

# Version Control

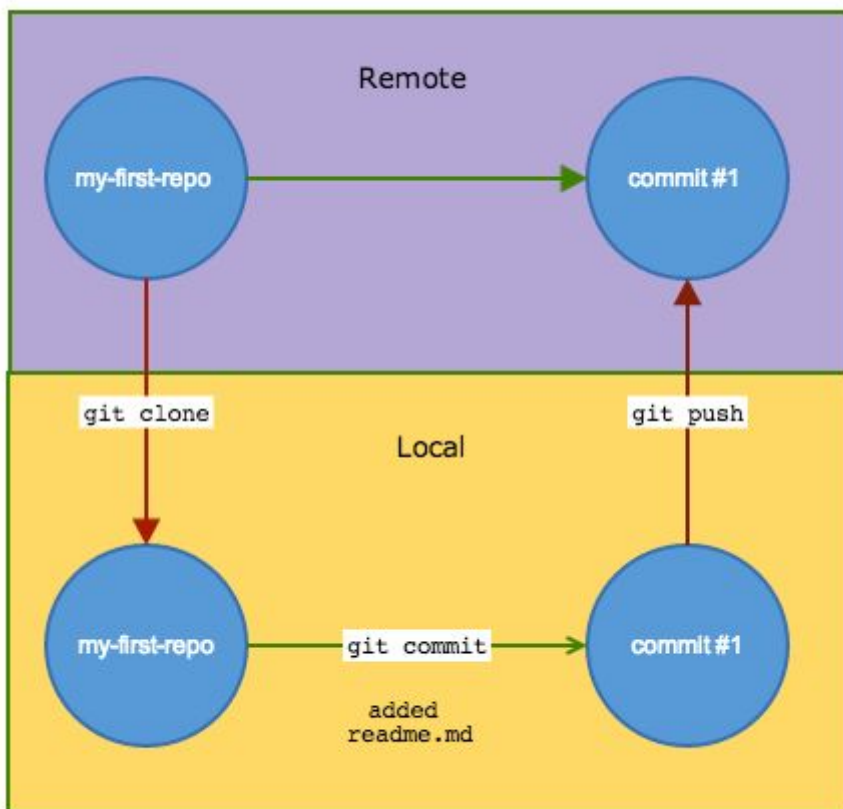
The definition of version control is “a system that records changes to a file or set of files over time so that you can recall specific versions later.”

It allows a developer to track what they are doing, recording changes so that

- Changes can be reverted if required
- Development can be carried out on a branch of the code that doesn't break the stable 'master branch'
- Software can be tagged as specific release versions
- Developers can collaborate and merge changes

The most commonly used version control software is [git](#) where all changes can be tracked locally but then 'pushed' to a remote git repository so that others can view code and contribute to code. Three commonly used platforms are github.com, gitlab.com and bitbucket.com. In this way a coder can develop locally making changes and recording those changes in the local git repository and push the changes to github for example so others can see the changes. In addition when major releases are made the specific version of the code corresponding to that release can be tagged and either downloaded or checked out.

