

Introduction to Scientific Computing

Version Control

Federico Raimondo
10/10/2019

fraimondo@uliege.be



@fraimondo



@RaimondoFede

Overview



- ▶ What are versions?
- ▶ Why you need versions?
- ▶ How can we organise versions?
- ▶ How can we collaborate and still be organised?
- ▶ Hands-on Git
- ▶ Demo on GitHub

Versions?



Name	✓	Date Modified	Size
▶ annals_neuro	✓	26 Apr 2018 at 00:19	
📄 Draft V1	✓	15 Sep 2016 at 15:03	
📄 Draft V2	✓	7 Nov 2016 at 16:10	
📄 Draft V3	✓	14 Nov 2016 at 15:44	
📄 Draft V4	✓	22 Nov 2016 at 08:18	
📄 Draft V5	✓	28 Nov 2016 at 09:46	
📄 Draft V6	✓	29 Dec 2016 at 18:23	
📄 Draft V7	✓	30 Dec 2016 at 11:58	
📄 Draft V8	✓	18 Jan 2017 at 21:41	
📄 Draft V8	✓	11 Jan 2017 at 09:49	
📄 Draft V9	✓	19 Jan 2017 at 10:58	
📄 Draft V10	✓	19 Jan 2017 at 22:25	
📄 Draft V10 ms.docx	✓	31 Jan 2017 at 09:53	
📄 Draft V10_In.docx	✓	24 Jan 2017 at 15:41	
📄 Draft V10-rb.docx	✓	24 Jan 2017 at 14:14	
📄 Draft V11	✓	9 Feb 2017 at 13:15	
📄 Draft V12-pnas	✓	13 Feb 2017 at 10:39	
📄 Draft V12-pnas js	✓	13 Feb 2017 at 21:13	
📄 Draft V12-pnas js (Jacobo's iMac's conflicted copy 2017-02-14)	✓	14 Feb 2017 at 10:15	
▶ ELife	✓	26 Apr 2018 at 00:20	
📄 figura_nota_leloir.ai	✓	12 Jan 2018 at 10:29	

Versions: why do we need them?



File 1: main_script.m

```
function analyse_data(subject)
    ... brain = read_brain_image(subject)
    ... [left, right] = split_brain(brain)
    ... c = compute_connectivity(left, right)
    ... return c
end
```

January

February

March

May

File 2: utils.m

```
function [left, right] = split_brain(brain)
    ... % Do complicated brain interpretation
end

function [c] = compute_connectivity(region1, region2)
    ... % Do complicated connectivity computation
end
```

```
function [left, right] = split_brain(brain)
    ... % Do ANOTHER complicated brain interpretation
end

function [c] = compute_connectivity(region1, region2)
    ... % Do complicated connectivity computation
end
```

