

# Practical Course Computational Workflows - 2025 -

Mark Polster and Julian Flesch

# Introduction to Computational Workflows

Prof. Sven Nahnsen

# Course Information

# Attendance

Attendance is mandatory!


We start at 09:15 every morning (lunch-break between 11:45 and 12:45)

There might be interfering meetings, that require our room.  
We will adapt our breaks to match that, where possible

Friday 03.10. is public holiday: No course day

# Structure of the course

Week 1: Analyzing data using nf-core pipelines / Intro to Nextflow



Week 2: Pipeline project



# Contact

In person during the course

via Mail:

[julian.flesch@uni-tuebingen.de](mailto:julian.flesch@uni-tuebingen.de)  
[mark.polster@uni-tuebingen.de](mailto:mark.polster@uni-tuebingen.de)

**always to both of us, please!**

# Missing a course date

If you are sick, inform us **before the course starts in the morning**

A doctor's note **must** always be handed in **via mail to Prof. Sven Nahnsen**  
[sven.nahnsen@uni-tuebingen.de](mailto:sven.nahnsen@uni-tuebingen.de)

Unexcused absence will lead to failing the course!

# Grading

Attendance is mandatory to get credits

All hand-ins are done via Github as a **Release**.

**Any work committed after the deadlines will not count!**

Each student is required to hand-in their own **week 1 assignments**

You will hand-in your course project **in teams of two** including:

- 1) A reproducible nextflow/nf-core workflow for RNA-sequencing analysis
- 2) A scientific report of roughly 5 pages



# Git Basics: Using your favorite LLM ...

... What is **git** and what are repositories?

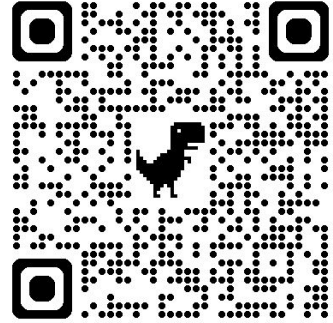
... What is a **fork** on Github?

... What is the difference between  
**origin** and **upstream**?

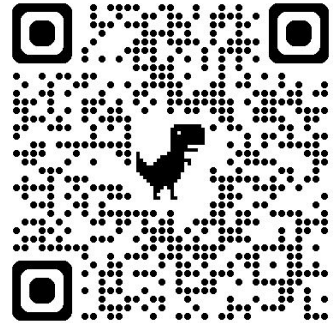


10

min



chat GPT



Gemini

# Your typical Day in the Computational Workflows Course

1. You pull today's updates from our **upstream**
2. Work on the assignments
3. Frequently commit and push to your **own fork**
4. Ask questions, if you have problems

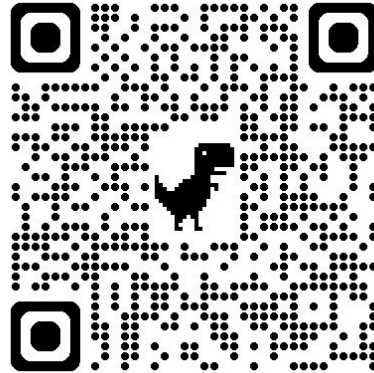


Let's get started!

Demo

# Let's get started!

1. Fork the assignment repository on github:



[github.com/qbic-pipelines/computational-workflows-2025](https://github.com/qbic-pipelines/computational-workflows-2025)

# Let's get started!

1. Fork the assignment repository
2. Set up ssh-key for Github
3. Clone your **personal fork** to your local machine
4. Set the **assignment repository** as upstream
5. Add us to your fork
6. Start working on today's jupyter notebook
7. Commit and push your work frequently!

```
$ mkdir -p ~/.ssh && cd ~/.ssh
$ ssh-keygen -t rsa
$ nano config

$ cd <your-workspace>
$ git clone git@<your-fork>
$ git remote add upstream <qbic>

$ git add <file1> <file2>
$ git commit -m "adds files"
$ git push origin main
```

# Practical Course

# Computational Workflows

Day 2

Mark Polster and Julian Flesch

# Upfront Information

Please add us to your forks on Github!

