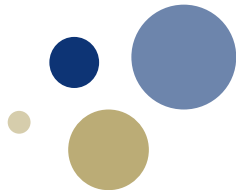




Norwegian University of
Science and Technology



Version control for researchers

Sigurd Hofsmo Jakobsen

Department of electric power engineering

October 6, 2016



Outline

Introduction to version control

Git

Example using git

Git work flow

Continuous integration

References



A basic example

- Only one person can work on a file at a time

Name



experiment.m



experiment_better_version.m



experiment_final.m



experiment_final_final_20161007...



experiment_maybe_better_2016...



A basic example

- Only one person can work on a file at a time
- Difficult to keep track on which file to use

Name



experiment.m



experiment_better_version.m



experiment_final.m



experiment_final_final_20161007...



experiment_maybe_better_2016...



A basic example

- Only one person can work on a file at a time
- Difficult to keep track on which file to use
- Difficult to compare the files

Name



experiment.m



experiment_better_version.m



experiment_final.m



experiment_final_final_20161007...



experiment_maybe_better_2016...



A basic example

- Only one person can work on a file at a time
- Difficult to keep track on which file to use
- Difficult to compare the files
- Do you have back up?

Name



experiment.m



experiment_better_version.m



experiment_final.m



experiment_final_final_20161007...



experiment_maybe_better_2016...



A basic example

- Only one person can work on a file at a time
- Difficult to keep track on which file to use
- Difficult to compare the files
- Do you have back up?
- Probably more issues

Name



experiment.m



experiment_better_version.m



experiment_final.m



experiment_final_final_20161007...

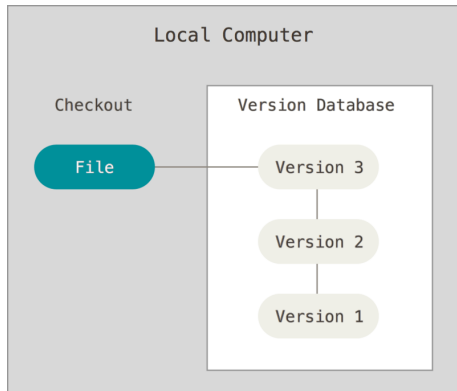


experiment_maybe_better_2016...



Smarter way of doing it

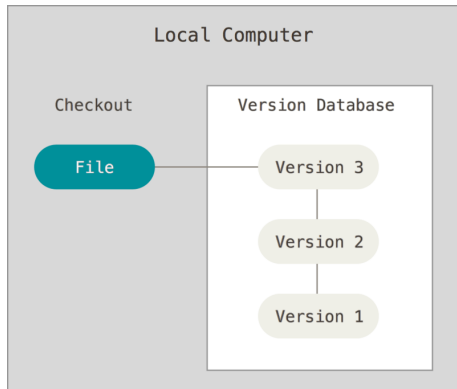
- Equivalent of storing your stuff on M





Smarter way of doing it

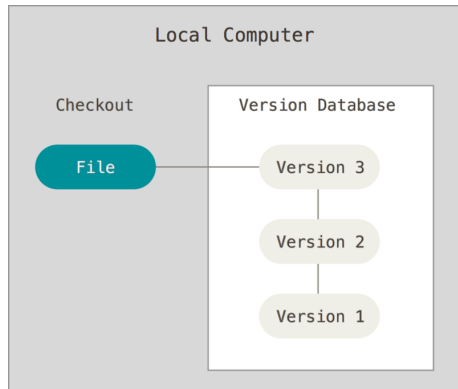
- Equivalent of storing your stuff on M
- Included in MAC





Smarter way of doing it

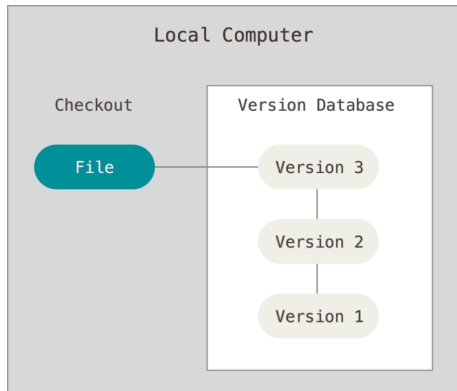
- Equivalent of storing your stuff on M
- Included in MAC
- Easy to set up





Smarter way of doing it

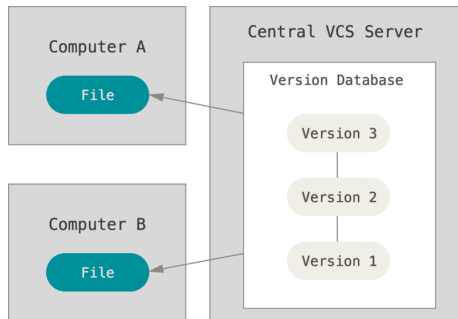
- Equivalent of storing your stuff on M
- Included in MAC
- Easy to set up
- Still difficult to collaborate





Centralized version control

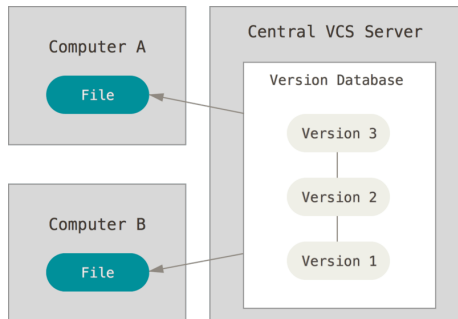
— Examples:





Centralized version control

- Examples:
 - CVS

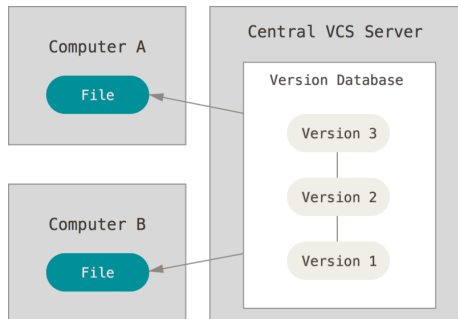




Centralized version control

— Examples:

- CVS
- subversion

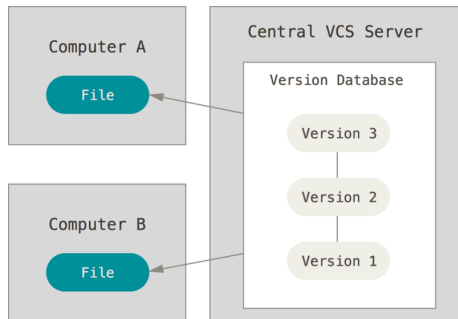




Centralized version control

— Examples:

