

**BioinformaticsClubPresentation.final.reallyFINAL.FINALFINAL.ppt**  
**or:**  
**how I learned to stop worrying and love version control**

A practical introduction to *git*

Paulo Filipe Pereira de Jesus Silva

e-mail: paulo.silva@ibmc.up.pt



# Version control

- Why?
  - Keep a history of the files or documents as they change along time
  - Avoid having many versions of scripts or documents lying around the disk
  - Go back to a previous version of a script
  - “Hmm, which file is the latest version?...”

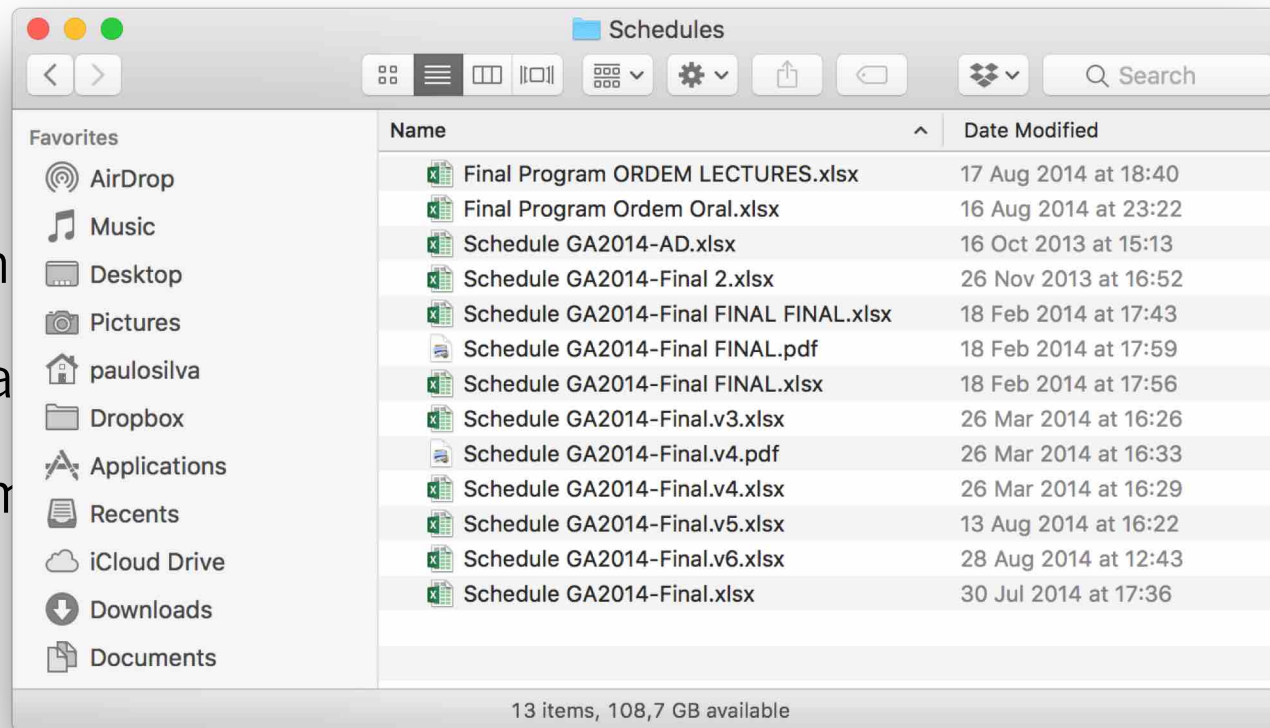
# Version control

- Why?
  - Keep a history of the files or documents as they change along time

– Avoid  
around

– Go ba

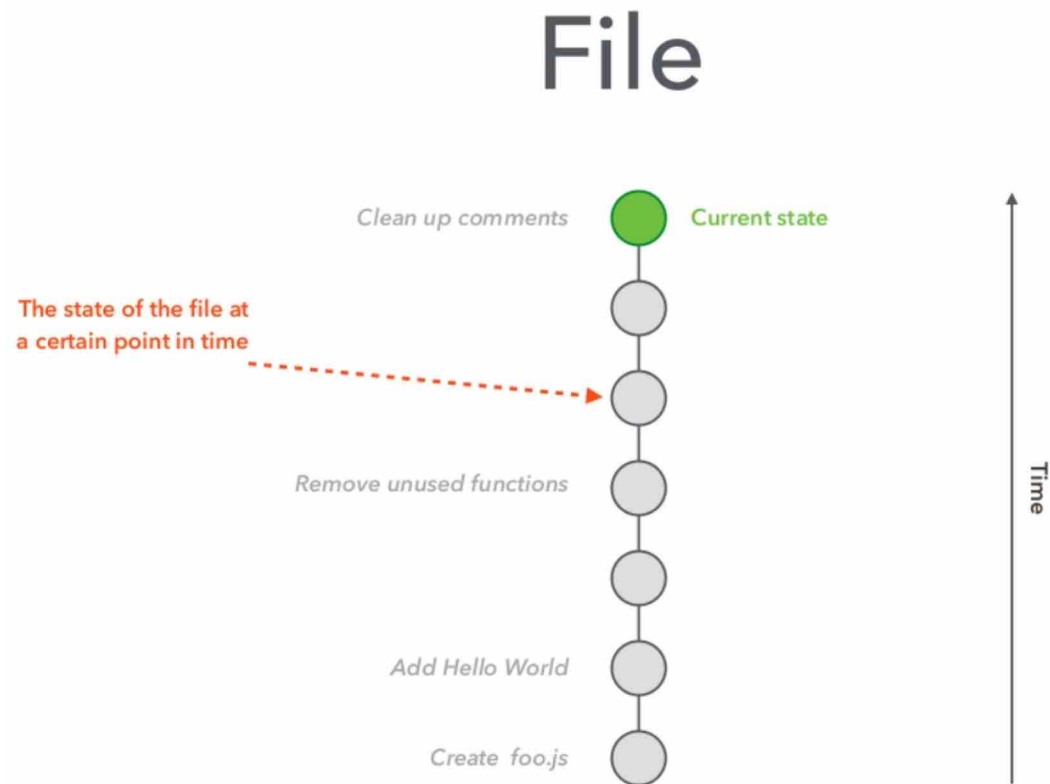
– “Hm



ing

# Version control

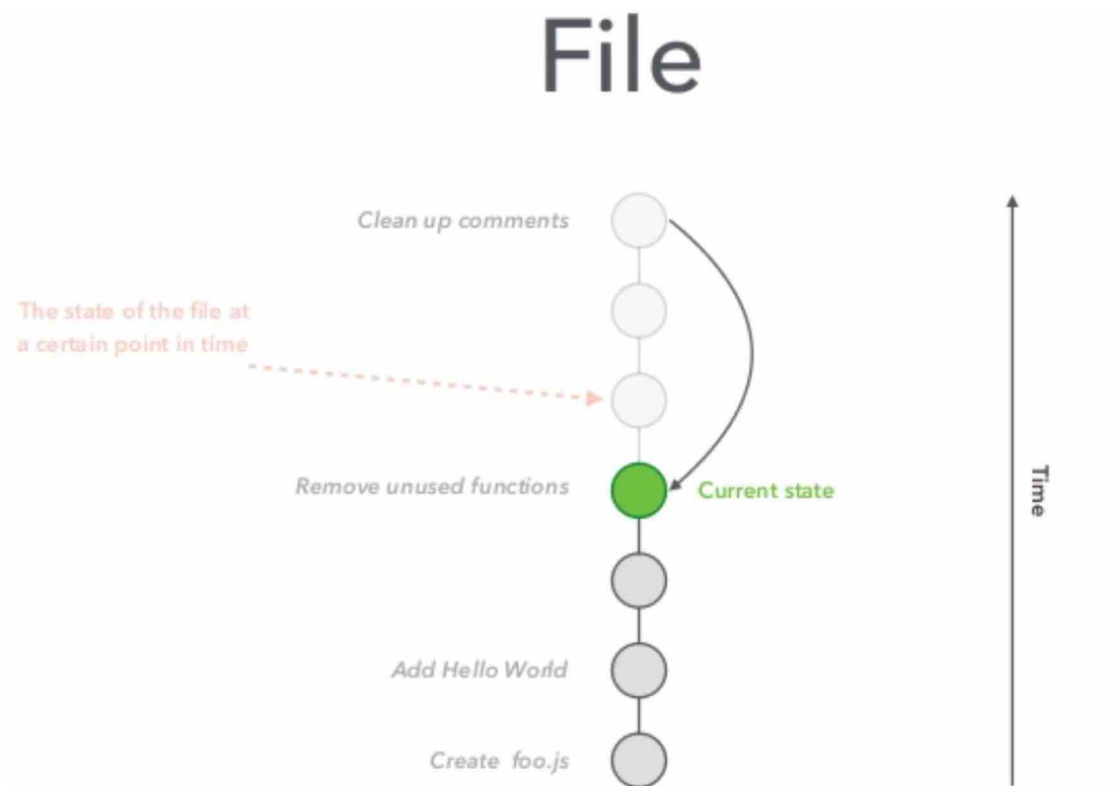
- Keeping a history of a file (or files) along time:



Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

# Version control

- Reverting to a previous version of a file:

