Computational tools for reproducibility

Antonio J. Berlanga-Taylor

25 April 2017

MRC-PHE Centre for Environment and Health

Imperial College London

Reproducibility in science

- Why now? What? How?
- Approaches and tools
- Work at EBS

- About me:
  - MBBS, MSc, DPhil (wet lab functional genomics)
  - CGAT Fellowship: transition to computational methods
  - No formal computer science training…

Computational methods in biomedical research

- Exponential increase in data generation
- Big data → Big noise
- Growing labs, collaborations, analysis complexity
- Need for new workflows, tools, forms of collaboration

Reproducible data analysis

(same results with the same data)

Crisis in Science

RESEARCH ARTICLE

Estimating the reproducibility of psychological science

Open Science Collaboration*,†
+ See all authors and affiliations

Science 28 Aug 2015:
Vol. 349, Issue 6251, aac4716
DOI: 10.1126/science.aac4716

BBC Sign in News Sport Weather iPlayer TV Ra

NEWS

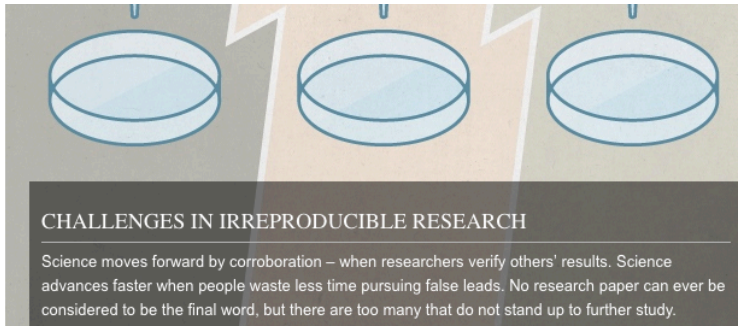
Home UK World Business Politics Tech Science Health Education Enterta

Science & Environment

Most scientists 'can't replicate studies by their peers'

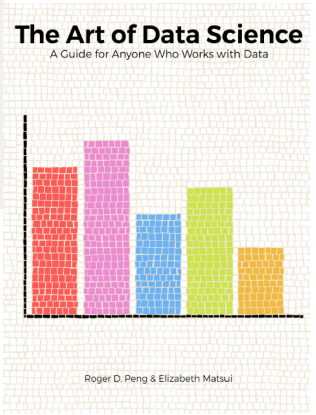
By Tom Feilden
Science correspondent, Today programme

- Psychology but also many other fields
- Current trends: concepts in place, higher entry points demanded from community and journals
- Reproducibility = the same results with the same data
- Replication = different cohort, organism, platform, etc.
- Open movement: data, code, publication



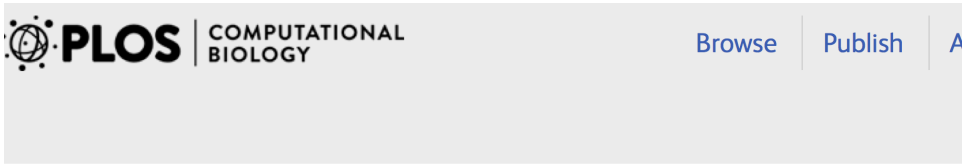
Data analysis (big and small)

- Reproducibility as a minimum standard
- Replicable
- Transparent
- Available
- Data + code



Idea + Question + Hypothesis → Experiments → Data + Analysis code → Memory + RAM + CPU → Results → New questions/Reformulation → Iteration

Standards and community expectations



OPEN ACCESS

EDUCATION

A Quick Guide to Organizing Computational Biology Projects

William Stafford Noble

EDITORIAL

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, Eivind Hovig

Published: October 24, 2013 • http://dx.doi.org/10.1371/journal.pcbi.1003285

OPEN ACCESS

COMMUNITY PAGE

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richa
Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Pau

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

Standards and community expectations

NATURE | EDITORIAL

Announcement: Transparency upgrade for Nature journals

The Nature journals continue journey towards greater rigour.