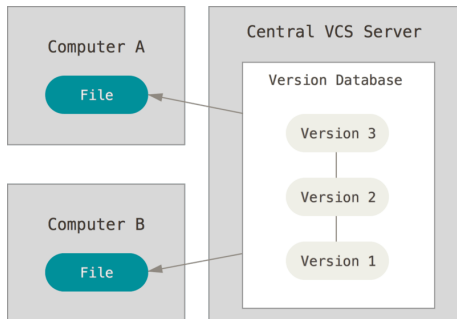
- CVS
- subversion
- perforce





Centralized version control

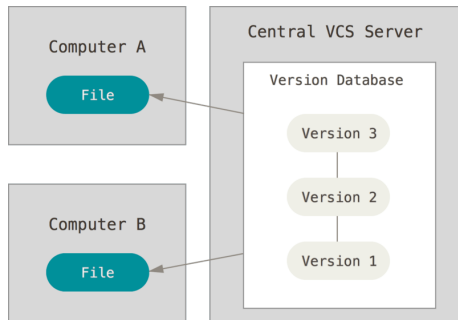
- Examples:
 - CVS
 - subversion
 - perforce
- Easy to collaborate





Centralized version control

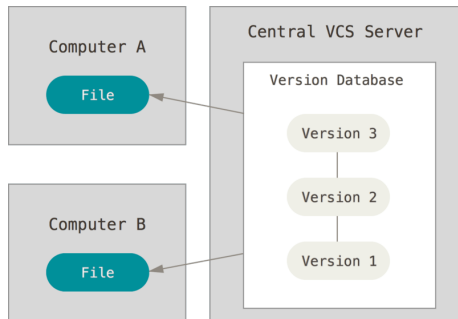
- Examples:
 - CVS
 - subversion
 - perforce
- Easy to collaborate
- Check out specific versions





Centralized version control

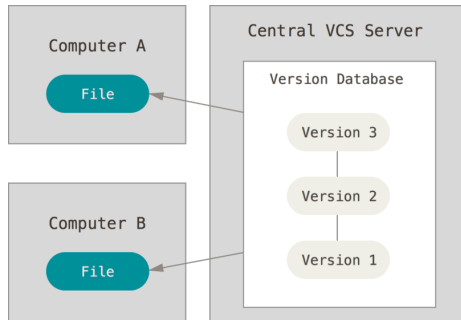
- Examples:
 - CVS
 - subversion
 - perforce
- Easy to collaborate
- Check out specific versions
- Single point of failure (N-0)





Centralized version control

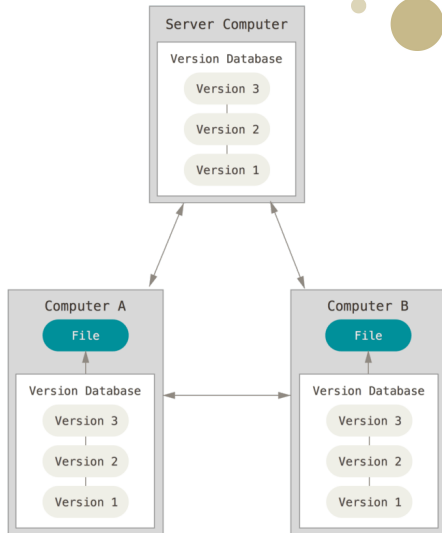
- Examples:
 - CVS
 - subversion
 - perforce
- Easy to collaborate
- Check out specific versions
- Single point of failure (N-0)
- If the server dies only checked out versions can be saved





Distributed version control

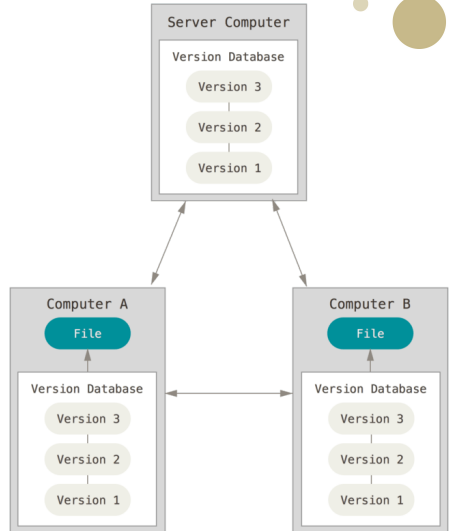
— Examples:





Distributed version control

- Examples:
 - Git

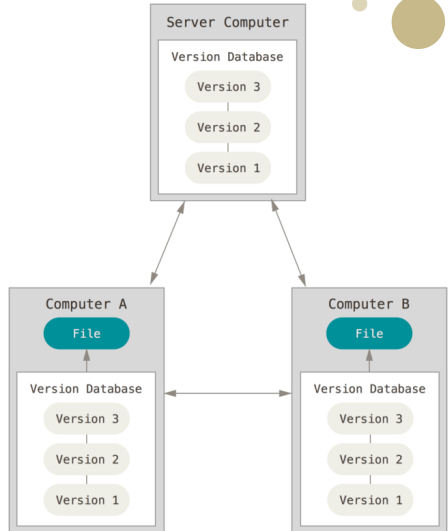




Distributed version control

— Examples:

- Git
- Mercurial

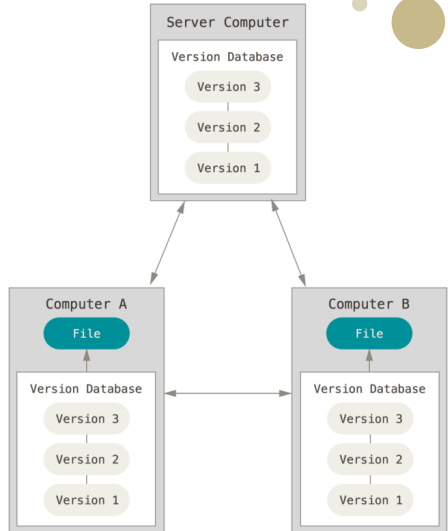




Distributed version control

— Examples:

- Git
- Mercurial
- Bazaar

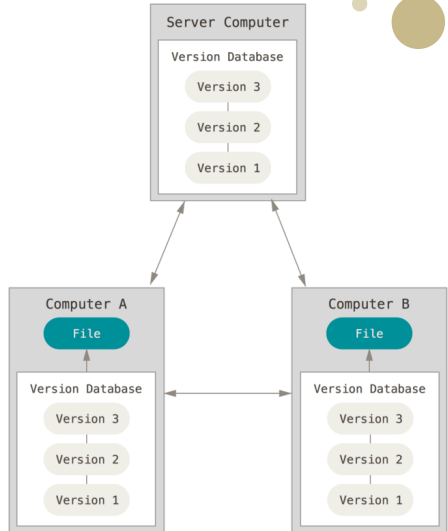




Distributed version control

— Examples:

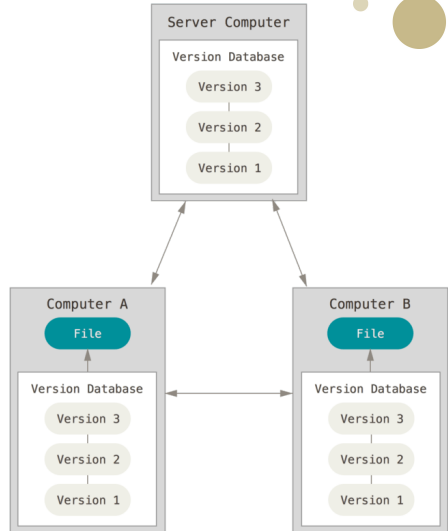
- Git
- Mercurial
- Bazaar
- Darcs





Distributed version control

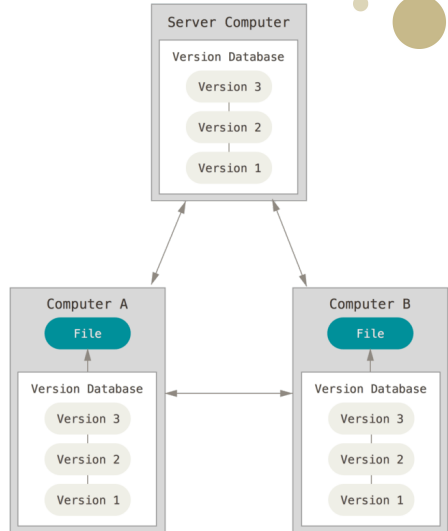
- Examples:
 - Git
 - Mercurial
 - Bazaar
 - Darcs
- Same advantages as centralized version control





Distributed version control

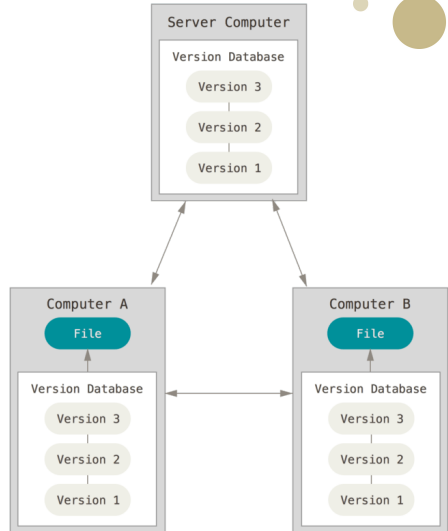
- Examples:
 - Git
 - Mercurial
 - Bazaar
 - Darcs
- Same advantages as centralized version control
- All users can reconstruct the project





Distributed version control

- Examples:
 - Git
 - Mercurial
 - Bazaar
 - Darcs
- Same advantages as centralized version control
- All users can reconstruct the project
- Easy to work against multiple servers





Should you use version control?

— Do you write code?

Dont do this:

```
% Read data set 1
data=read_data('data1.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```

```
% Read data set 1
data=read_data('data2.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```



Should you use version control?

- Do you write code?
 - If yes: use version control

Dont do this:

```
% Read data set 1
data=read_data('data1.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```

```
% Read data set 1
data=read_data('data2.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```



Should you use version control?

- Do you write code?
 - If yes: use version control
- Never copy code --+

Dont do this:

```
% Read data set 1
data=read_data('data1.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```

```
% Read data set 1
data=read_data('data2.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```



Should you use version control?

- Do you write code?
 - If yes: use version control
- Never copy code -_-+
- Write functions and keep them version controlled

Dont do this:

```
% Read data set 1
data=read_data('data1.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```

```
% Read data set 1
data=read_data('data2.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```



Should you use version control?

- Do you write code?
 - If yes: use version control
- Never copy code --+
- Write functions and keep them version controlled
- Remember code is a virtual laboratory

Dont do this:

```
% Read data set 1
data=read_data('data1.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```

```
% Read data set 1
data=read_data('data2.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```




Should you use version control?

- Do you write code?
 - If yes: use version control
- Never copy code --+
- Write functions and keep them version controlled
- Remember code is a virtual laboratory
- Versions can for instance be tagged with name of publication(reproducability for review)

Dont do this:

```
% Read data set 1
data=read_data('data1.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```

