

GIT - Global Information Tracker

Gergely Fekete

2022.06.01 - Lab Seminar

Features - What git can do?

- security copies
- synchronisation between machines
- work together
- version tracking

-  merged_operetta_layout_20200215.csv
-  merged_operetta_layout_20200210.csv
-  merged_operetta_layout_20200204.csv
-  merged_operetta_layout_20200115.csv
-  merged_operetta_layout_20200113-tmp.xlsx
-  merged_operetta_layout_20200113.csv
-  merged_operetta_layout_20191217-tmp.xlsx
-  merged_operetta_layout_20191217.csv
-  merged_operetta_layout_20191216.csv
-  merged_operetta_layout_20191213.csv
-  merged_operetta_layout_20191129.csv

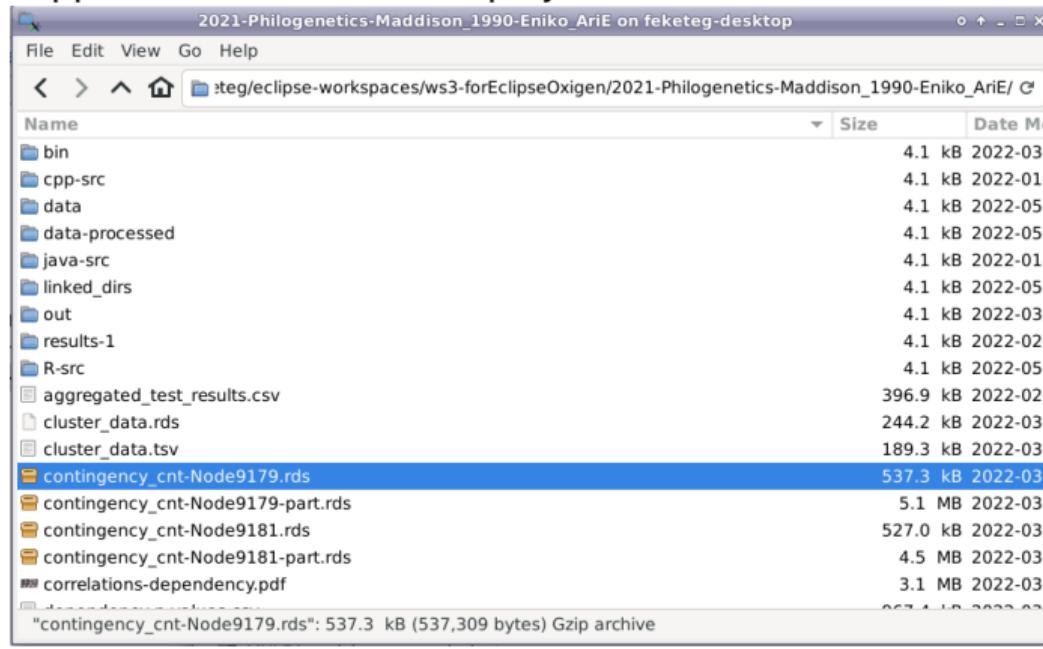




IDEA

Let's desing a version tracker

suppose that . . . we have a project folder and we want to desing a version tracking workflow

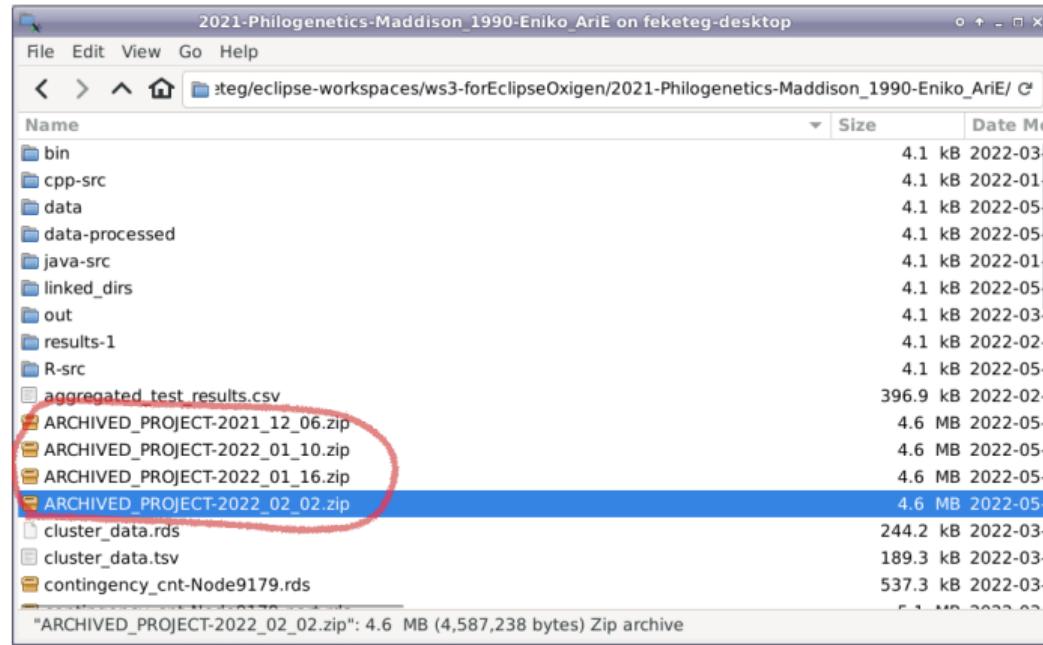


A screenshot of a file explorer window titled "2021-Philogenetics-Maddison_1990-Eniko_AriE on feketeg-desktop". The window shows a list of files and folders in a tree view. The root folder contains subfolders like bin, cpp-src, data, data-processed, java-src, linked_dirs, out, results-1, and R-src. Below these are several data files: aggregated_test_results.csv, cluster_data.rds, cluster_data.tsv, contingency_cnt-Node9179.rds, contingency_cnt-Node9179-part.rds, contingency_cnt-Node9181.rds, contingency_cnt-Node9181-part.rds, and correlations-dependency.pdf. The file "contingency_cnt-Node9179.rds" is selected and highlighted with a blue background. A status bar at the bottom indicates: "contingency_cnt-Node9179.rds: 537.3 kB (537,309 bytes) Gzip archive".

Name	Size	Date
bin	4.1 kB	2022-03-
cpp-src	4.1 kB	2022-01-
data	4.1 kB	2022-05-
data-processed	4.1 kB	2022-05-
java-src	4.1 kB	2022-01-
linked_dirs	4.1 kB	2022-05-
out	4.1 kB	2022-03-
results-1	4.1 kB	2022-02-
R-src	4.1 kB	2022-05-
aggregated_test_results.csv	396.9 kB	2022-02-
cluster_data.rds	244.2 kB	2022-03-
cluster_data.tsv	189.3 kB	2022-03-
contingency_cnt-Node9179.rds	537.3 kB	2022-03-
contingency_cnt-Node9179-part.rds	5.1 MB	2022-03-
contingency_cnt-Node9181.rds	527.0 kB	2022-03-
contingency_cnt-Node9181-part.rds	4.5 MB	2022-03-
correlations-dependency.pdf	3.1 MB	2022-03-



The Snapshots

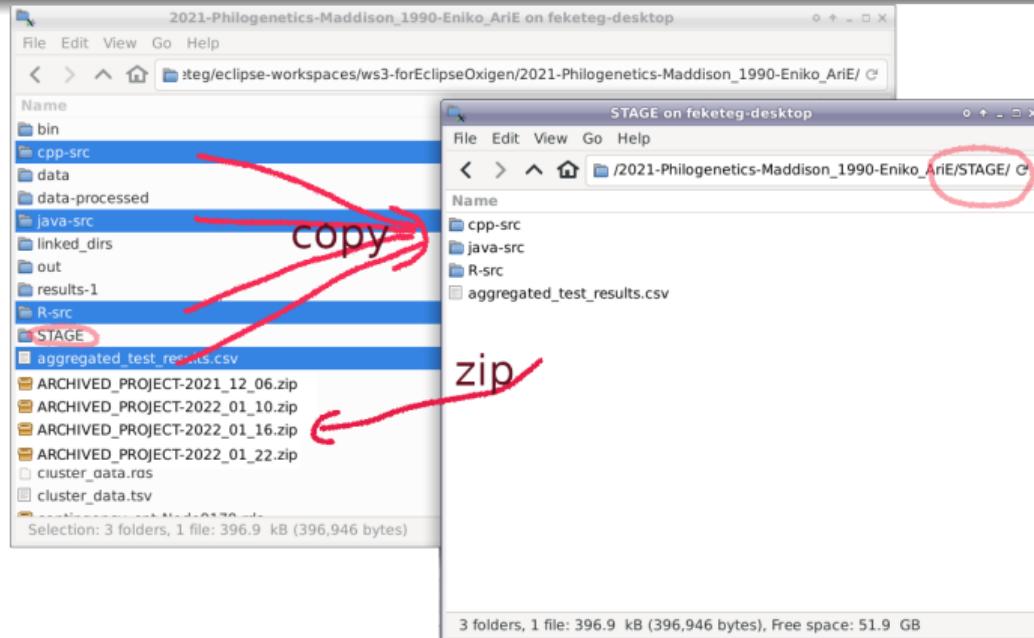


Name	Size	Date Modified
bin	4.1 kB	2022-03-
cpp-src	4.1 kB	2022-01-
data	4.1 kB	2022-05-
data-processed	4.1 kB	2022-05-
java-src	4.1 kB	2022-01-
linked_dirs	4.1 kB	2022-05-
out	4.1 kB	2022-03-
results-1	4.1 kB	2022-02-
R-src	4.1 kB	2022-05-
aggregated_test_results.csv	396.9 kB	2022-02-
ARCHIVED_PROJECT-2021_12_06.zip	4.6 MB	2022-05-
ARCHIVED_PROJECT-2022_01_10.zip	4.6 MB	2022-05-
ARCHIVED_PROJECT-2022_01_16.zip	4.6 MB	2022-05-
ARCHIVED_PROJECT-2022_02_02.zip	4.6 MB	2022-05-
cluster_data.rds	244.2 kB	2022-03-
cluster_data.tsv	189.3 kB	2022-03-
contingency_cnt-Node9179.rds	537.3 kB	2022-03-