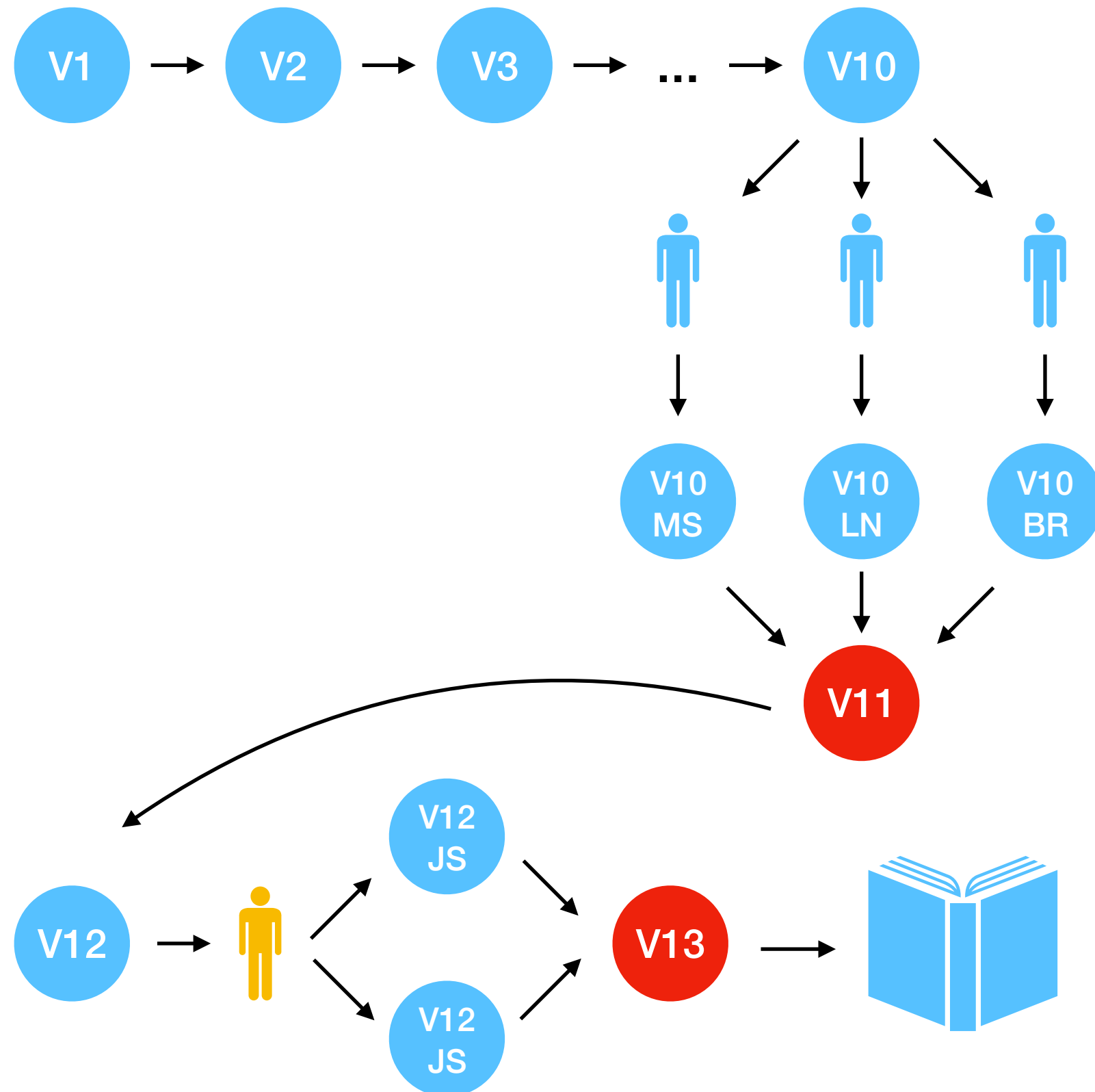
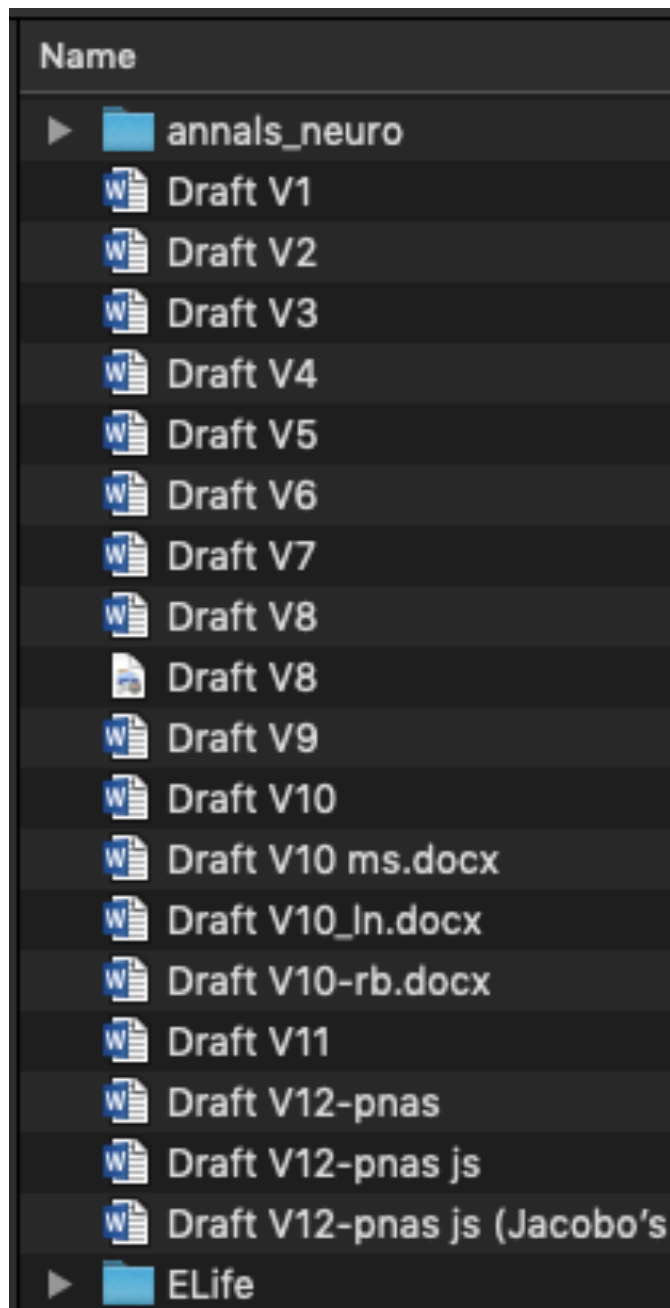
```
function [left, right] = split_brain(brain)
    ... % Do ANOTHER complicated brain interpretation
end

function [c] = compute_connectivity(region1, region2)
    ... % Do ANOTHER complicated connectivity computation
end
```

```
function [left, right] = split_brain(brain)
    ... % Do complicated brain interpretation
end

function [c] = compute_connectivity(region1, region2)
    ... % Do ANOTHER complicated connectivity computation
end
```