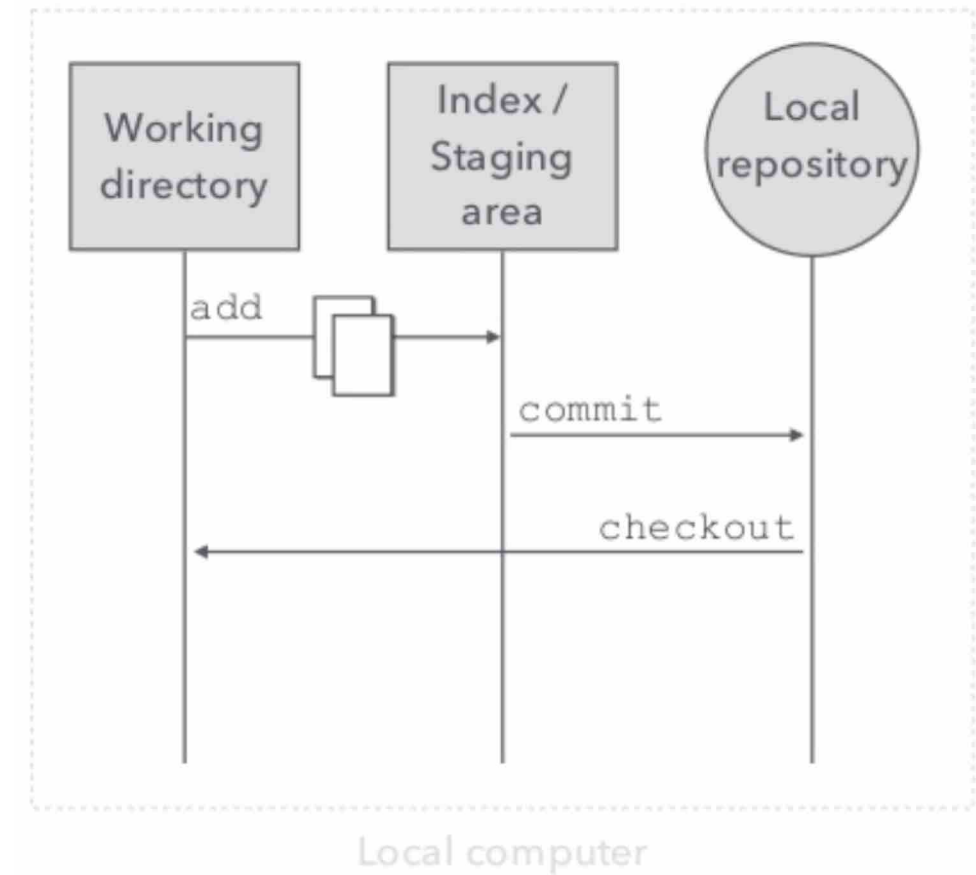


Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

# Version control

- Many different version control systems exist
  - cvs, subversion, mercurial, git
  - Track changes in Word doesn't count!
- *git* is currently the most widely used
  - Developed in 2005 by Linus Torvalds, the creator of Linux
  - Distributed version control system, where every user has a copy of the repository, including history
  - Can be a bit tricky sometimes (merge conflicts...)

