Manuscript Title

This manuscript (permalink) was automatically generated from computer-aided-biotech/better-cb@f5ca2a3 on November 12, 2020.

Authors

- **Benjamín J. Sánchez**
  0000-0001-6093-4110 · BenjaSanchez · BenjaSanchez
  Department of Bioengineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark

- **Daniela C. Soto**
  0000-0002-6292-655X · dcsoto
  Genome Center, MIND Institute, and Department of Biochemistry & Molecular Medicine, Davis, CA 95616,USA

Abstract

This should be the abstract.

Introduction

- Importance of computation in biological research and scope.
- Importance to reproducibility in computational biology.
- Mentioning previous literature and some of their major takeaways and topics not fully covered in here (what kind of languages to learn, advanced text editors/IDEs,
- The computational biology best-practices continuum: reproducibility at the personal level, sharing our research with others and over time.
- Addressing the audience: computational biologists researchers – data analysts, workflow designers, software developers, mathematical modelers, etc.
- Structure of this manuscript: mention Figure 1 and Figure 2.
  - **Figure 1:** Funnel structure for better computational biology.
  - **Figure 2:** Types of computational biology projects per level.

Level 1: Personal Research

Goal: How should you manage your computational biology project?

Topics: - Literate programming: R Markdown, Jupyter Notebooks - Version control: Git / GitHub (commits) - Software versioning and environment managers (Conda, python-env) - Modularize, snippets - Coding style: variable naming, linter, pep8, commenting - Programming practices: paradigms (object-oriented, procedural), assertions, pair-programming

Level 2: Collaboration

Goal: How to allow your collaborators to reproduce and interact with your research/software?

Topics: - Sharing notebooks: R Markdowns and Jupyter Notebooks (Binder, Google CoLab) - Sharing apps: Shiny apps, Dashboard. - GitHub (branching, pull requests) - Workflow automation: Snakemake (NextFlow, Make, Bash script) - Sharing data and metadata

Level 3: Community

Goal: How to develop and maintain a computational biology project with community feedback over time?

Topics: - GitHub releases and semantic versioning - Git Flow, GitHub Issues - Continuous integration and unit tests - Dependencies per user: pip-tools - Sharing software as Python packages, Conda/Bioconda or containers (Docker, Singularity) - Include license (MIT) and DOIs - Documentation: read the docs.

Case studies

- Example of computational biology project 1: RNA-seq analysis (workflow)

- Example of computational biology project 2: Genome-scale metabolic model (systems biology project)
- Example of computational biology project 3: Software development (computational tool)

Conclusion

Pending

Acknowledgments

Pending

References