Versioning: a manuscript writing example



Versioning: combining edits



V10

```
1 This is a text that I wrote over a long period of time.
2
3 Unfortunately, there are some mistakes. For example, in this
4 paragraph, I will forget one that is important to understand the key concept.
5
6 In this other paragraph, I should write the conclusions of the paper. I
7 should also depict the importance of this work and why I should be awarded
8 with the annual price of the Association for Computing Machinery also kowns as
9 Turing Award. However, I have a typo that will cool of the reviewers.
```

V10 MS

```
1 This is a text that I wrote over a long period of time.
2
3 Unfortunately, there are some mistakes. For example, in this
4 paragraph, I will forget one word that is important to
5 understand the key concept.
6
7 In this other paragraph, I should write the conclusions of the paper. I
8 should also depict the importance of this work and why I should be awarded
9 with the annual price of the Association for Computing Machinery also kowns as
10 Turing Award. However, I have a typo that will cool of the reviewers.
```

V10 LN

```
1 This is a text that I wrote over a long period of time.
2
3 Unfortunately, there are some mistakes. For example, in this
4 paragraph, I will forget one that is important to understand the key concept.
5
6 In this other paragraph, I should write the conclusions of the paper. I
7 should also depict the importance of this work and why I should be awarded
8 with the annual prize of the Association for Computing Machinery also kowns as
9 Turing Award. However, I have a typo that will cool of the reviewers.
```

