SouthGreen bioinformatics platform

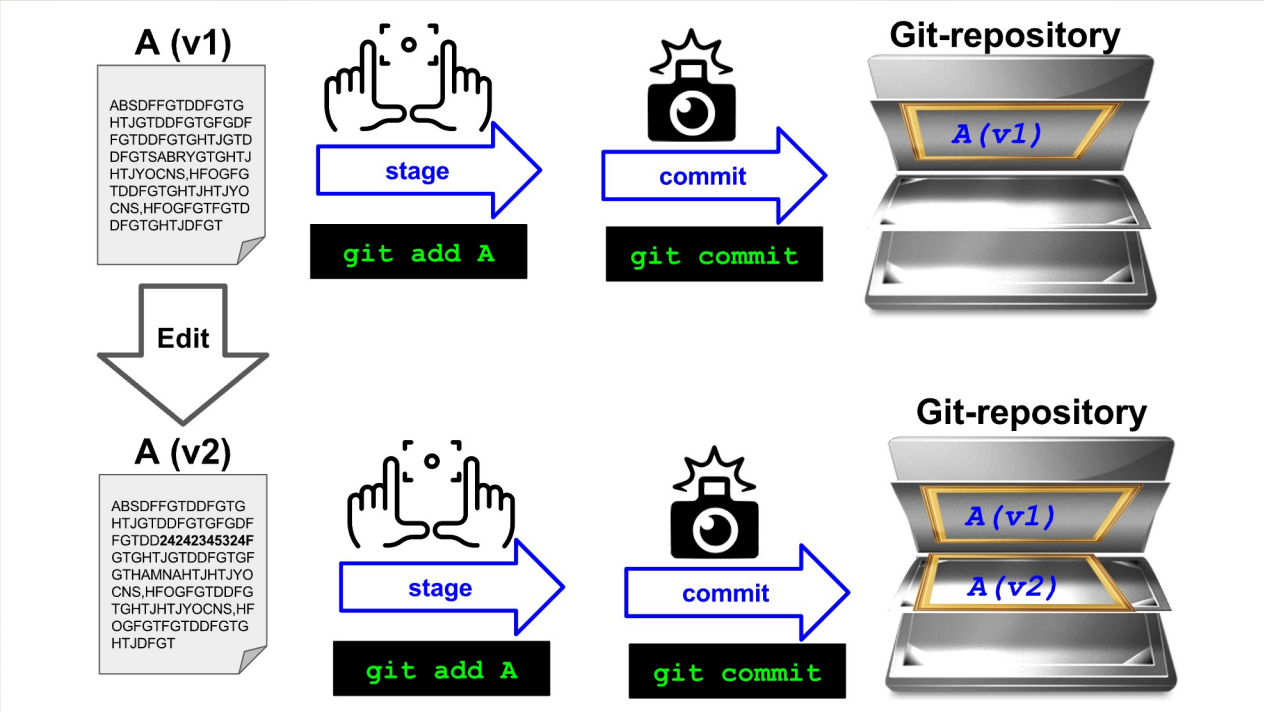
VERSION CONTROL WITH

Some of the material comes from the "CodeRefinery" project

Jacques Dainat Ph.D.

Version Control Systems (VCS)