@JulianFlesch      @mapo9

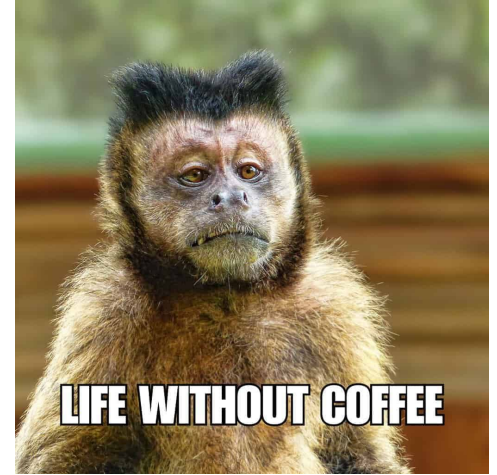
The screenshot shows the GitHub interface for a repository named 'computational-workflows-2025-prepared' owned by 'JulianFlesch'. The top navigation bar includes links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Settings' tab is highlighted with a red arrow labeled '1'. Below the navigation bar, the 'General' settings tab is selected. On the left sidebar, under the 'Access' section, the 'Collaborators' link is highlighted with a red arrow labeled '2'. The main content area shows the 'Repository name' as 'computational-workflows-2025-pr' with a 'Rename' button, and a checkbox for 'Template repository' which is currently unchecked.

# Upfront Information

Help yourself to coffee, if you want  
(list at our desk)

Push your work to Github regularly

Lunch break today from 12:00 to 13:00



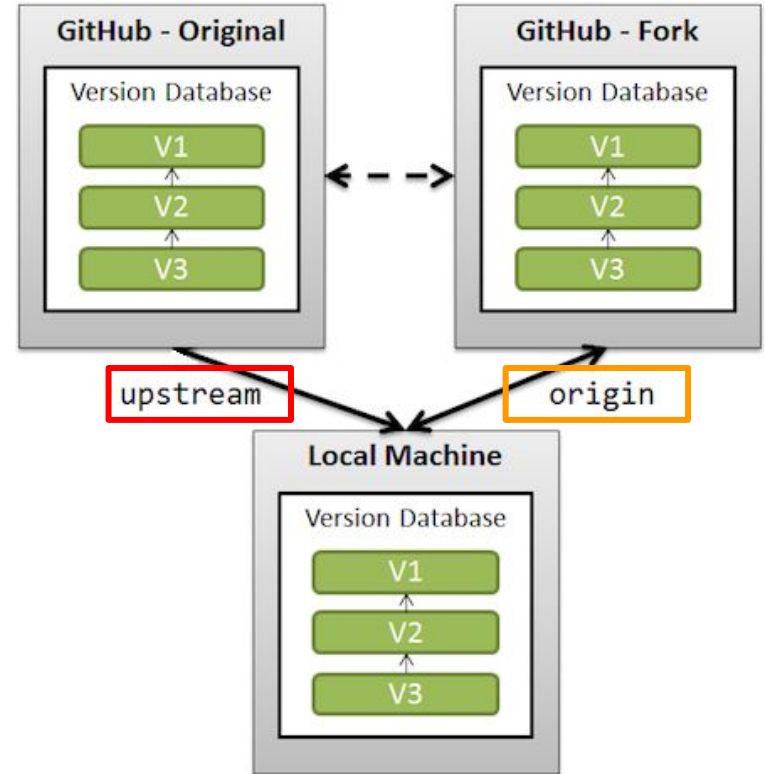


# Quick recap: Github Fork and Upstream

We update only the **upstream**

You work on **your fork**, keeping local and remote in sync and pulling in updates

(if we do not change files you worked on, there should be no *merge conflicts*)



# Getting your Material for today

**Step 1:** Save your work to **your fork**

```
$ git add <file1> <file2>

$ git commit -m "adds files"

$ git push
```

**Step 2:** Get today's notebook **from upstream**

```
# pull from upstream

$ git pull upstream main

# To leave vi:
# ":" -> "q" -> "w" -> "enter"
```