```
% Read data set 1
data=read_data('data2.csv'
               ');
df = process_data(data,
                  0.02, 1);
plot(df);
```



Outline

Introduction to version control

Git

Example using git

Git work flow

Continuous integration

References



About git

— Developed by the team behind Linux





About git

- Developed by the team behind Linux
- used by companies such as:





About git

- Developed by the team behind Linux
- used by companies such as:
 - Linux





About git

- Developed by the team behind Linux
- used by companies such as:
 - Linux
 - Microsoft





About git

- Developed by the team behind Linux
- used by companies such as:
 - Linux
 - Microsoft
 - Google





About git

- Developed by the team behind Linux
- used by companies such as:
 - Linux
 - Microsoft
 - Google
 - Android





About git

- Developed by the team behind Linux
- used by companies such as:
 - Linux
 - Microsoft
 - Google
 - Android
 - Facebook





About git

- Developed by the team behind Linux
- used by companies such as:
 - Linux
 - Microsoft
 - Google
 - Android
 - Facebook
 - Twitter





About git

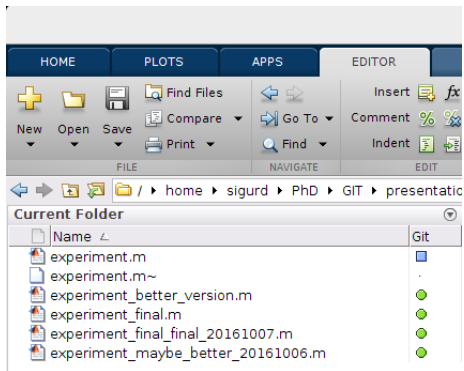
- Developed by the team behind Linux
- used by companies such as:
 - Linux
 - Microsoft
 - Google
 - Android
 - Facebook
 - Twitter
 - LinkedIn





Getting started with git

— Download git

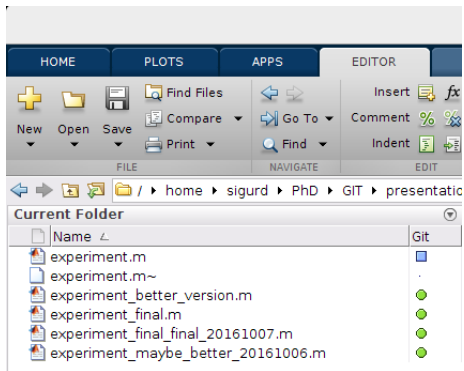




Getting started with git

— Download git

- www.git-scm.com

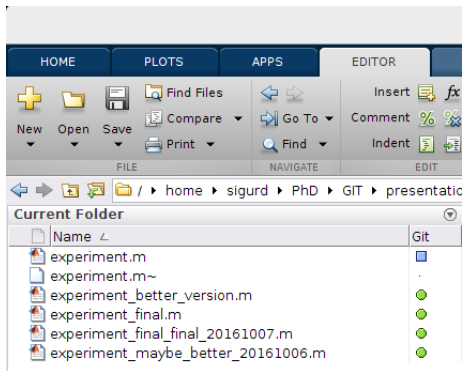




Getting started with git

— Download git

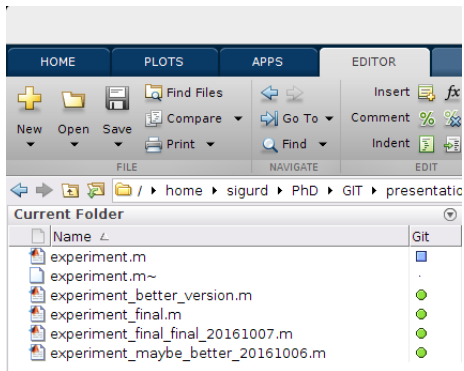
- www.git-scm.com
- Your favourite package manager





Getting started with git

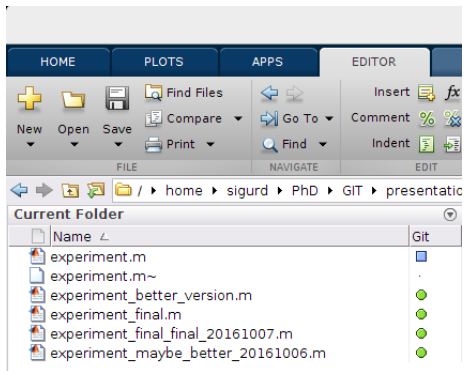
- Download git
 - www.git-scm.com
 - Your favourite package manager
- Get a server:





Getting started with git

- Download git
 - www.git-scm.com
 - Your favourite package manager
- Get a server:
 - Set up yourself





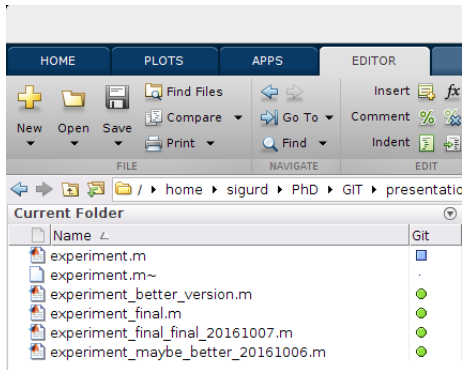
Getting started with git

— Download git

- www.git-scm.com
- Your favourite package manager

— Get a server:

- Set up yourself
- GitHub





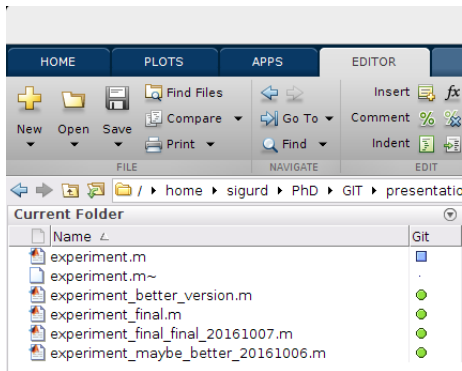
Getting started with git

— Download git

- www.git-scm.com
- Your favourite package manager

— Get a server:

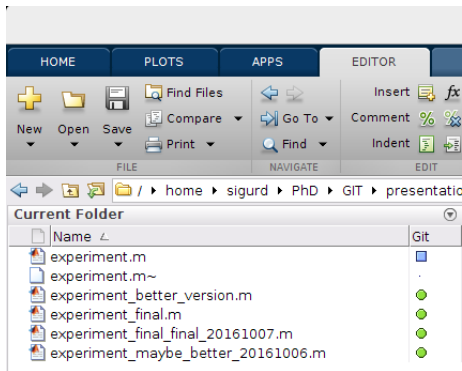
- Set up yourself
- GitHub
- BitBucket





Getting started with git

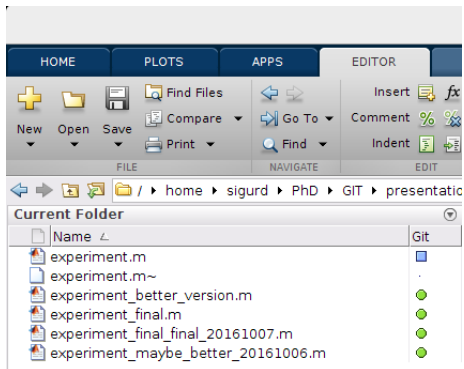
- Download git
 - www.git-scm.com
 - Your favourite package manager
- Get a server:
 - Set up yourself
 - GitHub
 - BitBucket
- Get a client:





Getting started with git

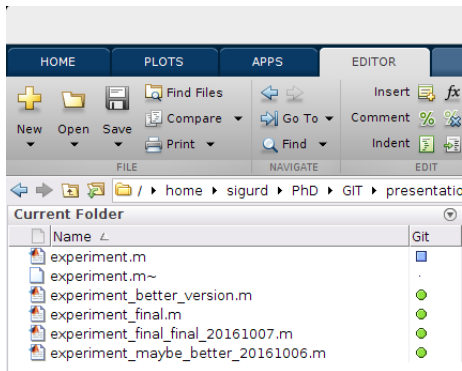
- Download git
 - www.git-scm.com
 - Your favourite package manager
- Get a server:
 - Set up yourself
 - GitHub
 - BitBucket
- Get a client:
 - Command line is already included





Getting started with git

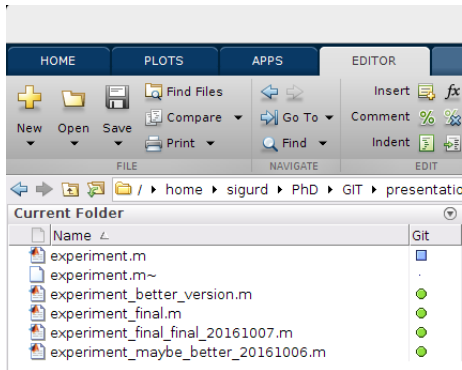
- Download git
 - www.git-scm.com
 - Your favourite package manager
- Get a server:
 - Set up yourself
 - GitHub
 - BitBucket
- Get a client:
 - Command line is already included
 - GitKraken





Getting started with git

- Download git
 - www.git-scm.com
 - Your favourite package manager
- Get a server:
 - Set up yourself
 - GitHub
 - BitBucket
- Get a client:
 - Command line is already included
 - GitKraken
 - Check out list at www.git-scm.com/downloads/guis





GitHub

— Largest host for git repositories





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money
- You can request an academic account at <https://education.github.com/>





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money
- You can request an academic account at <https://education.github.com/>
 - You get five closed source projects for free





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money
- You can request an academic account at <https://education.github.com/>
 - You get five closed source projects for free
 - Maybe other stuff too





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money
- You can request an academic account at <https://education.github.com/>
 - You get five closed source projects for free
 - Maybe other stuff too
- Wiki for your project





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money
- You can request an academic account at <https://education.github.com/>
 - You get five closed source projects for free
 - Maybe other stuff too
- Wiki for your project
- Place to discuss your project and fill bug reports





GitHub

- Largest host for git repositories
- For developers a git page is sometimes as important as a CV
- You can have as many open repositories you want free.
- Closed source projects cost money
- You can request an academic account at <https://education.github.com/>
 - You get five closed source projects for free
 - Maybe other stuff too
- Wiki for your project
- Place to discuss your project and fill bug reports
- Integrates with many cool services





BitBucket

— More or less same features as GitHub



BitBucket

- More or less same features as GitHub
- Wiki, bug reporting etc. different integrated services



BitBucket

- More or less same features as GitHub
- Wiki, bug reporting etc. different integrated services
- Free closed source repositories



BitBucket

- More or less same features as GitHub
- Wiki, bug reporting etc. different integrated services
- Free closed source repositories
- With git it is easy to change the remote, try both GitHub and BitBucket



Outline

Introduction to version control

Git

Example using git

Git work flow

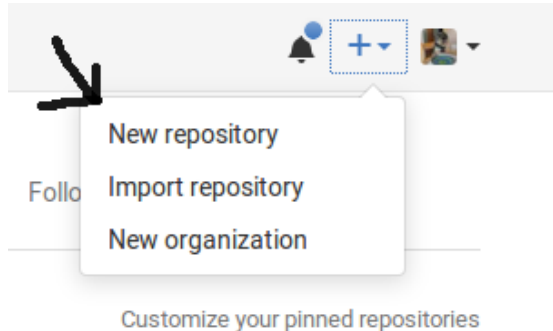
Continuous integration

References



Creating a git repository

— New repository button





Creating a git repository

- New repository button
- Create the name

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Hofsmo ▾

Repository name

Example



Great repository names are short and memorable. Need inspiration? How about **studious-engine**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository



Creating a git repository

- New repository button
- Create the name
- Create a description

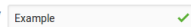
Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

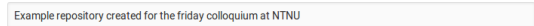


Repository name



Great repository names are short and memorable. Need inspiration? How about **studious-engine**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.

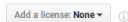
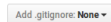


Private

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.



Create repository



Creating a git repository

- New repository button
- Create the name
- Create a description
- Open or public?

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Hofsmo

Repository name

Example



Great repository names are short and memorable. Need inspiration? How about [studious-engine](#).

Description (optional)

Example repository created for the friday colloquium at NTNU



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None**



Create repository



Creating a git repository

- New repository button
- Create the name
- Create a description
- Open or public?
- READMEs are cool

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

Example



Great repository names are short and memorable. Need inspiration? How about **studious-engine**.

Description (optional)

Example repository created for the friday colloquium at NTNU



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None**



Create repository




Creating a git repository

- New repository button
- Create the name
- Create a description
- Open or public?
- READMEs are cool
- Gitignore is useful

Create a new repository



A repository contains all the files for your project, including the revision history.

Owner	Repository name
 Hofsmo ▾	Example ✓

Great repository names are short and memorable. Need inspiration? How about [studious-engine](#).

Description (optional)

Example repository created for the friday colloquium at NTNU

- ☒  **Public**
Anyone can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **Python** ▾

Add a license: **None** ▾ ⓘ

Create repository



Creating a git repository

- New repository button
- Create the name
- Create a description
- Open or public?
- READMEs are cool
- Gitignore is useful
- Add license

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Hofsmo

Repository name

Example



Great repository names are short and memorable. Need inspiration? How about **studious-engine**.

Description (optional)

Example repository created for the friday colloquium at NTNU



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **Python**

Add a license: **GNU General Public License v3.0**



Create repository

Resulting repository

Hofsmo / Example
Unwatch 1
Star 0
Fork 0

Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Settings

Example repository created for the friday colloquium at NTNU — Edit

1 commit
1 branch
0 releases
1 contributor

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

Hofsmo Initial commit	Latest commit 14fe145 just now
.gitignore	Initial commit just now
LICENSE	Initial commit just now
README.md	Initial commit just now

README.md

Example

Example repository created for the friday colloquium at NTNU



Cloing the repository

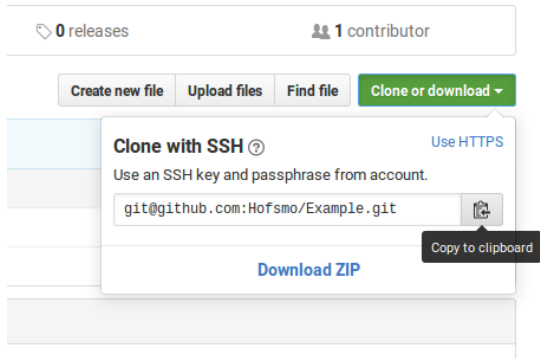
— using the terminal

