Data processing & automatization

ENBIT Workshop

Brussels, May 2018

Christophe Phillips,
GIGA Institute, ULiège, Belgium
c.phillips@uliege.be - http://www.giga.ulg.ac.be

Message from "Captain Obvious"

ENBIT Workshop

Brussels, May 2018

Christophe Phillips,
GIGA Institute, ULiège, Belgium
c.phillips@uliege.be - http://www.giga.ulg.ac.be

Job done ?

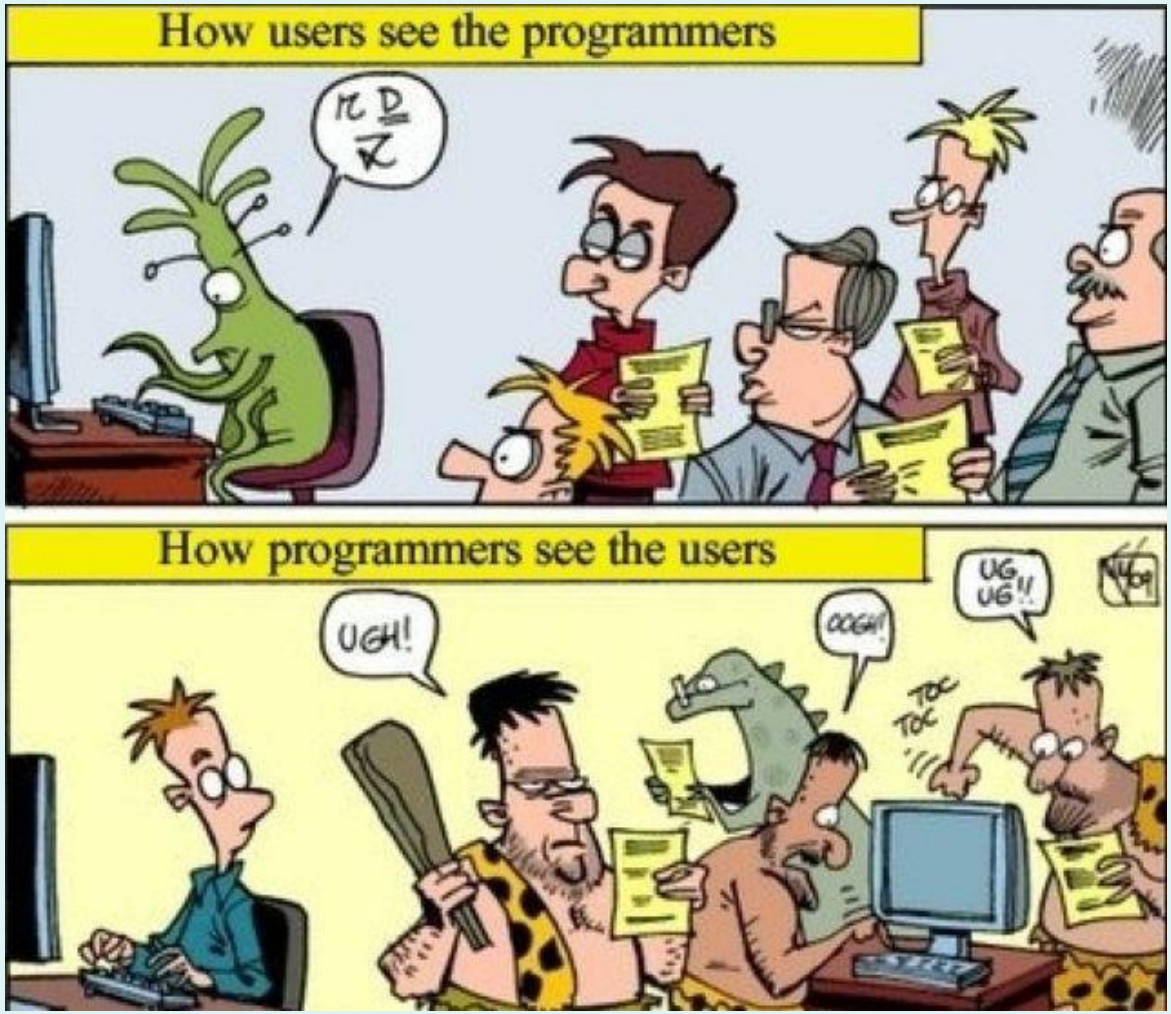
Methods people vs. Clinical people:

- M-people → happy when…
  - there is a theoretical solution,
  - evaluated and tested on some data,
  - relies on "a few" parameters to be adjusted

- C-people → happy when…
  - there is a 1-button software solution
  - works on any data
  - …all the time

Find some common ground!

Programmers vs. Users



Find some common ground!

Why?

Science relies on data and its analysis.

→ use & write scientific software!

Do we know

- what we want?

  → Mostly yes.
- how to calculate it?

  → We are working on it.
- how to build the “tool”?

  → Usually done “as it flows”!

→ Software/code development best practices!

Better code writing goal

Improve

- productivity of scientific programming,
- reliability of the resulting code.

→ speed up result production

→ boost confidence in results

→ ensure results reproducibility

→ increase your scientific impact

Best practices

1. Write programs for people, not computers
2. Let the computer do the work
3. Make incremental changes
4. Don’t repeat yourself or others
5. Plan for mistakes
6. Optimize software only after it works correctly
7. Document design and purpose, not mechanics
8. Collaborate

G. Wilson et al., *Best Practices for Scientific Computing*, PLOS Biology, 12:e1001745, 2014

Code & Document

1. Write programs for people
7. Document design and purpose, not mechanics

- Make names consistent, distinctive, and meaningful.
- Make code style, input/output and formatting consistent
- Break programs up into “simple modules”
- Document interfaces and reasons, not implementations (40% of file content!).

Code & Automatize

2. Let the computer do the work
4. Don’t repeat yourself or others

- never change data manually!
- do not type commands more than once
- script code for a “re-do this” call
- turn scripts into functions
- modularize code rather than copy-pasting bits.
- re-use code instead of rewriting it.

Code & develop

3. Make incremental changes
5. Plan for mistakes
8. Collaborate

- **use a version control system**.
- put everything that has been created manually in version control.
- automated testing of the code, in part or whole (unit, integration, regression tests)
- like manuscript writing, have colleagues review the code and/or write the code together

Can we trust our tools & results?

		Data	
		Same	Different
Code	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Kirstie Whitaker, https://github.com/WhitakerLab/ReproducibleResearch

How to evaluate/optimize ?

Several issues:

- Gold standard reference
  → rely on expert manual marking?
  → inter- & within-rater variability?

- Measure the match between prediction & GS
  → which metric(s)?
  → which one(s) really matter(s)?

- Parameter setting & optimization
  → train-test split & cross-validation or
  independent test set.
  → danger of overfitting & double dipping !

Data format & organization

Stick to **open data format**:

- Text & meta-data:
  - text → markdown file (`.md`)
  - array → tab-separated value file (`.tsv`)
  - key/value & structure → JSON file (`.json`)

- Images: NIfTI + JSON file (`.nii` + `.json`)