15 March 2017

POLICY FORUM | REPRODUCIBILITY

Enhancing reproducibility for computational methods

Victoria Stodden[1], Marcia McNutt[2], David H. Bailey[3], Ewa Deelman[4], Yolanda Gil[4], Brooks Hanson[5], Michael A.

+ See all authors and affiliations

PLOS | SUBMISSION

Good Enough Practices in Scientific Computing

Greg Wilson[1,‡*], Jennifer Bryan[2,‡], Karen Cranston[3,‡], Justin Kitzes[4,‡], Lex Nederbragt[5,‡], Tracy K. Teal[6,‡]

General principles and workflow suggestions

- Clear documentation of data, methods, code and results

  Anyone should be able to understand the logic, flow, project organisation and details of why and how things were done.

  - Mainly for yourself (!)
  - For your team and collaborators
  - For reusability at a later phase
  - Why, not just how

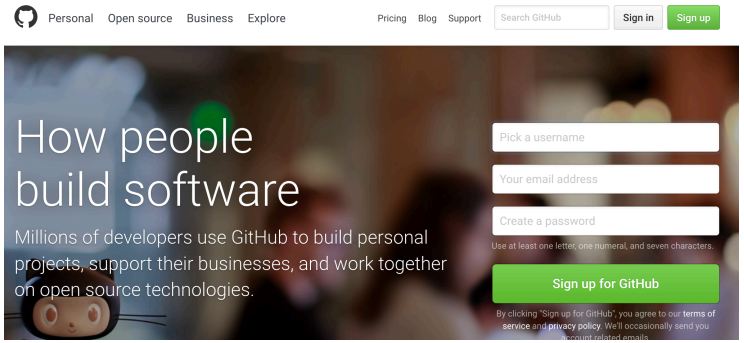
- Version control
- Freezing and packaging
  - Milestones and code
- Testing

General principles and workflow suggestions

- Documents:
  - Questions, hypotheses, data analysis plan
- Data (raw/original):
  - Without any processing and backed up
- Analysis steps/code:
  - Documented, reproducible, version controlled, backed up
- Results:
  - Clear methods and interpretation
- External software and computational environment:
  - Operating system, versions, dependencies, parameters used

Concepts and Tools: “big data”

- Open, community driven tools
- Large collaborative projects
- Many options available



Ruffus

Concepts and Tools

- Automation with flexibility:
  - Build analysis pipelines (chain tools and custom code to answer your question)
    - Production pipeline (e.g. any QC series of steps)
    - Question specific pipeline
  - Parameters changed, further exploration, new project with similar data?

Press the green button

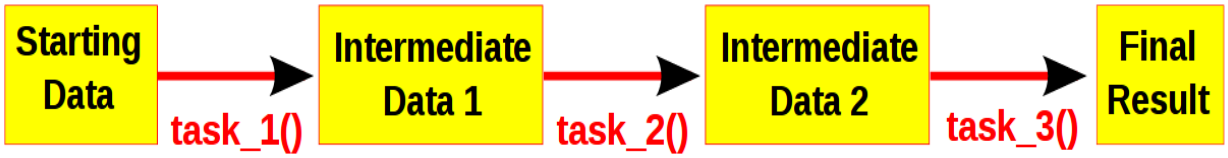
(famous last words)

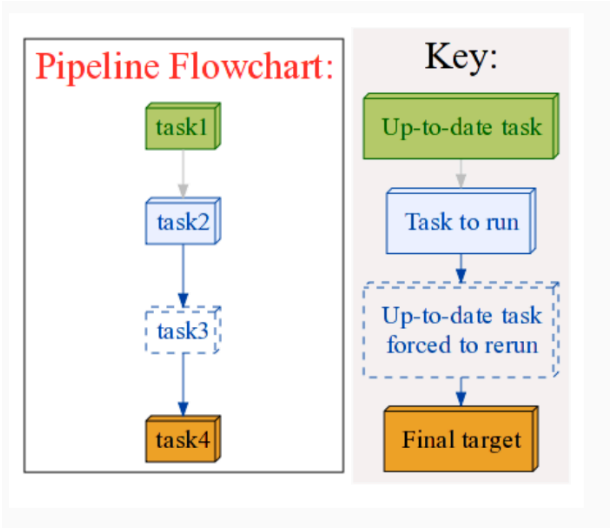
- Python as a general framework
- Packaging as an approach

Computational pipelines

Ruffus

python™





Just one example as several tools for pipelining are available.

