why we need to publish code!

- Make all analyses transparent
- It's part of the research work (imagine a mathematician publishing a theorem without a proof)
- Think over what we have done critically and in detail (clean-up of code)
- More opportunity for us and others to find potential bugs
- Collaboration vs competition
- Citations and reputation

Producing reader-friendly code

```
resultsTime<-resultsTime[order(resultsTime$P, decreasing=FALSE),]
for (i in 1:10){f<-data.frame(betas(ERA)[featureNames(ERA)==resultsTime$ILMNID[i],], pData(ERA)$TimeInst)
names(f)<-c("Probe", "Time")
f$Time<-as.numeric(f$Time)
p<-ggplot(f, aes(x=Time, y=Probe))+geom_point(aes(colour="red"))+ stat_smooth(method="lm", se=FALSE) + theme(legend.position
png(sprintf("/home/marzis/ERA/ERA_DATA/timedep/scatter/%s.png", resultsTime$ILMNID[i]))
print(p)
dev.off()}
```

Producing reader-friendly code

```
resultsTime<-resultsTime[order(resultsTime$P, decreasing=FALSE),]
for (i in 1:10){f<-data.frame(betas(ERA)[featureNames(ERA)==resultsTime$ILMNID[i],], pData(ERA)$TimeInst)
names(f)<-c("Probe", "Time")
f$Time<-as.numeric(f$Time)
p<-ggplot(f, aes(x=Time, y=Probe))+geom_point(aes(colour="red"))+ stat_smooth(method="lm", se=FALSE) + theme(legend.position
png(sprintf("/home/marzis/ERA/ERA_DATA/timedep/scatter/%s.png", resultsTime$ILMNID[i]))
print(p)
dev.off()}
```

```
# Scatter plots for top results
resultsTime<-resultsTime[order(resultsTime$P, decreasing=FALSE),]
for (i in 1:10){
    f<-data.frame(betas(ERA)[featureNames(ERA)==resultsTime$ILMNID[i],], pData(ERA)$TimeInst)
    names(f)<-c("Probe", "Time")
    f$Time<-as.numeric(f$Time)
    p<-ggplot(f, aes(x=Time, y=Probe))+geom_point(aes(colour="red"))+ stat_smooth(method="lm", se=FALSE) + theme(legend.posit
    png(sprintf("/home/marzis/ERA/ERA_DATA/timedep/scatter/%s.png", resultsTime$ILMNID[i]))
print(p)
dev.off()
}
```

Producing reader-friendly code

```
# Make scatter plots for top 10 results of variable "time of deprivation" and store in directory"/home/marzis/ERA/ERA_DATA/timedep/scatter/"

resultsTime <- resultsTime[order(resultsTime$P, decreasing = FALSE),]
for (i in 1:10){
    f <- data.frame(betas(ERA)[featureNames(ERA) == resultsTime$ILMNID[i],], pData(ERA)$TimeInst)
    names(f) <- c("Probe", "Time")
    f$Time <- as.numeric(f$Time)
    p <- ggplot(f, aes(x = Time, y = Probe))
        + geom_point(aes(colour = "red"))
        + stat_smooth(method = "lm", se = FALSE)
        + theme(legend.position = "none")
    png(sprintf("/home/marzis/ERA/ERA_DATA/timedep/scatter/%s.png", resultsTime$ILMNID[i]))
print(p)
dev.off()
}
```

How to keep track of your analyses

1. SAVE IT!
2. In particular: save it somewhere safe (not just your PC)
3. Keep organised directories for projects (otherwise finding scripts becomes a nightmare)
4. 1 „short“ descriptive main script, many macros/functions (nobody can keep track of 200 page scripts)
5. Save output data sets (don‘t just manually record results)
6. Version control

Version control systems

- Initially used in large software projects (many prgrammers)
- Keep track of changes
- Allow to revert to older versions
- Why are they more powerful than Dropbox or google docs?
  - Keep a history of changes made to your code and data („changesets“, memory efficient)
  - Allow multiple people to work on a project together

  -> No risk of overwriting each other‘s changes
  - Avoid file names like „ProjectX_Analysis_Part2_Version17_Sarah.R“…

Apache Subversion (SVN): Server-client model

- Central repository on a server
- Check out working copies from here
- Commit changes to server -> changes version number
- Only the master on the server stores the whole history
- Commands: add, commit, checkout, update
- Needs access to server
- Lots of files stored safely on the server
- Example:

  http://mammoet.iop.kcl.ac.uk/svnshared/sarah_shared

Git: Distributed version control

- Lets you keep clones of a repository under version control
- Every clone contains all history (in .git directory)
- Commands: clone, commit, push, fetch, merge
- No connection to server neccessary (no problem if the server dies)
- Useful for debugging and then „pushing“ to others
- Developed by Linus Torvalds
- Popular commercial repository: github
- Example: https://github.com/seandavi

Literate Programming

- Originally proposed by Donald Knuth in 1984
- Idea: Programming to follow natural human logic as opposed to the logic imposed by the programming language
  - Text interwoven with code chunks
  - Forces programmer to conceptualize logically
  - Compiles both text and code, generates output (tangle & weave)
  - Updates changes from source files or code whenever it's compiled
- Not just documentation of code!

