

Version Control with Git

Oliver Alka¹, Simone Lederer²

1 Applied Bioinformatics, University of Tübingen

2 Translational Bioinformatics, University of Tübingen

October 22th, 2020

Most important help



... btw ... do you know this awesome service?

... <https://lmgty.com>

The intention of this course is to bring the people with least IT skills to a point where they can start their masters courses in bioinformatics. If you're a fast one and manage to solve today's issues in a short time please be patient! Here are some interesting things you can read in the meantime:

- About Git:

<http://tom.preston-werner.com/2009/05/19/the-git-parable.html>

- Dilbert:

<http://dilbert.com>

- PhDomics:

<http://www.phdomics.com>

Git: Motivation

- Git is one software for VERSION CONTROL
- It's not the only one
 - CVS
 - Subversion
 - Mercurial
 - https://en.wikipedia.org/wiki/List_of_version_control_software
- However, maybe it's nowadays the most popular one ...

Git: Important Resources

- The docu
<https://git-scm.com>
- A nice tutorial
<https://try.github.io/levels/1/challenges/1>
- Maybe to be read first
<http://tom.preston-werner.com/2009/05/19/the-git-parable.html>

Git & Software Projects

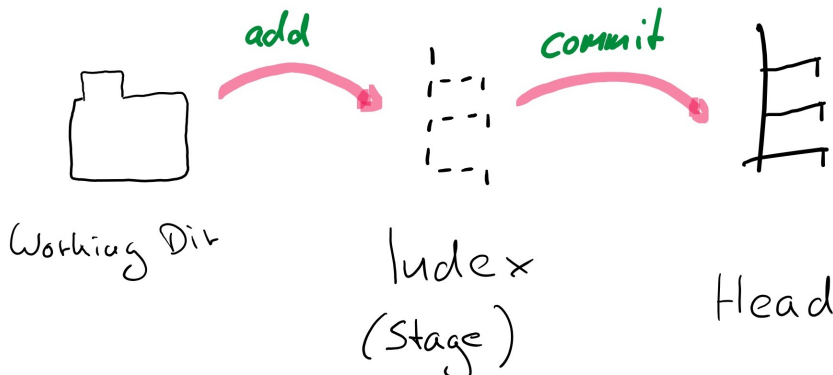
The screenshot shows the GitHub interface for the OpenMS/OpenMS repository. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below this, the repository name 'OpenMS / OpenMS' is displayed along with statistics: 47 Unwatched, 133 Stars, and 129 Forks. The main content area shows the repository description, a list of topics (openms, c-plus-plus, etc.), and a table of recent commits. The commit table includes columns for the commit hash, author, message, and time ago.

Commit Hash	Author	Message	Time Ago
4362344	simosachsenberg	Merge pull request #3767 from simosachsenberg/fix_warn_unused	2 days ago
e8f1e13	THIRDPARTY	rm DS_Store	6 months ago
#3654	simosachsenberg	Merge pull request #3654 from jscheffer/fix_dosygenJS	2 months ago
060f3cd	contrib	[NOP] submodule update to master	2 months ago
	dsc	Add additional instructions for the deb package	6 days ago
	share/OpenMS	correct names in HMD82StructMapping.txt	20 days ago
	src	[FIX] wrong critical section	2 days ago
	tools	Revert "[BUILD] travis cache restart"	16 days ago
	gRignore	[Update] gRignore	10 months ago
	gitmodules	[FEATURE] Add contrib and THIRDPARTY as submodules.	11 months ago
	travis.yml	Merge branch 'develop' into feature/ats_migrate	9 months ago

