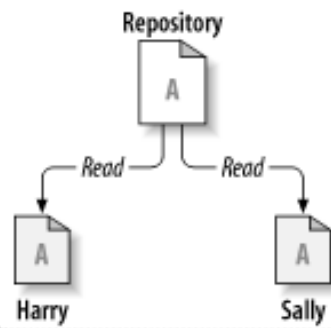


CeMM hackaton

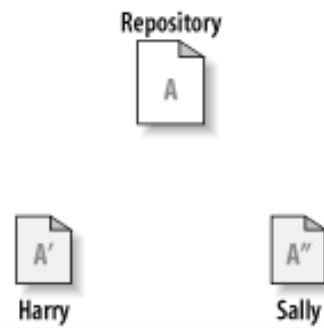
2015-03-30

<https://github.com/cemm-vienna/cemm-hackathon/tree/master/hackathons/2015-03-30>

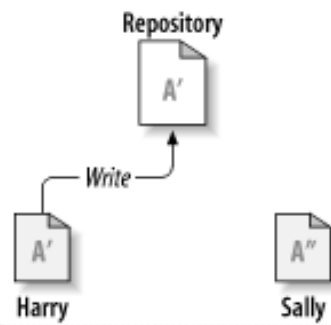
Two users read the same file



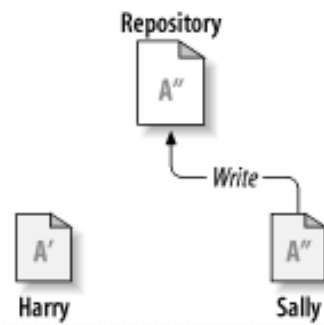
They both begin to edit their copies



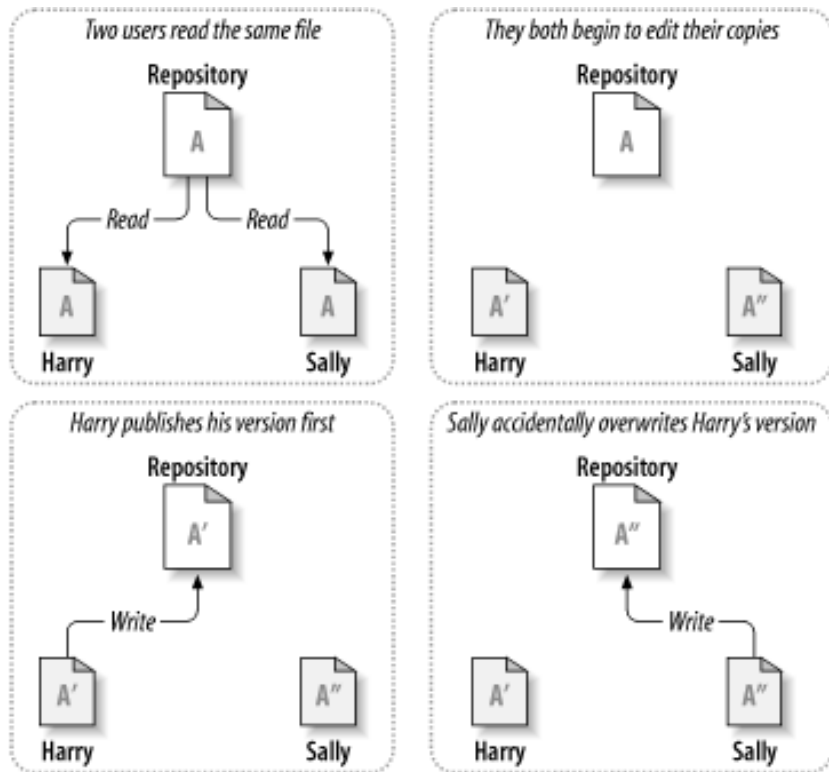
Harry publishes his version first

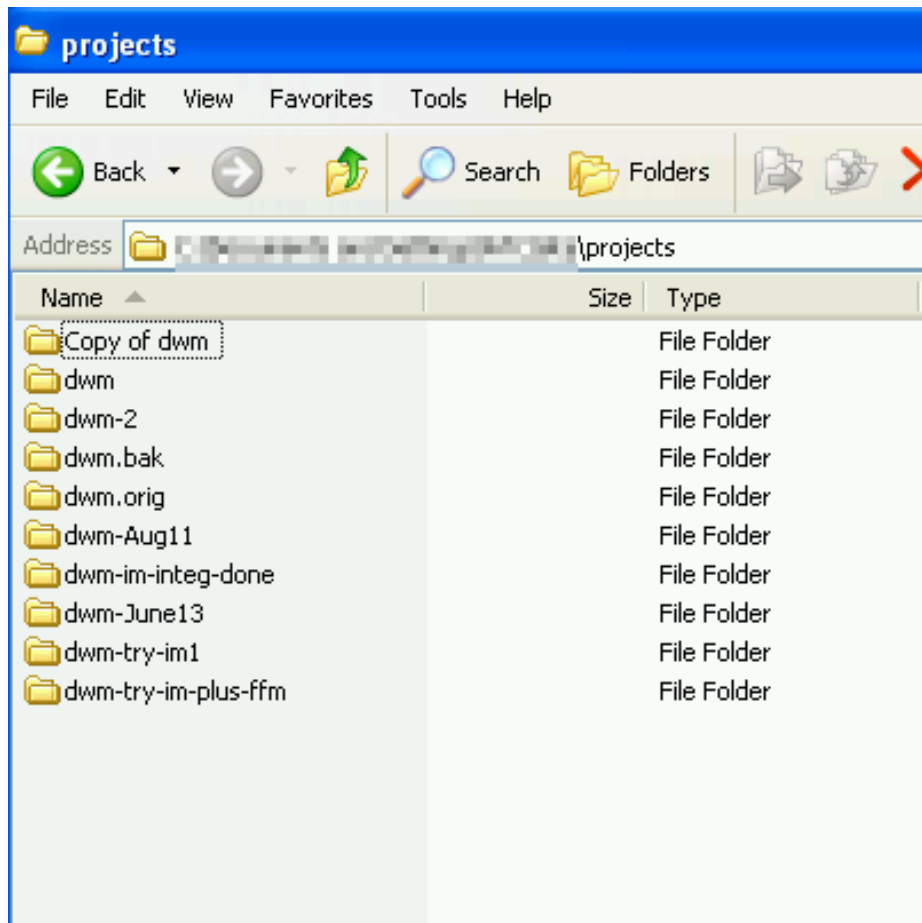


Sally accidentally overwrites Harry's version



Collaboration is hard





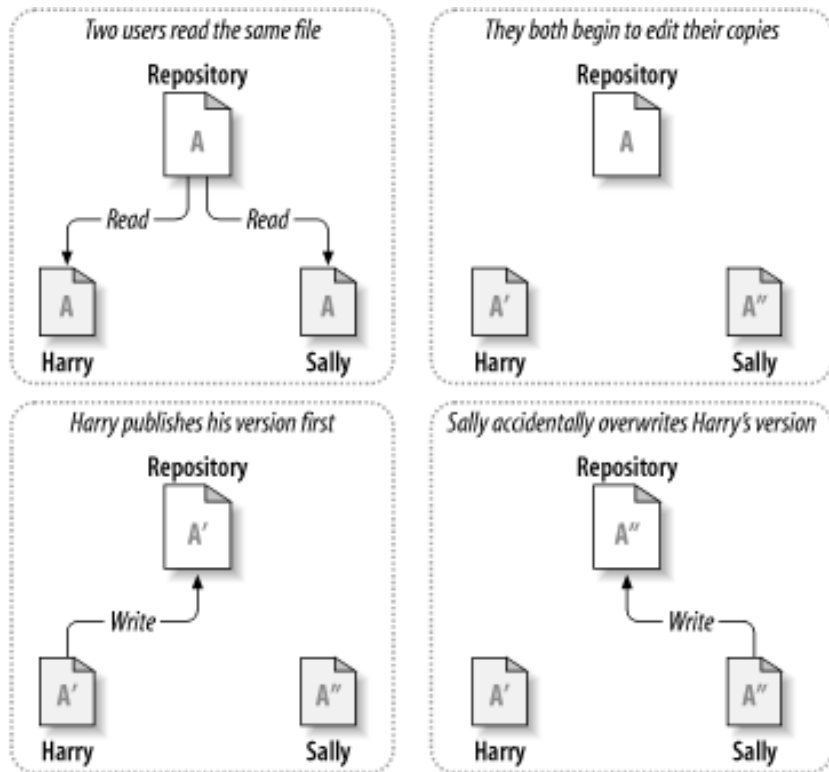
Collaboration is hard

Organization is hard

Collaboration is hard

Organization is hard

Retracing back
our steps is hard

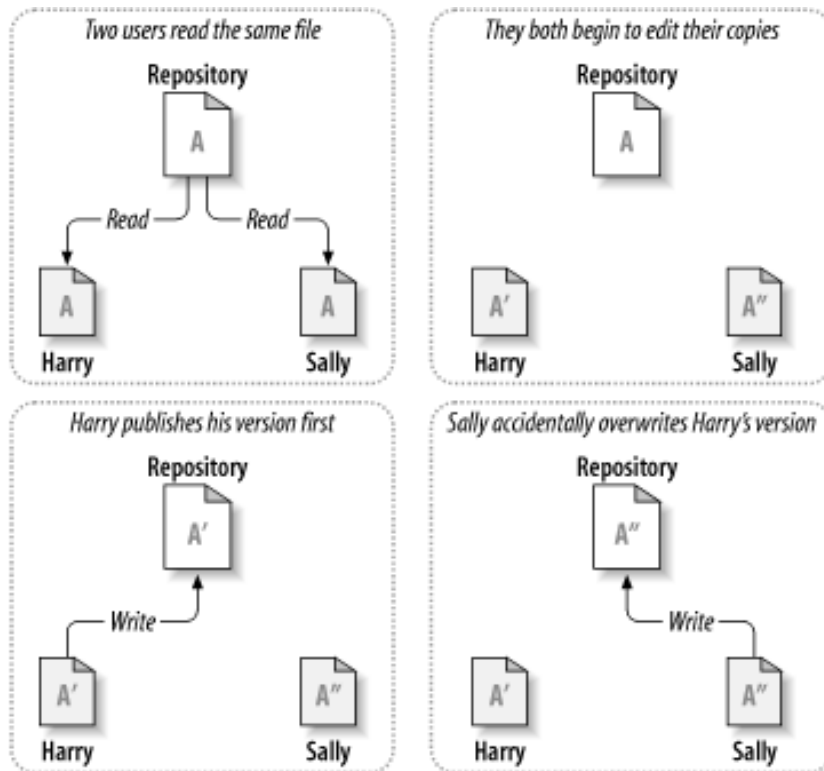


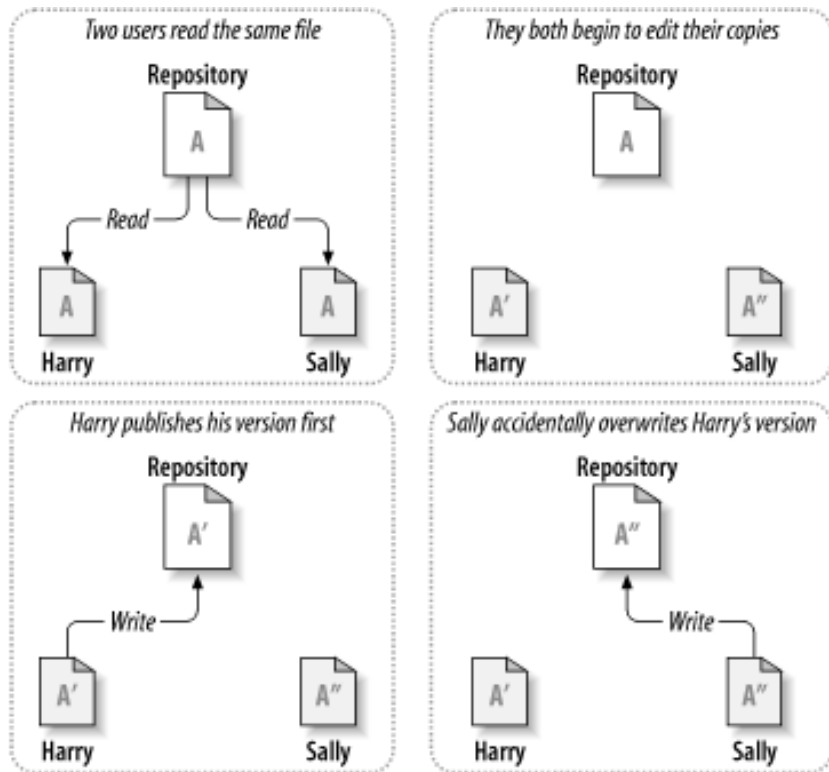
Collaboration is hard

Organization is hard

Retracing back
our steps is hard

