Contents

[1 Reproducible research: organizing your project 1](#_Toc79223242)

[1.1 Learn GitHub/Git 1](#_Toc79223243)

[1.2 Using GitHub and Git 1](#_Toc79223244)

[1.2.1 GitHub 1](#_Toc79223245)

[1.3 File System Structure (FSS) 3](#_Toc79223246)

[1.4 Notes 4](#_Toc79223247)

[1.5 Git add vs commit 6](#_Toc79223248)

# Reproducible research: organizing your project

The Bioinformatics Laboratory is in the process of changing its way of working to achieve a high level of organisation and reproducibility for all research and support projects. The use of GitHub and the File System Structure (FSS) are two interlinked approaches that we use in this context.

## Learn GitHub/Git

* Learn Github in 20 Minutes
  + Video: [https://www.youtube.com/watch?v=nhNq2kIvi9s](https://eur04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DnhNq2kIvi9s&data=04%7C01%7Ca.h.vankampen%40amsterdamumc.nl%7Cd96ee4a61ab84549ec0908d8c13bf687%7C68dfab1a11bb4cc6beb528d756984fb6%7C0%7C0%7C637471810587099881%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=6kKHK%2BEG3ndg9pdaQ9gMKsJwVzxWnynI85mcO1PSR1M%3D&reserved=0)
  + Notes and more information: [https://www.notion.so/Introduction-to-GitHub-202af6f64bbd4299b15f238dcd09d2a7](https://eur04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.notion.so%2FIntroduction-to-GitHub-202af6f64bbd4299b15f238dcd09d2a7&data=04%7C01%7Ca.h.vankampen%40amsterdamumc.nl%7Cd96ee4a61ab84549ec0908d8c13bf687%7C68dfab1a11bb4cc6beb528d756984fb6%7C0%7C0%7C637471810587109836%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=CAA2GHpC8NpWmAV7GWFALPcMK5dzVeZQX7W%2Bl5nssvE%3D&reserved=0)
* Learn Git In 15 Minutes
  + Video: [https://www.youtube.com/watch?v=USjZcfj8yxE](https://eur04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DUSjZcfj8yxE&data=04%7C01%7Ca.h.vankampen%40amsterdamumc.nl%7Cd96ee4a61ab84549ec0908d8c13bf687%7C68dfab1a11bb4cc6beb528d756984fb6%7C0%7C0%7C637471810587089933%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=c2AXLVG0D%2FW0iGK%2FwabzBQCEAtqYCnBENl1ppQAIPkM%3D&reserved=0)
  + Notes and more information: [https://www.notion.so/Introduction-to-Git-ac396a0697704709a12b6a0e545db049](https://eur04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.notion.so%2FIntroduction-to-Git-ac396a0697704709a12b6a0e545db049&data=04%7C01%7Ca.h.vankampen%40amsterdamumc.nl%7Cd96ee4a61ab84549ec0908d8c13bf687%7C68dfab1a11bb4cc6beb528d756984fb6%7C0%7C0%7C637471810587099881%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=aeaNib9RHxxoSG7Xa%2FrkbOgywT8u8QUak%2FTQAIh1SMI%3D&reserved=0)
* GitHub cheat sheet: <https://education.github.com/git-cheat-sheet-education.pdf>

## Using GitHub and Git

### GitHub

GitHub is used to manage your software. You have been invited to the EDS GitHub Repository by your supervisor. This is a organization repository on GitHub.com. Once you accept the invitation you will have access to all software currently being developed by the Bioinformatics Laboratory. Most of the software repositories are private (not public) and you are not allowed to redistribute any of the repositories since they may contain confidential information.

For some more information see also the following files in the Reproducibility repo (<https://github.com/EDS-Bioinformatics-Laboratory/Reproducibility>):

* GitHub\_Help.md
* GitHub\_BioLab-BestePractices.md
* GitHub\_Branch.md
* GitHub\_OpenIssues.md

<https://github.com/EDS-Bioinformatics-Laboratory/Reproducibility>

Now you can start using GitHub as explained below. First create a test repository (repo) directly on the GitHub website: <https://github.com/EDS-Bioinformatics-Laboratory>

* Click on the ‘New’ button
* Start the repo name with TEST\_ (to easily identify test repositories).
* Add a README file
* Add .gitignore (use the R template)
* Choose the GNU General Public License v3.0
* Add a description and keywords.
* Please remove this test repo if you don’t use it any more.

Git bash allows you to access GitHub from your own computer/laptop.

#### Install git bash

1. Download git bash: <https://git-scm.com/downloads>
2. Optionally you can download one of the GUI clients but using git bash (command line) will get you a better understanding of git.

After installing git you can start to use the test repo. There is a lot of documentation and tutorials on the internet.