- Track changes and versions
- System which **records snapshots** of a project

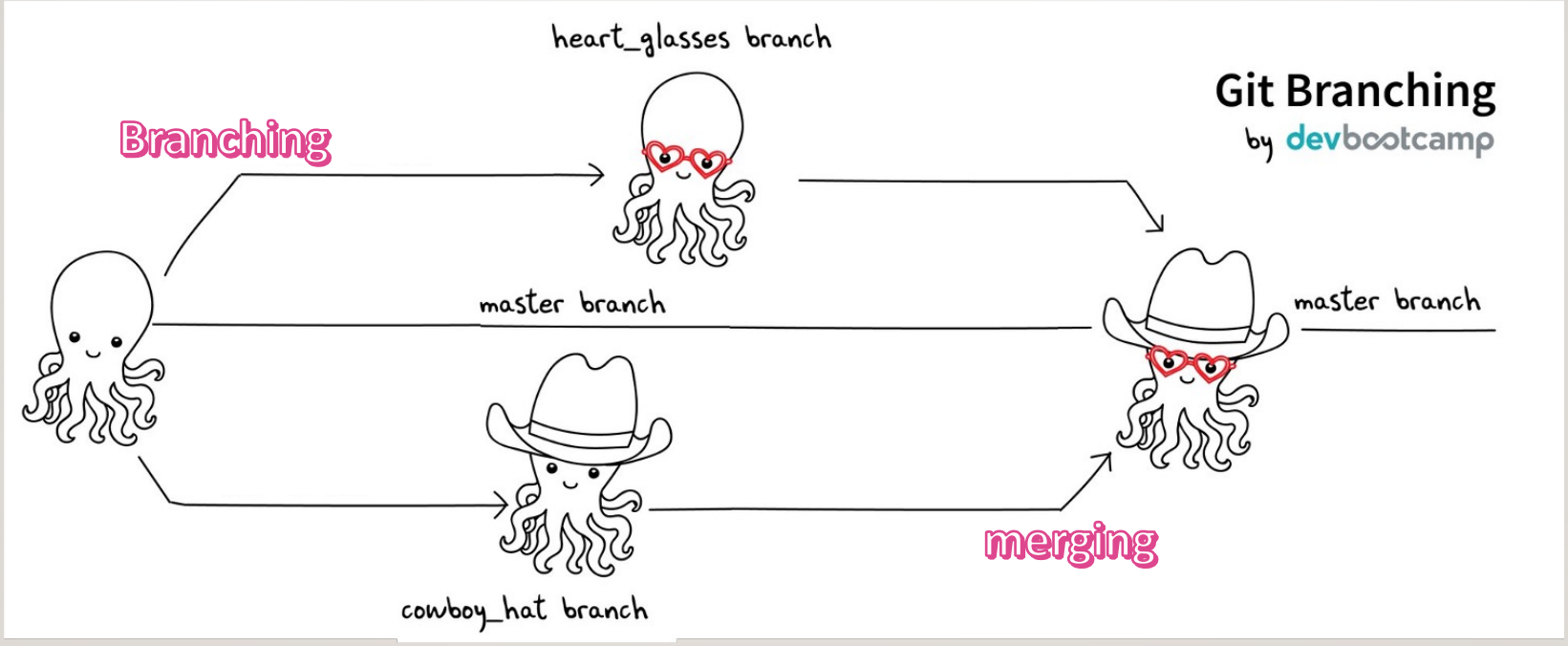


What we typically like to snapshot

- Software (this is how it started but Git/GitHub can track a lot more)
- Scripts
- Documents (plain text files much better suitable than Word documents)
- Manuscripts (Git is great for collaborating/sharing LaTeX or Quarto manuscripts)
- Configuration files
- Website sources
- Data (/!\ with huge data)

The essence of version control

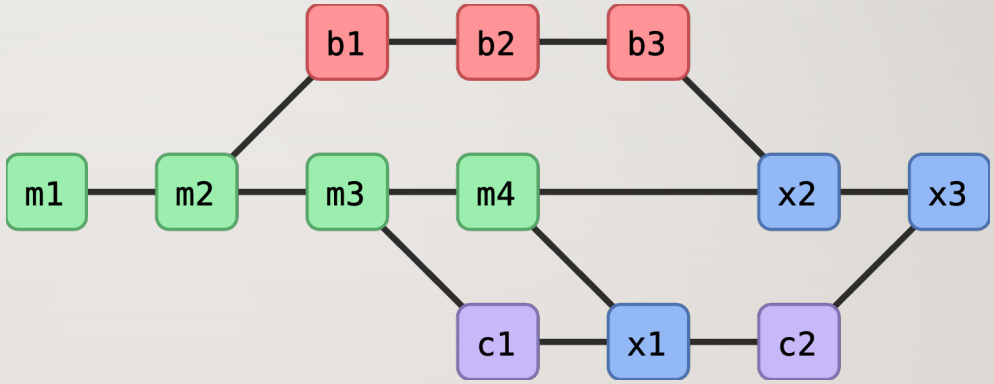
- Implements **branching**:
  - You can work on several feature branches and switch between them
  - Different people can work on the same code/project without interfering
  - You can experiment with an idea and discard it if it turns out to be a bad idea



- Implements **merging**:
  - Person A and B's simultaneous work can be easily combined

Roll-back functionality

• Mistakes happen - without recorded snapshots you cannot easily undo mistakes and **go back to a working version**.



Branching

• Often you need to work on **several issues/features in one code** - without branching this can be messy and confusing.

• You can simulate branching by copying the entire code to multiple places but also this will be messy and confusing.

Reproducibility

• How do you indicate which version of your code you have used in your paper?

• When you find a bug, how do you know **when precisely** this bug was introduced (Are published results affected? Do you need to inform collaborators or users of your code?).