"ARCHIVED_PROJECT-2022_02_02.zip": 4.6 MB (4,587,238 bytes) Zip archive

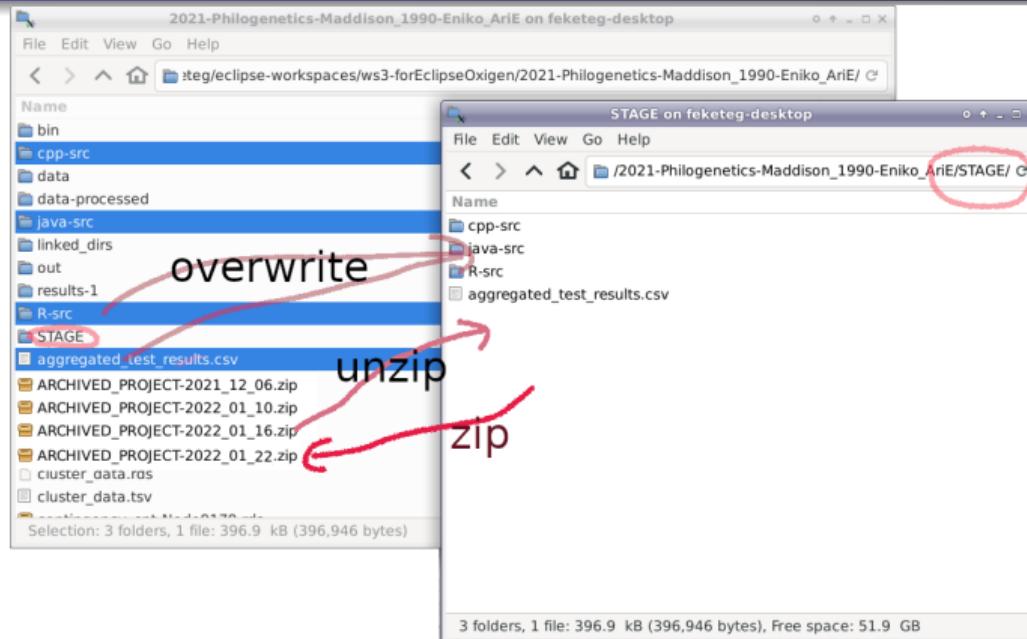


The Stage



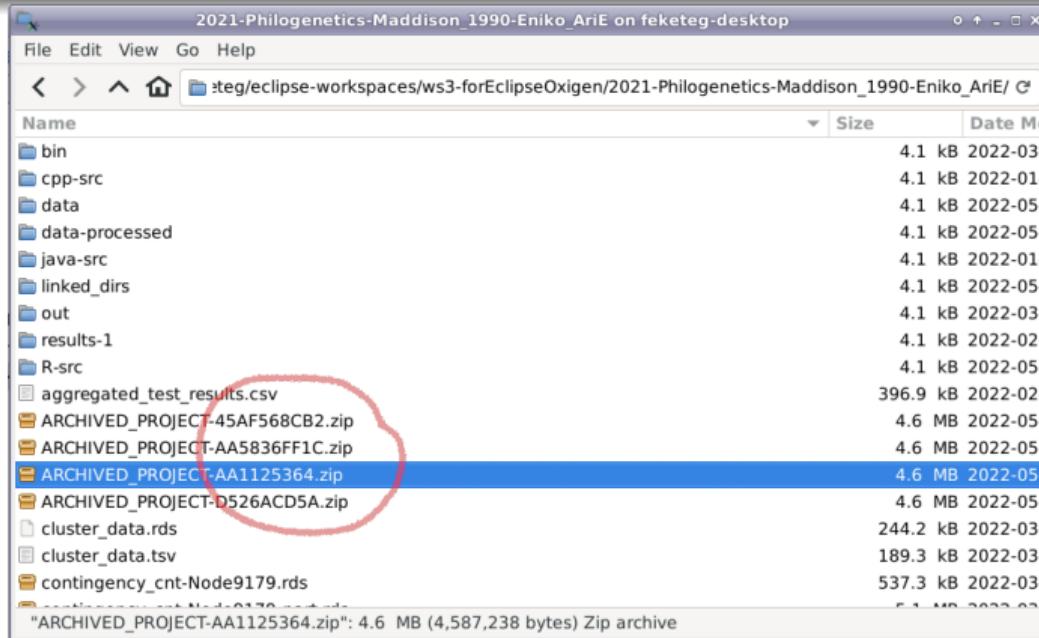
create an empty stage -> copy what you need to the stage -> zip the stage -> delete the stage

The Stage



create an empty stage -> extract there the previous zip -> overwrite some files from the actual project folder -> zip the stage -> delete the stage

The Hash



Name	Size	Date
bin	4.1 kB	2022-03
cpp-src	4.1 kB	2022-01
data	4.1 kB	2022-05
data-processed	4.1 kB	2022-05
java-src	4.1 kB	2022-01
linked_dirs	4.1 kB	2022-05
out	4.1 kB	2022-03
results-1	4.1 kB	2022-02
R-src	4.1 kB	2022-05
aggregated_test_results.csv	396.9 kB	2022-02
ARCHIVED_PROJECT-45AF568CB2.zip	4.6 MB	2022-05
ARCHIVED_PROJECT-AA5836FF1C.zip	4.6 MB	2022-05
ARCHIVED_PROJECT-AA1125364.zip	4.6 MB	2022-05
ARCHIVED_PROJECT-D526ACD5A.zip	4.6 MB	2022-05
cluster_data.rds	244.2 kB	2022-03
cluster_data.tsv	189.3 kB	2022-03
contingency_cnt-Node9179.rds	537.3 kB	2022-03
"ARCHIVED_PROJECT-AA1125364.zip": 4.6 MB (4,587,238 bytes) Zip archive		

Trivial naming methods causes name conflicts.

Hash is a unique hexadecimal number



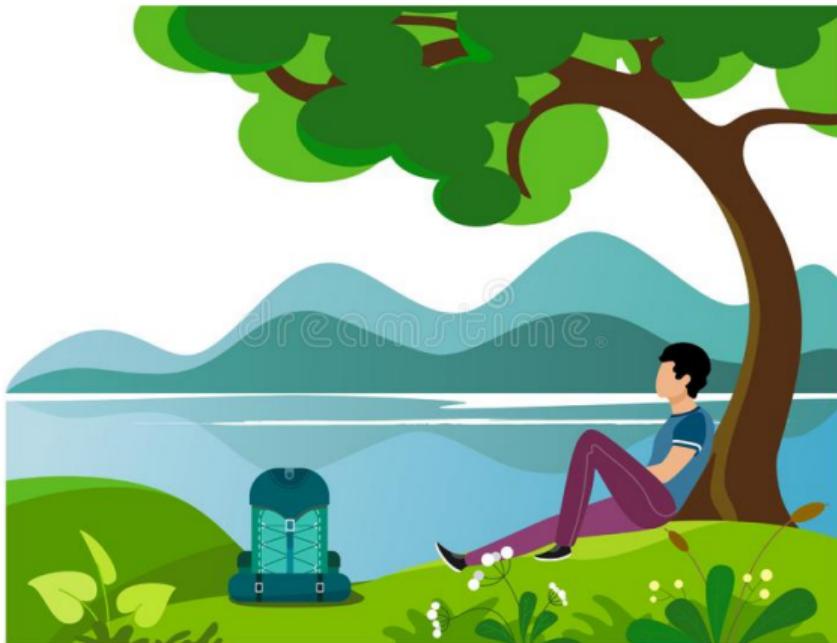
The Log and the Commit Message and The Commit

hash	author	date	message
4649520675e14bea	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 18:55:34 2020 +0100	20190717 no operetta info
280dbd863f1d5274	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 18:53:18 2020 +0100	new plates added + fix
01fc0c3f0eb2583d	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 18:49:16 2020 +0100	fix in WP-bakers-plate1
5f3541ba26ced7d5	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 17:57:12 2020 +0100	mol new columns clade,clinical isolate,control
07d502c30a444b68	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 17:53:10 2020 +0100	introduce merged operetta layout file
f190947bc82aa557	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 15:15:24 2020 +0100	fix formating again
54a51ca076f664c1	Fekete Gergo <fekger@gmail.com>	Tue Feb 25 15:11:11 2020 +0100	fix formating error
b8bfb3b9388b910e	Fekete Gergo <gergo@desktop1>	Tue Feb 25 15:03:48 2020 +0100	first files

The Log and the Commit Message and The Commit

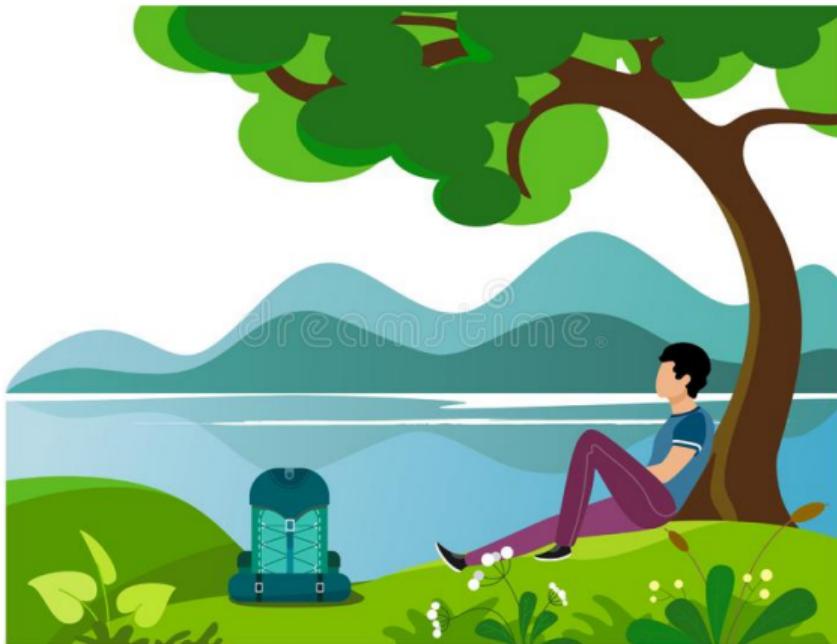
A **commit** means creating a (1) snapshot and (2) note it to the log table and (3) a reference to the parent(s)

