IPython Notebook and Tools for Reproducible Research

Andrew Quitadamo
Programming II

Overview

- Reproducible Research and Why it Matters

–

- The Anil Potti Saga

–

- IPython Notebooks

–

- Makefiles

–

- RMarkdown and Knittr

Download Anaconda

- Go to https://www.continuum.io/downloads
- Download Anaconda for OS X
- Follow the install instructions

What is Reproducible Research?

Reproducible research is the idea that data analyses, and more generally, scientific claims, are published with their data and software code so that others may verify the findings and build upon them. [1]

1

Why is reproducibility important?

- "Reproducibility Crisis"

–

- 47/53 "landmark" cancer studies couldn't be replicated [1]

–

- A 2014 paper reported only 25% reproducibility of CS papers in their study [2]

1 2

Why is reproducibility important?

- Research that isn't reproducible creates waste (money, time, effort)

–

- It can even affect patients

The Anil Potti Saga

- In 2006 Genomic signatures to guide the use of chemotherapeutics was published in Nature Medicine

–

- The group at Duke produced great results predicting chemosensitivity based on gene expression profiles

–

- Bioinformaticians at MD Anderson became interested in these results and attempted to use them

–

- When they followed the procedure in the paper, they didn't get similar results

–

- In fact their results looked totally different

The Anil Potti Saga (cont.)

- Forensic Bioinformatics time

–

- Attempt to recreate gene expression heatmaps for 7 chemotherapy drugs

–

- Compared list of genes and found an off-by-one error

–

- Bioinformaticians at MD Anderson ended up writing the documentation for the Duke software

–

- They replicated the off-by-one error by including a header in an input file

–

- They manage to match 6 of the 7 Heatmaps match, but only 3 of the 7 gene lists

–

- The list of sensitive samples and resistant samples were flipped in the Nat. Med paper

–

- Keith Baggerly and Kevin Coombes wrote a letter Nat. Med. and they published their code, which was reproducible

---

The Anil Potti Saga (cont.)

- The original authors responded saying Baggerly and Coombes are wrong

–

- Data published on the original authors web page had 144 samples in training/test data

–

- Baggerly and Coombes found only 84 unique samples, meaning samples were duplicated

–

- One sample was labeled resistant 3 times, and sensitive once

–

- The original authors redid their analysis with 95 “unique” samples

–

- They also took down the data from their website

–

- Of the 95 samples 15 were duplicated, 6 were labeled both as resistant and sensitive

---

The Anil Potti Saga (cont.)

- The Duke group published other papers, including one in the Journal of Clinical Oncology

–

- ERCC1, ERCC4 and DNA repair genes were found to be important

–

- ERCC1 and FANCM (DNA repair) aren’t measured on the microarray the authors used

–

- More papers using the same technique followed

The Anil Potti Saga (cont.)

- Clinical trials were started using the published gene signatures.

–

- Including the gene signatures that used flipped sensitive/resistant labels.

–

- People were literally receiving drugs that would not work

–

- Baggerly and Coombes publish a paper in The Annals of Applied Statistics addressing all the problems

–

- Duke suspends clinical trials, opens investigation

–

- Duke concludes their investigation and restarts the clinical trials

–

- Original authors published more data

–

- Every single sample was either mislabeled, or not in the data set they said they used

---

The Anil Potti Saga (cont.)

- A FOIA request was used to get the Duke report

–

- The investigation couldn't replicate the studies as published

–

- NCI removes funding from one clinical trial

–

- However Duke continues with its three clinical trials

–

- It comes to light that Anil Potti claimed on his CV that he was a Rhodes Scholar

–

- He wasn't

–

- 33 Biostatisticians send a letter to the NCI, Duke, ORI, DoD and to the press

–

- Duke suspends the trials

–

- Covered in NYT, NPR and elsewhere

The Moral of the Story

- This was a particularly bad combination of mistakes

–

- These mistakes are easy to make in bioinformatics research

–

- Simple mistakes are simple to fix. If you notice them. Documentation is important.

–

- Because of this all reports at MD Anderson are now 100% reproducible (written in Sweave).

Keith Baggerly's Talk

What can we do?

- Provide code

–

- Document the code

–

- Test the code

–

- Version control the code

–

- Provide data

–

- Be skeptical of results. Especially the good ones.

Reproducible Research Resources

- How to Avoid Having to Retract Your Genomics Analysis
- Myths of Computational Reproducibility
- Ten Simple Rules for Reproducible Computational Research
- Reproducible Research is Still a Challange
- Best Practices for Scientific Computing
- Tools and Techniques for Computational Reproducibility
- Five selfish reasons to work reproducibly

IPython Notebook

- IPython Notebooks provide a way to combine code, texts and plots.

–

- Similar to old-school lab notebooks, where results and methods are together. Somebody could look at your lab notebook and reproduce your analysis (in theory).

–

- Works in your browser, similar to the interactive Python shell on the command line.

Version 3.0.0

- IPython Notebooks aren't just for Python anymore. While earlier versions did have the ability to use different kernals, Version 3.0.0 makes using them alot easier.

–

- R, Julia, Perl, Bash, Spark, Haskell, Clojure, Go, Scala and many others.

NBViewer

- Provides a place to display and share IPython Notebooks with others

–

- Paper in Nature Genetics

–

- IPython Notebooks and NBViewer can be used to help create reproducible research.

–

- Plus its really cool.