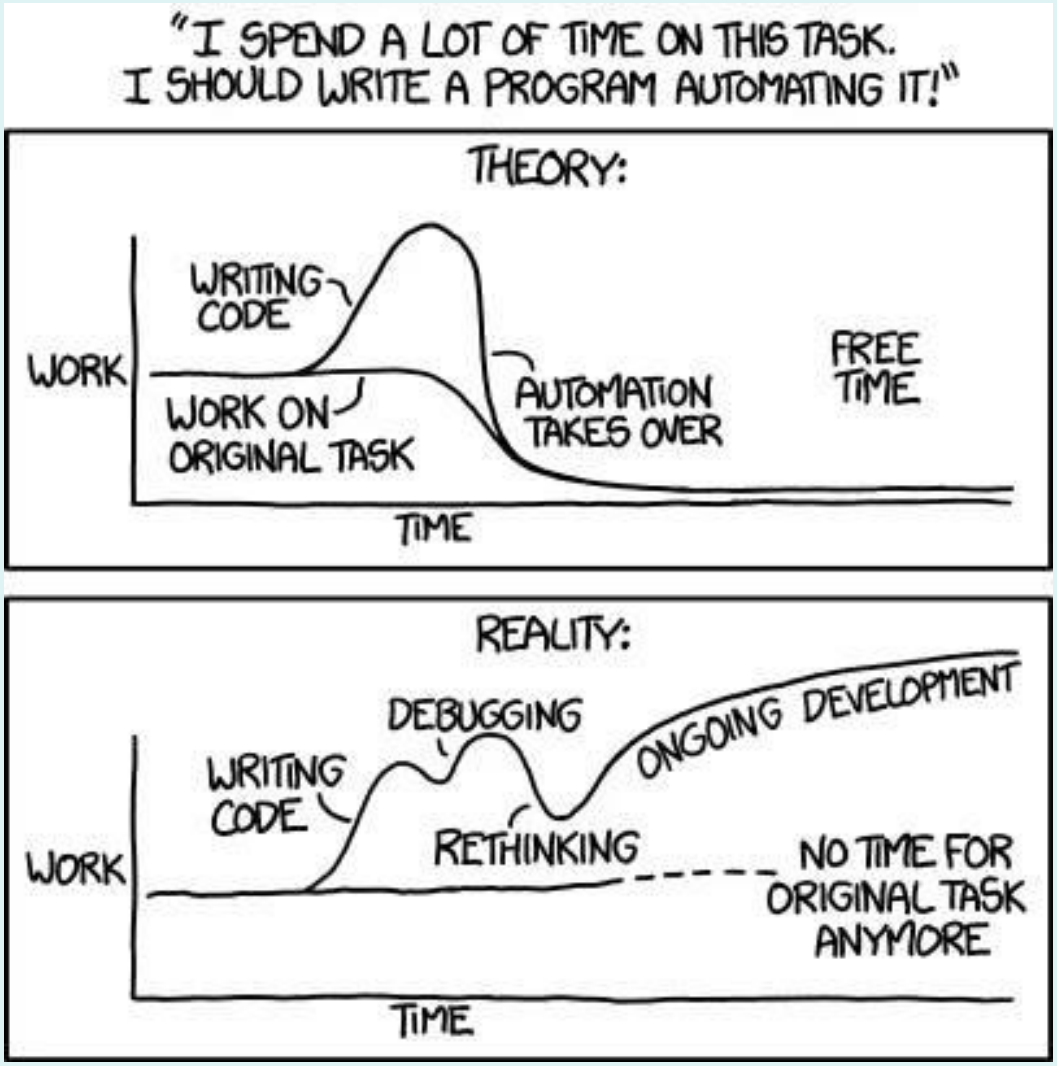
  Using BIDS formatting ?

`.md`, `.tsv` & `.json` files should be versioned!

Thank you for your attention!

Any question?

…and don’t forget



References

- G. Wilson et al., *Best Practices for Scientific Computing*, PLOS Biology, 12:e1001745, 2014 http://dx.doi.org/10.1371/journal.pbio.1001745

- J. D. Blischak et al., *A Quick Introduction to Version Control with Git and GitHub*, PLOS Computational Biology, 12(1): e1004668, 2016 http://dx.doi.org/10.1371/journal.pcbi.1004668

- https://github.com/ and http://gitlab.com

- https://en.wikipedia.org/wiki/Markdown
- https://en.wikipedia.org/wiki/Tab-separated_values
- https://en.wikipedia.org/wiki/JSON
- https://nifti.nimh.nih.gov/