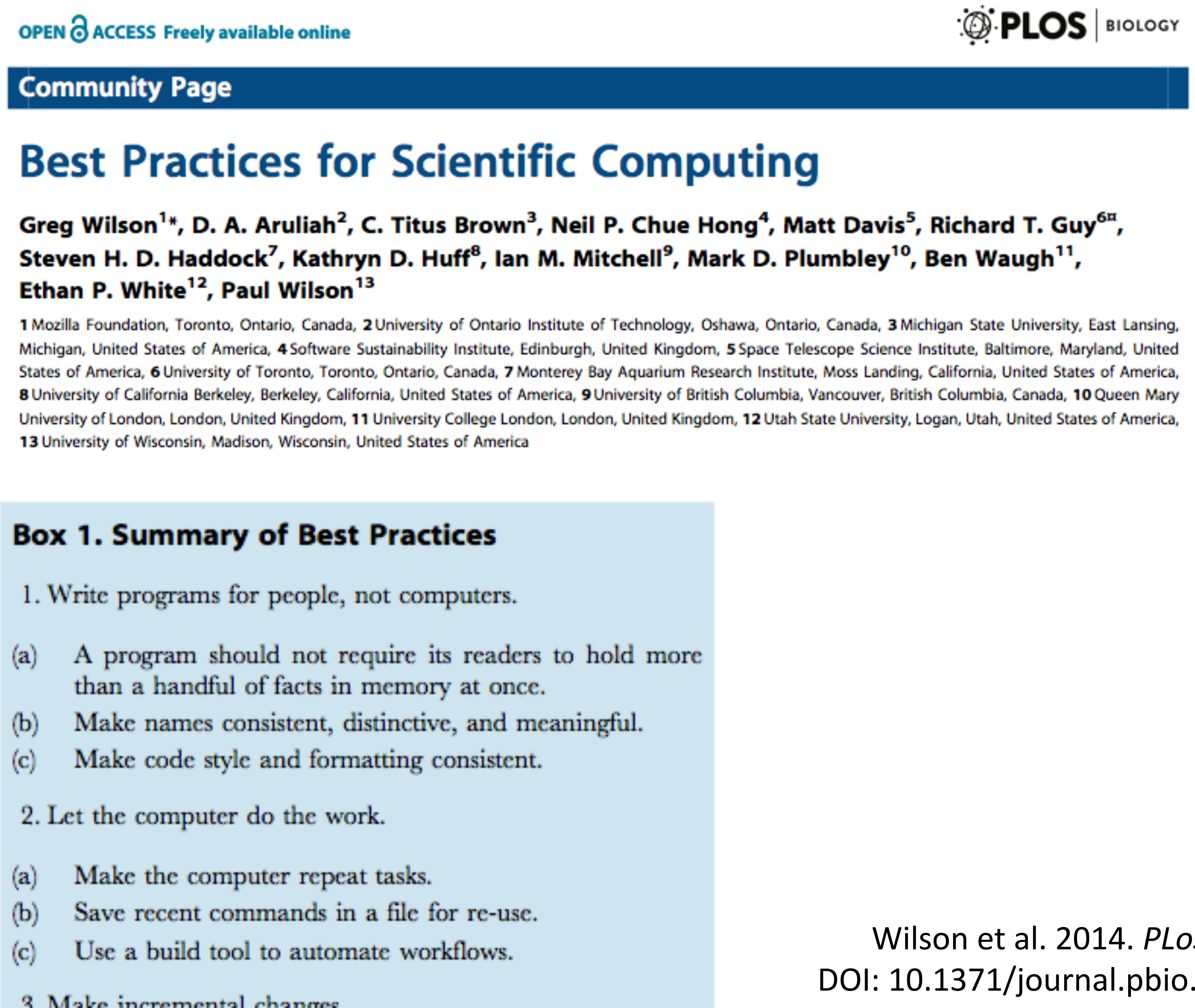
Data Management workflow with Rstudio & git

Lind & Cariveau

Data Management for Biologists

Spring 2018



OPEN ACCESS Freely available online

PLOS | BIOLOGY

Community Page

Best Practices for Scientific Computing

Greg Wilson[1]*, D. A. Aruliah[2], C. Titus Brown[3], Neil P. Chue Hong[4], Matt Davis[5], Richard T. Guy[6¤], Steven H. D. Haddock[7], Kathryn D. Huff[8], Ian M. Mitchell[9], Mark D. Plumbley[10], Ben Waugh[11], Ethan P. White[12], Paul Wilson[13]

1 Mozilla Foundation, Toronto, Ontario, Canada, **2** University of Ontario Institute of Technology, Oshawa, Ontario, Canada, **3** Michigan State University, East Lansing, Michigan, United States of America, **4** Software Sustainability Institute, Edinburgh, United Kingdom, **5** Space Telescope Science Institute, Baltimore, Maryland, United States of America, **6** University of Toronto, Toronto, Ontario, Canada, **7** Monterey Bay Aquarium Research Institute, Moss Landing, California, United States of America, **8** University of California Berkeley, Berkeley, California, United States of America, **9** University of British Columbia, Vancouver, British Columbia, Canada, **10** Queen Mary University of London, London, United Kingdom, **11** University College London, London, United Kingdom, **12** Utah State University, Logan, Utah, United States of America, **13** University of Wisconsin, Madison, Wisconsin, United States of America

Box 1. Summary of Best Practices

1. Write programs for people, not computers.

(a) A program should not require its readers to hold more than a handful of facts in memory at once.
(b) Make names consistent, distinctive, and meaningful.
(c) Make code style and formatting consistent.

2. Let the computer do the work.

(a) Make the computer repeat tasks.
(b) Save recent commands in a file for re-use.
(c) Use a build tool to automate workflows.

3. Make incremental changes

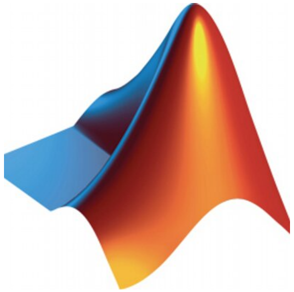
Wilson et al. 2014. *PLoS Biology*
DOI: 10.1371/journal.pbio.1001745

Why `code`?

- **Advantages**:
  - Raw data remain unmodified
  - Can modify repeatedly with easy “undo”
  - Provides record of manipulation
    - good for others
    - *great* for originator (information entropy strikes)

- **Disadvantages**:
  - scripting == programming
  - not all scripting languages (e.g. *R*) are good at big data manipulation and aggregation

Scripting/programming



Why `R`?

- **Advantages**:
  - open-source
  - ecological standard
  - built for visualization & analysis
  - *other people’s code & packages*
  - Integrated development environment (RStudio)

- **Disadvantages**:
  - open-source
  - not always memory-efficient

To R Studio…