```
~/P/G/presentations (vc|+7...) $ cd ~/PhD/GIT/  
~/P/GIT $
```



Cloing the repository

- using the terminal
 - Copy the url





Cloing the repository

- using the terminal
 - Copy the url
 - Run command

```
~/P/G/presentations (vc|+7...) $ cd ~/PhD/GIT/  
~/P/GIT $ git clone git@github.com:Hofsmo/Example.git
```




Cloing the repository

— using the terminal

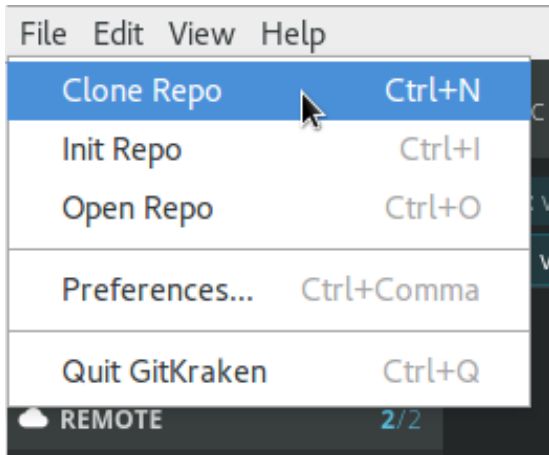
- Copy the url
- Run command
- Done

```
~/P/G/presentations (vc|✚...) $ cd ~/PhD/GIT/  
~/P/GIT $ git clone git@github.com:Hofsmo/Example.git  
Cloning into 'Example'...  
Enter passphrase for key '/home/sigurd/.ssh/id_rsa':  
remote: Counting objects: 5, done.  
remote: Compressing objects: 100% (5/5), done.  
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (5/5), 12.78 KiB | 0 bytes/s, done.  
Checking connectivity... done.
```



Cloing the repository

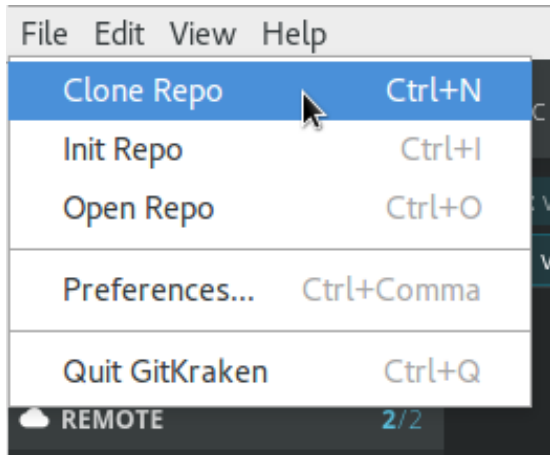
- using the terminal
 - Copy the url
 - Run command
 - Done
- Using a GUI(GitKraken in this case)





Cloing the repository

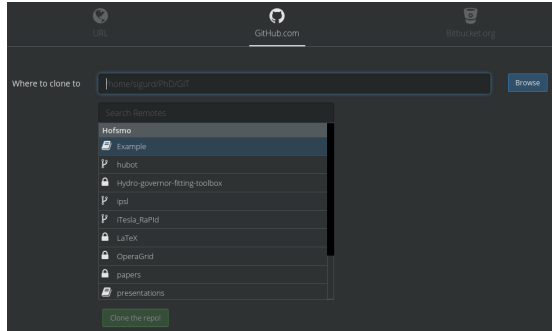
- using the terminal
 - Copy the url
 - Run command
 - Done
- Using a GUI(GitKraken in this case)
 - Find clone repo button





Cloing the repository

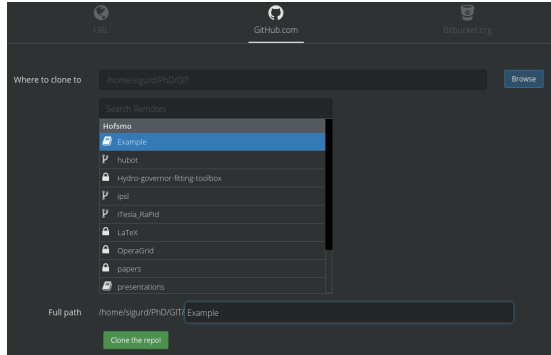
- using the terminal
 - Copy the url
 - Run command
 - Done
- Using a GUI(GitKraken in this case)
 - Find clone repo button
 - Select repository





Cloing the repository

- using the terminal
 - Copy the url
 - Run command
 - Done
- Using a GUI(GitKraken in this case)
 - Find clone repo button
 - Select repository
 - Decide where to put it and clone





Staging our first file

