2025 June 5th



## Introduction to Git

ROUSSEAU Baptiste SEILER Julien











- What is Git?
- Key concepts
- How to use the Git command?
- The Burger project
  - Create a repository
  - Add files
  - Commit changes
  - Modify files
  - Check status
  - Revert changes
  - Use branches





The **French Institute of Bioinformatics**, or **IFB**, is the national bioinformatics infrastructure that provides support, deploys services, organizes training, and carries out innovative developments for the life sciences communities.

The IFB offers **a national-scale computing infrastructure** in addition to the services provided regionally, known as the Core Cluster.

The Core Cluster is a computing infrastructure co-managed by system administrators and bioinformaticians from 5 IFB platforms.

**4640** cores (9280 threads) 62TB of RAM 10 42 GPUs 2PB of scratch storage 2 PB of project storage (with backup) more than 600 tools



The Core Cluster offers three modes of access to its resources.



#### SSH

core.cluster.france-bioinformatique.fr

A connection node for launching SLURM tasks



The Core Cluster offers three modes of access to its resources.



### **Galaxy** https://usegalaxy.fr

French instance of Galaxy 3,174 tools available 6 thematic subdomains:

- Workflow4Metabolomics
- ProteoRE
- Covid19
- Metabarcoding
- Met4J
- MNHN

TiaaS : Training infrastructure as a Service



The Core Cluster offers three modes of access to its resources.



### **Open OnDemand**

https://ondemand.cluster.france-bioinformatique.fr/

Web portal for accessing computing resources Interactive applications:

- JupyterLab
- RStudio
- XFCE Desktop
- And more to come

#### File manager

SLURM task launching and monitoring

- You will learn to use git by working with it
- You only need a web browser and an Internet connection to follow this training.
- Each time you see this icon, you will have something to do

https://ondemand.cluster.france-bioinformatique.fr/











Username	
username	
Password	User
password	





powered by







.

### This is an interactive training



>	😑	ଲ୍ଟ ୪ 🗾	•	( )	₫ + ©
Apps ▼ Files ▼ Jobs ▼ Clu	usters 🔹 Interactive Apps 👻 🗐 My Interactive Sessions	<th>• 🕜 Help •</th> <th>Logged in as jseiler</th> <th>🕩 Log Out</th>	• 🕜 Help •	Logged in as jseiler	🕩 Log Out
me / My Interactive Sessions /	JupyterLab: Core				
teractive Apps	JupyterLab: Core				
sktops	This app will launch a JupyterLab server inside a SLURM job.				
Desktop: Core	Reservation				
rvers	2522_fairomics	\$			
JupyterLab: Core	Account				
RStudio Server: Core	tp_2522_fairomics_181496	*			
	Partition				
teractive Apps [Sandbox]	fast	\$			
sktops	Number of CPUs				
Desktop	2	0			
	A least one CPU is required for the JupyterLab webserver				
	Amount of memory				
	2G				
	Unit can be specified using the suffix M, G or T (default is megabytes).				
	GPUs				
	No GPU	\$			
	Number of GPUs				
	0				
	Apps - Files - Jobs - Clime         me       / My Interactive Sessions         eractive Apps         ktops         Desktop: Core         vers         JupyterLab: Core         RStudio Server: Core         eractive Apps [Sandbox]         sktops         Desktop	Apps Files Jobs Clusters Interactive Apps   Apps Files JupyterLab: Core   eractive Apps JupyterLab: Core   Besktop: Core   yers 2522_fairomics_181496   Partition   fast   Number of CPUs   2   A least one CPU is required for the JupyterLab webserver   Amount of memory   26   Unit can be specified using the suffix M, G or T (default is megabytes).   GPUs   Number of GPUs   1   1   1   1   26   20   21   22   23   24   25   26   27   28   29   30   30   30	Image: Price of Device Apps   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions   Partition   Studio Server: Core   Partition   fast   fast   * Number of CPUs   2   2   2   2   2   2   2   Core   Partition   fast   fast   * Number of CPUs   2	Piles * Jobs * Clusters * Interactive Apps * PMy Interactive Sessions    Apps * Files * Jobs * Clusters * Interactive Apps * PMy Interactive Sessions   Apps * Files * Jobs * Clusters * Interactive Apps * PMy Interactive Sessions    Partition    eractive Apps [Sandbox]   pesktop   Desktop   Partition   fast   pesktop   Desktop   Desktop   Desktop   Desktop   QupyterLab: Core   Reservation   2522_fairomics_181496   \$ Partition    fast    Desktop   Desktop   Desktop   Desktop                        Partition fast     etaps    Desktop    Partition fast     fast    0    Number of CPUs  Cale on the Specified using the suffix M, 0 or T (default is megabytes). GPUs Number of GPUs Number of GPUs  Number of GPUs<	Image: Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions (*) Develop * @ Help * & Logged in as joeler   Apps * Files * Jobs * Clusters * Interactive Apps * My Interactive Sessions (*) Develop * @ Help * & Logged in as joeler   ractive Apps and the power of My Interactive Sessions / JupyterLab: Core   ractive Apps JupyterLab: Core   ractive Apps [Sandbox] Keservation   wers 2   Desktop: Account   fast •   Number of CPUs 2   2 0   Aleast one CPU Is required for the JupyterLab webserver   Amount of memory   2G   Unit can be specified using the suffix M, G or T (default is megabytes).   CPUs   Number of GPUs   0   0