Versioning: Patching



```
diff --git a/text_v1.txt b/text_v2.txt
index c552c88..5f28b03 100644
--- a/text_v1.txt
+++ b/text_v2.txt
@@ -1,7 +1,8 @@
 This is a text that I wrote over a long period of time.

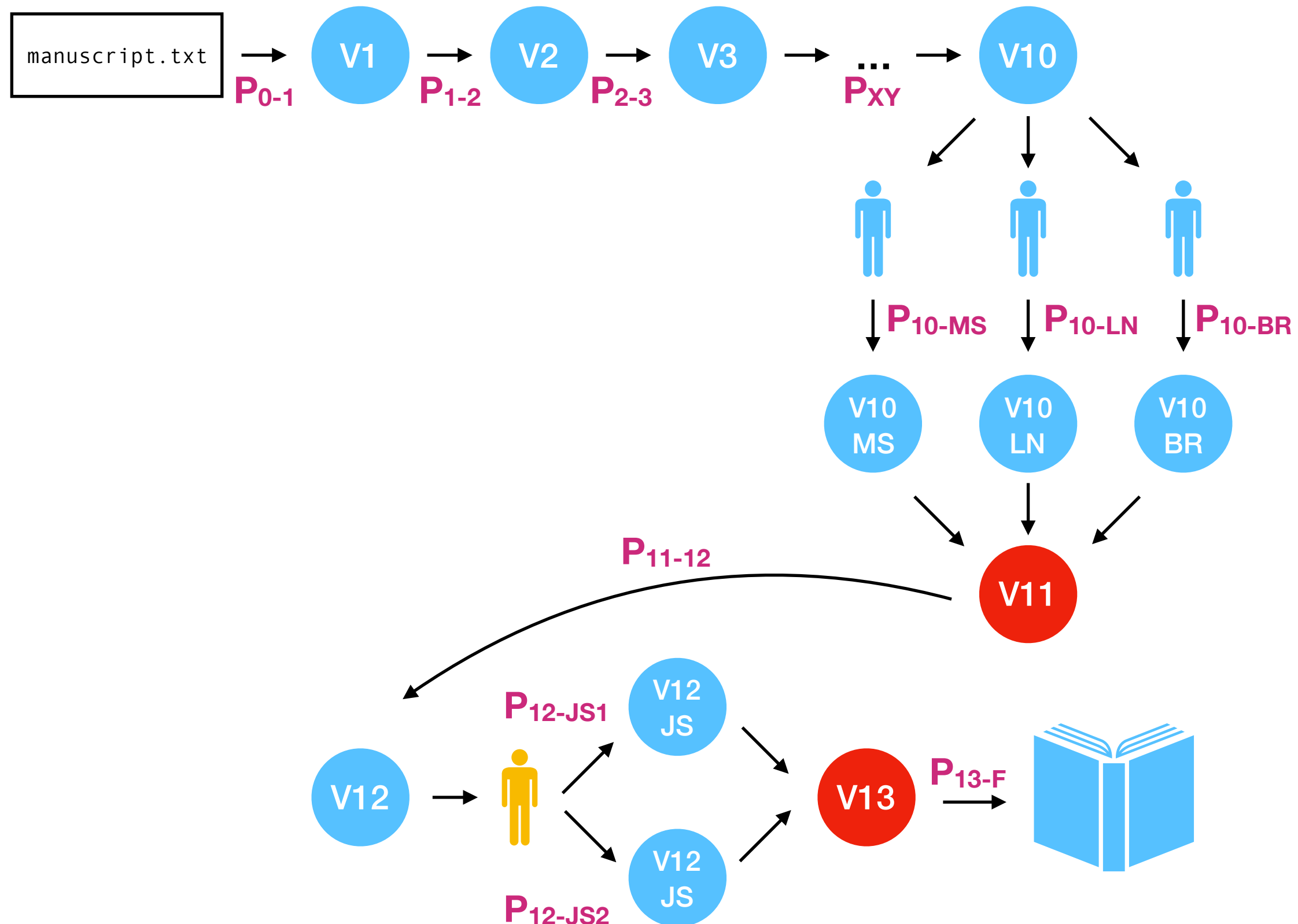
 Unfortunately, there are some mistakes. For example, in this
-paragraph, I will forget one that is important to understand the key concept.
+paragraph, I will forget one word that is important to
+understand the key concept.

 In this other paragraph, I should write the conclusions of the paper. I
 should also depict the importance of this work and why I should be awarded
(END)
```

```
diff --git a/text_v1.txt b/text_v3.txt
index c552c88..4871fc7 100644
--- a/text_v1.txt
+++ b/text_v3.txt
@@ -5,5 +5,5 @@ paragraph, I will forget one that is important to understand the key concept.

 In this other paragraph, I should write the conclusions of the paper. I
 should also depict the importance of this work and why I should be awarded
-with the annual price of the Association for Computing Machinery also kowns as
+with the annual prize of the Association for Computing Machinery also kowns as
 Turing Award. However, I have a typo that will cool of the reviewers.
\ No newline at end of file
(END)
```