* <https://guides.github.com/>
* <https://docs.github.com/en>

Here are the steps to get you going:

1. Make a directory on your computer. For example ‘testProject’ and go into this directory
2. Start git bash in this directory. You can do this by starting git bash and then use the command ‘cd’ or (in Windows) you can right click the mouse and select ‘git bash here’.
3. Next you can copy the GitHub repository that you made (eg. TEST\_project) to this folder by using the following command:

git clone https://github.com/EDS-Bioinformatics-Laboratory/TEST\_project

1. You can now go inside the directory TEST\_project. In this directory you will see all the files that are also in your GitHub repo (not that many at the moment). Inspect the contents of the files. Also note that README.md is a MarkDown file (<https://guides.github.com/features/mastering-markdown/>). You will also see a .git directory (note that this might be hidden).
2. Also change the working directory in git bash (cd TEST\_project).
3. Next add a new file (for example an R or Python script) to this folder.
4. You can now synchronize the directory with GitHub with the following commands (git status is not really required but allows you to see what happens to some extend):
   1. git status
   2. git add .
   3. git status
   4. git commit -m “short description” -m “long description”
   5. git status
   6. git push
   7. git status
5. Now have a look at your GitHub repository again and you will see that the file has been added. You will also see a strange string like 35e0928. This is the commit ID and can be used to identify this specific version of the repository.

## File System Structure (FSS)

Below you find the git instructions to use the so-called File System Structure (FSS) with GitHub/git.

On the GitHub website you will also find the Reproducibility repo:

<https://github.com/EDS-Bioinformatics-Laboratory/Reproducibility>

This repo contains some information, presentations, and publications about reproducibility. I encourage you to have a look at the latest version of the PowerPoint ReproducibleResearch\_vx.pptx. This repo also contains the FSS template. The following steps instruct you to make it ready for your own project.

First make a GitHub repository for your project as described above but do not add any files except for the GNU license (your supervisor might have done this for your already). Let’s assume this repo is named RNAseq.

1. Now first clone the complete Reproducibility repository

git clone https://github.com/EDS-Bioinformatics-Laboratory/Reproducibility

1. Go into the directory Reproducibility and remove the .git directory (note that this might be hidden)
2. Go into the directory Reproducibility/ExampleFolderStructure. Here you will find the FSS template: yyyymmdd\_ProjectName. Copy only this directory to an appropriate place on your laptop (your supervisor has actually done this for you in Dropbox).
3. Rename the folder. For example, 20210120\_RNAseq-Analyses. Next go to the Processing directory in the FSS \20210120\_RNAseq-Analyses\Processing. The Processing directory should contain all software (e.g., R scripts, Python) that will be synchronized with the corresponding GitHub repo.
4. Start git bash in the Processing directory and give the following commands (use git status to check):
   1. git init
   2. git remote add origin [URL of repo]
   3. git pull origin main
   4. git add .
   5. git commit -m "short description" -m "long description"
   6. git push -u origin main

The step (f) might give an error because of a ‘mixup’ of branch names (main vs master), then give the following commands:

git rm –cached -r .

git branch main

git checkout main

git merge master

git push origin main

From now on you can use the following git commands to keep your project directory and GitHub synchronized (preferably, you do this at the end of every day).

1. Go to the Processing directory
2. git pull https://github.com/EDS-Bioinformatics-Laboratory/RNAseq (only if there were changes on the GitHub repo that are not yet in your local directory)
3. git add .
4. git commit -m "short description" -m "long description"
5. git push

This is basically it. Don’t forget to rename the gitignore-template.txt to .gitignore and update its content. Also carefully read the files HELP\_FileNameConventions.md and HELP-FileSystemStructure.md in \20210120\_RNAseq-Analyses

GitHub and git allows you to do much more but in the beginning you will not need it. However, the use of branches might be useful. See for example the file GitHub\_Help.md in the Reproducibility repo.

## Notes

**Use of branches**

git branch show all branches of repo

git checkout [branch name] go to specific branch

git push origin [branch-name] push to github (after git add/git commit)

On GitHub: go to the branch and click compare to show differences with master branch

To merge [branch-name] into master:

* Git checkout master
* Git merge [branch-name]

Resolving conflicts: [www.atlassian.com/git/tutorials/using-branches](http://www.atlassian.com/git/tutorials/using-branches)

Tips for collaboration and best practices: [www.atlassian.com/git/tutorials](http://www.atlassian.com/git/tutorials)

Branches for support: don’t merge back into the master branch. Every branch is a project.

**Authorization**

<https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/>

<https://docs.github.com/en/github/authenticating-to-github/creating-a-personal-access-token>

**Setting origin**

git remote add origin [repo name]

**Moving master to main**

https://www.r-bloggers.com/2020/07/5-steps-to-change-github-default-branch-from-master-to-main/

git branch -m master main

git push -u origin main

**Automatic creation of repository from within R/Rstudio**

Aldo Jongejan has a R script that creates a new repository for you when working from R/RStudio. You wouldn’t have to go to the GitHub website first to create a new repository.

https://github.com/aldojongejan/R\_tests.git - zie getFSS.r

Example in R:

# Start new analysis

source("getFSS.r")

startNewRepo("20210119\_FleurPeters\_CLL\_Dasatinib", analysisDir = "Dasatinib")

## Git add vs commit



