HowTo

Exercise Material
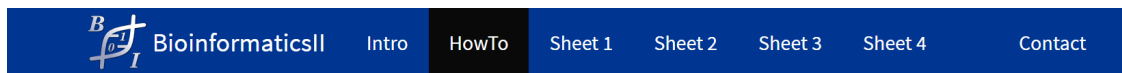
All exercise material can be accessed via this webpage.

Exercises are provided as a practical addition to the lectures and will help you to better understand the topics of the lecture.

Results will be discussed in the weekly exercise sessions.

Setup

You can access individual exercise sheets using the navigation bar.

All exercise sheets are made for self-learning and provide questions, hints and a master-solution.

There are no deadlines when exercises should be finished. Keep in mind though that exercise discussions will be held every week for a certain exercise sheet. We advise you to do one exercise-sheet every week to not lose track of the discussions.

A detailed schedule is available on our Ilias page.

Programming Tasks

Some exercise-sheets have programming tasks.

Note

We will explain how to do the programming tasks during the first exercise session

Whenever a programming task is available you will be redirected to our Github Classroom page:

Exercise 3 - Programming assignment

Programming assignments are available via Github Classroom and contain automatic tests.

We recommend doing these assignments as they will help you further your understanding of this topic.

Access the Github Classroom link: Programming Assignment: Sheet 01.

In Github Classroom we provide automated testing for solutions written in python. You are of course welcome to use whatever language you prefer, but the automatic tests will not work for other languages.

The link provided will lead you to the specific assignment.

Bioinformatics II Exercises

Accept the assignment —

Exercise sheet 01: Hidden Markov models

Once you accept this assignment, you will be granted access to the `exercise-sheet-01-hidden-markov-models-RickGelhausen` repository in the Bioinformatics2-teaching organization on GitHub.

Accept this assignment

Once you accept the prompt shown on Github, a custom Github repository will be created for yourself, with your Github Name as a suffix.

You're ready to go!

You accepted the assignment, **Exercise sheet 01: Hidden Markov models**.

Your assignment repository has been created:

https://github.com/Bioinformatics2-teaching/exercise-sheet-01-hidden-markov-models-RickGelhausen

We've configured the repository associated with this assignment (update).

Note: You may receive an email invitation to join Bioinformatics2-teaching on your behalf. No further action is necessary.

If you follow this link you will be able to see your personal Github repository. This will contain all the necessary files and descriptions to complete the programming task. The task description will be given in the README.md which is also visible below the source code.

The important file for you will be the `exercise-sheet.py` file. It contains skeletons for all the functions you need to write with additional explanations.

You can either edit this file directly via Github using the edit button, or clone the repository to your PC and edit it with your preferred IDE.

Recommended good practices

For this course, you will mainly need the basic `git clone`, `git add`, `git commit`, `git push` and `git pull` commands.

Note

Here we have included some best practices helping you solve the exercises as efficiently as possible. For demonstration purposes we will use the first exercise sheet.

First, clone the assignment repository.

git clone git@github.com:Bioinformatics-teaching/exercise-sheet-01-biology-basics-{userID}.git

Now you should have a local copy of your git repository. Within the directory you will find a file called `exercise_sheet1.py`. It contains the skeleton functions of the Programming assignment. Open it with your favorite text editor or IDE and fill in the skeleton functions such that they return the expected result.

You can now check the status of your repository again via:

git status

It will show you that you have `Changes not staged for commit`, namely the modified `exercise_sheet1.py`. The following commands will add the changes and commit them.

git add exercise_sheet1.py

git commit -m "solved sheet"

Now that you set up everything locally you can push your changes to GitHub, thus triggering the **Autograding**

git push origin main

Note

If you want to test locally whether you solved the Programming Task correctly you can use `pytest`. Just install it via:

pip install pytest

`cd` into the directory where the `exercise_sheet1.py` lives and call:

pytest

Once you have solved the programming task, you can push your changes to the repository, which will trigger the automatic testing of your code.

In the following video it is shown how to check if the tests succeeded

As you can see Programming Tasks a) and b) were solved perfectly. However the tests for part c) failed. In such a case you would have to check the part c) again and fix it. In some cases the autograding will also print hints about what was tested and how your output differed from the expected result.

Warning

Autograding might contain errors. If you are sure that you solved the task correctly but the autograding fails, let us know either in the exercise session or in the ILIAS forum.