Documenting our
progress is hard





Collaboration is hard

Organization is hard

Retracing back
our steps is hard

Documenting our
progress is hard

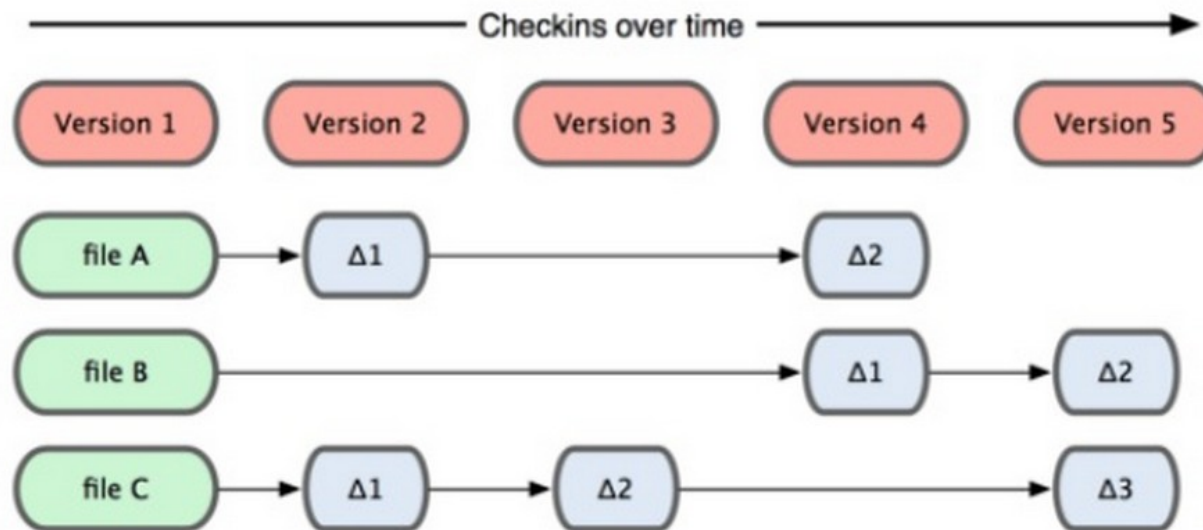


All of these are in
increasing demand

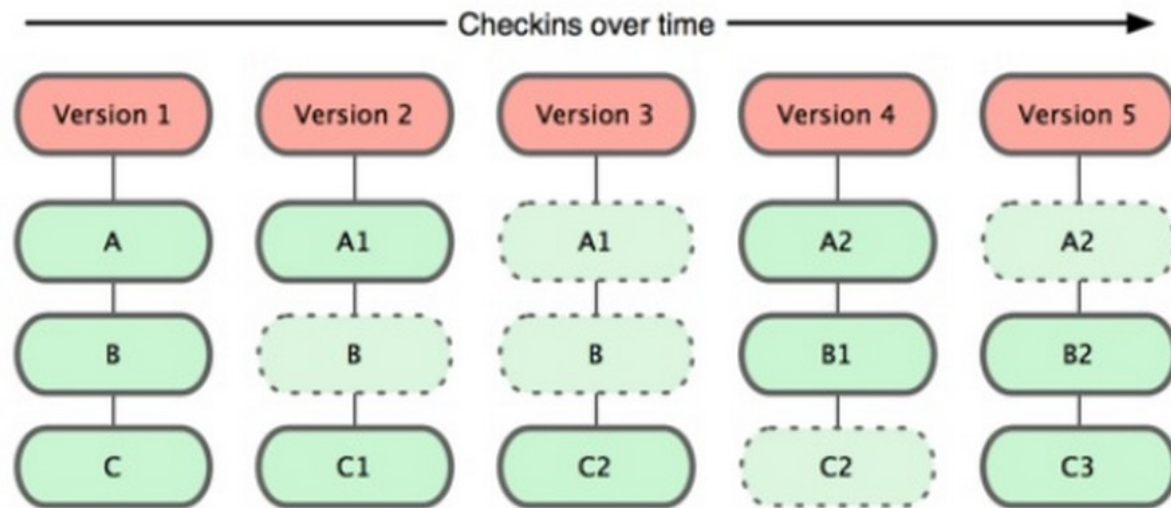
Overview

- Introduction to **version control** and **git**
- Introduction to **Markdown**
- Overview of **Github**'s features
- Experimenting with git and Github (~30 min)
- Git and Markdown for **collaborative research**
- **Discussing** usage of git CeMM-wide

Version control



Version control



Version control

- Efficient collaboration
- Traceability
- Reproducibility
- Storage and backup
- ...

git

git implements version control



workflow

your local repository consists of three "trees" maintained by git. the first one is your **Working Directory** which holds the actual files. the second one is the **Index** which acts as a staging area and finally the **HEAD** which points to the last commit you've made.