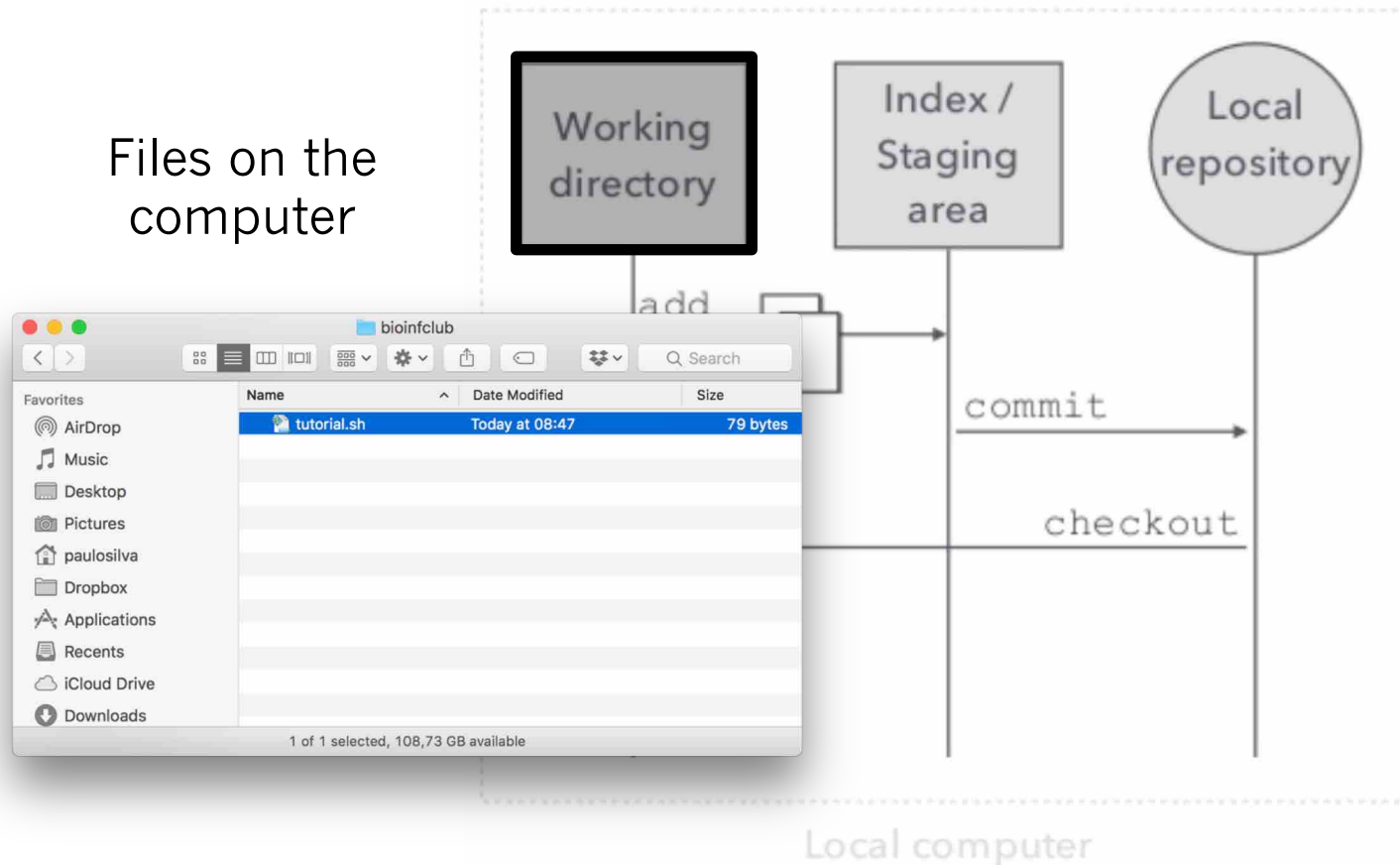
# Overview of *git*



Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

# Overview of *git*

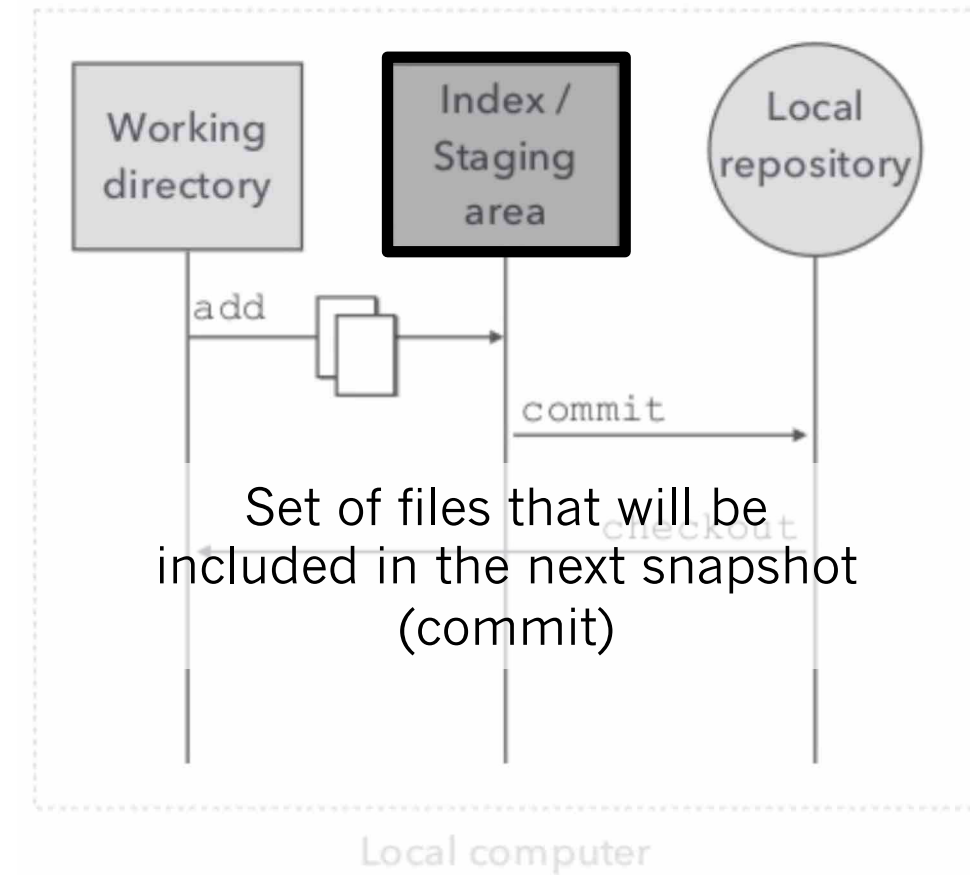
Files on the computer



Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

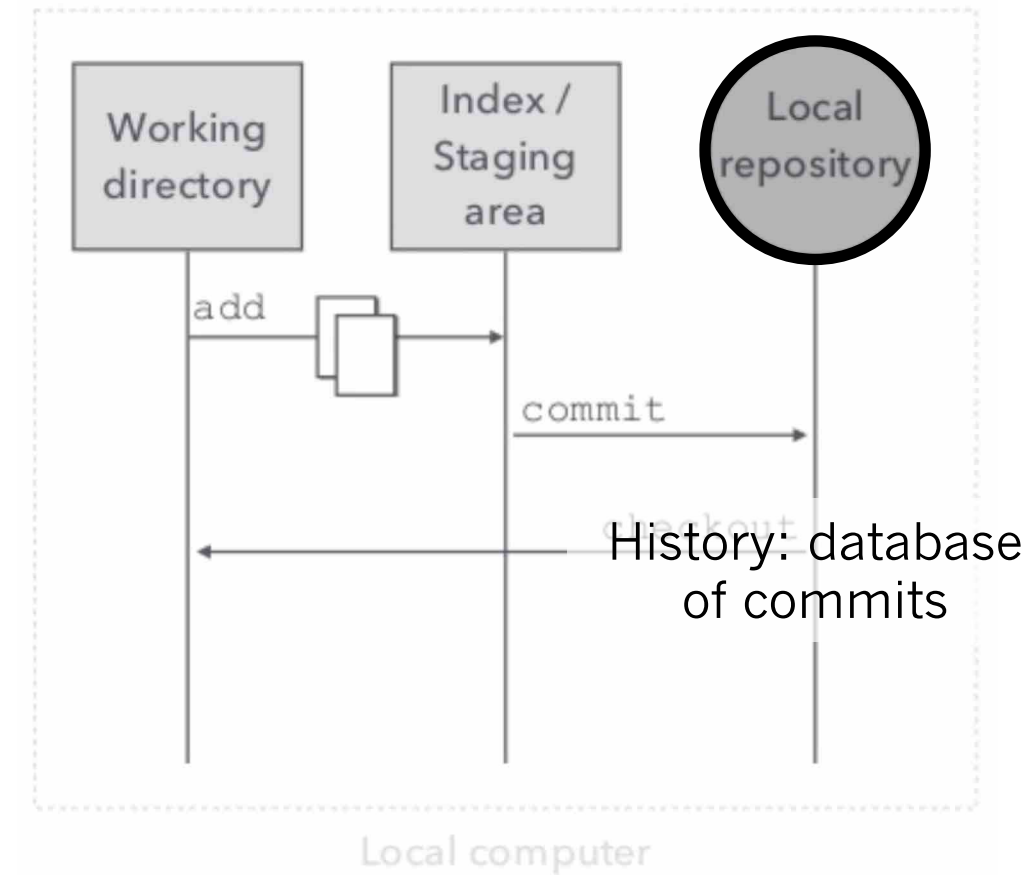


# Overview of *git*



Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

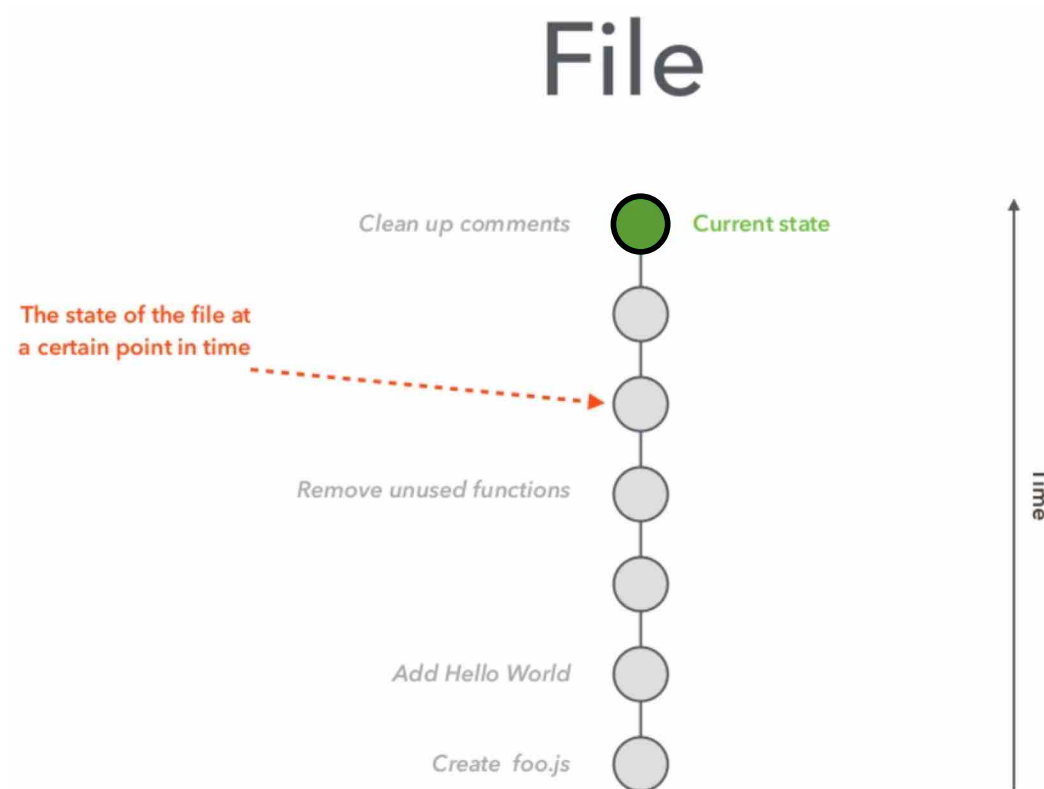
# Overview of *git*



Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

# Overview of *git*

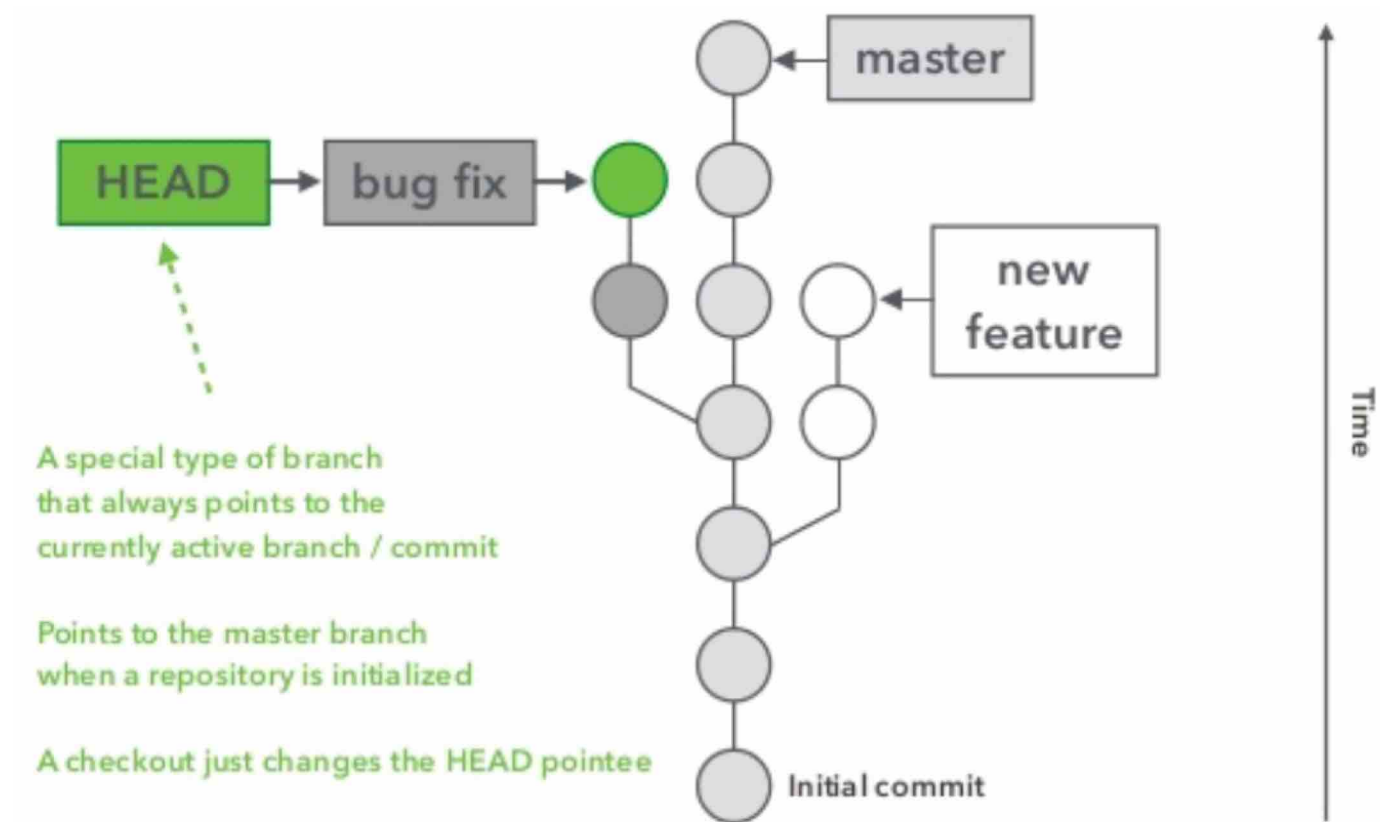
- History: database of commits



Source: <https://www.slideshare.net/HiroYukiVincentYamaz/an-introduction-to-git-59148054/8>