Collaboration

With version control, none of these are needed anymore (or have much simpler answers):

- *"I will just finish my work and then you can start with your changes."*
- *"Can you please send me the latest version?"*
- *"You never got the code I send by email? Maybe the spam filter marked it as malicious?"*
- *"Where is the latest version?"*
- *"Which version are you using?"*
- *"Which version have the authors used in the paper I am trying to reproduce?"*

Version Control Systems (VCS)

VCS => To keep track of changes and versions

Subversion / Mercurial / **Git** => One of the most popular VCS in use today

Bitbucket / GitHub / GitLab => hosting services for Git repositories



```
> git log
commit 9fc68014300136d412f350d25309e3c209dbeeb3
Author: Jacques Dainat <jacques.dainat@gmail.com>
Date:   Fri Mar 3 17:50:41 2023 +0100

    add github usernames

commit 1b399d4f42b9700af3daa07d4d9f511b43d5b701
Author: jhayer <juliette.hayer@gmail.com>
Date:   Tue Jan 17 15:38:27 2023 +0100

    fixed condition to run krakentools extract
```

Unique identifier

Who

When

What

One commit

One commit

Version Control System (VCS)

Differences ?

```
commit 9fc68014300136d412f350d25309e3c209dbeeb3
Author: Jacques Dainat <jacques.dainat@gmail.com>
Date:   Fri Mar 3 17:50:41 2023 +0100

    add github usernames

commit 1b399d4f42b9700af3daa07d4d9f511b43d5b701
Author: jhayer <juliette.hayer@gmail.com>
Date:   Tue Jan 17 15:38:27 2023 +0100

    fixed condition to run krakentools extract
```

```
> git diff 9fc68014300136d412f350d25309e3c209dbeeb3 1b399d4f42b9700af3daa07d4d9f511b43d5b701
```

File

Changes

```
--- a/README.md
+++ b/README.md
@@ -38,5 +38,5 @@ d

 ## Author and contributors

-Juliette Hayer
-Jacques Dainat
+Juliette Hayer  (@jhayer)
+Jacques Dainat  (@Juke34)
```

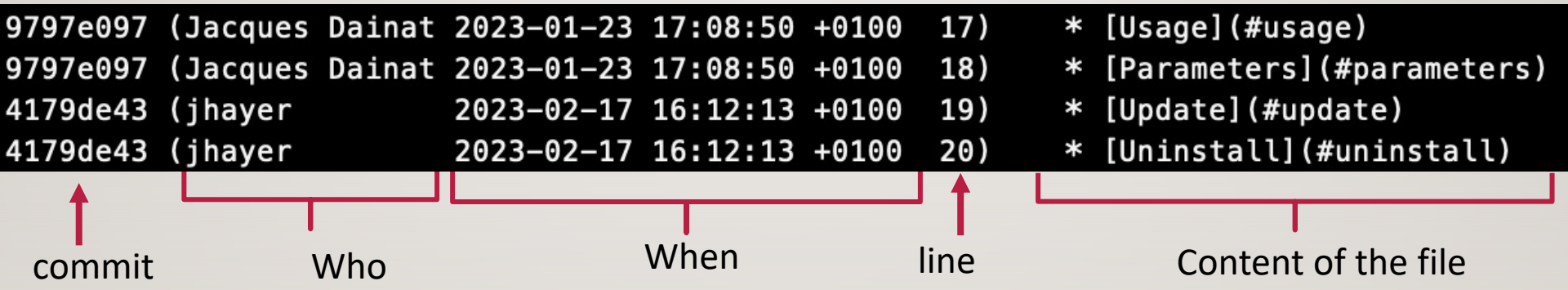
Version Control System (VCS)

Using git blame, you can see the entire git revision of the file with respect to each line.

Git blame will come in handy when you want to know which commit is responsible for the changes and which author has committed those changes

```
> git blame myfile.nf
```

```
9797e097 (Jacques Dainat 2023-01-23 17:08:50 +0100  17)    * [Usage](#usage)
9797e097 (Jacques Dainat 2023-01-23 17:08:50 +0100  18)    * [Parameters](#parameters)
4179de43 (jhayer         2023-02-17 16:12:13 +0100  19)    * [Update](#update)
4179de43 (jhayer         2023-02-17 16:12:13 +0100  20)    * [Uninstall](#uninstall)
```



Difficulties of version control

Despite the benefits, let's be honest, there are some difficulties:

• One more thing to learn (it's probably worth it and will save you more time in the long run; basic career skill).

• Difficult if some people don't want to use it (in the worst case, you can version control on your side and send them versions).

• Advanced things can be difficult, a bit too many gotchas (basics are often enough, ask others for help when needed).

A real-life example

Let us explore an **existing Git repository** on GitHub.

The goal here is not to teach GitHub yet (we will explain some of the concepts later), but rather to get a glimpse of the wider picture and see the social aspect to know what our end goal is.

As an example we can explore a famous Git repository which was used to produce the Event Horizon Telescope images:

https://github.com/achael/eht-imaging