

## **Bio-3027 / Bio-8027 Assignment Guidelines**

### ***What's this about?***

Learning programming is best done by practicing on real-world examples relevant to you. The goal of this assignment is therefore to get you to combine and demonstrate what you have learned so far about Python and scientific programming overall.

**Objective:** You will program a short pipeline to perform an analysis relevant to your project, whether it be a MSc. or PhD thesis, or another scientific project. This assignment serves as your exam for this course, and the grade will be Pass/Fail.

This means you have a lot of freedom to design and construct this pipeline - with respect to the code format (e.g. script or notebook, or combination of these) or the packages you use (e.g. pandas, numpy, new packages not covered in the course that you found, etc.), and of course freedom to choose what your pipeline does. So, you'll have to do quite some thinking before you even start coding. Come find me or Ines to talk through your assignment ideas, we can help you through the thinking process.

There are a few requirements for this assignment, followed by guidelines (aka recommendations to help you secure that Pass). We will also go through these in class.

### **Hard requirements:**

Before you submit your assignment, make sure:

1. Your code runs. It should run from start to finish smoothly, with no errors that cause early exits.
2. Your code is documented. Make sure you have a header for your script, or headings for sections of your notebook. Give your functions meaningful docstrings, and use comments to clarify your code.
3. The pipeline should be described. Describe the objective and design of your pipeline in the script/notebook directly or as a separate document like a README.

### **Guidelines:**

Depending on the nature of your project, your pipeline could include some or all of following:

- Data handling: file reading, file writing, data parsing, working with data structures
- Variety of data structures: show that you have mastery of data types and structures (when to use them, how to use them)
- Modular design: break up your code into logical chunks, use functions to perform actions such that you don't have to repeat yourself, minimize hard coding
- Use of data analysis packages: show you know how to import, read up on, and use packages, can be the ones we cover in class, or ones you've found that are more suited to your data and project
- Summary of results: either through visual plots or summarising charts, depending on the goal of the analysis