Genotype QC → Imputation → GWAS → Variant annotation

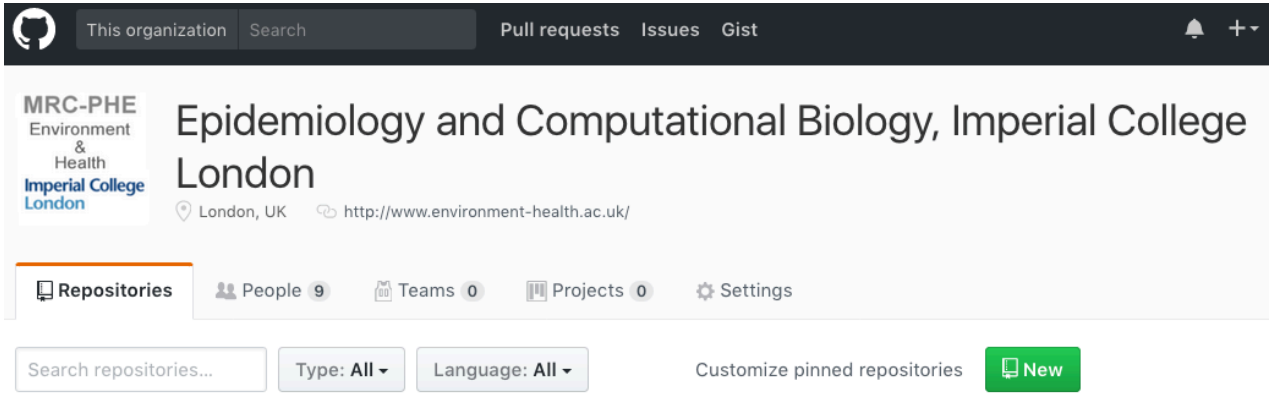
Multiple platforms, cohorts, questions

Work at EBS: EpiCompBio

- Reproducibility principles
- Adopting, implementing, developing best practice
- Reducing barriers to achieve this:

  Tools + Science + Learning

- Collaborative bioinformatics workflows and methods



Work at EBS

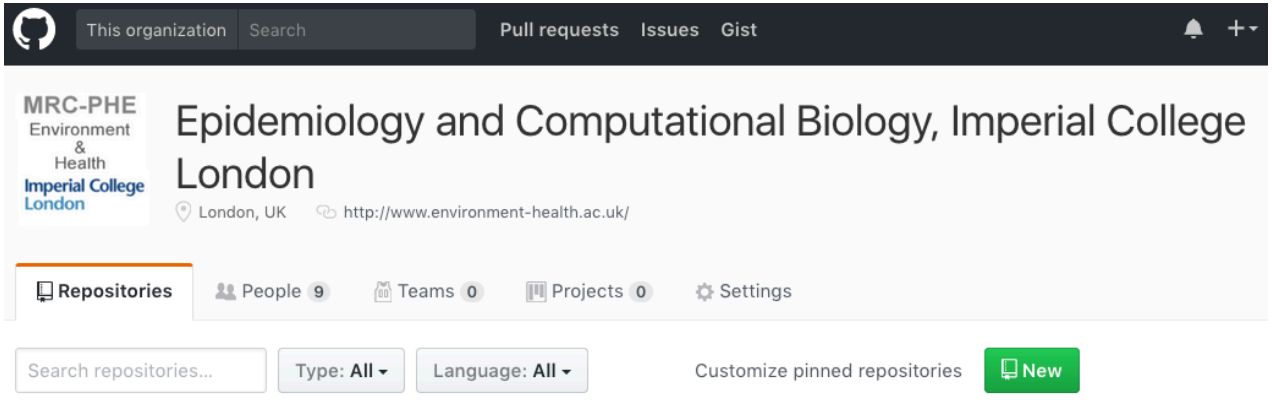
- Currently directed at “power” users, familiarity with:
  - Unix systems and command line interface (grep, sed, awk, bash …)
  - non-GUI text editor
  - Programming language (Python, Perl, Julia, etc.)
  - HPC use
  - Specific tools used: git, Ruffus, Make, Travis CI, Read the Docs,
  - R, Matlab, SQL …
  - Field specific knowledge