#### This is an interactive training







.

What is Git?



### Git is

## a command line tool

### a version control system

decentralized and distributed command line (Git) developed in C, Bash and Perl Open Source (GNU GPL 2)

- track changes to a file/folder or a set of files/folder
- navigation in the history of modifications
- sharing of changes

no need for a server

multi-user





Where does Git come from?



Git was invented in 2005 by Linus Torvalds.



## Git is the successor of many similar tools like cvs or subversion.

git means "unpleasant person" (Linus like to name his projects after himself...)





## A few examples

- Follow the steps of modification of a program
- Test a complex change and be able to go back easily
- Working with others on a project
- Invite collaborators on a project
- Contribute to OpenSource projects





#### How to use Git?



- Git is a command line tool
- the command is git
- you can use it from shell terminal







For each project you track with Git, Git maintains a repository at the root of the project in a **.git** folder

The **git** command let you interact with this repository.





## Working directory



Your working folder contains the files and folders that make up your project.

Git can modify these files to update them or present them to you at different versions of the project through its index.

## **Git repository**

.git

Your Git repository contains the entire history of your project. All file versions, all modifications, etc.

This is the **.git** folder at the root of your working folder.







A Git repository will allow you to track the **history of changes** in your project. Each change is first recorded in **an index** (or indexed) to form **a collection of changes**.

This collection of changes is then validated (or committed) in your repository Each commit is a new version or revision in your project history.







To use the latest version of Git, load the command using **module** 

```
$ module load git
$ git --version
git version 2.40.1
```





## In your console, type git

```
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
<command> [<args>]
```

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone	Clone a repository into a new directory
init	Create an empty Git repository or reinitialize an existing one

```
work on the current change (see also: git help everyday)
add Add file contents to the index
mv Move or rename a file, a directory, or a symlink
restore Restore working tree files
rm Remove files from the working tree and from the index
sparse-checkout Initialize and modify the sparse-checkout
```

#### [...]

'git help -a' and 'git help -g' list available subcommands and some concept guides. See 'git help <command>' or 'git help <concept>' to read about a specific subcommand or concept. See 'git help git' for an overview of the system.