add & commit

You can propose changes (add it to the **Index**) using

```
git add <filename>
```

```
git add *
```

This is the first step in the basic git workflow. To actually commit these changes use

```
git commit -m "Commit message"
```

Now the file is committed to the **HEAD**, but not in your remote repository yet.

pushing changes

Your changes are now in the **HEAD** of your local working copy. To send those changes to your remote repository, execute

```
git push origin master
```

Change *master* to whatever branch you want to push your changes to.

If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with

```
git remote add origin <server>
```

Now you are able to push your changes to the selected remote server

branching

Branches are used to develop features isolated from each other. The *master* branch is the "default" branch when you create a repository. Use other branches for development and merge them back to the master branch upon completion.



update & merge

to update your local repository to the newest commit, execute

```
git pull
```

in your working directory to *fetch* and *merge* remote changes.

to merge another branch into your active branch (e.g. master), use

```
git merge <branch>
```

in both cases git tries to auto-merge changes. Unfortunately, this is not

always possible and results in *conflicts*. You are responsible to merge

those *conflicts* manually by editing the files shown by git. After

changing, you need to mark them as merged with

```
git add <filename>
```

before merging changes, you can also preview them by using

```
git diff <source_branch> <target_branch>
```

tagging

it's recommended to create tags for software releases. this is a known concept, which also exists in SVN. You can create a new tag named *1.0.0*

by executing

```
git tag 1.0.0 1b2e1d63ff
```

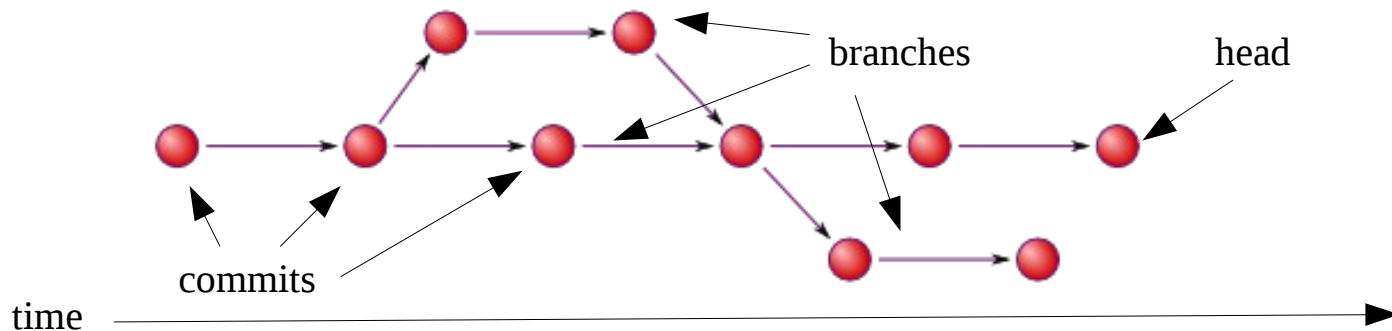
tag

commit id

the *1b2e1d63ff* stands for the first 10 characters of the commit id you want to reference with your tag. You can get the commit id by looking at the...

git vocabulary review

- repository – the files, structure and history of a project
- branch
- head – the tip of a branch (the one you add changes to)
- commit – an individual point where changes were made and saved
- push – the action of submitting commits (changes)
- pull – the action of retrieving commits
- clone – the action of duplicating a repository
- diff – a comparison of changes between commits



Sounds awesome,
what's the catch?