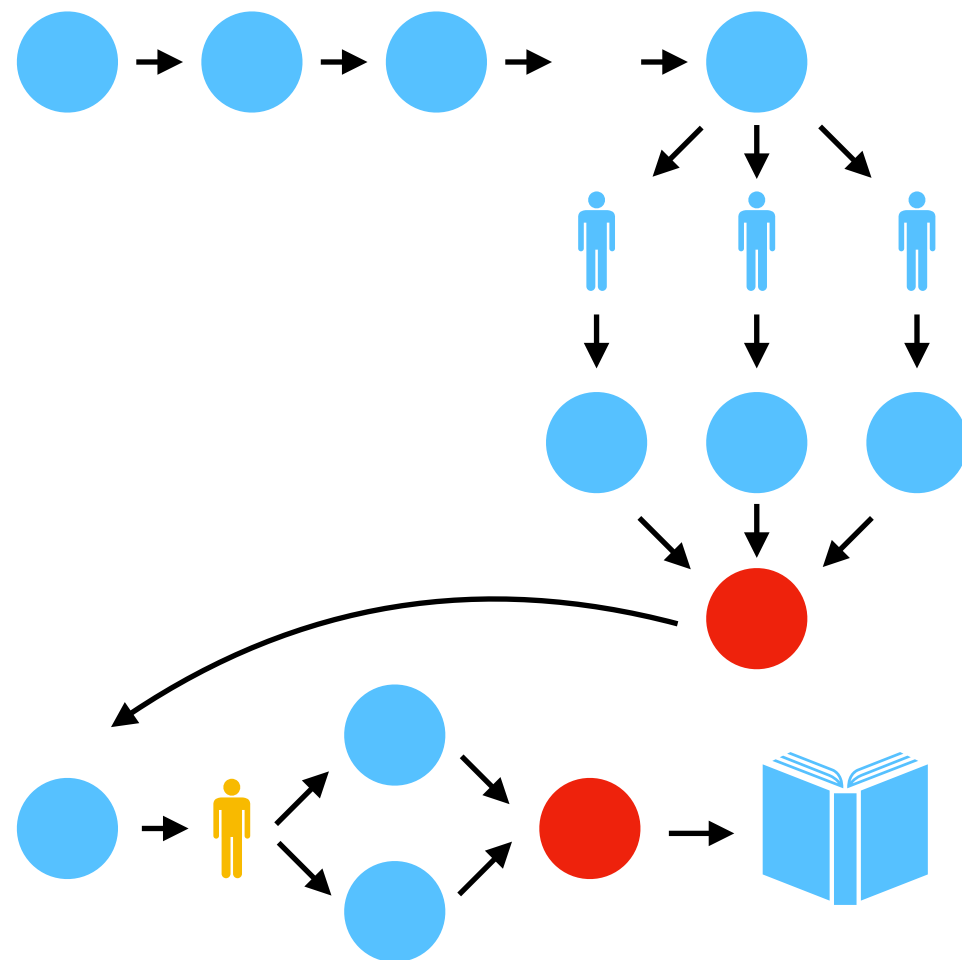
Version control: a collection of patches