Summarise



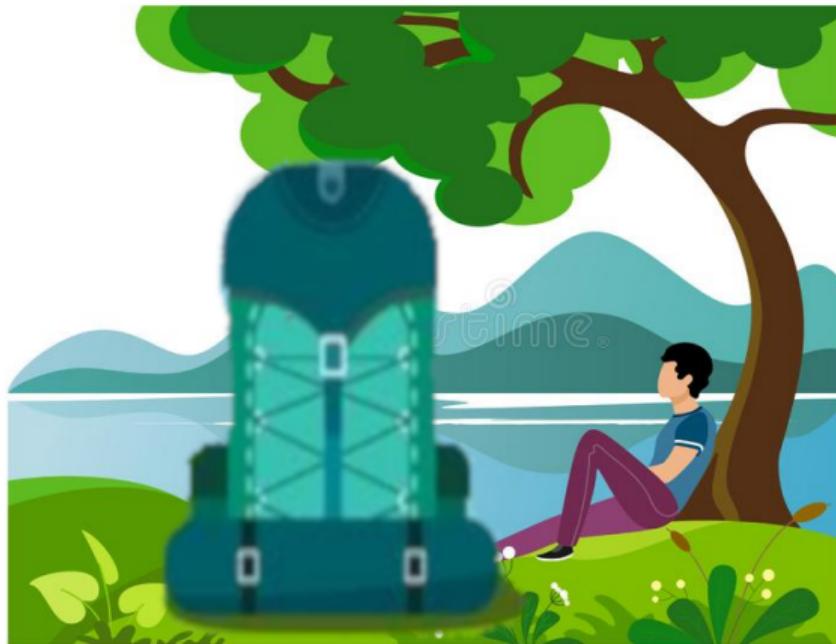
- stage
- snapshot
- commit
- hash
- log
- **repository**
- git status
- git add
- git commit
- git log

Summarise



- stage
- snapshot
- commit
- hash
- log
- **repository**
- git status
- git add
- git commit
- git log

Resource Requirements



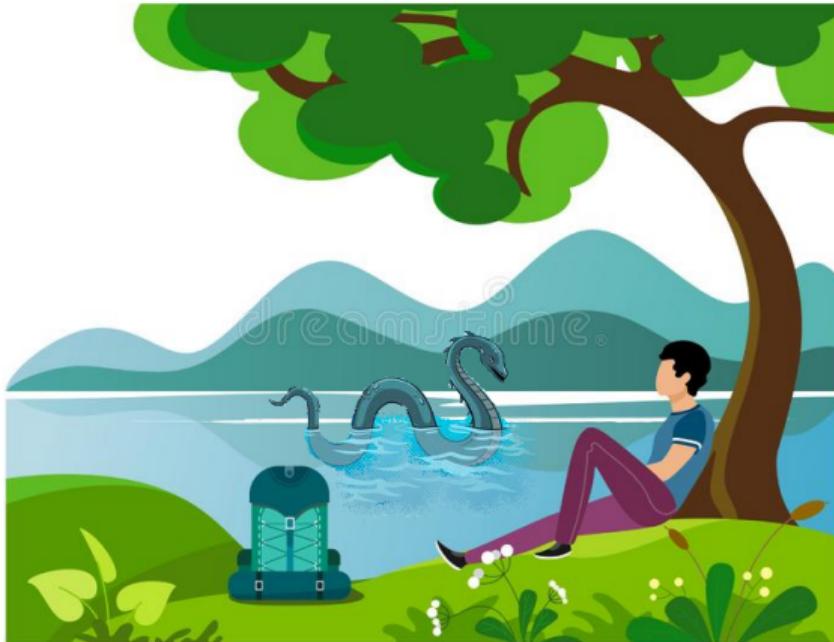
Does the lot of snapshots need a lot of space?

No

Is it fast?

Yes

The scary part



- how to upload on a server?
- how to download ?
- how to roll back ?
- how to manage the simultaneous work of more users ?
- branching?
- which is the current snapshot?

The not so scary part

how to roll back to an older version?

- git checkout extract the zip you want

how to upload on a server?

- git push simple upload all zip-s
- some of the zip-s has a copy on the server yet. The hash based name helps.
- commit first, then push

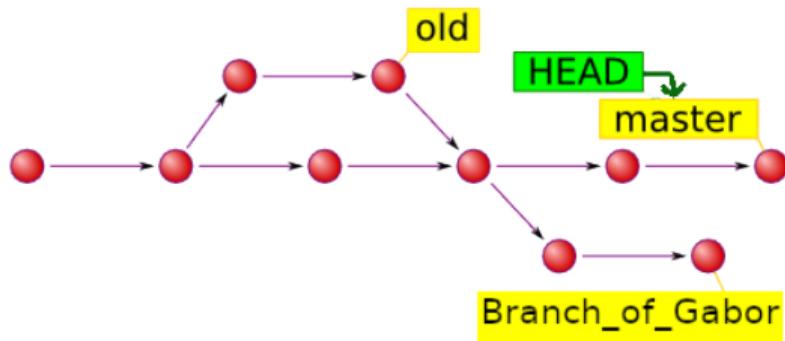
how to download from a server?

- simple download zip-s you don't have yet
- git fetch only download zips
- git pull download zips, and extract the **current**





That is the question:
which is the current snapshot?

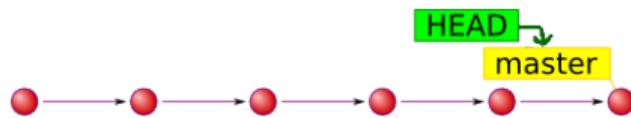


That is the question:
which is the current snapshot?

- red: commit objects : snapshots + ...
- yellow: Branches = pointers to commit objects
- green: the HEAD = pointer to pointer

- Different snapshot is current for you and for the co-workers.
- the HEAD points to the current *branch* and it points to the current snapshot
- how is it managed ?

Situation 1 : The simplest case



- red: commit objects
- yellow: Branches
- green: the HEAD

- If you work alone on only one machine with no rollbacks , no branches
- master branch always points to the current commit
- master steps automatically by commit
- HEAD always points to the master

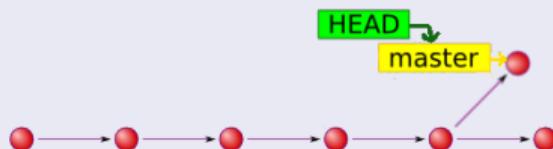
Situation 2 : Rollback



Something is wrong, we step back



The next commit creates a bifurcation; The last commit turns to a dead end

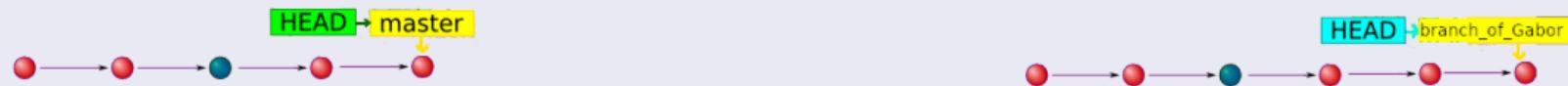


Situation 3 : Working together

Gabor wants to join my work. We create a virtual copy of all files: a branch



We both are working on two separate machines, separate working copies



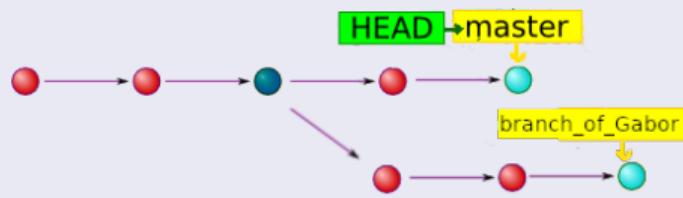
We synchronise : push, push, fetch, fetch



There is only one head per machine!



Merge



Merge

```

#1/usr/bin/python3
from itertools import product
from random import choice
from time import sleep

class GameOfLife(object):
    def __init__(self, width, height):
        self.size = (width, height)
        self.random_world()

    def __str__(self):
        width, height = self.size
        return '\n'.join(
            ''.join(
                self.draw_cell(x, y) for x in range(width)
            ) for y in range(height)
        )

    def __iter__(self):
        return self

    def __next__(self):
        self.evolve_world()
        return self

    def evolve_cell(self, cell):
        alive = cell in self.live_cells
        neighbours = self.count_neighbours(cell)
        return neighbours == 3 or (alive and neighbours == 2)

    def count_neighbours(self, cell):
        x, y = cell
        deltas = set(product([-1, 0, 1], repeat=2)) - set((0, 0))
        neighbours = [(x + dx, y + dy) for (dx, dy) in deltas]
        return sum(neighbour in self.live_cells for neighbour in neighbours)

    def evolve_world(self):
        width, height = self.size
        world = product(range(width), range(height))
        self.live_cells = {cell for cell in world if self.evolve_cell(cell)}
        self.random_world()

    def random_world(self):
        width, height = self.size

```



```

from itertools import product
from random import choice
from time import sleep

class GameOfLife:
    """Conway's Game of Life"""

    def __init__(self, width, height):
        self.size = (width, height)
        self.random_world()

    def __str__(self):
        width, height = self.size
        return '\n'.join(
            ''.join(
                self.draw_cell(x, y) for x in range(width)
            ) for y in range(height)
        )

    def __iter__(self):
        return self

    def __next__(self):
        self.evolve_world()
        return self

    def evolve_cell(self, cell):
        alive = cell in self.live_cells
        neighbours = self.count_neighbours(cell)
        return neighbours == 3 or (neighbours == 2 and alive)

    def count_neighbours(self, cell):
        x, y = cell
        deltas = set(product([-1, 0, 1], repeat=2)) - set((0, 0))
        neighbours = [(x + dx, y + dy) for (dx, dy) in deltas]
        return sum(neighbour in self.live_cells for neighbour in neighbours)

    def evolve_world(self):
        width, height = self.size
        world = product(range(width), range(height))
        self.live_cells = {cell for cell in world if self.evolve_cell(cell)}

