

Genomes Workflow - LBCM

00 - Introduction

General guidelines

Tuesday 15th July, 2025

Contents

1	Overview	3
1.1	Module Objectives	3
2	Background	3
2.1	General Context	3
3	Workflow & Methods	3
3.1	The repository	3
3.2	Available scripts	4
3.2.1	Global Scripts	4
3.2.2	Scripts for General Users	4
3.3	General pipeline structure	4
3.4	Modules organization	4
4	Best Practices	4

1 Overview

1.1 Module Objectives

This module covers:

Learning goals

- Repository organization.
- Available scripts.
- General pipeline structure.
- Modules organization.

2 Background

2.1 General Context

Key concepts:

Why this matters

- The integration between biological knowledge and IT or bioinformatics tools can be harsh.
- The presence and interference of multiple key concepts of each area creates a big wall on newcomers.
- Explain key features and provide support in a organized way can help to circumvent this initial barrier.

3 Workflow & Methods

3.1 The repository

The repository is structured aiming to facilitate user operation and base codes and modules maintenance. The main separation happens in the global folder, that stores templates, bibliography files and scripts for project managers usage.

Key parameters

```
genomes_workflow_lbcm_ufrgs
|-- CHANGELOG.md
|-- README.md
|-- global
|   |-- bibliography
|   |-- scripts
|   '-- templates
|-- modules
|   '-- 00_introduction
|-- requirements.txt
|-- scripts
|   |-- README.md
|   |-- build_course.py
|   |-- config.yaml
|   |-- interactive_builder.py
|   |-- main.py
|   |-- requirements.txt
|   '-- utils
'-- setup.py
```

3.2 Available scripts

3.2.1 Global Scripts

- **main.py:** Terminal choice selection to execute the possible tools scripts, all related to repository creation and maintenance.
- **tools/create_module.py:** Creates a new module under the **modules/** folder, based on the available template file named **main_template.tex** and **preamble_template.tex**. Also creates a base **README.md** file and an **images** folder. Receives the name of the module to be created (folder name, supposed to be short and not formatted), the title as intended to be displayed to the final user and a subtitle, if needed.
- **tools/set_vim_config.py:** Changes the current **.vimrc** file to a custom one, that can be found on the **global/templates/** folder. Includes **UltiSnips** and **VimTeX** shortcuts and **QOL** settings for module production.

3.2.2 Scripts for General Users

- **build_course.py:** General command line interface for course building. Accepts single modules, complete courses or custom module selection. Imports configuration from the **config.yaml** file and receives user arguments depending on final intention.
- **interactive_builder.py:** Similar in function to the script above, although it provides a simple interface to facilitate user usage and interaction.

3.3 General pipeline structure

WORK IN DEVELOPMENT

3.4 Modules organization

00 - Introduction - Similar to the main **README.md** file, this module aims to give a general perception on the repository and what composes it. Also provides some general explanation on scripts and modules currently implemented.

01 - General guidelines and workflow organization - This module presents a short introduction to Linux logic and terminal usage, as well as conda environment creation and logic. Lastly approaches some good practices on workflow organization.

4 Best Practices

- **Data backup:** Always keep original data copies, especially when working on sequenced data.
- **Version control:** Track analysis versions and parameters if working on the repository construction and maintenance.
- **Documentation:** Record all analysis steps and decisions if working directly on the project.
- **Reproducibility:** Use consistent environments, seeds and terminology, referencing external bibliography when necessary.