Version control: a collection of patches



manuscript.txt



P_{0-1}

P_{1-2}

P_{2-3}

P_{XY}

P_{10-MS}

P_{10-LN}

P_{10-BR}

P_{11-12}

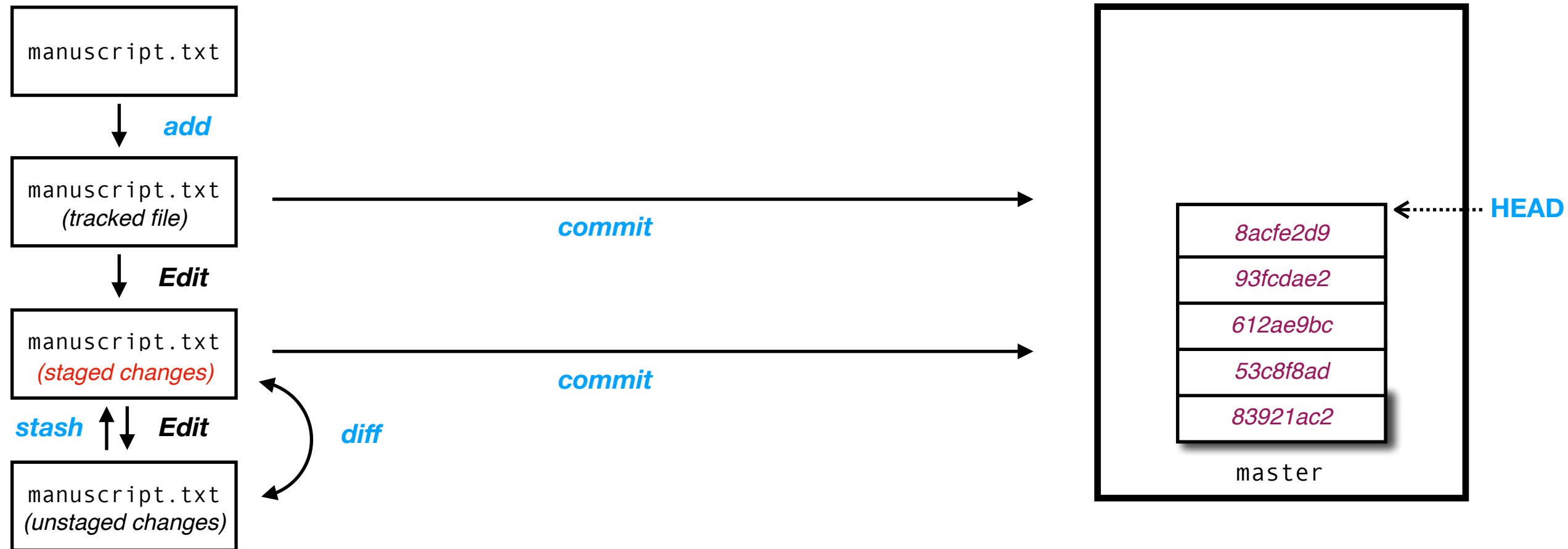
P_{12-JS1}

P_{12-JS2}

P_{13-F}

GIT: A ~~patches~~ organiser

Version Control System



```
> git add manuscript.txt
> git commit -m "Added the empty file"
> git diff
> git stash
> git status
> git checkout HEAD~2
```

GIT: A Version Control System



```
[mia → vcclass git:(master) ✕ git add manuscript.txt
[mia → vcclass git:(master) ✕ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   manuscript.txt

[mia → vcclass git:(master) ✕ git commit -am "Added the empty file"
[master (root-commit) a145144] Added the empty file
 1 file changed, 1 insertion(+)
 create mode 100644 manuscript.txt
[mia → vcclass git:(master) git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
[mia → vcclass git:(master) git diff
[mia → vcclass git:(master) echo 'First line' >> manuscript.txt
[mia → vcclass git:(master) ✕ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

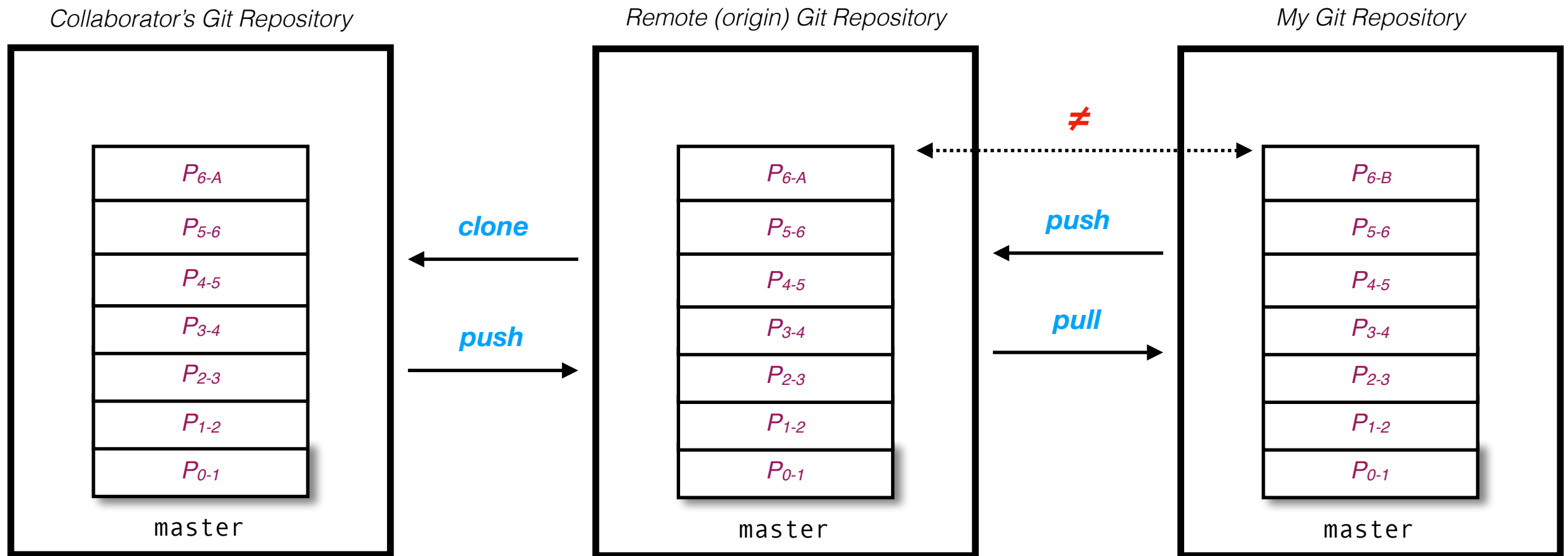
    modified:   manuscript.txt

no changes added to commit (use "git add" and/or "git commit -a")
[mia → vcclass git:(master) ✕ git diff
[mia → vcclass git:(master) ✕ git stash
Saved working directory and index state WIP on master: a145144 Added the empty file
```

```
> git add manuscript.txt
> git commit -m "Added the empty file"
> git diff
> git stash
> git status
> git checkout HEAD~2
```

```
diff --git a/manuscript.txt b/manuscript.txt
index 8b13789..701526b 100644
--- a/manuscript.txt
+++ b/manuscript.txt
@@ -1,2 @@
+First line
(END)
```