An overview of the local <-> remote model is shown in this picture

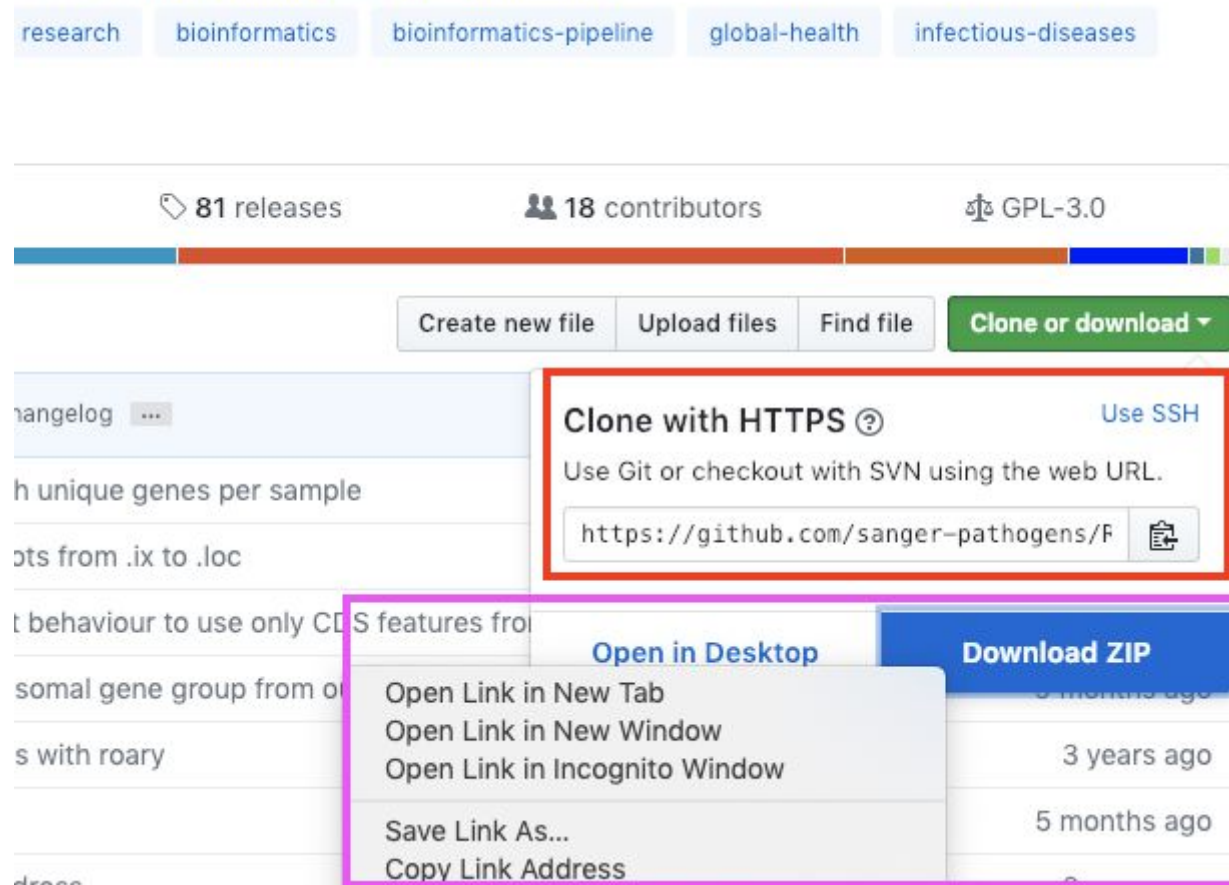


In this diagram a user can **clone** the remote repository, make changes, **commit** these changes and then **push** them back to the remote server so that others can view and/or download them.

To keep it simple, this tutorial will describe how to get software from a remote repository and checkout specific versions.

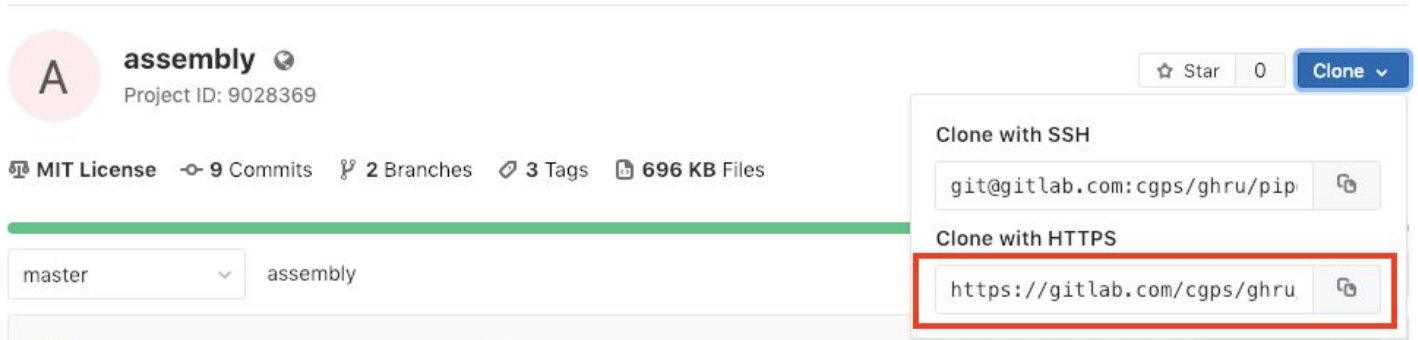
On github a repository has a button which gives options to download the repository or a link to clone the project

