Good Coding Practices

Rodrigo Leal Cervantes

University of Oxford

18 of October 2018

Introduction

See the paper: Wilson et al., *Best practices for Scientific Computing*, PLoS biology **12**(1), (2014).

- Scientists typically spend 30% of their time developing software, but...
- ...90% or more are primarily self-taught!
- Software is another type of experimental apparatus, and it should be checked and used as carefully as any physical apparatus.
- We are unaware of tools and practices that would allow us to write code that is more reliable and easier to mantain.

Summary of 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.

Summary of 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.

Summary of 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.