GIT: A *distributed* Version Control System



- > `git push origin master`
- > `git clone https://github.com/fraimondo/vcclass.git`
- > `git pull origin master`

GIT: conflict resolution



```
From https://github.com/fraimondo/vcclass
* branch          master      -> FETCH_HEAD
   29aba7c..dabb4c8  master      -> origin/master
Auto-merging mars.txt
CONFLICT (content): Merge conflict in manuscript.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
1  This is a text that I wrote over a long period of time.
2
3  Unfortunately, there are some mistakes. For example, in this
4
5  Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
6  <<<<<<< HEAD (Current Change)
7  paragraph, I will forget one word that is important to understand the key concept.
8  =====
9  paragraph, I will forget one that is of utter importance to interpret
10 the main notion.
11 >>>>>>> dabb4c8c450e8475aee9b14b4383acc99f42af1d (Incoming Change)
12
13 In this other paragraph, I should write the conclusions of the paper. I
14 should also depict the importance of this work and why I should be awarded
15 with the annual price of the Association for Computing Machinery also known as
16 Turing Award. However, I have a typo that will cool of the reviewers.
```

```
> git add manuscript.txt
> git commit -am "Integrated comments"
> git push origin master
```

What else can we do with git?



- ▶ Find who did what: `blame`
- ▶ Try something “big” without ruining master: `branch`
- ▶ Save the revision with a better name: `tag`
- ▶ Find where we broke the code: `bisect`
- ▶ And of course, a big mess! Do not hesitate to contact the community.

Resources



- ▶ Hosting:
 - GitHub: <https://github.com>
 - GitLab: <https://gitlab.uliege.be>
- ▶ Tutorials/Lectures:
 - A 3h lecture by Elizabeth Dupre: <https://elizabeth-dupre.com/git-course/>
 - Software carpentry on Git: <https://swcarpentry.github.io/git-novice/>
 - Mozilla on Git: <https://mozilla.github.io/open-leadership-training-series/articles/get-your-project-online/introducing-github-for-collaborative-work-and-version-control/>
 - Git by Kirstie Whitaker: <https://kirstiejane.github.io/friendly-github-intro/>

Hands-on!



1. Create a GitHub account and tell me the username *.
2. Clone <https://github.com/fraimondo/vcclass.git>
3. Create a file “[NAME].m” inside and add some MATLAB code **.
4. Edit the file “functions.m” and replace the content of the function to return your name.
5. Make sure that all the changes are now safe and versioned in the repository.

* I will invite you to collaborate

** replace [NAME] with your name



GitHub/GitLab