<http://sanger-pathogens.github.io/Roary>

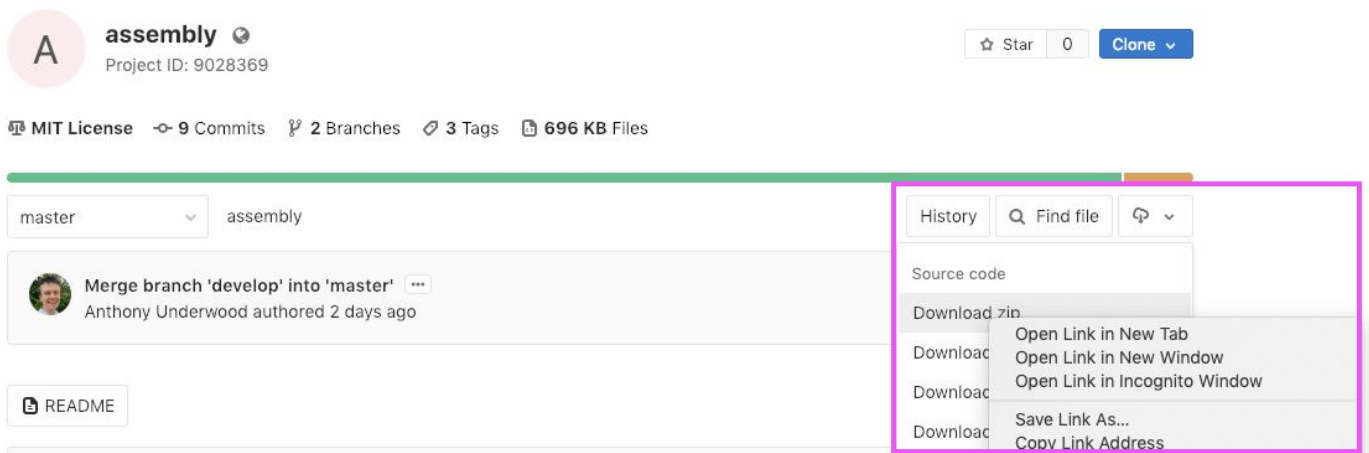


- In this image the link for clone the project is shown in the red box.
  - On the linux machine where you would like to clone the project type git clone and paste the link `git clone https://github.com/sanger-pathogens/Roary.git`
  - This will create a directory named the same as the project in this case Roary with the code cloned within it.
- To download the project in a static non-version controlled version - right click on the Download ZIP button and copy the link address (shown in the pink box)
  - On the linux machine where you would like to clone the project to type wget and paste the link `wget https://github.com/sanger-pathogens/Roary/archive/master.zip`
  - To access the code unzip the file `unzip master.zip`  
This will create a new directory named after the project `Roary-master`. The code for the project will be found within this directory but is static but can not be updated with future changes

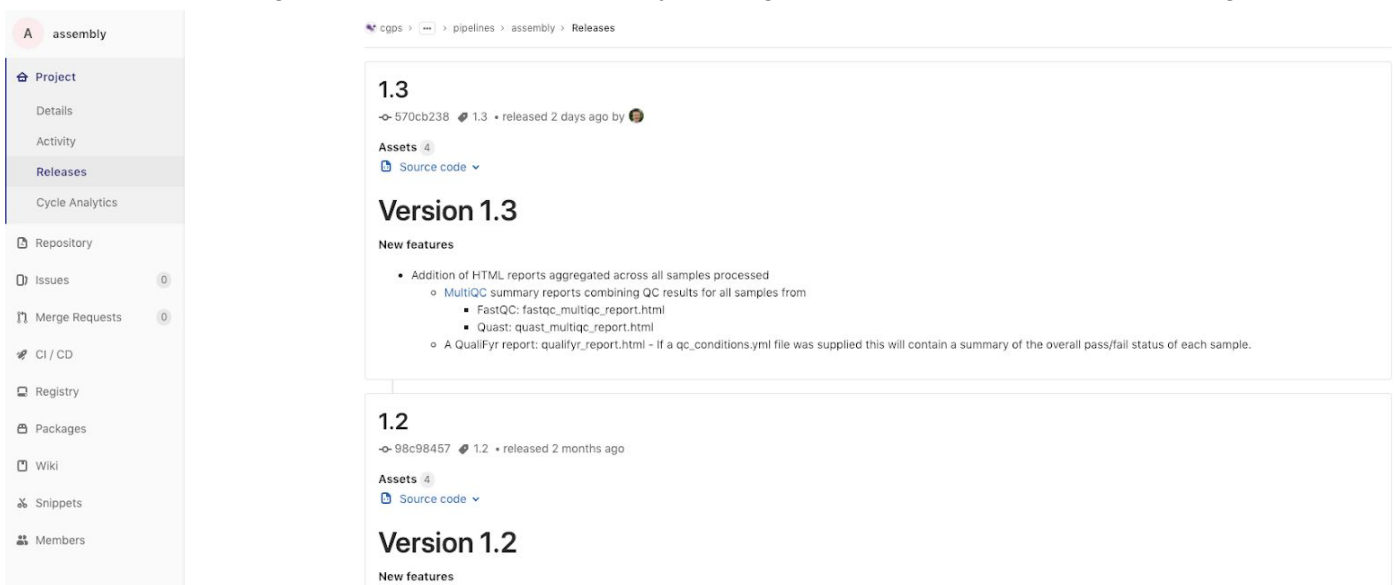
A similar interface exists on gitlab. Here for the assembly workflow project the project can be cloned from the link shown by clicking on the clone button



or downloaded as static files from the download icon



Releases (specific tags of the code) can be seen by clicking on 'Releases' in the left hand navigation bar



This enables checking out of specific versions of the code as follows

1. Check out the project and change directories to the project

```
git clone https://gitlab.com/cgps/ghru/pipelines/assembly.git  
cd assembly
```

2. List available tags

```
git tag -l
```

```
1.1
```

```
1.2
```

```
1.3
```

3. Checkout a specific tag

```
git checkout tags/1.3
```

The console will report that You are in 'detached HEAD' state. Although this sounds painful, don't worry this is what happened when a particular 'commit' (point in time within the project) is checked out

4. In the future when a new release is made, it is easy to update the code as follows by pulling new changes from the remote repository and checking out the new version

```
git pull  
git checkout tags/1.4
```