The **git** program allows you to run commands to manage your Git repository

\$ git <command> <arguments>





## **Defining your identity**



Your identity will be associated with the changes you make in your repositories It is defined in the file ~/.gitconfig or %USERPROFILE%\.gitconfig

\$ git config --global user.name "Your name"
\$ git config --global user.email your@email
\$ git config --global init.defaultBranch main





To illustrate this training, we will work on a burger recipes project.

Create a **burgers** folder in your home folder



\$ mkdir burgers





Create your repository



Go to your burgers folder and run the **git init** comand

```
$ cd burgers
$ git init
Initialized empty Git repository in /Users/seilerj/burgers/.git
$ ls -a
. . . .git
```







creates a Git repository

If the specified working dir does not exist, it will be created.

Without parameters, the command creates a Git repository for the current folder.







## Let's cook a burger



Create the file **doublecheese.txt** in the burgers folder and write down the list of ingredients to make a double cheese.









#### Add a file to your repository









Add a file to your repository



Index a file in your Git repository

Index the doublecheese.txt file in your repo with the **git add doublecheese.txt** command

\$ git add doublecheese.txt





## Save changes



Validate this modification in order to save it in your repository with the **git commit** command

\$ git commit -m "Birth of the double cheese"
[main (root-commit) bb0188d] Birth of the double cheese
1 file changed, 7 insertions(+)
create mode 100644 doublecheese.txt











•

Modify a file

There is no tomatoes in the double cheese! Correct the file doublecheese.txt









A new burger

Add the file bigmac.txt to your burgers project





steak salad tomatoes onions pickle ketchup mustard







# Where are we now?



## Check what has changed in your project with the **git status** command

















Index the change on the doublecheese.txt file with the **git** add command

\$ git add doublecheese.txt









Check what has changed in your project with the git status command

```
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  modified: doublecheese.txt
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .ipynb_checkpoints/
      bigmac.txt
```




Add a new file to the repo



Add the new file bigmac.txt with the **git** add command

\$ git add bigmac.txt







Check what has changed in your project with the git status command

```
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: bigmac.txt
    modified: doublecheese.txt
```

Untracked files: (use "git add <file>..." to include in what will be committed) .ipynb\_checkpoints/





Validate the changes

## Run the **git commit** command



\$ git commit -m "Add the big mac and correct the double cheese"
[main 299a6b2] Add the big mac and correct the double cheese
2 files changed, 7 insertions(+), 1 deletion(-)
create mode 100644 bigmac.txt











.

Let's introduce an error



Edit the **bigmac.txt** file to add a bad ingredient.











Restore the last valid version of the file with the git checkout command

\$ git checkout bigmac.txt







## Did you say revision number?

Each commit to your repository creates a new revision or version of your project. Each revision is actually **a set of changes**.





Use the git log command to view the revisions of your local repository



\$ git log commit 299a6b210eed54e9f4c164b85ecbcb9ed899e6eb (HEAD -> main) Author: Julien SEILER <seilerj@igbmc.fr> Date: Tue May 16 12:14:35 2023 +0200

Add the big mac and correct the double cheese

commit bb0188df5bf0c3cb3a152e52e22df1249d52e2be
Author: Julien SEILER <seilerj@igbmc.fr>
Date: Tue May 16 11:49:37 2023 +0200

Birth of the double cheese







Revert to the initial version of your project (just the doublecheese.txt) using the git checkout <id rev> command

\$ git checkout bb0188df5bf0c3cb3a152e52e22df1249d52e2be
Note: switching to 'bb0188df5bf0c3cb3a152e52e22df1249d52e2be'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this [...] \$ 1s doublecheese.txt





































Use the command git checkout main

```
$ git checkout main
Previous HEAD position was bb0188d Birth of the double cheese
Switched to branch 'main'
$ ls
bigmac.txt doublecheese.txt
```





















Our fast-food restaurant wants to go **ORGANIC**. We need to change all our recipes!!!









Create a new branch



Create a new organic branch on your repository

\$ git branch organic





Create a new branch

Consult the branches available on your repository

# \$ git branch \* main organic





Change branch



Use the **git** checkout command to switch branches.