Principles are what matter however:

- Questions, people and project dictate tools and workflows
- Implement in whichever way is best

EpiCompBio

- Unix philosophy, Python-centric
- JISCMail: email communication and history trail
- Version control and collaboration:
  - EpiCompBio at GitHub

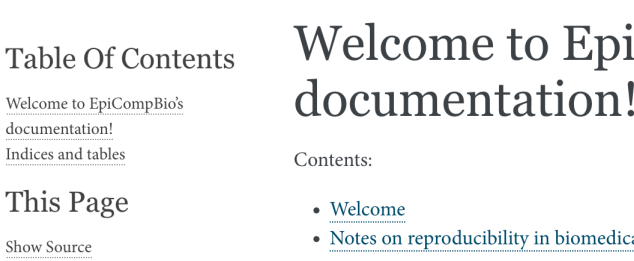


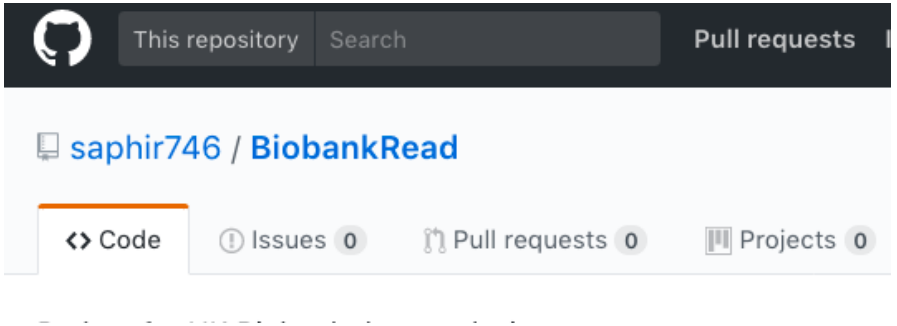
Concepts and Tools: Python as glue

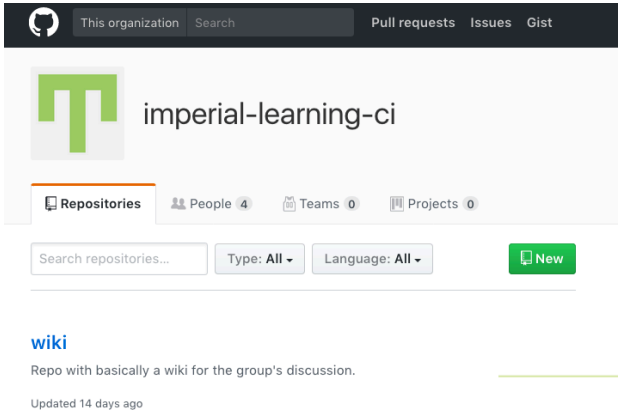
- Python programming, packaging and pipelining
- restructuredText and Sphinx for reporting
- Travis and tox for testing
- Pip, Conda and Docker for dependencies and environment management
- GitHub for version control
- ReadtheDocs for documentation (of software)
- Zenodo (for archiving code and generating a DOI)

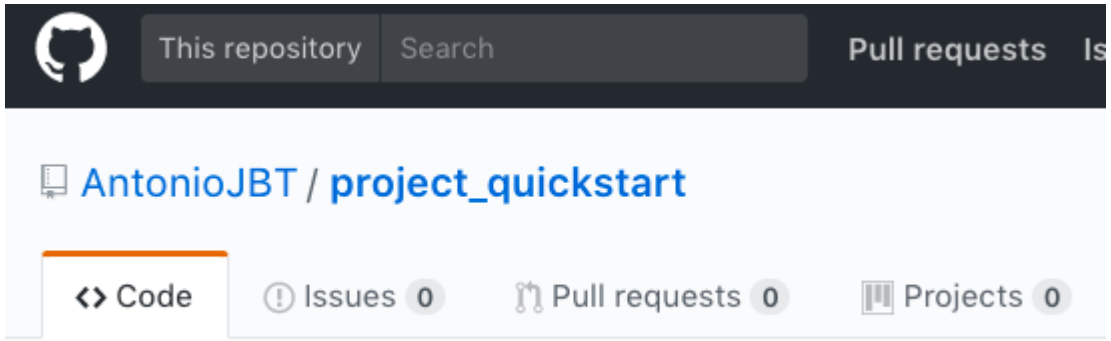
Imperial HPC + UK MEDBIO support: compute, back-up, queries, installation.