```



```

from itertools import product
from random import choice
from time import sleep

class GameOfLife:
    """Conway's Game of Life
    by Ben Nuttall
    """

    def __init__(self, width, height):
        self.size = (width, height)
        self.random_world()

    def __str__(self):
        width, height = self.size
        return '\n'.join(
            ''.join(
                self.draw_cell(x, y) for x in range(width)
            ) for y in range(height)
        )

    def __iter__(self):
        # make the class iterable
        return self

    def __next__(self):
        self.evolve_world()
        return self

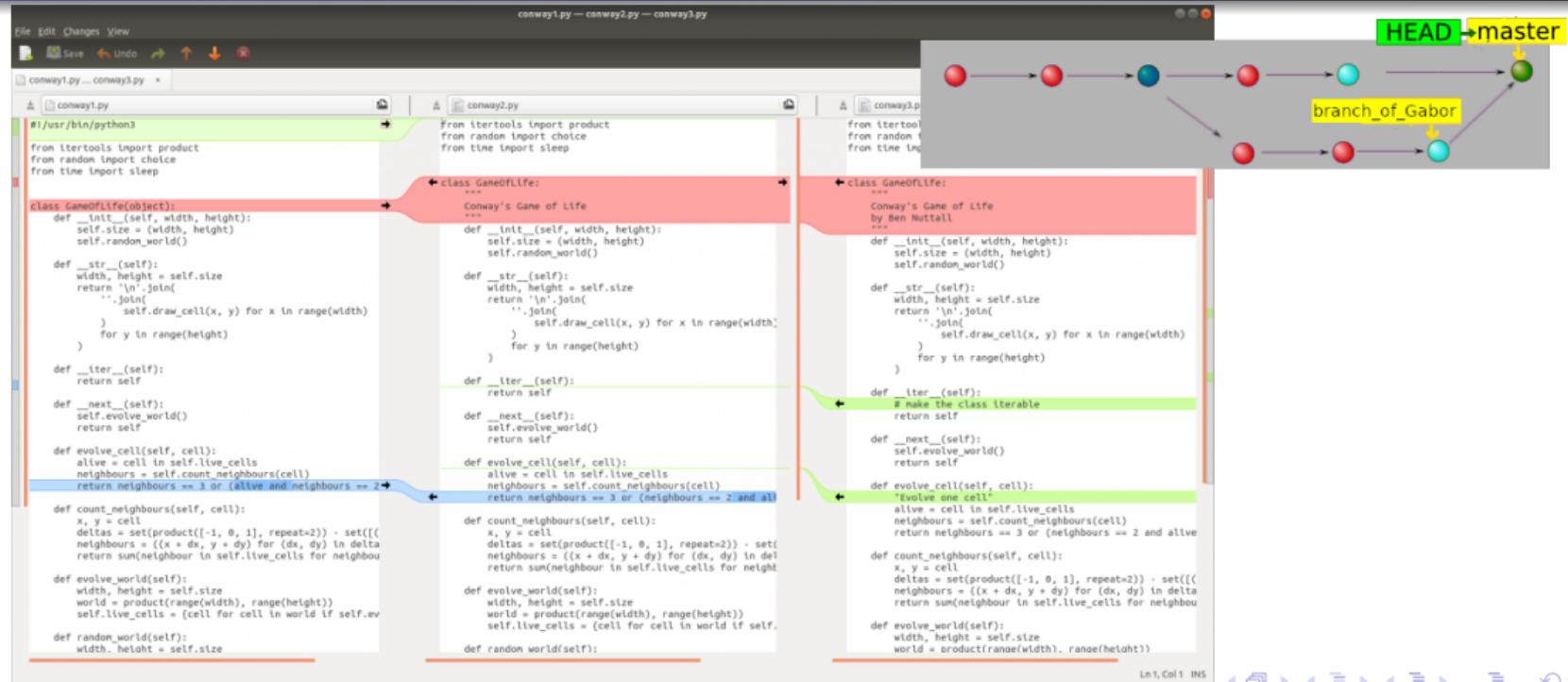
    def evolve_cell(self, cell):
        """Evolve one cell"""
        alive = cell in self.live_cells
        neighbours = self.count_neighbours(cell)
        return neighbours == 3 or (neighbours == 2 and alive)

    def count_neighbours(self, cell):
        x, y = cell
        deltas = set(product([-1, 0, 1], repeat=2)) - set((0, 0))
        neighbours = [(x + dx, y + dy) for (dx, dy) in deltas]
        return sum(neighbour in self.live_cells for neighbour in neighbours)

    def evolve_world(self):
        width, height = self.size
        world = product(range(width), range(height))
        self.live_cells = {cell for cell in world if self.evolve_cell(cell)}

```

Merge



The image shows a code editor interface with three tabs: `conway1.py`, `conway2.py`, and `conway3.py`. The code in each tab is identical, representing a Conway's Game of Life implementation.

```


```

#!/usr/bin/python3
from itertools import product
from random import choice
from time import sleep

class GameOfLife(object):
 def __init__(self, width, height):
 self.size = (width, height)
 self.random_world()

 def __str__(self):
 width, height = self.size
 return '\n'.join(
 ''.join(
 self.draw_cell(x, y) for x in range(width)
) for y in range(height)
)

 def __iter__(self):
 return self

 def __next__(self):
 self.evolve_world()
 return self

 def evolve_cell(self, cell):
 alive = cell in self.live_cells
 neighbours = self.count_neighbours(cell)
 return neighbours == 3 or (alive and neighbours == 2)

 def count_neighbours(self, cell):
 x, y = cell
 deltas = set(product([-1, 0, 1], repeat=2)) - set([(0, 0)])
 neighbours = [(x + dx, y + dy) for (dx, dy) in deltas]
 return sum(neighbour in self.live_cells for neighbour in neighbours)

 def evolve_world(self):
 width, height = self.size
 world = product(range(width), range(height))
 self.live_cells = {cell for cell in world if self.evolve_cell(cell)}
 self.world = world

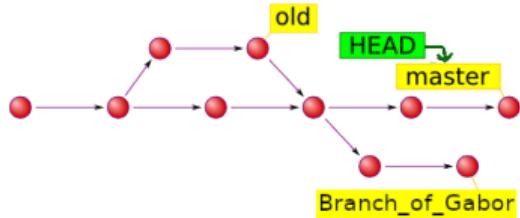
 def random_world(self):
 width, height = self.size

```

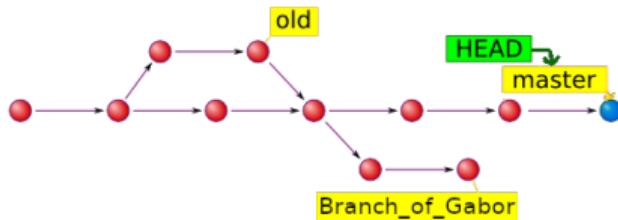

```

On the right side of the interface, there is a graphical representation of a branching history. It shows a main horizontal line representing the `master` branch. A green circle at the end of this line is labeled `HEAD`. A yellow circle on the line is labeled `branch_of_Gabor`. A purple arrow points from the `branch_of_Gabor` node to a red circle, which then has a purple arrow pointing to a blue circle. The blue circle is connected by a purple arrow to a red circle, which is then connected by a purple arrow to a cyan circle. Finally, a purple arrow points from the cyan circle to a green circle, which is labeled `HEAD`.

Branches and the HEAD pointer



Effect of git commit



- default branch name is master (can be renamed)
- branches are moving pointers
- current branch steps automatically by commit
- HEAD points to the current branch
- the new commit object will know which one is its parent

GITG Screenshot

The screenshot shows the Eclipse IDE interface with a Git commit history and a code diff view.

Git Commit History:

- All commits
- Branches:**
 - master (2 ahead)
 - origin/master
 - v1.0 (x)
- Remotes:**
 - origin (master)
- Tags:**

Commits listed from top to bottom:

- case study slides added (feketegergo, A minute ago)
- I created the first 15 slides of the new pdf (feketegergo, 15 hours ago)
- I checked out the github project into a new eclipse p (feketegergo, 22 hours ago)
- x (feketegergo, Feb 26 2020, 1...)
- Merge branch 'master' of https://github.com/feketegergo/present. Fekete Gergo (feketegergo, Feb 26 2020, 1...)
- new image (Fekete Gergo, Feb 26 2020, 1...)
- new screenshot (feketegergo, Feb 26 2020, 1...)
- Screenshots (Fekete Gergo, Feb 26 2020, 0...)
- ??? (Fekete Gergo, Feb 25 2020, 1...)
- ??? (Fekete Gergo, Feb 25 2020, 1...)
- _20190717 no operetta info (Fekete Gergo, Feb 25 2020, 1...)
- new plates added + fix (Fekete Gergo, Feb 25 2020, 1...)
- fix in WP-bakers-plate1 (Fekete Gergo, Feb 25 2020, 1...)
- mol.new.columns.clade.clinical.isolate.control (Fekete Gergo, Feb 25 2020, 1...)

Code Diff View:

```
diff --git a/  b/
--- a/  ...
+++ b/
@@ -50,5 +50,5 @@
 50  - \date{2020.02.26 - Lab Meeting}
 54 + \date{2022.06.01 - Lab Seminar}
 55 
 56 
 53 57 % This is only inserted into the PDF information catalog. Can be left
 54 58 % out.
 55 
 59 - \subject{GIT and GitHub}
 59 + \subject{GIT the Global Information Tracker}
```

GITG Screenshot- Commit Window

The screenshot shows the GITG commit window interface. On the left, there are three sections: **Staged** (empty), **Unstaged** (containing document.pdf and document.tex), and **Untracked** (containing pictures/gitg-screenshot-1.png). The **Unstaged** section is currently selected. The right pane displays the content of document.tex. A specific line of code is highlighted with a green background:

```
@@ -457,6 +457,15 @@ Yes
\end{frame}

...
457 457
458 458
459 459
460 +
461 + \begin{frame}
462 +   \frametitle<presentation>{GITG Screenshot}
463 +
464 +   \includegraphics[height=180pt]{pictures/gitg-
screenshot-1.png}
465 +
466 + \end{frame}
467 +
468 +
469 469
470 470
471 \begin{frame}
```

At the bottom of the window, there are buttons for Skip commit hooks, Discard selection, Stage selection, and Commit. Below the window, there is a toolbar with various icons.

