Litterate programming

Arnaud Legrand and Jean-Marc Vincent

Scientific Methodology and Performance Evaluation
ENS Lyon, November 2016

Outline

1. Reporting Results
   - An IMRAD Report
   - Good Practice for Setting up a Laboratory Notebook
2. Emacs Demo of How to Keep Things Tidy

Structure

Research articles are often structured in this basic order:

- **Introduction** Why was the study undertaken? What was the research question, the tested hypothesis or the purpose of the research?
- **Methods** When, where, and how was the study done? What materials/hardware were used? How was it configured?
- **Results** What answer was found to the research question; what did the study find? Was the tested hypothesis true? Present useful results in a synthetic way with a logical order.
- **Discussion** What might the answer imply and why does it matter? How does it fit in with what other researchers have found? What are the possible bias and points to improve? What are the perspectives for future research?

Such structure facilitates literature review and is a very effective way to convey information.

If the report is a few pages long then an abstract is required.

Outline

1. Reporting Results
   - An IMRAD Report
   - Good Practice for Setting up a Laboratory Notebook

2. Emacs Demo of How to Keep Things Tidy

Step 0: Taking Notes

Document your:

- Hypotheses: keep track of your ideas/line of thoughts
- Experiments: details on how and why an experiment was run, including failed or ambiguous attempts.
- Initial analysis or interpretation of these experiments: was the outcome conform to the expectation or not? does it (in)validate the hypothesis?
- Organization: keep track of things to do/fix/test/improve

Structure:

1. General information about the document and organization conventions (e.g., directory structure, notebook structure, experimental result storing mechanism, . . .)
2. Documentation of commonly used commands and of how to set up experiments (e.g., git cloning, environment deployment, connection to machines, compiling scripts)
3. Experiment results can be either structured by dates (⇝ add tags) or by experiment campaigns (⇝ add date/time)

Which format should I use ?

- Wikis are encouraged to favor collaboration but I do not find them really effective
- Blogging systems are also a way of managing such notebook but they should rather be considered as an effective way to share information with others
- I recommend to use basic plain-text format and to structure it hierarchically

  Here is a link to an excerpt of the journal of one of my PhD student, managed with git/org-mode. More detailed links are given in slide 12.

Last but not least:

Provide links to Raw Data!!!

When/How Often Should I Use it?

I have a very intense usage (demo to general journal and specific BOINC journal) and I tend to capture a lot of information but you do not have to be as extreme as I am. Here are a few advices:

- Spending more than an hour without at least writing what you're working on is not right...
    - Take a 5 minutes break and ask yourself what you're doing, what is keeping you busy and where all this is leading you
- While working on something, you will often notice/think about something you should fix/improve but you just don't want to do it now. Take 20 seconds to write a TODO entry.
- There are moments where you have to wait for something (compiling, deployment, ...). It is generally the perfect time for improving your notes (e.g., detail the steps to accomplish a TODO entry).
- By the end of the day: daily (and weekly) review!
    - Update your lists, write what the next steps are
    - Summarize in a 2-4 lines (for your advisor) what you did, what was difficult, what you learnt.

Step 1: Sharing Code and Data

What kinds of systems are available?

- "Good" - The cloud (Dropbox, Google Drive, Figshare)
- Better - Version control systems (SVN, Git and Mercurial)
- "Best" - Version control systems on the cloud (GitHub, Bitbucket)

Depends on the level of privacy you expect but you probably already know these tools. **Few handle GB files...**

Is this enough?

1. Use a workflow that documents both data and process
2. Use the machine readable CSV format
3. Provide raw data and meta data, not just statistical outputs
4. Never do data manipulation and statistical tests by hand
5. Use R, Python or another free software to read and process raw data (ideally to produce complete reports with code, results and prose)

Courtesy of Adam J. Richards

Step 2: Literate Programming

Donald Knuth: explanation of the program logic in a natural language interspersed with snippets of macros and traditional source code.

I'm way too 3133t to program this way but that's
exactly what we need for writing a reproducible article/analysis!

Org-mode (requires emacs)

My favorite tool.

- plain text, very smooth, works both for html, pdf, . . .
- allows to combine all my favorite languages even with sessions

Ipython notebook

If you are a python user, go for it! Web app, easy to use/setup. . .

KnitR (a.k.a. Sweave)

For non-emacs users and as a first step toward *reproducible papers*:

- Click and play with a modern IDE (e.g., Rstudio)

Outline

1. Reporting Results
   - An IMRAD Report
   - Good Practice for Setting up a Laboratory Notebook

2. Emacs Demo of How to Keep Things Tidy

Literate Programming on a Daily Basis

Mastering Emacs

- `C-g`: get me out of here!
- `C-_`: undo
- Activate CUA keys in the Options menu

Mastering Org-mode

- `Tab` will fold/unfold stuff
- `C-c C-c`: do something (context-sensitive) where you are
- `<s` + `Tab`, `<b`, `<l`, `<r`, `<h`, ... for creating code blocks
- `C-c C-e`: export
- `C-c c`: capture content
- `C-c C-o` / `C-c l` / `C-c C-l`: open/store/insert links
- `C-c C-a`: attach a file
- `C-c C-d`: set deadline, `C-c C-t`: TODO/DONE

Emacs/Org-mode Recap

Key features

- Plain text makes it very robust and human readable
- Allow to mix any language and has a notion of session that makes its use very effective
- Allow to produce both `html documents`, classical LaTeXarticles, beamer slides, odt documents, . . . Native pretty printing on Github

A Few Links to Learn More

- *Org for beginners*, *my emacs configuration* and *tricks for Mac OS X users*
- A *script* capturing and gathering many information into a *single result document*
- A *laboratory notebook* with notes about all the experiments performed since the beginning of the project
- *Litterately conducting experiments* using org-mode