EpiCompBio: repositories and tools









Learning: steep curve but worth it

- Postgrad/early post-doc? Now may be the time
- Many free online, self-directed learning options available
  - MOOCS, language tutorials
  - Stack Overflow and many communities
- The person next to you, peers, colleagues, etc.
- Ask/Organise short courses:
  - Software Carpentry, Data Carpentry
  - Start a learning group:
    - Get a book, tackle problems together

Summary

- Reproducibility is a key step to high quality science
- It is becoming (?) harder:
  - Difficult questions, big data, big teams
- Concepts are the same, practice is changing
- There are many tools available:
  - Not just for ‘power’ users
  - It’s OK if you don’t use any of them. You can achieve the same, you probably already do. Principles are what matter.
  - Think abour your current practice, reinforce positives, learn new skills, adopt new tools where needed.
- There are many learning opportunities available
- Go open (!): data, code, papers

Acknowledgements

- Paul Elliott, David Mosen-Ansorena, Deborah Schneider-Luftman, Evangelos Evangelou, Abbas Dehghan, Ioanna Tzoulaki

MRC-PHE
Centre for Environment & Health

Public Health England | MRC Medical Research Council | Imperial College London | King's College London

ukMedSeq

University of Oxford

CGAT
COMPUTATIONAL GENOMICS ANALYSIS AND TRAINING

Thanks!

- Comments, questions, suggestions?
- Contribute at EpiCompBio:
  - Create a GitHub account and email me to add you as a member at: a.berlanga@imperial.ac.uk
- Join an internal discussion list (email me to add you)
- Do your own thing! In your own account, own group, etc.