```
~/P/G/Example (master|...) $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        example/

nothing added to commit but untracked files present (use "git add" to track)
```

— In the terminal



Staging our first file

```
~/P/G/Example (master|...) $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        example/

nothing added to commit but untracked files present (use "git add" to track)
```

- In the terminal
 - git status



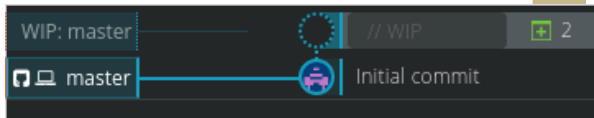
Staging our first file

```
~/P/G/Example (master|...) $ git add example/preprocess_data.py  
~/P/G/Example (master|●...) $
```

— In the terminal

- git status
- stage file using
git add

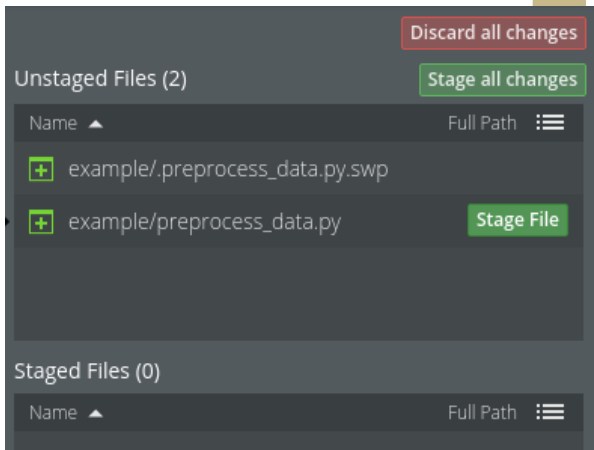
Staging our first file



- In the terminal
 - git status
 - stage file using git add
- In the GUI

Staging our first file

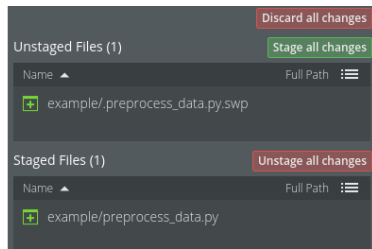
- In the terminal
 - `git status`
 - stage file using `git add`
- In the GUI
 - Push "Stage File" button





Ignoring editor generated file

- Remember the second file?



```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   example/preprocess_data.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    example/.preprocess_data.py.swp
```



Ignoring editor generated file

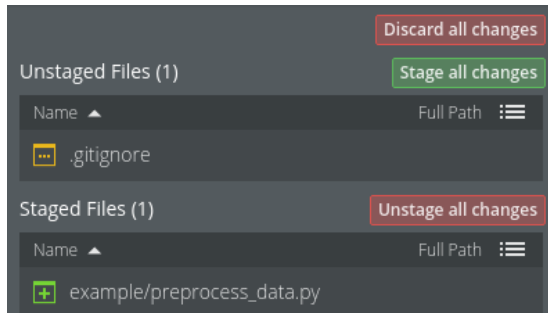
- Remember the second file?
- Tell git to ignore it

```
1
2 # Spyder project settings
3 .spyderproject
4
5 # Rope project settings
6 .ropeproject
7
8 # Ignore vim stuff
9 *.swp
10 *.swp
NORMAL > master > .gitignore
```



Ignoring editor generated file

- Remember the second file?
- Tell git to ignore it
- Git ignores it





Ignoring editor generated file

```
~/P/G/Example (master|🍌🔑) $ git add .gitignore
```

- Remember the second file?
- Tell git to ignore it
- Git ignores it
- Stage the modified .gitignore



Our first commit

— In terminal:

```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   .gitignore
        new file:   example/preprocess_data.py
```



Our first commit

— In terminal:

- Enter git commit

```
~/P/G/Example (master|●) $ git commit
```




Our first commit

— In terminal:

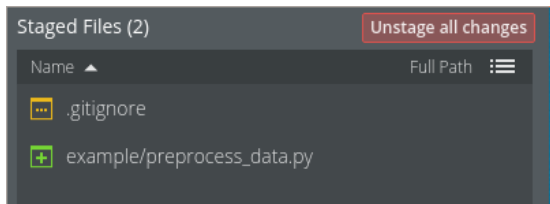
- Enter git commit
- Write and save commit message

```
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 On branch master
5 Your branch is up-to-date with 'origin/master'.
6
7 Changes to be committed:
8   modified:   .gitignore
9   Added swp and swo to gitignore
10  new file:   example/preprocess_data.py
11  First implementation of function
```



Our first commit

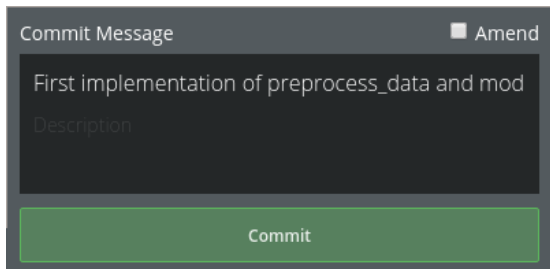
- In terminal:
 - Enter git commit
 - Write and save commit message
- In GUI:





Our first commit

- In terminal:
 - Enter git commit
 - Write and save commit message
- In GUI:
 - Write commit message





Our first commit

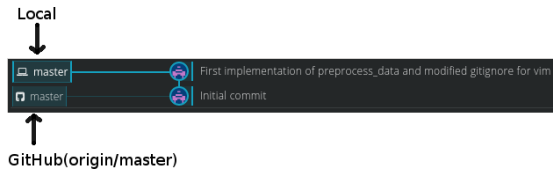
- In terminal:
 - Enter `git commit`
 - Write and save commit message
- In GUI:
 - Write commit message
 - Push commit button

A screenshot of a 'Commit Message' dialog box. The title bar says 'Commit Message' and there is a checkbox labeled 'Amend' in the top right corner. The main area is a dark text input field containing the text 'First implementation of preprocess_data and mod' followed by 'Description' on a new line. At the bottom of the dialog is a large green button labeled 'Commit'.



Push the changes to GitHub

- GitHub (remote) is now behind our local copy





Push the changes to GitHub

- GitHub (remote) is now behind our local copy
- Push changes to GitHub