## \$ git checkout organic Switched to branch 'organic'









organic





We go organic!



organic steak organic salad organic tomatoes organic onions organic pickles organic ketchup organic mustard

```
$ git commit -a -m "organic mac"
[organic a7a6f6e] organic mac
1 file changed, 7 insertions(+), 7 deletions(-)
```













We go organic!



organic steak organic cheese organic onions organic pickles organic ketchup organic mustard

```
$ git commit -a -m "organic cheese"
[organic 1ed7510] organic cheese
1 file changed, 6 insertions(+), 6 deletions(-)
```













Let's go back to our main branch

Return to the main branch with the git checkout command

\$ git checkout main
Switched to branch 'main'











We can retrieve the changes saved on the organic branch with the **git merge** command

```
$ git merge organic
Updating 299a6b2..1ed7510
Fast-forward
bigmac.txt | 14 ++++++-----
doublecheese.txt | 12 +++++-----
2 files changed, 13 insertions(+), 13 deletions(-)
```













The Git cycle



#### Working directory







git command	Description
init	Creating a repository for a project/folder
add	Indexing of a modification or addition of a file or folder
rm	Deleting a file or folder
mv	Moving a file or folder
status	Visualization of the repository status
diff	Viewing changes between two revisions or between a revision and the current version
checkout	Retrieving a file from the repository
log	Consultation of the list of revisions (commits) registered on the repository





- 1. Delete the bigmac and save the change in your repository
- 2. Add a burger and save the change to your repository
- 3. Restore the bigmac to your working folder









1. Delete the bigmac and save the change in your repository

```
$ git rm bigmac.txt
$ git commit -m "bye bye big mac"
[main 916c075] bye bye big mac
1 file changed, 7 deletions(-)
delete mode 100644 bigmac.txt
```







2. Add a burger and save the change to your repository

\$ git add newburger.txt
\$ git commit -m "add a new burger"
[main 26f9033] add a new burger
1 file changed, 1 insertion(+)
create mode 100644 newburger.txt "







3. Restore the bigmac to your working folder

```
$ git checkout <rev> bigmac.txt
Updated 1 path from 69a5a31
$ git commit -a -m "return of the big mac"
[main 9a4a1c6] return of the big mac
1 file changed, 7 insertions(+)
create mode 100644 bigmac.txt
```

<rev> is the last revision at which the bigmac.txt was present







To share our work and be able to collaborate with other people, we need to use a "code forge"







#### **Connect to Github**

Product  $\lor$  Solutions  $\lor$  Open Source  $\lor$  Pricing Q Search or jump to ... Sign in Sign up Let's build from here The world's leading Al-powered developer platform. Sign up for GitHub Start a free enterprise trial > <> Email address Trusted by the world's leading organizations ¥ крмд P&G TELUS 3M (A) Mercedes-Benz SAP Ê Productivity Accelerate innovation Our Al-nowered platform increases the

INISTITUT FRANCAIS DE BIOINFORMATIONE


## $\mathbf{O}$

### Sign in to GitHub

Password	Forgot password?
5	Sign in
S	Sign in
Sign in u	Sign in

Terms Privacy Docs Contact GitHub Support Manage cookies Do not share my personal information



### Your Dashboard

FRANCE

INISTITUT FRANKCAIS DE BIOINFORMATIONE



```
74
```

In a terminal in Jupyter

\$ ssh-keygen -t ed25519 -C "<u>your\_email@example.com</u>" \$ Generating public/private ALGORITHM key pair. \$ Enter a file in which to save the key (/home/YOU/.ssh/id\_ALGORITHM):[Press enter] \$ Enter passphrase (empty for no passphrase): [Type a passphrase] \$ Enter same passphrase again: [Type passphrase again]

https://docs.github.com/fr/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-a dding-it-to-the-ssh-agent