GITG Screenshot- Commit Window

```
>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   document.pdf
    modified:   document.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    pictures/git-head+after-commit.png
    pictures/git-head-1.png
    pictures/git-mater-head-rollback-and-branch.png
    pictures/git-mater-head-rollback.png
    pictures/git-mater-head-simplest.png
    pictures/git-mater-working-simultanously-1.png
    pictures/git-mater-working-simultanously-2.png
    pictures/git-mater-working-simultanously-3.png
    pictures/git-mater-working-simultanously-4.png
    pictures/project_folder_stageing-2.png
    pictures/to-be-or-not-to-be-that-is-the-question.jpg

no changes added to commit (use "git add" and/or "git commit -a")
```

GIT

GIT is a tool to save folders , and handle versions

GitHub

GitHub is a webserver where you can save

- GIT works without GitHub
- GitHub is only one of the many git servers
- Git servers adds the ability to share your stuff

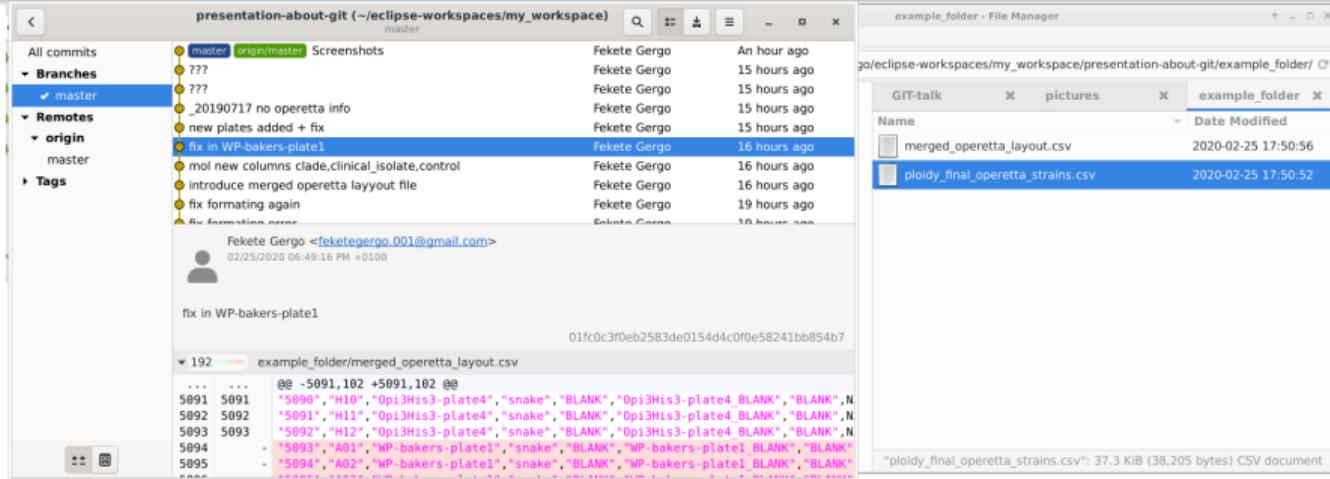
What are versions?

- Sometimes we edit a file continuously and want to keep its earlier versions

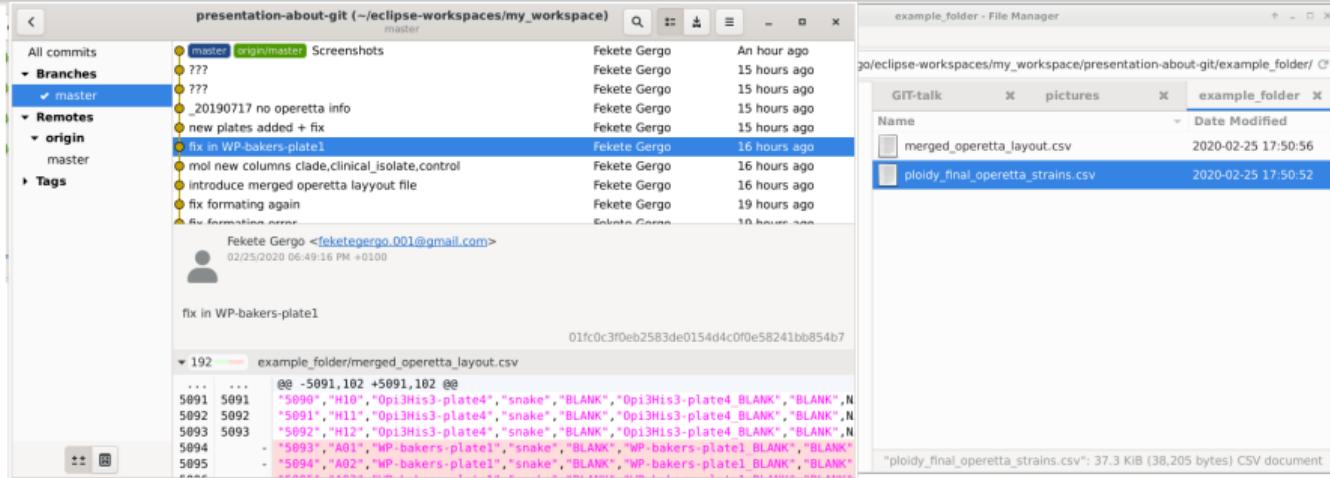
merged_operetta_layout_20200215.csv
 merged_operetta_layout_20200210.csv
 merged_operetta_layout_20200204.csv
 merged_operetta_layout_20200115.csv
 merged_operetta_layout_20200113-tmp.xls
 merged_operetta_layout_20200113.csv
 merged_operetta_layout_20191217-tmp.xlsx
 merged_operetta_layout_20191217.csv
 merged_operetta_layout_20191216.csv
 merged_operetta_layout_20191213.csv
 merged_operetta_layout_20191129.csv

- the state of the art solution
 - have one file in the working directory
 - store the old versions 'hidden' in a **repository**
- What is a **repository**?
 - a simple subfolder
 - The folder name is '.git'.
 - It is a hidden folder
 - You have to start git to see the content

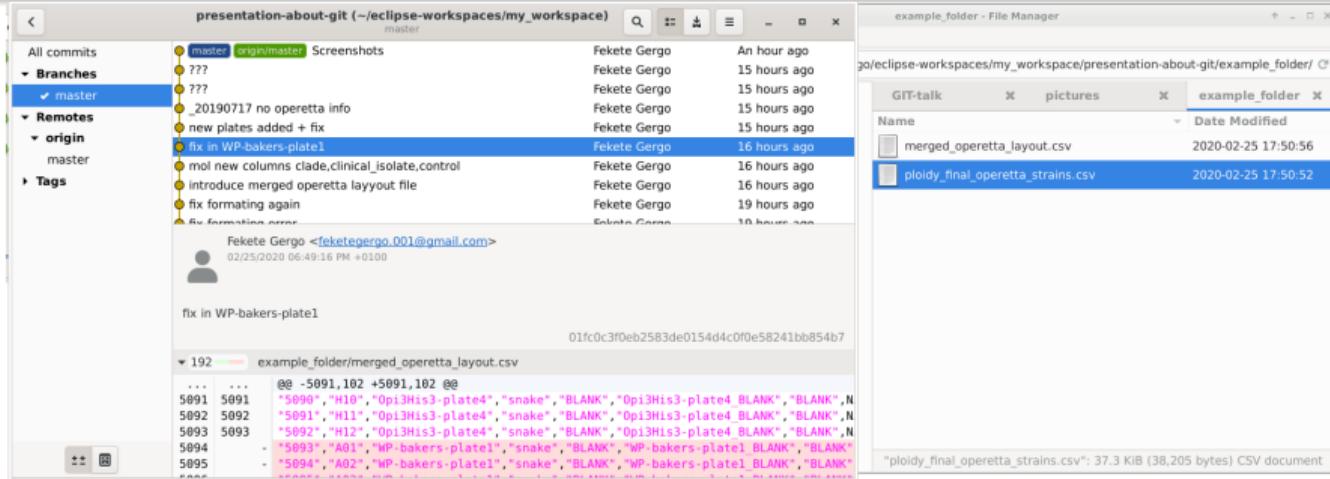
- the state of the art solution
 - have one file in the working directory
 - store the old versions 'hidden' in a **repository**
- What is a **repository**?
 - a simple subfolder
 - The folder name is '.git'.
 - It is a hidden folder
 - You have to start git to see the content



- normally you see only the 2 important files
- If you need the old versions you can turn on the repository browser.
- Each ball represents a previous version
- the term of the 'balls' is commit/revision/version
- You can delete files from the working directory. The repo keeps it.



- normally you see only the 2 important files
- If you need the old versions you can turn on the repository browser.
- Each ball represents a previous version
- the term of the 'balls' is commit/revision/version
- You can delete files from the working directory. The repo keeps it.



- normally you see only the 2 important files
- If you need the old versions you can turn on the repository browser.
- Each ball represents a previous version
- the term of the 'balls' is commit/revision/version
- You can delete files from the working directory. The repo keeps it.

Name	Size	Date Modified
plody_final_operetta_stain_190807_mod2_verGenTmp.csv	36.6 KIB	2019-12-20 22:45:22
plody_final_operetta_stain_190807_mod2.csv	37.0 KIB	2019-10-23 16:35:17
plody_final_operetta_stain_190807_mod.csv	37.3 KIB	2019-08-07 18:40:33
merged_operetta_layout_20200224.csv	3.7 MB	2020-02-24 16:53:11
merged_operetta_layout_20200215.csv	3.7 MB	2020-02-15 20:23:35
merged_operetta_layout_20200210.csv	3.6 MB	2020-02-10 13:24:00
merged_operetta_layout_20200204.csv	3.6 MB	2020-02-04 16:04:49
merged_operetta_layout_20200115.csv	3.7 MB	2020-01-15 15:09:45
merged_operetta_layout_20200113-tmp.xls	4.0 MB	2020-01-14 09:14:32
merged_operetta_layout_20200113.csv	2.9 MB	2020-01-13 18:07:00
merged_operetta_layout_20191217-tmp.xls	1018.0 KIB	2019-12-18 10:53:08
merged_operetta_layout_20191217.csv	2.4 MB	2019-12-17 12:02:21
merged_operetta_layout_20191216.csv	2.4 MB	2019-12-16 13:11:23
merged_operetta_layout_20191213.csv	2.4 MB	2019-12-13 13:30:20
merged_operetta_layout_20191229.csv	2.4 MB	2019-11-29 16:36:46
merged_operetta_layout_20190807.csv	2.4 MB	2019-08-07 17:49:06
merged_operetta_layout_20190805.csv	2.4 MB	2019-08-05 13:36:59
merged_operetta_layout_20190803.csv	2.4 MB	2019-08-03 14:56:55
merged_operetta_layout_20190724.csv	2.0 MB	2019-07-24 14:36:08
merged_operetta_layout_20190717_no_operetta_info.csv	691.0 KIB	2019-07-17 14:18:48
merged_operetta_layout_20190215.csv	1.6 MB	2019-02-15 17:50:50
merged_operetta_layout_20181122.csv	1.6 MB	2018-11-22 17:44:42
merged_operetta_layout_20181108.csv	1.6 MB	2018-11-10 10:49:19
merged_operetta_layout_20180911.csv	1.3 MB	2018-09-11 13:30:37
celium_periplo_genotype_170616_v5_withKO15.csv	56.7 KIB	2018-11-19 10:08:57
af_strains_morphology_plody.csv	18.6 MB	2019-01-16 16:33:18
af_strains_morphology2.csv	16.0 MB	2018-10-11 14:35:53
af_strains_morphology.csv	8.3 MB	2018-08-28 13:42:32

29 items: 94.3 MB (96,844,291 bytes). Free space: 14.1 TiB

- working with messy folders is slower and confusing
- it causes errors
- It is waste of time and money.