```
~/P/G/Example (master|●) $ git push origin master
```

Where to push



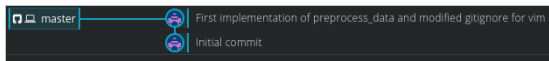
What to push





Push the changes to GitHub

- GitHub (remote) is now behind our local copy
- Push changes to GitHub
- GitHub and local copy are now equal





Outline

Introduction to version control

Git

Example using git

Git work flow

Continuous integration

References



Introducing a git work flow

- Working directly on the master branch is not recommended



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue
 2. Create issue branch



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue
 2. Create issue branch
 3. Work on issue branch until it works



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue
 2. Create issue branch
 3. Work on issue branch until it works
 4. Create a pull request



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue
 2. Create issue branch
 3. Work on issue branch until it works
 4. Create a pull request
 5. Merge issue branch to develop



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue
 2. Create issue branch
 3. Work on issue branch until it works
 4. Create a pull request
 5. Merge issue branch to develop
 6. When develop is stable merge into master



Introducing a git work flow

- Working directly on the master branch is not recommended
- In this section I will present a git flow inspired working flow I use at SINTEF:
 1. Create issue
 2. Create issue branch
 3. Work on issue branch until it works
 4. Create a pull request
 5. Merge issue branch to develop
 6. When develop is stable merge into master
- Example will follow




Create an issue

Hofsmo / Example

Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)



Add decimate


Write

Preview

AA B i “ < > ↻ ⋮ ≡ ☰ ↶ @ 📌

Hi,
It would be nice if the function also decimated the signal.

Attach files by dragging & dropping or [selecting them](#).

 Styling with Markdown is supported

Submit new issue


Labels

enhancement

Milestone

No milestone

Assignees

 Hofsmo

Sigurd Hofsmo Jakobsen (NTNU)

Version control for researchers

October 6, 2016

23 / 31



Create an issue

Add decimate #1



Hofsmo opened this issue a minute ago · 0 comments



Hofsmo commented a minute ago

Owner



Hi,
It would be nice if the function also decimated the signal.



Hofsmo added the **enhancement** label a minute ago



Hofsmo self-assigned this a minute ago



Create the feature branches

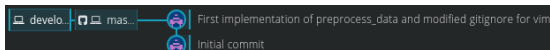
- Branch the develop branch from master