Literate programming in R

- Code, analysis and interpretation all in one document
- **Sweave**
  - Allows to embed R code and output in LaTeX documents
  - Dynamic reports
- **Knitr**
  - Successor of Sweave
  - Better modularization
  - Code integrated into LaTeX, LyX, HTML, Markdown etc.
  - Supports other programming languages as well (e.g. Python, C++. Perl)

```
\documentclass[a4paper]{article}

\title{Sweave Example 1}
\author{Friedrich Leisch}

\begin{document}

\maketitle

In this example we embed parts of the examples from the
\texttt{kruskal.test} help page into a \LaTeX{} document:

<<>>=
data(airquality, package="datasets")
library("stats")
kruskal.test(Ozone ~ Month, data = airquality)
@
which shows that the location parameter of the Ozone
distribution varies significantly from month to month. Finally we
include a boxplot of the data:

\begin{center}
<<fig=TRUE,echo=FALSE>>=
library("graphics")
boxplot(Ozone ~ Month, data = airquality)
@
\end{center}

\end{document}
```

LaTeX commands

R code

```
\documentclass[a4paper]{article}

\title{Sweave Example 1}
\author{Friedrich Leisch}

\begin{document}

\maketitle

In this example we embed parts of the examples from the
\texttt{kruskal.test} help page into a \LaTeX{} document:

<<>>=
data(airquality, package="datasets")
library("stats")
kruskal.test(Ozone ~ Month, data = airquality)
@
which shows that the location parameter of the Ozone
distribution varies significantly from month to month. Finally we
include a boxplot of the data:

\begin{center}
<<fig=TRUE,echo=FALSE>>=
library("graphics")
boxplot(Ozone ~ Month, data = airquality)
@
\end{center}

\end{document}
```

Sweave Example 1

Friedrich Leisch

April 29, 2014

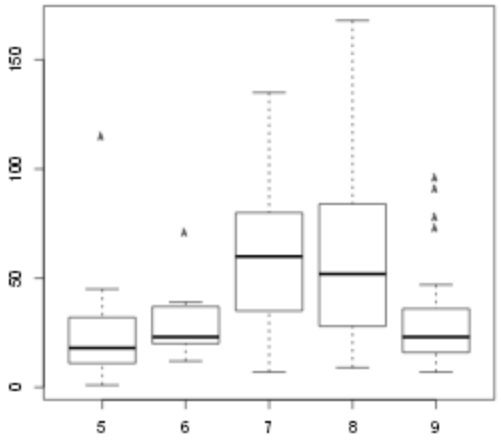
In this example we embed parts of the examples from the `kruskal.test` help page into a LaTeX document:

```
> data(airquality, package="datasets")
> library("stats")
> kruskal.test(Ozone ~ Month, data = airquality)

        Kruskal-Wallis rank sum test

data:  Ozone by Month
Kruskal-Wallis chi-squared = 29.2666, df = 4, p-value = 6.901e-06
```

which shows that the location parameter of the Ozone distribution varies significantly from month to month. Finally we include a boxplot of the data:



LaTeX output

R code

R output

Sweave Example 1

Friedrich Leisch

April 29, 2014

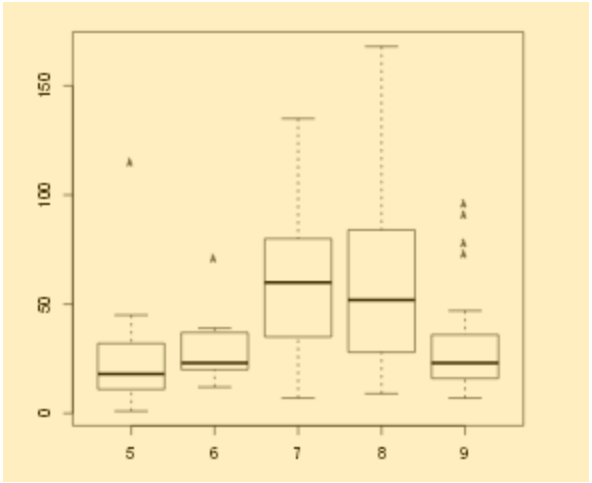
In this example we embed parts of the examples from the `kruskal.test` help page into a LaTeX document:

```
> data(airquality, package="datasets")
> library("stats")
> kruskal.test(Ozone ~ Month, data = airquality)
```

```
        Kruskal-Wallis rank sum test

data:  Ozone by Month
Kruskal-Wallis chi-squared = 29.2666, df = 4, p-value = 6.901e-06
```

which shows that the location parameter of the Ozone distribution varies significantly from month to month. Finally we include a boxplot of the data:



Markdown

- Created in 2004 by John Gruber
- Text formatting syntax
- Text to html conversion tool (same name as syntax)
- Goal: easy-to-read & easy-to write

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>

<body>
<h1>Markdown example</h1>

<p>This is a simple example of a Markdown document.</p>

<p>Use a blank link between paragraphs.
You can use a bit of <strong>bold</strong> or <em>italics</em>. Use backticks to indicate
<code>code</code> that will be rendered in monospace.</p>

<p>Here's a list:</p>

<ul>
<li>an item in the list</li>
<li>another item</li>
<li>yet another item</li>
</ul>

<p>You can include blocks of code using three backticks:</p>

<p><code>
x &lt;- rnorm(100)
y &lt;- 2*x + rnorm(100)
</code></p>
```

Karl Broman, „knitr in a nutshelll“,
http://kbroman.github.io/knitr_knutshell/

````
# Markdown example

This is a simple example of a Markdown document.