In a terminal in Jupyter

\$ ls -al ~/.ssh								
total 28								
drwx	2	tdenecker	tdenecker	4096	Apr	4	10:40	•
drwx	30	tdenecker	tdenecker	12288	Apr	4	10:37	• •
-rw	1	tdenecker	tdenecker	419	Apr	4	10:40	id_ed25519
-rw-rr	1	tdenecker	tdenecker	106	Apr	4	10:40	id_ed25519.pub
-rw-rr	1	tdenecker	tdenecker	1063	Jul	13	2023	known_hosts

```
$ cat ~/.ssh/id_ed25519.pub
```

Copy the output



Jupyter

\$\_

### Add your SSH key

thomasdenecker Thomas Denecker	×		
😔 Set status			
A Your profile			
Your repositories		Your personal account	r (thomasdenecker)     Go to your personal profile       2 Switch settings context +
III Your projects		A Public profile	SSH keys
🔀 Your Copilot		8 Account	This is a list of SSH keys associated with your account. Remove any keys that you do not recognize
Your organizations		Appearance	Authentication keys
Your enterprises		A Notifications	
☆ Your stars			
♡ Your sponsors		Billing and plans	~
• Your gists		⊡ Emails	
⊥ Upgrade		⑦ Password and authentication (۱۹) Sessions	
Try Enterprise	Free	₽ SSH and GPG keys	
☐ Feature preview		Drganizations	Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.
ô Settings		Enterprises	
유 GitHub Support 딨 GitHub Community		U Moderation	✓ GPG keys New GPG key

Sign out



•

1.1.1

### Add your SSH key

Your personal account ₹ Swi	Go to your personal profile
우 Public profile	Add new SSH Key
Appearance     Accessibility	Title
수 Notifications	Key type
Access	Authentication Key 🗢
Billing and plans	, Key
🖂 Emails	Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-
1 Password and authentication	nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'
((1)) Sessions	
SSH and GPG keys	
Drganizations	
Enterprises	
J Moderation	Add SSH key

### Create a new repository

× thomasdenecker Thomas Denecker Set status 8 Your profile న<sup>+</sup> Add account Your repositories Your projects 8 Your Copilot Your organizations Your enterprises ☆ Your stars ♡ Your sponsors Your gists Try Enterprise (Free) 冯 Feature preview 诊 Settings A GitHub Support 😡 GitHub Community

Sign out





	@IFB-ElixirFr
0	thomas.denecker.free.fr
6	https://orcid.org/0000-0003-1421-7641
$\mathbb{X}$	@DeneckerThomas
in	in/thomas-denecker

### Achievements



Find a repository Type - Langua	age - Sort - 🗐 New
AIR_Bioinfo Public	🟠 Star 👻
vémonstration d'outils de bioinfo dans le cadre d'un projet	
HTML 🛱 10 💱 4 친 Other Updated on Jul 15, 2021	
ronYeasts (Public)	Starred -
onYeasts is a tool suite dedicated to the analysis of iron in pathogenic easts 🛃 📚	
R 🏠 3 🥸 BSD 3-Clause "New" or "Revised" License Updated on Dec 8, 2022	
START-R (Public)	☆ Star 👻
imple Tool for Analysis of Replication Timing with R	
R 🏠 2 🖞 2 🤹 BSD 3-Clause "New" or "Revised" License Updated on Oct 17, 2023	
START-R-OLD Private	📩 Star 👻
オ1 Updated on Feb 5, 2018	
Club-Bioinfo Public	📩 Star 👻
communications lors du Club Bioinformatique de l'I2BC	

.