```
~/P/G/Example (master|✓) $ git checkout -b develop  
Switched to a new branch 'develop'  
~/P/G/Example (develop|✓) $
```



Create the feature branches

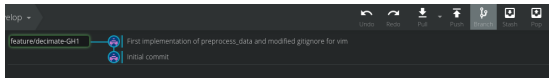
- Branch the develop branch from master
- Develop and master point to the same commit





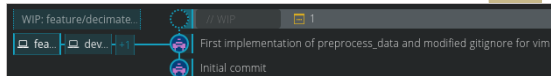
Create the feature branches

- Branch the develop branch from master
- Develop and master point to the same commit
- Branch the feature branch from develop





Create a pull request

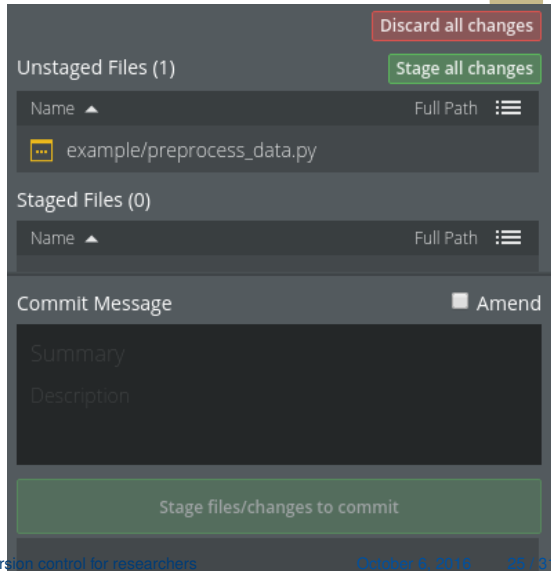


- Now we can see that we have unstaged changes



Create a pull request

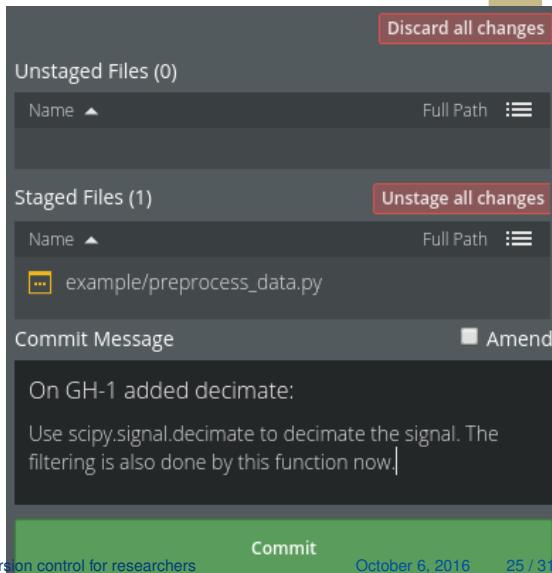
- Now we can see that we have unstaged changes
- Stage the changes





Create a pull request

- Now we can see that we have unstaged changes
- Stage the changes
- Commit the changes





Create a pull request

- Now we can see that we have unstaged changes
- Stage the changes
- Commit the changes
- GH-1 is a reference to the issue

Add decimate #1



Hofsmo opened this issue an hour ago · 0 comments



Hofsmo commented an hour ago

Owner



Hi,
It would be nice if the function also decimated the signal.



Hofsmo added the **enhancement** label an hour ago



Hofsmo self-assigned this an hour ago



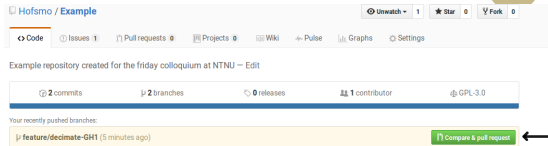
Hofsmo added a commit that referenced this issue 21 seconds ago

On GH-1 added decimate:

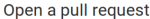
Ba99eed

Create pull request

— Create a pull request



- Create a pull request
- Choose develop as base






Create pull request

- Create a pull request
- Choose develop as base
- Create the pull request

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: `develop` ... compare: `feature/decimate-GH1` ✓ **Able to merge.** These branches can be automatically merged.



On GH-1 added decimate:

Write Preview AA^{*} B i “ ” < > ☰ ☲ ☳ ☴ ☵ ☶ ☷ ↶ ↷

Use `scipy.signal.decimate` to decimate the signal. The filtering is also done by this function now.

Attach files by dragging & dropping or [selecting them](#).

Styling with Markdown is supported

Create pull request ←

Labels None yet

Milestone No milestone

Assignees No one—assign yourself



Create pull request

- Create a pull request
- Choose develop as base
- Create the pull request
- Delete the branch

On GH-1 added decimate: #2

The screenshot shows a GitHub pull request interface. At the top, a purple 'Merged' badge is followed by the text 'Hofsmo merged 1 commit into develop from feature/decimate-gh1 3 minutes ago'. Below this, a summary bar shows 'Conversation 0', 'Commits 1', and 'Files changed 1'. The main content area includes a comment from 'Hofsmo' 5 minutes ago: 'Use scipy.signal.decimate to decimate the signal. The filtering is also done by this function now.' Below the comment is a commit history section showing 'On GH-1 added decimate: ...' by 'Ba09eed'. Further down, a message states 'Hofsmo merged commit 61dfe82 into develop 3 minutes ago' with a 'Revert' button. A light blue informational box with a lightbulb icon says 'Avoid bugs by automatically running your tests.' and provides a 'Learn more' link. At the bottom, a purple box with a checkmark icon states 'Pull request successfully merged and closed' and 'You're all set—the feature/decimate- branch can be safely deleted.' To the right of this box is a 'Delete branch' button, which is highlighted by a black arrow pointing down from the 'Learn more' box.



Create pull request

- Create a pull request
- Choose develop as base
- Create the pull request
- Delete the branch
- Close the issue (Could also have written "close GH-1" earlier to do this

Add decimate #1

Open Hofsmo opened this issue an hour ago · 0 comments



Hofsmo commented an hour ago

Owner



Hi,
It would be nice if the function also decimated the signal.



Hofsmo added the **enhancement** label an hour ago



Hofsmo self-assigned this an hour ago



Hofsmo added a commit that referenced this issue 30 minutes ago

On GH-1 added decimate:

8a89eed



Write

Preview

AA[~] B *i* “ ” < > @

Leave a comment

Attach files by dragging & dropping or [selecting them](#).

Styling with Markdown is supported

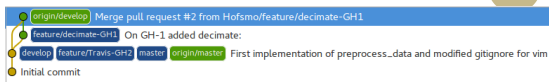
Close issue

Comment



Create pull request

- Create a pull request
- Choose develop as base
- Create the pull request
- Delete the branch
- Close the issue (Could also have written "close GH-1" earlier to do this
- The git tree





Outline

Introduction to version control

Git

Example using git

Git work flow

Continuous integration

References



Continuous integration

— Did anyone notice?

On GH-1 added decimate: #2

Merged Hofsmo merged 1 commit into `develop` from `feature/decimate-GH1` 3 minutes ago

Conversation 0 Commits 1 Files changed 1

Hofsmo commented 5 minutes ago Owner + 👤 ✎

Use `scipy.signal.decimate` to decimate the signal. The filtering is also done by this function now.

On GH-1 added decimate: ... Sa@9eed

Hofsmo merged commit `61dfe82` into `develop` 3 minutes ago Revert

→ **Avoid bugs by automatically running your tests.** Learn more ×

Continuous integration can help catch bugs by running your tests automatically. Merge your code with confidence using one of our continuous integration providers.

Pull request successfully merged and closed Delete branch

You're all set—the `feature/decimate` branch can be safely deleted.



Continuous integration

- Did anyone notice?
- For GitHub travis is a good alternative

On GH-1 added decimate: #2

Merged Hofsmo merged 1 commit into `develop` from `feature/decimate-GH1` 3 minutes ago

Conversation 0 Commits 1 Files changed 1

Hofsmo commented 5 minutes ago Owner +😊 ✎

Use `scipy.signal.decimate` to decimate the signal. The filtering is also done by this function now.

On GH-1 added decimate: ... Sa@Seed

Hofsmo merged commit `61dfe82` into `develop` 3 minutes ago Revert

→ **Avoid bugs by automatically running your tests.** Learn more ×

Continuous integration can help catch bugs by running your tests automatically. Merge your code with confidence using one of our continuous integration providers.

Pull request successfully merged and closed Delete branch

You're all set—the `feature/decimate..` branch can be safely deleted.



Continuous integration

- Did anyone notice?
- For GitHub travis is a good alternative
- Jenkins also works with GitHub and supports MATLAB unlike travis

On GH-1 added decimate: #2

Merged Hofsmo merged 1 commit into `develop` from `feature/decimate-GH1` 3 minutes ago

Conversation 0 Commits 1 Files changed 1

Hofsmo commented 5 minutes ago Owner +😊 ✎

Use `scipy.signal.decimate` to decimate the signal. The filtering is also done by this function now.

On GH-1 added decimate: ... Sa@geed

Hofsmo merged commit `61dfe82` into `develop` 3 minutes ago Revert

→ **Avoid bugs by automatically running your tests.**
 Continuous integration can help catch bugs by running your tests automatically.
 Merge your code with confidence using one of our continuous integration providers. Learn more ×

Pull request successfully merged and closed
 You're all set—the `feature/decimate.` branch can be safely deleted. Delete branch



Continuous integration

- Did anyone notice?
- For GitHub travis is a good alternative
- Jenkins also works with GitHub and supports MATLAB unlike travis
- BitBucket has bamboo, but may cost money

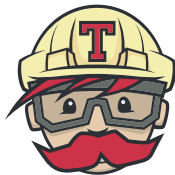
On GH-1 added decimate: #2

The screenshot shows a GitHub pull request interface. At the top, a purple 'Merged' badge is followed by the text 'Hofsmo merged 1 commit into develop from feature/decimate-GH1 3 minutes ago'. Below this, a summary bar shows 'Conversation 0', 'Commits 1', and 'Files changed 1'. The main content area shows a comment from 'Hofsmo' 5 minutes ago: 'Use scipy.signal.decimate to decimate the signal. The filtering is also done by this function now.' Below the comment is a commit entry 'On GH-1 added decimate: ...' by 'Sa@geed'. Another entry shows 'Hofsmo merged commit 61dfe82 into develop 3 minutes ago' with a 'Revert' button. A blue arrow points to a light blue box with a lightbulb icon containing the text: 'Avoid bugs by automatically running your tests. Continuous integration can help catch bugs by running your tests automatically. Merge your code with confidence using one of our continuous integration providers. Learn more'. At the bottom, a purple box with a checkmark icon states: 'Pull request successfully merged and closed. You're all set—the feature/decimate. branch can be safely deleted. Delete branch'.



Travis

- Travis is free for open repositories

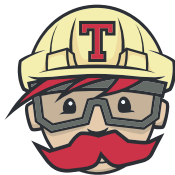


Travis CI



Travis

- Travis is free for open repositories
- It is controlled through a configuration file named `.travis.yml`

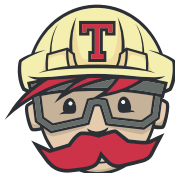


Travis CI



Travis

- Travis is free for open repositories
- It is controlled through a configuration file named `.travis.yml`
- Does not support MATLAB



Travis CI



Outline

Introduction to version control

Git

Example using git

Git work flow

Continuous integration

References



References

- Most pictures and a lot of information from:
`www.git-scm.com/book/en/v2/`
- `www.github.com`
- `www.bitbucket.org`
- This presentation:
`https://github.com/Hofsmo/presentations/tree/vc`
- `https://github.com/Hofsmo/Example`