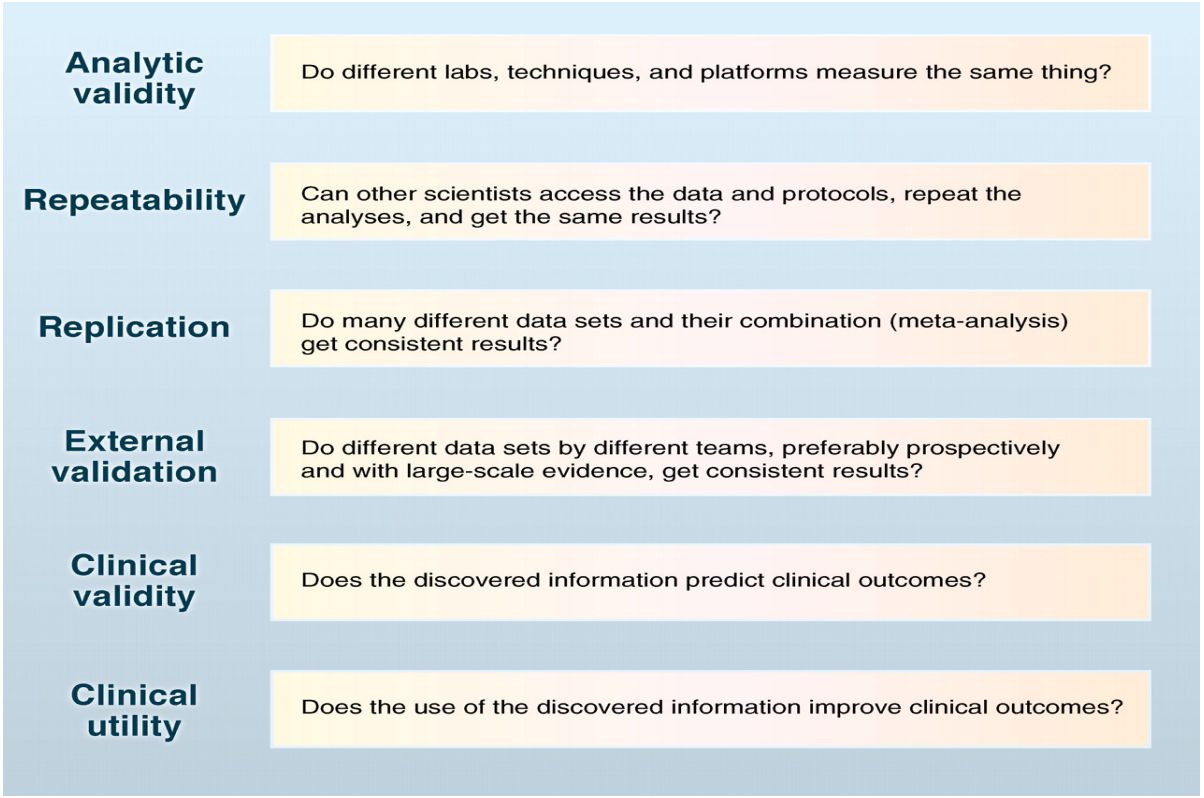
Additional slides

Reducing barriers to reproducibility

Demo project_quickstart

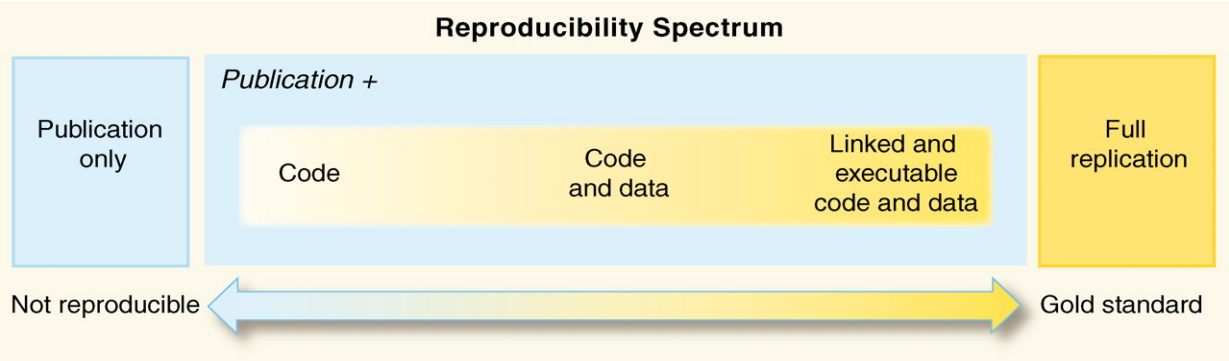
```
SUPER/
|-- code
|   |-- COPYING
|   |-- Dockerfile
|   |-- KNOWN_BUGS
|   |-- LICENSE
|   |-- MANIFEST.in
|   |-- README.rst
|   |-- SUPER
|   |   |-- SUPER.R
|   |   |-- SUPER.py
|   |   |-- module_SUPER.py
|   |   `-- version.py
|   |-- SUPER.ini
|   |-- THANKS.txt
|   |-- TODO.rst
|   |-- docs
|   |   |-- conf.py
|   |   |-- getting_started.rst
|   |   `-- index.rst
|   |-- external_dependencies.txt
|   |-- requirements.txt
|   |-- run_travis_tests.sh
|   |-- setup.cfg
|   |-- setup.py
|   |-- tests
|   |   |-- file_to_compare_against.gold_std
|   |   |-- sample_data.data
|   |   |-- test_style.py
|   |   `-- tests.yaml
|   |-- tox.ini
|   `-- version.py
|-- data
|   |-- external
|   |-- processed
|   `-- raw
|-- manuscript
|   |-- DAP_SUPER.rst
|   |-- cover_letter_SUPER.rst
|   |-- file_with_include.rst
|   |-- lab_notebook_SUPER.rst
|   |-- manuscript_SUPER.rst
|   `-- substitution_vars.rst
`-- results_1

10 directories, 33 files
```