# Overview of *git*

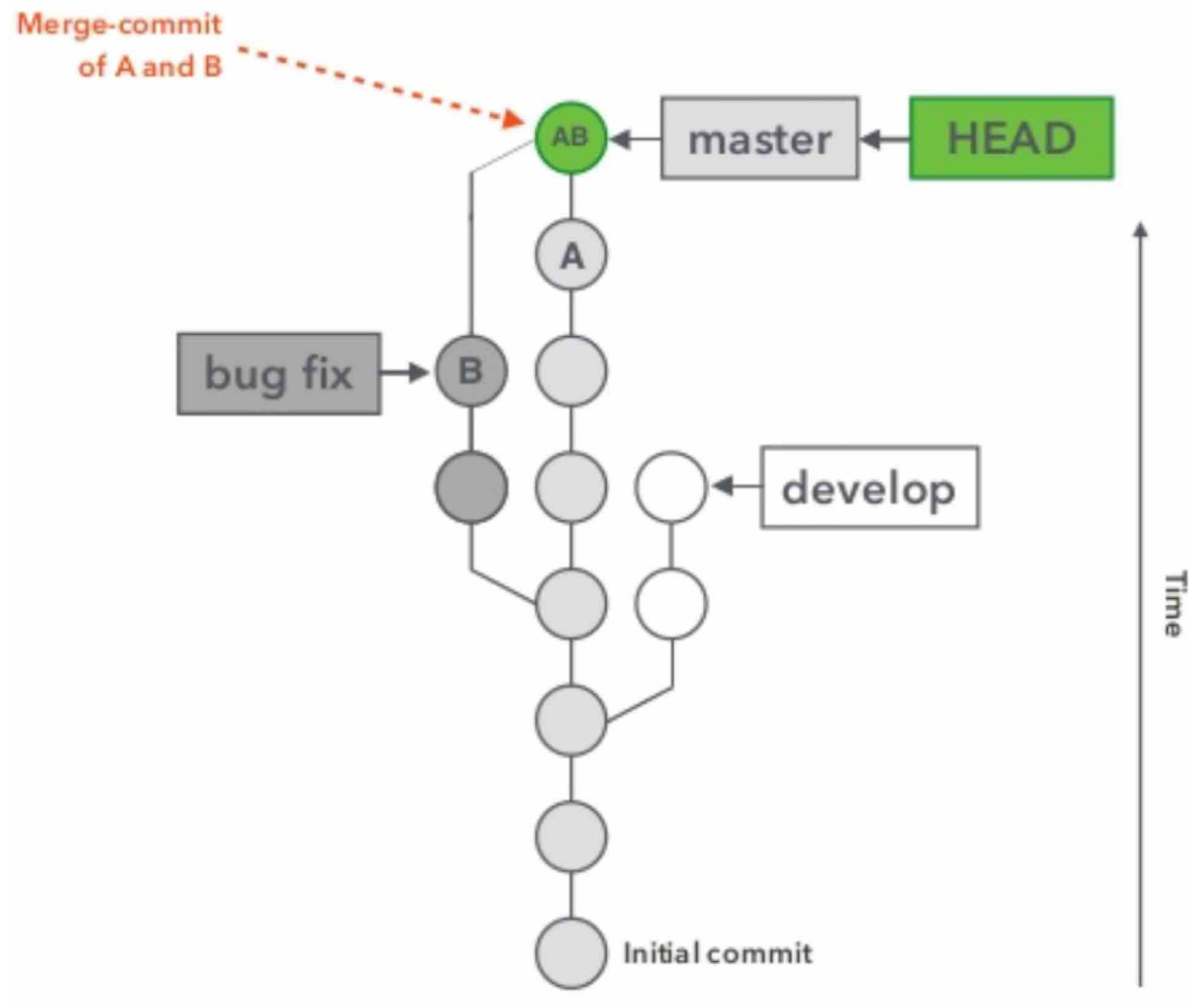
- Branching



Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

# Overview of *git*

- Merging



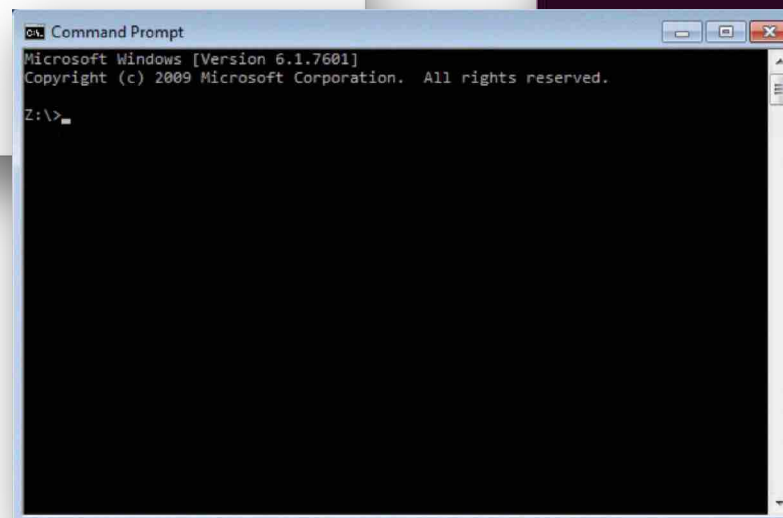
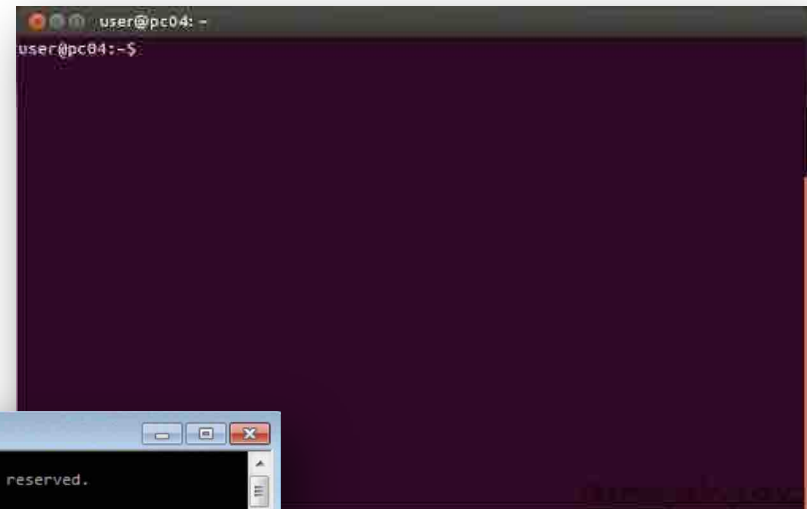
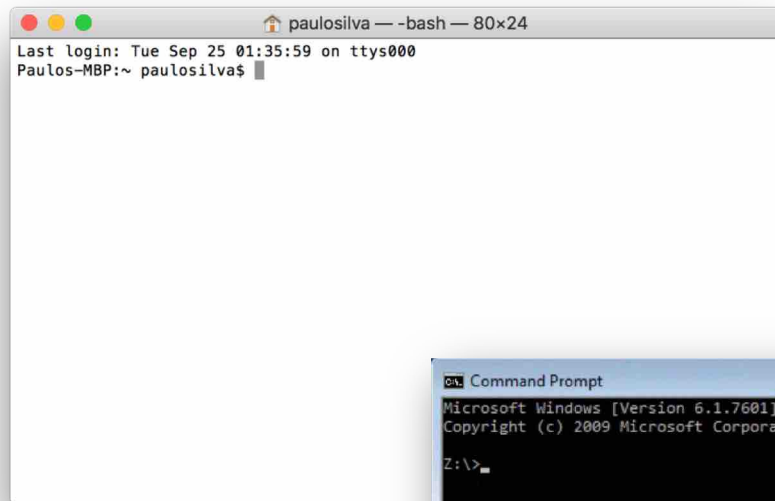
Source: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054/8>

# Practical *git*

- Open your laptops or follow on-screen!

# Practical *git*

- Start your terminals



# Practical *git*

- Create a folder to hold a local repository

```
mkdir bioinfclub
```

```
cd bioinfclub
```

- Now initialize a repository in the new folder

```
git init
```

- You have created your first repository! Now let's add something to it



# Practical *git*

- Create a text file or script and add content, e.g.:

```
#!/bin/sh  
echo "Hello World!"
```

  - Try something in your favorite language! Or copy that script you were working on just before you came to this talk
- Save it in the `bioinfclub` folder (mine is `tutorial.sh`)



Source: <https://www.slideshare.net/jelabra/introduction-to-git-44244608>

# Practical *git*

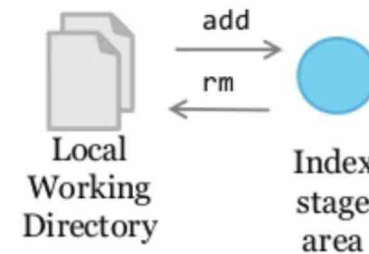
- Check the status of the local repository:

```
git status
```

- Now add the file to the index and check the status again:

```
git add tutorial.sh
```

```
git status
```



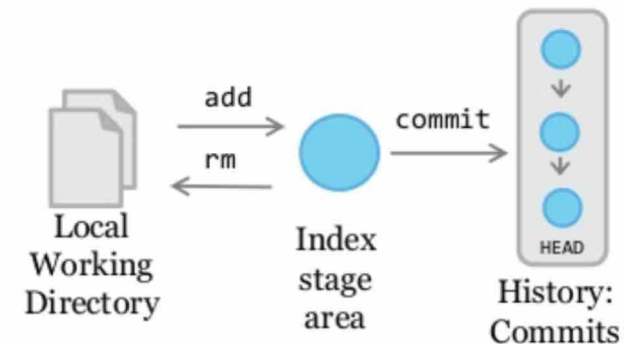
Source: <https://www.slideshare.net/jelabra/introduction-to-git-44244608>

# Practical *git*

- You're now ready to make your first commit! Think of an inspiring commit message, or use the following:

```
git commit -m 'My first commit'
```