Sounds awesome,
what's the catch?

(We can only track properly plain text)

Markdown

- Markup language, very human-readable
- Plain text documents
- Highly compatible
- Very easy to convert to other formats
(MS word, tex, pdf, html, ...)
- Don't spend time formatting → it comes for free
- .md extension

Markdown

text using Markdown syntax	the text viewed in a browser
<pre>Heading ===== Sub-heading ----- ### Another deeper heading Paragraphs are separated by a blank line. Let 2 spaces at the end of a line to do a line break Text attributes <i>italic</i>, bold, `monospace`, ~~strikethrough~~ . A [link](http://example.com). <<< No space between] and (>>> Shopping list: * apples * oranges * pears Numbered list: 1. apples 2. oranges 3. pears The rain---not the reign---in Spain.</pre>	<p>Heading</p> <hr/> <p>Sub-heading</p> <p>Another deeper heading</p> <p>Paragraphs are separated by a blank line.</p> <p>Let 2 spaces at the end of a line to do a line break</p> <p>Text attributes <i>italic</i>, bold, monospace, strikethrough.</p> <p>A link.</p> <p>Shopping list:</p> <ul style="list-style-type: none">• apples• oranges• pears <p>Numbered list:</p> <ol style="list-style-type: none">1. apples2. oranges3. pears <p>The rain—not the reign—in Spain.</p>

1 Git, Markdown and Github

4 ****Making CeMM a git-powered institute****

6 30th of March, 2015 - Level 5 meeting room.

10 The current increase in demand for both computational skills in biology and reproducibility in research, imposes challenges in the way research projects are conducted, and its methods and results are stored and shared.

12 [Git](<http://git-scm.com/>) as a technology has the power to provide control of project changes, collaborative editing, and with technologies built on top of git, sharing projects and learning with them is made possible with [love, Internet style](<https://www.youtube.com/watch?v=Xe1TZaElTAs>).

15 # Tentative program

- 16 1. Introduction to version control and git
- 17 2. Introduction to Markdown
- 18 3. Overview of Github's features (issues, wiki, Github pages, organizations, gists)
- 19 4. Playing with git and Github (30 min hands-on)
- 20 5. Git and Markdown for collaborative research (workflow)
- 21 6. Using git CeMM-wide
- 22 + Proposal
- 23 + Discussion

25 ### Hands-on

26 #### Setup

27 Depending on the experience of the attendees these steps might need to be done initially:

- 29 1. Set up git
- 30 2. Create Github account + add keys
- 31 3. Join CeMM team
- 32 4. Get a text editor

34 #### Experimenting

35 Depending on the number and experience of the attendees we can have several activities (individual or in groups):

- 36 +
- 37 + Contribute to a tutorial (``cemm-vienna/tutorials``) on:
 - 38 + "introduction to programming" (*e.g.* unix commands tips, project organization, bioinformatics file types)
 - 39 + bioinformatic workflows (*e.g.* practical principles of read alignment, variant calling, transcriptome assembly, epigenome data processing)
 - 40 + basic plotting tutorial (*e.g.* R introduction, ggplot, matplotlib)
- 41 + Converting a own-existing project into a repository and upload it to Github (personal/private accounts for now)
- 42 + ...

45 **Note**:

- 46 + The tutorials would be a one-page markdown document that becomes part of the tutorials wiki.
- 47 + ****The goal**** is not to have a very detailed and polished document, but ****to experiment with git**** in the process. It can be finished later.
- 48 + For many things it could be simply a gathering of existing resources, rather than rewriting stuff, or a prototypical example of a task so people that already have this could simply make a markdown document with it.