Create a new renov	sitory			
A repository contains all project	t files, including the revision history. Already have a p	proiect repository		
elsewhere? Import a repository.				
Required fields are marked with	n an asterisk (*).			
Owner *	Repository name *			
Thomasdenecker	burgers			
Inomasuchecker · J	Surgers is available.			
Great repository names are sho	art and memorable. Need inspiration? How about fuz	zv=hanniness ?		
Breachepository numes are sho		Ly happiness .		
Description (optional)				
You choose who can see	and commit to this repository.			
Initialize this repository with:	and commit to this repository.			
Initialize this repository with: Add a README file This is where you can write a lo	and commit to this repository. ng description for your project. <u>Learn more about READMEs</u>	è.		
Initialize this repository with: Add a README file This is where you can write a lo Add .gitignore	and commit to this repository. ng description for your project. Learn more about READMEs	à.		
Add a gitignore template: None v	and commit to this repository. ng description for your project. <u>Learn more about READMEs</u>	b.		
Add a gitignore template: None v Choose which files not to track from	and commit to this repository. ng description for your project. <u>Learn more about READMEs</u> n a list of templates. <u>Learn more about ignoring files</u> .	ð.		
Add .gitignore     .gitignore template: None      Choose a license	and commit to this repository. ng description for your project. <u>Learn more about READMEs</u> n a list of templates. <u>Learn more about ignoring files</u> .	b.		
Add a gitignore gitignore template: None  Choose a license License: None	and commit to this repository. ng description for your project. <u>Learn more about READMEs</u> n a list of templates. <u>Learn more about ignoring files</u> .	i.		
Add a README file This is where you can write a lo Add a README file This is where you can write a lo Add .gitignore .gitignore template: None ~ Choose which files not to track from Choose a license License: None ~ A license tells others what they can	and commit to this repository. Ing description for your project. <u>Learn more about READMEs</u> In a list of templates. <u>Learn more about ignoring files</u> . and can't do with your code. <u>Learn more about licenses</u> .	i.		





### git remote

# Push an existing repository from the command line

\$ git remote add origin git@github.com:<repository\_URL>/burgers.git
\$ git push -u origin main

8	A <sup>+</sup>
Set up GitHub Copilot	Add collaborators to this repository
Use GitHub's AI pair programmer to autocomplete suggestions as you code.	Search for people using their GitHub username or email address.
Get started with OltHub Copilot	Invite collaborators
QUICK SETUP — IT YOU'VE done This kind of thing before	burgers.git r
Get started by creating a new file or uploading an existing file. We recommend every it	repository include a <u>README</u> , <u>LICENSE</u> , and <u>.gitignore</u> .
or create a new repository on the command line	
echo "# burgers" >> README.md gli anit gli ang README.md gli commit - m frigri commit" gli branch -H main gli trancte add origin glitggithub.com:thomasdenecker/burgers.git gli punt-u origin main	<u>ی</u>
or push an existing repository from the command line	
git remote add origin git@github.com:thomasdenecker/burgers.git	ى
git pranch -H main git push -u origin main	
or import code from another repository	
	st.
You can initialize this repository with code from a Subversion, Mercurial, or TFS project	



### git clone

Clone a remote repository to your local machine.

git clone <repository\_URL>

Q Go	o to file	t	Add file	•	<> Co	de 🚽
	Local		(	Codes	oaces	
ol versior	➢ Clone					?
e 1,Pream	HTTPS SSH	GitHub	CLI			
te new se	git@github.com	:IFB-Elix	∷irFr/Wasm4	4Learn	-con¹	Q
ease 2.1.	Use a password-p	protected	SSH key.			
of condu	덮 Open with Git	Hub Desk	top			
CENSE	🗄 Download ZI	Р				
ne updated	d			6	months	s ago



### git push

Send local commits to the remote repository.

git push <remote\_name> <branch\_name>





After Pushing





### git push

Fetch the latest changes from the remote repository and merge them into your local branch.

git pull <remote\_name> <branch\_name>







### git remote

Show or add remote repositories.

git remote -v
git remote add <remote\_name> <remote\_URL>



### git fetch

Fetch branches and commits from the remote repository, but do not merge the changes into your local branch.

git fetch <remote\_name>







# Thank you for your attention !