Back to the top

GIT

GIT is a tool to save folders , and handle versions

- now we know what are versions
- Let's see why to save folders instead of files

Why to save folders?

Believe me! It is a result of 35 years of evolution and design.

Why to save folders?

Imagine a project where are

- experimental layout file
- result files from a microscope

They belong together. It is nice to connect them.

- actually it does not save full folder. You can select some files to save together.
- The principal concept is 'commit together what belongs together'

Why to save folders?

Imagine a project where are

- experimental layout file
- result files from a microscope

They belong together. It is nice to connect them.

- actually it does not save full folder. You can select some files to save together.
- The principal concept is 'commit together what belongs together'

Why to save folders?

Imagine a project where are

- experimental layout file
- result files from a microscope

They belong together. It is nice to connect them.

- actually it does not save full folder. You can select some files to save together.
- The principal concept is 'commit together what belongs together'

Back to the top again

GIT

GIT is a tool to save folders , and handle versions

- now we know what are versions
- we understand that commit many files together is clever
- What is the GIT tool?

Back to the top again

GIT

GIT is a tool to save folders , and handle versions

- now we know what are versions
- we understand that commit many files together is clever
- What is the GIT tool?

What is the GIT tool?

- actually git is not one tool: it is a protocol/standard
- There are a lot of git programs you can install.
 - Linux and Mac have preinstalled git
 - RStudio contains a git client
 - every IDE contains a git client (C, JAVA, python editors ...)
 - gitg (graphical UI - linux, windows, mac)
 - git SMC (Windows git client)
 - Git Bash (Windows git terminal)
 - every IDE contains a git client (C, JAVA, python editors ...)

What is the GIT tool?

- actually git is not one tool: it is a protocol/standard
- There are a lot of git programs you can install.
- Linux and Mac have preinstalled git
- RStudio contains a git client
- every IDE contains a git client (C, JAVA, python editors ...)
- gitg (graphical UI - linux, windows, mac)
- git SMC (Windows git client)
- Git Bash (Windows git terminal)
- every IDE contains a git client (C, JAVA, python editors ...)

What is the GIT tool?

- actually git is not one tool: it is a protocol/standard
- There are a lot of git programs you can install.
- Linux and Mac have preinstalled git
- RStudio contains a git client
- every IDE contains a git client (C, JAVA, python editors ...)
- gitg (graphical UI - linux, windows, mac)
- git SMC (Windows git client)
- Git Bash (Windows git terminal)
- every IDE contains a git client (C, JAVA, python editors ...)

Let's see how to use it

The screenshot shows an RStudio interface with the following details:

- File menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools.
- Toolbar:** Go to Reference, Stage, Revert, Ignore.
- Left pane (Code Editor):** An R script titled "script-07-create_plate_map_in...". The code includes sections for graphics.off(), rm(list=ls()), library(Tidyverse), source("script/script-03-tryGLMM/functions.R"), and various data processing steps involving mol_tbl<-read_csv() and mutate() functions. It also includes sections for creating figures about images of WT and GLMM models.
- Middle pane (File Browser):** Shows a tree view of files and folders. The "mol" folder contains many R scripts such as script-04-preprocess_calmolph_outp.R, script-05-GLMM-normalizer.R, and script-06-GLMM-normalizer.R. Other folders like "figures", "histograms", and "plots" are also visible.
- Right pane (File Browser):** A detailed view of the "mol" folder showing files like "script-04-preprocess_calmolph_outp.R" with a size of 3.4 KB and modified on Feb 11, 2020, at 12:25 PM.
- Status Bar:** Natural language support not running in an English locale.

Let's see how to use it

The screenshot shows the RStudio interface with a "Review Changes" window open. The left pane lists staged changes, and the right pane shows a commit message field and a diff viewer.

Changes pane:

- test1.txt
- test2.txt
- tmp1.txt
- tmp5.csv
- info/methodsdba.txt
- M script/script-04-preprocess_calmorph_output/script-07-create_plate_map_from_merged_operetta_layout.R
- M script/script-04-preprocess_calmorph_output/script-08-ploidy.R
- M script/script-04-preprocess_calmorph_output/script-22-create_figure_about_images_of_WT_anc.R
- M script/script-05-GLMM-normalisation/script_03_01-create_joblist.R
- M script/script-05-GLMM-normalisation/script_03_02-fit_GLMM_models.R

Commit message field:

Commit message

Amend previous commit

Show buttons: Staged (radio button selected), Unstaged, Context, Line, Ignore Whitespace, Stage All, Discard All

Diff View:

```
@@ -18,20 +18,24 @@ stopifnot(!any(duplicated( plates_tbl$filename)))
18 18 #####
19 19 #####
20 20 #####
21 21 #####
22 22 #####
23 mol_tbl<-read_csv("/home/feketeg/mounted-shares/raid-of-group/PROJECTS/YeastMorphology-2015-
FarkasZoli_KovacsKaroly/DATA_of_microscope/merged_operetta_layout_20200215.csv",
24   col_types = cols( .default = col_character(), transformant=col_logical()))
23 mol_tbl<-read_csv("/home/feketeg/mounted-shares/raid-of-group/PROJECTS/YeastMorphology-2015-
FarkasZoli_KovacsKaroly/DATA_of_microscope/merged_operetta_layout_20200225.csv",
24   col_types = cols( .default = col_character(),
transformant=col_logical(),is_control_per_plate=col_logical()))
25 mol_tbl<-mol_tbl %>%
26   mutate(Filename= gsub("\\\\$", "", Filename)) %>% # a felesleges backslash karaktereket leveszem
oszlop vegeir ol
27 27   rename(filename=Filename)
28 28
29 29
```

Let's see how to use it

RStudio: Review Changes (on feketep-desktop)

Changes History Filter ▾ Stage Revert Ignore

Commit message

Amend previous commit Commit

```
18 18
19 19 #####
20 20 #####
21 21 #####
22 22 #####
23 mol_tbl<-read_csv("/home/feketep/mounted-shares/raid-of-group/PROJECTS/YeastMorphology-2015-
FarkasZoli_KovacsKaroly/DATA_OF_MICROSCOPE/merged_operetta_layout_20104225.csv",
24 col_types = cols( .default = col_logical(),
23 mol_tbl<-read_csv("/home/feketep/mounted-shares/raid-of-group/PROJECTS/YeastMorphology-2015-
FarkasZoli_KovacsKaroly/DATA_OF_MICROSCOPE/merged_operetta_layout_20104225.csv",
24 col_types = cols( .default = col_character(),
transformation=col_logical(),is_control_per_plate=col_logical()))
25 25 mol_tbl<-mol_tbl %>%
26 26   mutate(Filename=gsub("\\\\\\\\", "", Filenames)) %>% # a felesleges backslash karaktereket levesz
27 27   unzip_vegevol
28 28   rename(filename=Filename)
29 29 %>
```

- select files to the stage
- unfullowed/followed files
- diff-s
- commit msg + button
- push/pull button

Terminology

- commit = save it (to the local repository)
- stage = files selected for save
- push = upload to the server
- pull = download from the server

Let's see how to use it - History

The screenshot shows the RStudio interface with the 'History' tab selected in the top-left corner. The main workspace displays a script titled 'RStudio: Review Changes (on feketegeorge)' containing R code for creating a plate map from merged operetta layout files. The 'GIT' panel on the right shows a local repository named 'Morphology-2018-Karczi' with a single commit from 'feketegeorge' dated 2020-02-18 at 16:08. The commit message is 'A merged_operetta_layout file datamat kovet'. The commit details show the SHA '67002c71' and the author information.

```

RStudio: Review Changes (on feketegeorge)
Changes History master + script-07-create_plate_map_from_merged_operetta_layout.R + 0
Subject: A merged_operetta_layout file datamat kovet
Author: feketegeorge <feketegeorge@desktp1.brc> Date: 2020-02-18 16:08 SHA: 67002c71
Az xisz mentő scriptbe bettem egy gcd hivat, mert OutOfMemory-val néha megszűr.
Beméne fejlesztettem egy hibát az xisz körből, azt javítottam.
nem tudom mit csináljam
A script-07-create_plate_map_from_merged_operetta_layout.R fájban volt egy hiba az fejlesztőről <feketegeorge@desktp1.brc> 2020-02-03
A mergedOperettaLayout file egy csomot valtoztattam es a kod kovette a valtozásokat.
feketegeorge <feketegeorge@desktp1.brc> 2020-01-21
feketegeorge <feketegeorge@desktp1.brc> 2020-01-07
Macsos idén elvan nincs commitok

```

Environment is empty

Commits 1-7 of 7

File Edit Code View Plots Session Help

File Edit Code View Plots Session Help

File Plot Packages Help Viewer

New Folder Delete Rename Home + user-BRANCH Morphology-2018-Karczi script script-04-preprocess_calmorph_output