- You now have a local repository with one file committed
- Check the status again



Source: <https://www.slideshare.net/jelabra/introduction-to-git-44244608>

# Practical *git*

- Let's change the file. Open the file you created previously and add or delete something:

```
echo 'echo "Hello again!"' >> tutorial.sh
```

- Now add the file to the index and do another commit:

```
git add tutorial.sh
```

```
git commit -m 'Update for my project'
```

# Practical *git*

- The changed file has been added to the history of your local repository, which has now two commits. You can check the history with:

```
git log
```

- Oh no! My previous version of my file was working and this one doesn't! How do I get it back?

# Practical *git*

- You can get a previous commit back by doing a checkout with the commit hash code:

```
git checkout <commithash>
```

- You can also examine the differences between the current and previous versions of the file:

```
git diff <commithash>
```

# Practical *git*

- A colleague has asked for help on a task and adapting your script might just work, but you also want to keep your existing script. Let's create a branch:

```
git branch adapted
```

```
git checkout adapted
```

- Open the script and add or delete something:

```
echo 'echo "Help for my colleague"' >> tutorial.sh
```

# Practical *git*

- Save the script and add the file to the index and commit:  

```
git add tutorial.sh
```

```
git commit -m 'Update to help my colleague'
```
- You can go back to the previous version by doing:  

```
git checkout master
```
- And switch again to the friend-helping-version with:  

```
git checkout adapted
```

