Making Implementations Available for the Research Community

Neil D. Lawrence

Departments of Neuro- and Computer Science, University of Sheffield, U.K.

Talk at Validation in Statistics and Machine Learning, Berlin

6th October 2010

Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

Nicole's Introduction

2. difficulties to **reproduce** published results.

Reproducible Research

> *An article about computational science in a scientific publication is* **not** *the scholarship itself, it is merely the* **advertising** *of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.*

Buckheit and Donoho (1995)

The Idea

- Make research in the “computational sciences” reproducible.
- Researchers provide their code allowing all figures from their paper to be reproduced.
- For me: first asked to provide code for a 2003 *Bioinformatics* paper.
- **This talk:** motivations for why that is the right thing and how we are trying to carry it out.

Reproducible Research

Examples from Buckheit and Donoho (1995) with my commentary.

- *Burning the Midnight Oil*
  Good practise suggests the first thing a researcher should try is a "toy problem". Once the toy problem is working, researcher moves onto a real problem. To make it work perhaps tweaks are needed to generate the next set of results. Does the tweaked code also generate the same results as the original code on the toy problem? Is it possible to produce all results with exactly the same algorithm? The problem reoccurs when you try a new data set. Did you tweak your algorithm for the second data set? Does it still work on the first? I've followed up research where it turned out some results were on normalized data and others weren't. This wasn't in the text.

Reproducible Research

- *The Stolen Briefcase*
  Backed up storage can be expensive. Often we write results to local drives. If the local drive crashed, could you recreate the results stored their using your existing code?
  If a hard drive crashed containing the written results can you recover everything using your code? If not why not? Did your research really add to "Human Knowledge"?

Actually Boss ...

- *Who's on First?*
  - Does the Prof's idea really work?
  - When was the last time you queried student's good result?
  - Do students hack beyond the original idea to make things work?
  - Bootstrap particle filters require significant annealing of the likelihood to work, but many people don't seem to know this — students do. Alchemy!!
  - If it's not working how easy is it for the Prof to examine their code?
- *A la Récherche des Parametres Perdues*

Reanimating Suspended Work

- *A Year is a Long Time in this Business*
  - How easy is it for you to return to suspended projects?
  - If you want to resurrect something, can you remember how you did it?
  - If a new student arrives to build on a previous students work, can you get them started without access to the old student?
  - When you return to your own software after some time, you experience it much like a newcomer does.

Claerbout and Reproducible Research

- Works on Seismic Imaging.
- His main point: the deliverable is not the sub-surface image, but the software that create the image.
- For us it is vital that we understand that the journal paper is only part of the deliverable.
- Even if we are measured by citations and publications, our real worth is contribution to knowledge.
- Research which is not reproducible leads to **Orphaned Software**.

Orphaned Software and Orphaned Research

- Software which has no maintainer or documentation.
- Happens in companies all the time (just think of all the different word processing softwares: wordwise, wordstar, ...).
- Also happens in your group: postdoc leaves, visitor/intern collaborates and then goes. Student moves to industry.
- For companies societal contribution ceases to exist when the product is terminated. (although open source is ameliorating this somewhat)
- This *cannot* be allowed to happen with research.
- **Orphaned Research** is a big problem too!!
- Reproducible research is one answer.
- There is an Interaction with Free Software but also independent.

Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

Changing Times

- Times are changing rapidly.
- Printed distribution of scholarly work hails from 17th Century: Royal Society etcetera.
- It's *unclear* what academic knowledge distribution will look like in 50 years time.
- It's *clear* (we hope!) that science will still be contributing to society.
- We should be focusing on that contribution not current metrics of quality.
- Journal publications are important but so is the underlying scholarship.
- **Quick aside:** historical availability of numerical algorithms and statistical software.

Aside: Would Mathworks Exist Today?

- Developed by Cleve Moler as a simple interface to EISPACK and LINPACK, predecessors of LAPACK.
- At the time (1984) distribution of software was difficult.
- Main contribution of MATLAB: nice interface for EISPACK/LINPACK and distribution of resulting code.
- Today: Open source projects could have handled each stage. See `scipy`, `octave`, `R`, Weka.
- In Statistics R (Ihaka and Gentleman, 1996) seems to have replaced SAS & S-PLUS for academic statisticians.
- SAS and S-PLUS have academic origins but were lost to the community through formation of companies to distribute.
  - Today maybe commercial software should focus on what they are good at: interfaces for less technically able.
- We must not loose control of our underlying software.

Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

Bioconductor

- Bioconductor (Gentleman et al., 2004) provides another example of what's possible when academics work together to create software frameworks.
- Open source system for computational biology and the *de facto* standard for microarray analysis.
- We've shipped two packages through bioconductor:
  - tigre: `http://www.bioconductor.org/help/bioc-views/release/bioc/html/tigre.html`
  - puma: `http://www.bioconductor.org/help/bioc-views/release/bioc/html/puma.html`
- Excellent for computational biology.
- Documentation using Sweave (Leisch, 2002) which allows R code to be integrated into LaTeX.

Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

What do we do?

Two suggested actions for your research.

1. Develop a systematic way of writing code within the group so that code can be rapidly disseminated.
2. On submission of the paper freeze a version of the code and supply it with the submission.

These two actions will improve the quality of code written and the quality of science produced.

Our System for Source Release

- **Rigid directory structure**
  - Parent directory (projects) then:
  - Subdirectory for each project including.
  - Each project directory has `tex`, `matlab`, `html` subdirectories.
  - `tex` directory has subdirectories for each paper or talk (e.g. `nips10` or `techreport` plus a subdirectory for figures (`diagrams`) and one for useful shared talk slides (`talks`).
  - `html` directory contains stub for software home page including examples of how to run the code.
  - `matlab` directory contains the code. We use consistent formatting for each file's comments. Each `.m` file is tagged to indicate it is in the toolbox. Also contains `.txt` files like `readme.txt`, `additionalfiles.txt` and `ignorefiles.txt`.

Our System for Source Release

- **Directory structure is all stored within SVN available to collaborators and students.**
  - We use `http://www.assembla.com`
- **On software release, we use a python script file to process all `.m` files.**
  - A new subdirectory under `matlab` is created with the software release, e.g. `mlprojects/fgplvm/matlab/FGPLVM0p1`.
  - This is also copied to the web along with an `index.html` augmented with all the toolbox dependencies.

Problems: Dreaded Dependencies

- **Theory:** release software every time you release code.
- **Practice:** you keep updating it, do you want to re-release the whole thing.
- **Solution:** subdivide your software into toolboxes. e.g. `kern`, `mltools`, `ndlutil`, `gp`.
  - Unfortunately you tend to update these quite regularly too, necessitating regular releases. `kern` toolbox has 28 releases in 6 years!
  - It is a pain for your users having to download all these toolboxes.
- Each `matlab` directory has a file for loading in relevant toolboxes, e.g. `fgplvmToolboxes.m`.
- Also have `.m` (`importTool.m`, `closeTool.m`) files for managing the path and including appropriate toolbox (e.g. latest release, current working or legacy toolbox).
- *If* rules are stuck to, software release takes about 30 seconds!

Problems: Create Individuals

- We work with very creative talented people: collaborators, post-docs, students.
- They don't always understand the need for rigid systems.
- They want to try new things: git, python etc.
- Sometimes they are right!
- You have to be flexible and make judgement calls.

Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

MATweave

- A couple of tricks to allow integration of MATLAB/Octave and LaTeX.
- Relies on the fact that MATLAB contains block comments since R14 and Octave since 3.2.
- Begin comment in MATLAB/Octave is %{ and end comment is %}.
- Just need to add a new environment to LaTeX for MATLAB/Octave code.

```
\newenvironment{matlab}{\comment}{\endcomment}
\newenvironment{octave}{\comment}{\endcomment}
\newenvironment{matlabv}{\verbatim}{\endverbatim}
\newenvironment{octavev}{\verbatim}{\endverbatim}
```

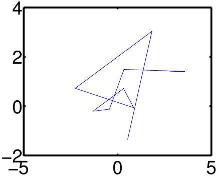
Example MATweave Code

```
%{
% these lines are commented for a MATLAB/Octave parse
\documentclass{article}
\usepackage{verbatim}
\usepackage{graphicx}
\newenvironment{octavev}{\verbatim}{\endverbatim}
\begin{document}
\begin{octavev}
%}
  % these lines are in a LaTeX verbatim environment.
  plot(randn(10, 1), randn(10, 1), ’b’);
  set(gca, ’linewidth’, 4, ’fontsize’, 44);
  print -depsc myGaussian.eps
  system(’epstopdf myGaussian.eps’);
%{
\end{octavev}
\includegraphics[width=4cm]{myGaussian}
\end{document}
%}
```

Compiling MATweave

- First run the MATLAB/Octave with
  - `octave -eval source\ example.tex`
    or `matlab < example.tex`
    pdflatex example.tex
- This gives the result:

```
%}
  % these lines are in a LaTeX verbatim environment.
  plot(randn(10, 1), randn(10, 1), 'b');
  set(gca, 'linewidth', 4, 'fontsize', 44);
  print -depsc myGaussian.eps
  system('epstopdf myGaussian.eps');
%{
```



Outline

Motivation

Changing Times

Bioconductor

Our System

MATweave

Conclusions

Conclusions

- It's about habits, not rules.
- It's about good practise: like spell checking.
- It's about courtesy to other researchers.
- It's about keeping track of students and visitors work.
- It's something you should all be doing.
- It's about **making research reproducible**.

References I

J. B. Buckheit and D. L. Donoho. WaveLab and reproducible research. Technical report, Stanford University,

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biol*, 5(10):R80, 2004. [URL]. [DOI].

R. Ihaka and R. C. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5: 299–314, 1996.

F. Leisch. Sweave, Part I: Mixing R and LaTeX: A short introduction to the Sweave file format and corresponding R functions. *R News*, 2(3):28–31, 2002.