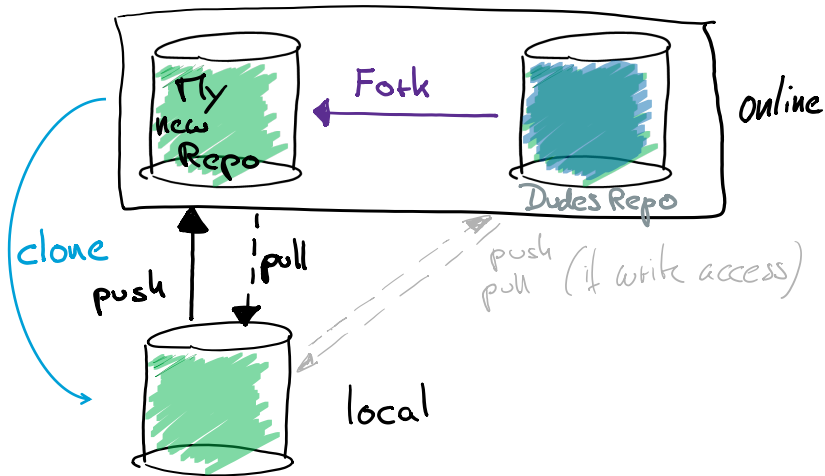
Version Control is indispensable for bigger software projects where a lot of developers are working together (e.g. <https://github.com/OpenMS/OpenMS>), but also helps with versioning in smaller projects/applications/scripts. Use it!

Git & Software Projects

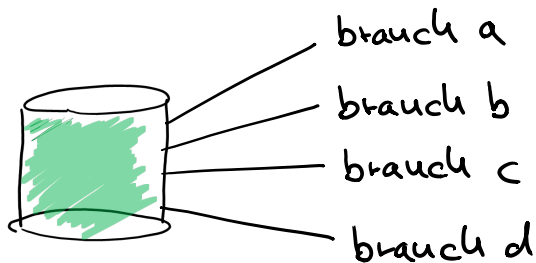
You can depict a local repository by three trees. The first one is the working directory, which holds the files. The second is an index which is used as staging area. The last one is head, which points to your last commit. Git saves the changes you make and stores it in the index/head based on adding and commit the changes.



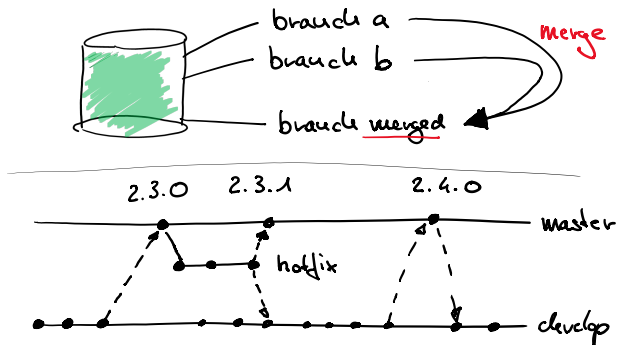
Git & Software Projects



Git & Software Projects



Git & Software Projects



Git & Software Projects

Prerequisites:

Create an account for Github (<https://github.com>)

Install git <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Note: You can use the VirtualBox Image (Linux course) - if you like.

Create a new empty directory and change into it.

Note: If you run into a problem when using git, please read the info/error message in the terminal carefully - usually it is self explaining. If you get stuck - give google a chance. If nothing helps, please shoot me an email!

How to proceed

In the first set, we have a look at basic git commands.

We create a local repository and play around with it.

The following slides are separated into commands and Task sets.

Please have a peak at the Task Set first!

Then check the following "important commands", play around with them and try to solve the given tasks.

Git: Important Commands

```
$ git init
```

- Create a new empty repository
- Play around in the directory
- Check the git status

```
$ git status
```

Git: Important Commands

- Create a new text file and write something into it ...
- ... and check the status again

```
$ git status
```

Git: Important Commands

```
$ git add
```

- Purpose: Prepare changes for integration into your repo

```
$ git add -A
```

- Add everything that is not ignored

Git: Important Commands

```
$ git commit
```

- Purpose: Integrate added changes into your repo

```
$ git commit -m "important-commit-message"
```

- Commit - adding a commit message directly

```
$ git commit --amend
```

- Correct last commit wrt files and message

Git: Important Commands

```
$ git reset HEAD <file>
```

- Unstage a file (Undo adding)