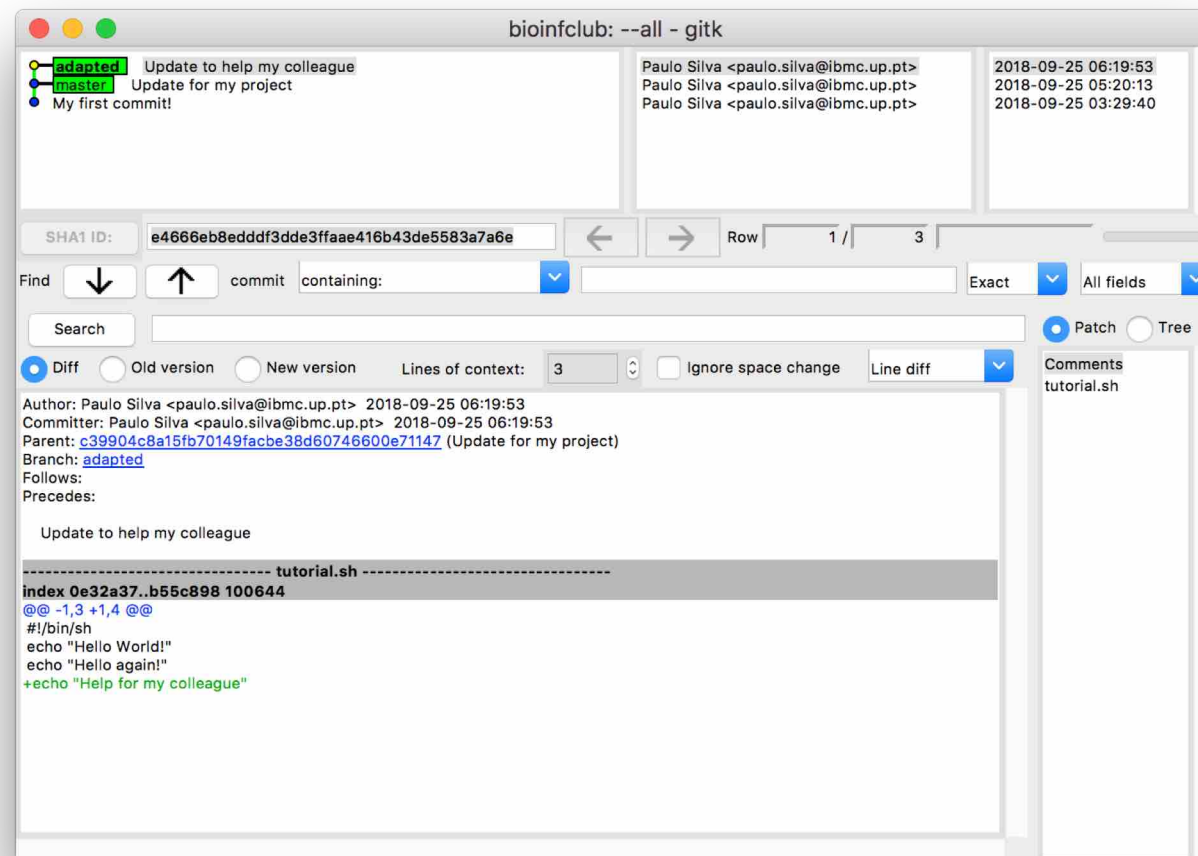


# Practical *git*

- To help visualize what we've been doing during this exercise, there are many git GUIs available, but git comes with one.

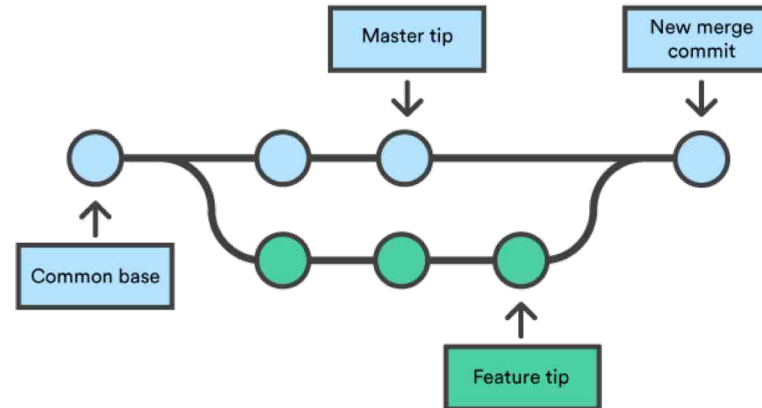
Run:

`gitk`



# Advanced *git*

- Merging
  - Fast-forward
  - 3-way merge
- Rebasing
- Pushing, pulling and fetching from remote repositories
- Reset, revert, clean, tag, reflog
- git cheat sheet:
  - [http://rogerdudler.github.io/git-guide/files/git\\_cheat\\_sheet.pdf](http://rogerdudler.github.io/git-guide/files/git_cheat_sheet.pdf)



Source: <https://www.atlassian.com/git/tutorials/using-branches/git-merge>

# More info on using *git*

- Official documentation: <https://git-scm.com/doc>
- Tutorials and guides:
  - Atlassian git tutorials: <https://www.atlassian.com/git/tutorials>
  - DataCamp Introduction to Git for Data Science: <https://www.datacamp.com/courses/introduction-to-git-for-data-science>
  - git – the simple guide: <http://rogerdudler.github.io/git-guide/>
- Presentations:
  - Introduction to git – Universidad de Oviedo: <https://www.slideshare.net/jelabra/introduction-to-git-44244608>
  - An Introduction to Git - Hiroyuki Vincent Yamazaki: <https://www.slideshare.net/HiroyukiVincentYamaz/an-introduction-to-git-59148054>
- There are papers!
  - Ten Simple Rules for Taking Advantage of Git and GitHub (PMID:27415786)
  - A Quick Introduction to Version Control with Git and GitHub (PMID: 26785377)