Reproducibility and validation in computational workflows



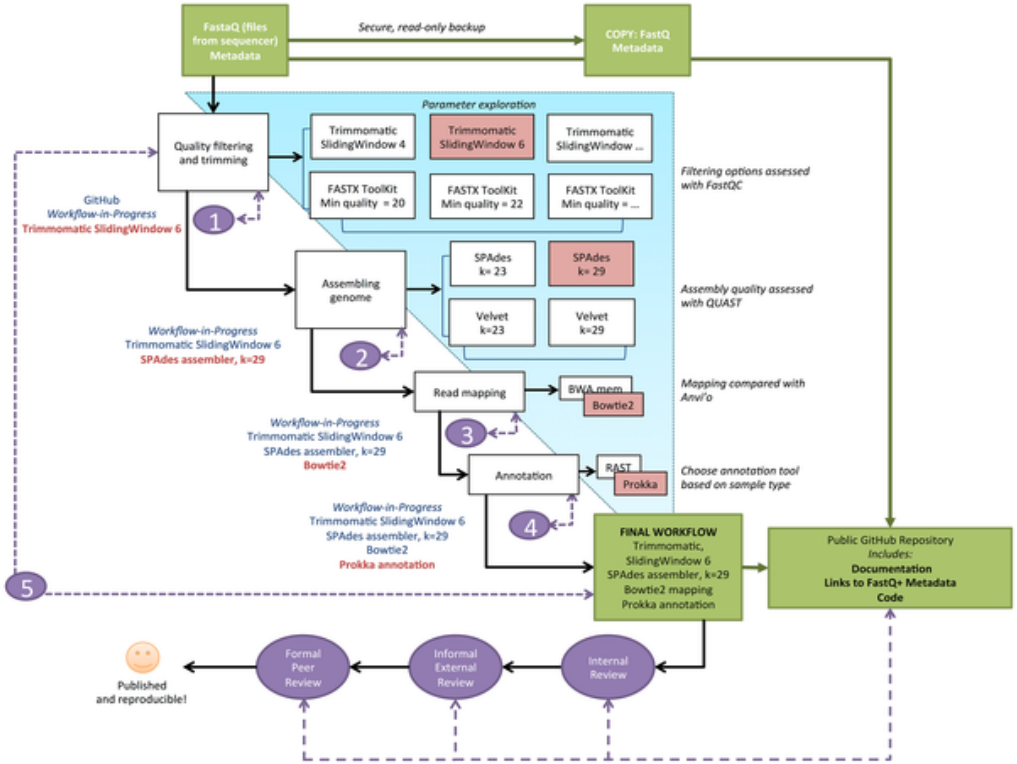
John P. A. Ioannidis, and Muin J. Khoury Science 2011;334:1230-1232

Reproducibility and validation in computational workflows



Roger D. Peng Science 2011;334:1226-1227

Reproducibility and validation in computational workflows



Shade A, Teal TK (2015) Computing Workflows for Biologists: A Roadmap. PLoS Biol 13(11): e1002303. doi:10.1371/journal.pbio.1002303
http://journals.plos.org/plosbiology/article?id=info:doi/10.1371/journal.pbio.1002303

Reproducibility and validation in computational workflows

Type	On what?	By whom?
Self	• Every parameterized step in a workflow • Final, complete workflow • Final batch script	• User(s) who develops the analysis workflow
Internal	• Final, complete workflow • Final batch script	• At least one colleague in the research group • Research group leader/principal investigator (PI)
External	• Final, complete workflow • Final batch script	• Crowdsourcing (e.g., GitHub/BitBucket/R community) • Informal review (ArchiveX, PeerJ PrePrint) • Reviewers and editor of a submitted manuscript

doi:10.1371/journal.pbio.1002303.t002

Shade A, Teal TK (2015) Computing Workflows for Biologists: A Roadmap. PLoS Biol 13(11): e1002303. doi:10.1371/journal.pbio.1002303
http://journals.plos.org/plosbiology/article?id=info:doi/10.1371/journal.pbio.1002303