Git: Important Commands

```
$ git log
```

- Purpose: Check commit history

Git: Task Set 1

- 1 Download the following archive: `https://www.dropbox.com/s/y2bwsfvv2ctn6a9/awesome_project.zip?dl=0`
- 2 Extract `awesome_project.zip` somewhere in your file system
- 3 Initialize `awesome_project/` to a Git repository
- 4 Add source files, but not the config and `.class` files
- 5 Make a commit
- 6 Try to find out how to ignore config and `.class` files (So they won't be added with `git -A`)
- 7 If you are fast: Reset your repo to previous commit (Try to find out how to use `git reset`)

Git: Some Preliminary Work

- Purpose: Get a copy of a our work project in your GitHub account (peek at Task2)
- To do this we fork the repository (repo) of interest.
- Go the the online repository and use the fork button. Now you have a fork and can edit it as you like!

lukaszimmermann / vorkurs

Watch 1 Star 0 Fork 42

Code Issues 0 Pull requests 15 Actions Projects 0 Wiki Security Insights

No description, website, or topics provided.

3 commits 4 branches 0 packages 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

lukaszimmermann	Merge pull request #17 from Julia-97/master	Latest commit 5d68647 on 1 Oct 2019
old_stuff	Sample files	3 years ago
src	Sample files	3 years ago
README	Sample files	3 years ago
readme	change readme	6 months ago

README

afzhfgjhkgzgd

Git: Important Commands

```
$ git clone
```

- Purpose: Get a local copy of your remote repo (project)
- How does this work?
- Let's get help

```
$ git clone -h
```

Git: Important Commands

```
$ git status
```

- Purpose: Check for changes in your repo ...
- Well, you'll get it in a moment!

Git: Important Commands

```
$ git push
```

- Purpose: Upload your local changes into the remote *parent repo*
- There are different reasons to do this such as
 - 1 Save your work at a secure place
 - 2 Give others the chance to see your work
 - 3 Enable collaboration with others on the same work

Git: Important Commands

```
$ git pull
```

- Purpose: Integrate changes from the remote *parent repo*

Git: Important Commands

```
$ git remote -v
```

- Lists all known remote repos

Git: Important Commands

```
$ git branch -a
```

- See all branches that the local repo knows about

Git: Important Commands

```
$ git checkout <branch>
```

- Switch to branch

```
$ git checkout -b <new_branch>
```

- Create and switch to new_branch

```
$ git merge <branch-to-be-merged-into-current-one>
```

- Merge commits from a different branch into the current one

Git: Task Set 2

- 1 Fork the repository of Lukas:
`https://github.com/lukaszimmermann/vorkurs`
- 2 Clone your forked repository
- 3 Checkout new branch for modification
- 4 Make commit to remove old_stuff/ (to the new branch!)
- 5 Push the branch to your remote
- 6 Add Lukas repository as a remote (Hint: `git remote add`)
- 7 Make a pull request of your changes to Lukas repository (online)

Git: Important Things Most Likely Omitted

- Branching - Why is branching so powerful, and how is it used?

Please check:

`https://nvie.com/posts/`

`a-successful-git-branching-model`

`https://guides.github.com/introduction/flow/`

- Checking out other branches
- Merging changes from others

Mergeconflicts : TaskSet 3

- Fork

https://github.com/klarareichard/vorkurs_merging

- Clone your Fork.
- Create new branch modification

```
$ git checkout -b modification
```

- insert "Hello World" as first line into mergeconflict.txt, change second line to "This line won't cause a mergeconflict anymore". Add an additional line: "This is an additional line" to the end of the file.
- add and commit your changes
- switch to branch master
- create another branch other-modification and switch to it

Mergeconflicts : TaskSet 3

- Change second line of mergeconflict.txt to "This line will cause a merge conflict". Add a file "newfile.txt".
- Commit and switch to master

```
$ git merge modification
```

```
$ git merge other-modification
```

- open mergeconflict.txt and resolve mergeconflict. Commit the result.
- show branches and commits as a graph

```
$ git log --graph --oneline --all
```

Thank you