IPython Notebook Resources

- NBViewer
- Interesting IPython Notebooks
- Bioinformatics with Python Cookbook
- Ben Langmead's Computational Genomics Class
- An Introduction to Applied Bioinformatics

Makefiles

- GNU Make allows you to automatically execute rules, and specify dependencies for targets

–

- A simple Makefile looks like:

```
targetfile: dependancyfile
    rule_to_create_targetfile
```

Here is an actual example from one of our projects:

```
data/Homo_sapiens.GRCh37.75.gtf:
    wget -P ./data ftp://ftp.ensembl.org/pub/grch37/release-81/\
            gtf/homo_sapiens/Homo_sapiens.GRCh37.75.gtf.gz
    gunzip data/Homo_sapiens.GRCh37.75.gtf.gz

data/gene_positions: data/Homo_sapiens.GRCh37.75.gtf
    python code/extract_gene_position.py data/Homo_sapiens.GRCh37.75.gtf data/gene_positions
```

Why Make

- Allows you to specify dependancies

- Unlike a shell script, Make only reruns rules when necessary

- You can use Make with any language

- Make is already installed on Mac OSX, Linux and Unix systems

---

Make Resources

- Karl Broman's Minimal Make Tutorial
- Why Use Make
- Make for Reproducible Data Analysis

Knitr and RMarkdown

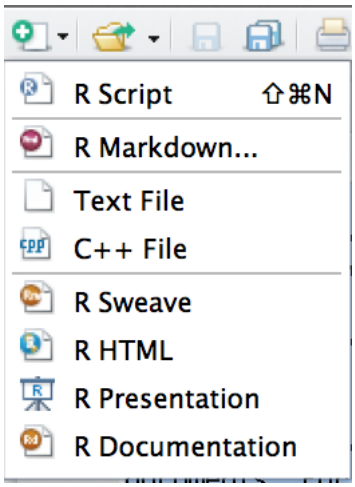
- Knitr can be used to combine text and R code to produce dynamic reports

_

- The R code is embedded in an RMarkdown document, and can be rerun by anyone

---

How to Create a RMarkdown Document



Knitr Example

```
---
title: "Example Knitr Document"
author: "Andrew Quitadamo"
date: "January 27, 2016"
output: html_document
---

This is an R Markdown document.
```

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content
as well as the output of any embedded R code chunks within the document.
You can embed an R code chunk like this:

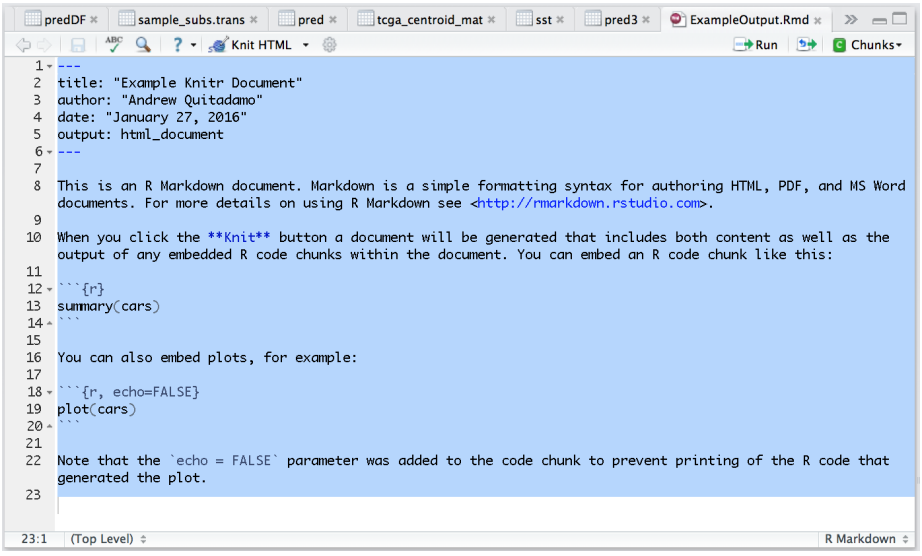
```
 ```{r}
summary(cars)
 ```
```

You can also embed plots, for example:

```
 ```{r, echo=FALSE}
plot(cars)
 ```
```

Note that the `echo = FALSE` parameter was added to the code chunk
to prevent printing of the R code that generated the plot.

Knit



HTML Output

Example Knitr Document

Andrew Quitadamo

January 27, 2016

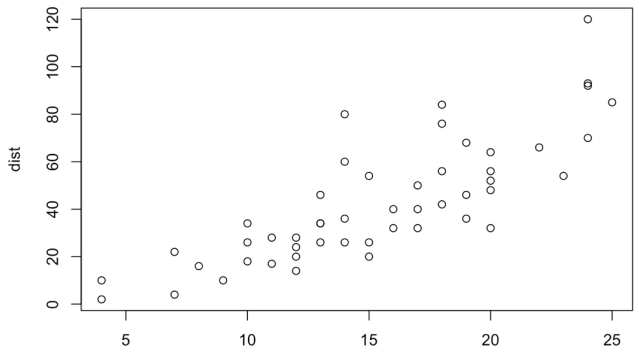
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

You can also embed plots, for example:



Knitr Resources

- Knitr in a Knutshell
- Knitr Showcase
- Knitr Homepage

Other Tools for Reproducible Research

- FigShare
- Data Dryad
- GitHub
- BitBucket