- script-04-preprocess_calmorph_output.R
- script-04-preprocess_calmorph_output\$script-07-create_plate_map_from_merged_operetta_layout.R
- script-04-preprocess_calmorph_output\$script-08-ploddy.R
- script-04-preprocess_calmorph_output\$script-09-big_tbl.R
- script-04-preprocess_calmorph_output\$script-10-merge.R
- script-05-04-preprocess\$script-03_01-create_glist.R
- script-05-04-preprocess\$script-04-preprocess_calmorph_output.R
- script-05-04-preprocess\$script-05-04-preprocess_calmorph_output.R
- script-05-04-preprocess\$script-06-merge.R
- script-05-04-preprocess\$script-07-merge.R
- script-05-04-preprocess\$script-08-ploddy.R
- script-05-04-preprocess\$script-09-big_tbl.R
- script-05-04-preprocess\$script-10-test_outlier_filter.R
- script-05-04-preprocess\$script-11-test_outlier.R
- script-05-04-preprocess\$script-12-test_outlier.R
- script-05-04-preprocess\$script-13-test_outlier.R
- script-05-04-preprocess\$script-14-test_outlier.R
- script-05-04-preprocess\$script-15-test_outlier.R
- script-05-04-preprocess\$script-16-test_outlier.R
- script-05-04-preprocess\$script-17-test_outlier.R
- script-05-04-preprocess\$script-18-test_outlier.R
- script-05-04-preprocess\$script-19-test_outlier.R
- script-05-04-preprocess\$script-20-test_outlier.R
- script-05-04-preprocess\$script-21-test_outlier.R
- script-05-04-preprocess\$script-22-test_outlier.R
- script-05-04-preprocess\$script-23-test_outlier.R
- script-05-04-preprocess\$script-24-test_outlier.R
- script-05-04-preprocess\$script-25-test_outlier.R
- script-05-04-preprocess\$script-26-test_outlier.R
- script-05-04-preprocess\$script-27-test_outlier.R
- script-05-04-preprocess\$script-28-test_outlier.R
- script-05-04-preprocess\$script-29-test_outlier.R
- script-05-04-preprocess\$script-30-test_outlier.R
- script-05-04-preprocess\$script-31-test_outlier.R
- script-05-04-preprocess\$script-32-test_outlier.R
- script-05-04-preprocess\$script-33-test_outlier.R
- script-05-04-preprocess\$script-34-test_outlier.R
- script-05-04-preprocess\$script-35-test_outlier.R
- script-05-04-preprocess\$script-36-test_outlier.R
- script-05-04-preprocess\$script-37-test_outlier.R
- script-05-04-preprocess\$script-38-test_outlier.R
- script-05-04-preprocess\$script-39-test_outlier.R
- script-05-04-preprocess\$script-40-test_outlier.R
- script-05-04-preprocess\$script-41-test_outlier.R
- script-05-04-preprocess\$script-42-test_outlier.R
- script-05-04-preprocess\$script-43-test_outlier.R
- script-05-04-preprocess\$script-44-test_outlier.R
- script-05-04-preprocess\$script-45-test_outlier.R
- script-05-04-preprocess\$script-46-test_outlier.R
- script-05-04-preprocess\$script-47-test_outlier.R
- script-05-04-preprocess\$script-48-test_outlier.R
- script-05-04-preprocess\$script-49-test_outlier.R
- script-05-04-preprocess\$script-50-test_outlier.R
- script-05-04-preprocess\$script-51-test_outlier.R
- script-05-04-preprocess\$script-52-test_outlier.R
- script-05-04-preprocess\$script-53-test_outlier.R
- script-05-04-preprocess\$script-54-test_outlier.R
- script-05-04-preprocess\$script-55-test_outlier.R
- script-05-04-preprocess\$script-56-test_outlier.R
- script-05-04-preprocess\$script-57-test_outlier.R
- script-05-04-preprocess\$script-58-test_outlier.R
- script-05-04-preprocess\$script-59-test_outlier.R
- script-05-04-preprocess\$script-60-test_outlier.R
- script-05-04-preprocess\$script-61-test_outlier.R
- script-05-04-preprocess\$script-62-test_outlier.R
- script-05-04-preprocess\$script-63-test_outlier.R
- script-05-04-preprocess\$script-64-test_outlier.R
- script-05-04-preprocess\$script-65-test_outlier.R
- script-05-04-preprocess\$script-66-test_outlier.R
- script-05-04-preprocess\$script-67-test_outlier.R
- script-05-04-preprocess\$script-68-test_outlier.R
- script-05-04-preprocess\$script-69-test_outlier.R
- script-05-04-preprocess\$script-70-test_outlier.R
- script-05-04-preprocess\$script-71-test_outlier.R
- script-05-04-preprocess\$script-72-test_outlier.R
- script-05-04-preprocess\$script-73-test_outlier.R
- script-05-04-preprocess\$script-74-test_outlier.R
- script-05-04-preprocess\$script-75-test_outlier.R
- script-05-04-preprocess\$script-76-test_outlier.R
- script-05-04-preprocess\$script-77-test_outlier.R
- script-05-04-preprocess\$script-78-test_outlier.R
- script-05-04-preprocess\$script-79-test_outlier.R
- script-05-04-preprocess\$script-80-test_outlier.R
- script-05-04-preprocess\$script-81-test_outlier.R
- script-05-04-preprocess\$script-82-test_outlier.R
- script-05-04-preprocess\$script-83-test_outlier.R
- script-05-04-preprocess\$script-84-test_outlier.R
- script-05-04-preprocess\$script-85-test_outlier.R
- script-05-04-preprocess\$script-86-test_outlier.R
- script-05-04-preprocess\$script-87-test_outlier.R
- script-05-04-preprocess\$script-88-test_outlier.R
- script-05-04-preprocess\$script-89-test_outlier.R
- script-05-04-preprocess\$script-90-test_outlier.R
- script-05-04-preprocess\$script-91-test_outlier.R
- script-05-04-preprocess\$script-92-test_outlier.R
- script-05-04-preprocess\$script-93-test_outlier.R
- script-05-04-preprocess\$script-94-test_outlier.R
- script-05-04-preprocess\$script-95-test_outlier.R
- script-05-04-preprocess\$script-96-test_outlier.R
- script-05-04-preprocess\$script-97-test_outlier.R
- script-05-04-preprocess\$script-98-test_outlier.R
- script-05-04-preprocess\$script-99-test_outlier.R
- script-05-04-preprocess\$script-100-test_outlier.R
- script-05-04-preprocess\$script-101-test_outlier.R
- script-05-04-preprocess\$script-102-test_outlier.R
- script-05-04-preprocess\$script-103-test_outlier.R
- script-05-04-preprocess\$script-104-test_outlier.R
- script-05-04-preprocess\$script-105-test_outlier.R
- script-05-04-preprocess\$script-106-test_outlier.R
- script-05-04-preprocess\$script-107-test_outlier.R
- script-05-04-preprocess\$script-108-test_outlier.R
- script-05-04-preprocess\$script-109-test_outlier.R
- script-05-04-preprocess\$script-110-test_outlier.R
- script-05-04-preprocess\$script-111-test_outlier.R
- script-05-04-preprocess\$script-112-test_outlier.R
- script-05-04-preprocess\$script-113-test_outlier.R
- script-05-04-preprocess\$script-114-test_outlier.R
- script-05-04-preprocess\$script-115-test_outlier.R
- script-05-04-preprocess\$script-116-test_outlier.R
- script-05-04-preprocess\$script-117-test_outlier.R
- script-05-04-preprocess\$script-118-test_outlier.R
- script-05-04-preprocess\$script-119-test_outlier.R
- script-05-04-preprocess\$script-120-test_outlier.R
- script-05-04-preprocess\$script-121-test_outlier.R
- script-05-04-preprocess\$script-122-test_outlier.R
- script-05-04-preprocess\$script-123-test_outlier.R
- script-05-04-preprocess\$script-124-test_outlier.R
- script-05-04-preprocess\$script-125-test_outlier.R
- script-05-04-preprocess\$script-126-test_outlier.R
- script-05-04-preprocess\$script-127-test_outlier.R
- script-05-04-preprocess\$script-128-test_outlier.R
- script-05-04-preprocess\$script-129-test_outlier.R
- script-05-04-preprocess\$script-130-test_outlier.R
- script-05-04-preprocess\$script-131-test_outlier.R
- script-05-04-preprocess\$script-132-test_outlier.R
- script-05-04-preprocess\$script-133-test_outlier.R
- script-05-04-preprocess\$script-134-test_outlier.R
- script-05-04-preprocess\$script-135-test_outlier.R
- script-05-04-preprocess\$script-136-test_outlier.R
- script-05-04-preprocess\$script-137-test_outlier.R
- script-05-04-preprocess\$script-138-test_outlier.R
- script-05-04-preprocess\$script-139-test_outlier.R
- script-05-04-preprocess\$script-140-test_outlier.R
- script-05-04-preprocess\$script-141-test_outlier.R
- script-05-04-preprocess\$script-142-test_outlier.R
- script-05-04-preprocess\$script-143-test_outlier.R
- script-05-04-preprocess\$script-144-test_outlier.R
- script-05-04-preprocess\$script-145-test_outlier.R
- script-05-04-preprocess\$script-146-test_outlier.R
- script-05-04-preprocess\$script-147-test_outlier.R
- script-05-04-preprocess\$script-148-test_outlier.R
- script-05-04-preprocess\$script-149-test_outlier.R
- script-05-04-preprocess\$script-150-test_outlier.R
- script-05-04-preprocess\$script-151-test_outlier.R
- script-05-04-preprocess\$script-152-test_outlier.R
- script-05-04-preprocess\$script-153-test_outlier.R
- script-05-04-preprocess\$script-154-test_outlier.R
- script-05-04-preprocess\$script-155-test_outlier.R
- script-05-04-preprocess\$script-156-test_outlier.R
- script-05-04-preprocess\$script-157-test_outlier.R
- script-05-04-preprocess\$script-158-test_outlier.R
- script-05-04-preprocess\$script-159-test_outlier.R
- script-05-04-preprocess\$script-160-test_outlier.R
- script-05-04-preprocess\$script-161-test_outlier.R
- script-05-04-preprocess\$script-162-test_outlier.R
- script-05-04-preprocess\$script-163-test_outlier.R
- script-05-04-preprocess\$script-164-test_outlier.R
- script-05-04-preprocess\$script-165-test_outlier.R
- script-05-04-preprocess\$script-166-test_outlier.R
- script-05-04-preprocess\$script-167-test_outlier.R
- script-05-04-preprocess\$script-168-test_outlier.R
- script-05-04-preprocess\$script-169-test_outlier.R
- script-05-04-preprocess\$script-170-test_outlier.R
- script-05-04-preprocess\$script-171-test_outlier.R
- script-05-04-preprocess\$script-172-test_outlier.R
- script-05-04-preprocess\$script-173-test_outlier.R
- script-05-04-preprocess\$script-174-test_outlier.R
- script-05-04-preprocess\$script-175-test_outlier.R
- script-05-04-preprocess\$script-176-test_outlier.R
- script-05-04-preprocess\$script-177-test_outlier.R
- script-05-04-preprocess\$script-178-test_outlier.R
- script-05-04-preprocess\$script-179-test_outlier.R
- script-05-04-preprocess\$script-180-test_outlier.R
- script-05-04-preprocess\$script-181-test_outlier.R
- script-05-04-preprocess\$script-182-test_outlier.R
- script-05-04-preprocess\$script-183-test_outlier.R
- script-05-04-preprocess\$script-184-test_outlier.R
- script-05-04-preprocess\$script-185-test_outlier.R
- script-05-04-preprocess\$script-186-test_outlier.R
- script-05-04-preprocess\$script-187-test_outlier.R
- script-05-04-preprocess\$script-188-test_outlier.R
- script-05-04-preprocess\$script-189-test_outlier.R
- script-05-04-preprocess\$script-190-test_outlier.R
- script-05-04-preprocess\$script-191-test_outlier.R
- script-05-04-preprocess\$script-192-test_outlier.R
- script-05-04-preprocess\$script-193-test_outlier.R
- script-05-04-preprocess\$script-194-test_outlier.R
- script-05-04-preprocess\$script-195-test_outlier.R