Use a blank link between paragraphs.
You can use a bit of **bold** or _italics_. Use backticks to indicate
`code` that will be rendered in monospace.

Here's a list:

- an item in the list
- another item
- yet another item

You can include blocks of code using three backticks:

```
x <- rnorm(100)
y <- 2*x + rnorm(100)
```
````

Karl Broman, „knitr in a nutshelll“,
http://kbroman.github.io/knitr_knutshell/

chunks.Rmd

Knit HTML | Chunks

```
R Code Chunks
==========

With R Markdown, you can insert R code
chunks including plots:

```{r qplot, fig.width=4, fig.height=3,
message=FALSE}
# quick summary and plot
library(ggplot2)
summary(cars)
qplot(speed, dist, data=cars) +
      geom_smooth()
```
```

RStudio: Preview HTML

Preview: ~/chunks.html | Save As | Publish

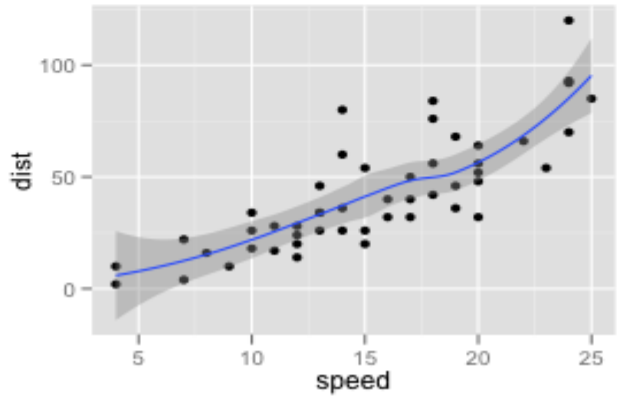
R Code Chunks

With R Markdown, you can insert R code chunks including plots:

```
# quick summary and plot
library(ggplot2)
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2
##  1st Qu.:12.0   1st Qu.: 26
##  Median :15.0   Median : 36
##  Mean   :15.4   Mean   : 43
##  3rd Qu.:19.0   3rd Qu.: 56
##  Max.   :25.0   Max.   :120
```

```
qplot(speed, dist, data = cars) + geom_smooth()
```



R Studio Support, https://support.rstudio.com/hc/en-us/articles/200552086-Using-R-Markdown

OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Editorial

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve[1,2*], Anton Nekrutenko[3], James Taylor[4], Eivind Hovig[1,5,6]

1. For Every Result, Keep Track of How It Was Produced
2. Avoid Manual Data Manipulation Steps
3. Archive the Exact Versions of All External Programs Used
4. Version Control All Custom Scripts
5. Record All Intermediate Results, When Possible in Standardized Formats
6. For Analyses That Include Randomness, Note Underlying Random Seeds

Institute of Psychiatry at The Maudsley

KING'S College LONDON

OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Editorial

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve[1,2*], Anton Nekrutenko[3], James Taylor[4], Eivind Hovig[1,5,6]

7. Always Store Raw Data behind Plots
8. Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
9. Connect Textual Statements to Underlying Results
10. Provide Public Access to Scripts, Runs, and Results

Sources

- Roger Peng, “Reproducible Research” Coursera online course, https://www.coursera.org/course/repdata
- Randall LeVeque, “High Performance Scientific Computing”, Coursera online course, https://www.coursera.org/course/scicomp
- Martin Morgan, “Reproducible Research in R / Bioconductor”, http://www.bioconductor.org/help/course-materials
- Roger Peng, "Reproducible research and Biostatistics." *Biostatistics* 10.3 (2009): 405-408.
- Keith A.Baggerly and Kevin R. Coombes, "Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology." *The Annals of Applied Statistics* (2009): 1309-1334.
- Geir Kjetil Sandve et al. "Ten simple rules for reproducible computational research." *PLoS computational biology* 9.10 (2013): e1003285.
- Jelte M. Wicherts et al. "The poor availability of psychological research data for reanalysis." *American Psychologist* 61.7 (2006): 726.
- Randall LeVeque, “Top Ten Reasons to Not Share Your Code (and why you should anyway)”, http://faculty.washington.edu/rjl/pubs/topten/topten.pdf
- R Studio Support, https://support.rstudio.com/hc/en-us/articles/200552086-Using-R-Markdown
- Karl Broman, „knitr in a nutshelll“, http://kbroman.github.io/knitr_knutshell/
- Friedrich Leisch, Sveawe User Manual