Let's see how to use it- History

RStudio: Review Changes (on feketeg-deSKTOP)

Changes History master • ① script-07-create_plate_map_from_merged_operetta_layout.R

Subject	Author	Date	SHA
A merged_operetta_layout file datamat koveti	feketegergo <feketegergo@desktop1.brc>	2020-02-18	67002c71
Az xlxs mentési scriptbe betettem egy grep hibast, mert OutOfMemory-val néha megallt.	feketegergo <feketegergo@desktop1.brc>	2020-02-11	c84bad60
Bemér felejtettem egy hibát az xlxs kiroban, azt javítottam.	feketegergo <feketegergo@desktop1.brc>	2020-02-10	97709d8a
nem tudom mit csináltam	feketegergo <feketegergo@desktop1.brc>	2020-02-10	aaba8779
A script-07-create_plate_map_from_merged_operetta_layout.R fileban volt egy hiba az ger feketegergo <feketegergo@desktop1.brc>	feketegergo <feketegergo@desktop1.brc>	2020-02-03	cde7ed8d
A mergedOperettaLayout file egy csomot változtott és a kod kovette a változásokat.	feketegergo <feketegergo@desktop1.brc>	2020-01-21	f0d9e7a3
újcommit idben minden rövid commitnak hozzá	feketegergo <feketegergo@desktop1.brc>	2020-01-07	2011-081a

Commits 1-7 of 7

SHA 67002c71
Author feketegergo <feketegergo@desktop1.brc>
Date 2020-02-18 16:08
Subject A merged_operetta_layout file datamat koveti
Parent b7142471

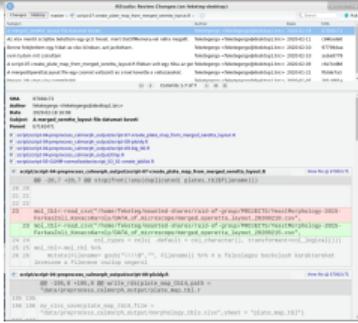
```
① script/script-04-preprocess_calmorph_output/script-07-create_plate_map_from_merged_operetta_layout.R
① script/script-04-preprocess_calmorph_output/script-08-ploidy.R
① script/script-04-preprocess_calmorph_output/script-09-big_tbl.R
① script/script-04-preprocess_calmorph_output/tmp.R
① script/script-05-GLMM-normalisation/script_03_01-create_joblist.R

② script/script-04-preprocess_calmorph_output/script-07-create_plate_map_from_merged_operetta_layout.R [View file @ 67002c71]
② -20,7 +20,7 @@ stopifnot(!any(duplicated( plates_tbl$filename)))
20 20
21 21
22 22
23 mol_tbl<-read.csv("/home/feketeg/mounted-shares/raid-of-group/PROJECTS/YeastMorphology-2015-
FarkasZoli_KovacsKaroly/DATA_of_microscope/merged_operetta_layout_20200210.csv",
23 mol_tbl<-read.csv("/home/feketeg/mounted-shares/raid-of-group/PROJECTS/YeastMorphology-2015-
FarkasZoli_KovacsKaroly/DATA_of_microscope/merged_operetta_layout_20200215.csv",
24 col_types = cols( .default = col_character(), transformant=col_logical())
25 mol_tbl<-mol_tbl %>%
26   mutate(Filename= gsub("\\\\\$","", Filename)) %>% # a felesleges backslash karaktereket
leveszi a Fileneve oszlop vegéről

③ script/script-04-preprocess_calmorph_output/script-08-ploidy.R [View file @ 67002c71]
③ -195,8 +195,8 @@ write_rds(plate_map_tbl4,path =
"data/preprocess_calmorph_output/plate_map_tbl.r
195 195
196 my_xlsx_save(plate_map_tbl4,file =
"data/preprocess_calmorph_output/morphology_tbls.xlsx",sheet = "plate_map_tbl")
197 197
```



Let's see how to use it- History



- each row is a commit with
 - date
 - author
 - comment
 - commit ID
- list of files modified in the selected commit
- diffs : for each file it shows what is modified

Principles

- If you want to roll back
 - You have to commit first
 - git will replace the actual files with the old ones
 - You can not rollback only one file.
- You can go back to an old version and then return to the latest version
- If you want to go somewhere you have to tell the ID of the version
- the last commit called **HEAD**
- the 'go to' command is **checkout**

Example

```
git checkout 67002c71
```

```
git checkout HEAD
```

Principles

bad news

sometimes you need to type commands

GitHub

GitHub

GitHub is a webserver where you can save

- GIT saves things to the local repository on your machine
- GIT can upload everything to a remote git-server
- GitHub is one of the git servers

GitHub

GitHub is a webserver where you can save

- GIT saves things to the local repository on your machine
 - local repository needs to exist
- GIT can upload everything to a remote git-server
 - does not upload the actual files.
 - uploads the repository: all versions
- GitHub is one of the git servers
 - Bitbucket
 - Gitlab
 - We can have our own git server

GitHub

GitHub is a webserver where you can save

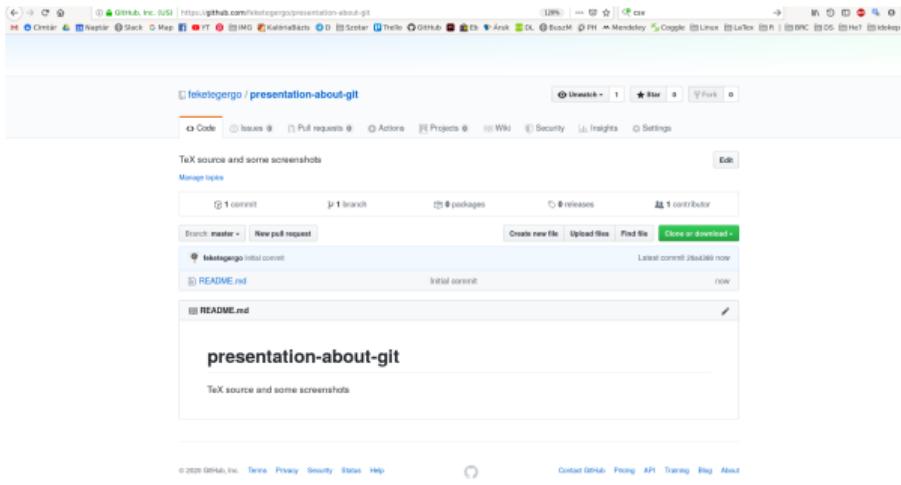
- GIT saves things to the local repository on your machine
 - local repository needs to exist
- GIT can upload everything to a remote git-server
 - does not upload the actual files.
 - uploads the repository: all versions
- GitHub is one of the git servers
 - Bitbucket
 - Gitlab
 - We can have our own git server

GitHub

GitHub is a webserver where you can save

- GIT saves things to the local repository on your machine
 - local repository needs to exist
- GIT can upload everything to a remote git-server
 - does not upload the actual files.
 - uploads the repository: all versions
- GitHub is one of the git servers
 - Bitbucket
 - Gitlab
 - We can have our own git server

- basic git servers just stores the repository
- GitHub provides additional Web interface



Start a GitHub Project

- go to github.com , register a user
- create a new project. Tick the 'initialised' checkbox.
- copy paste the url of the project
- start a terminal, go to the parent folder.
- use 'git clone <url>' command
 - now you have an initialised local repository in the folder
 - the **clone** command automatically connected it to the GitHub repo.
 - you can commit files.
 - if you press the 'push' button or give the 'git push' command, then everything will be uploaded.
 - If another user modified the files on the server the 'git pull' command download it

Start a GitHub Project

- go to `github.com` , register a user
- create a new project. Tick the 'initialised' checkbox.
- copy paste the url of the project
- start a terminal, go to the parent folder.
- use '`git clone <url>`' command
- now you have an initialised local repository in the folder
- the **clone** command automatically connected it to the GitHub repo.
- you can commit files.
- if you press the 'push' button or give the '`git push`' command, then everything will be uploaded.
- If another user modified the files on the server the '`git pull`' command download it

Tricky things start here

- If you try to upload a file, what is modified by another user....
- It is called 'conflict'
- The operation 'merge' can fix the problem
- normally git merge it automatically
- pull first then push

Tricky things start here

- If you try to upload a file, what is modified by another user....
- It is called 'conflict'
- The operation 'merge' can fix the problem
- normally git merge it automatically
- pull first then push

Git Data Transport Commands

<http://csteelle.com>