51 # Prepare beforehand

52 ## Register to [Github](<https://github.com>)

Git, Markdown and Github

Making CeMM a git-powered institute

30th of March, 2015 - Level 5 meeting room.

The current increase in demand for both computational skills in biology and reproducibility in research, imposes challenges in the way research projects are conducted, and its methods and results are stored and shared.

[Git](#) as a technology has the power to provide control of project changes, collaborative editing, and with technologies built on top of git, sharing projects and learning with them is made possible with [love](#), [Internet style](#).

Tentative program

1. Introduction to version control and git
2. Introduction to Markdown
3. Overview of Github's features (issues, wiki, Github pages, organizations, gists)
4. Playing with git and Github (30 min hands-on)
5. Git and Markdown for collaborative research (workflow)
6. Using git CeMM-wide
 - Proposal
 - Discussion

Hands-on

Setup

Depending on the experience of the attendees these steps might need to be done initially:

1. Set up git
2. Create Github account + add keys
3. Join CeMM team

Git, Markdown and Github

Making CeMM a git-powered institute

30th of March, 2015 - Level 5 meeting room.

The current increase in demand for both computational skills in biology and reproducibility in research, imposes challenges in the way research projects are conducted, and its methods and results are stored and shared.

[Git](#) as a technology has the power to provide control of project changes, collaborative editing, and with technologies built on top of git, sharing projects and learning with them is made possible with [love](#), [Internet style](#).

Tentative program

1. Introduction to version control and git
2. Introduction to Markdown
3. Overview of Github's features (issues, [wiki](#), [Github](#) pages, organizations, [gists](#))
4. Playing with git and [Github](#) (30 min hands-on)
5. Git and Markdown for collaborative research ([workflow](#))
6. Using git [CeMM-wide](#)
 - Proposal
 - Discussion

Hands-on

Setup

Depending on the experience of the attendees these steps might need to be done initially:

1. Set up git
2. Create [Github](#) account + add keys
3. Join [CeMM](#) team
4. Get a text editor

Experimenting

Depending on the number and experience of the attendees we can have several activities (individual or in groups): + + Contribute to a

tutorial ([cemm-vienna/tutorials](#)) on: + "introduction to programming" (e.g. [unix](#) commands tips, project organization, [bioinformatics](#) file types) + [bioinformatic workflows](#) (e.g. practical principles of read alignment, variant calling, transcriptome assembly, epigenome data processing) + basic plotting tutorial (e.g. [R](#) introduction, [ggplot](#), [matplotlib](#)) + Converting a own-existing project into a repository and upload it to [Github](#) (personal/private accounts for now) + ...

Note: + The tutorials would be a one-page markdown document that becomes part of the tutorials [wiki](#). + **The goal** is not to have a very detailed and polished document, but **to experiment with git** in the process. It can be finished later. + For many things it could be simply a gathering of existing resources, rather than rewriting stuff, or a prototypical example of a task so people that already have this could simply make a markdown document with it.

Prepare beforehand

Register to [github](#)

- [github setup & configuration tutorial](#)

Get a text editor

General-purpose, cross-platform editors

- [Atom](#)
- [Sublime text](#)
- [Gedit](#)

Windows-only

- [Notepad ++](#)

Advanced

- [Vim](#)
- [Emacs](#)

Install git

Git for Mac

- [Download & Installation] (<http://git-scm.com/download/mac>)
- [Graphical user interface] (<https://mac.github.com/>)

Git for Windows

- [Download & Installation] (<http://git-scm.com/download/win>)
- [Graphical user interface] (<https://windows.github.com/>)

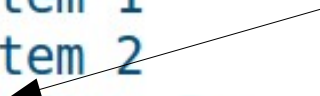
Markdown syntax

```
1
2 # This is a title
3
4 ## This is a subtitle
5
6 ### This is a subsubtitle
7
8 #### ...got it?
9 |
```

```
1 *I'm italic*
2
3 **I'm bold**
4
5 ***I have the best of both***
6 |
```

```
1 *Unordered list*
2
3 * Item 1
4 * Item 2
5   - Item 2a
6     + Item 2b
7   + Item 3
8
9 *Ordered list*
10
11 1. Item 1
12 2. Item 2
13 3. Item 3
14   1. Item 3a
15   2. Item 3b
16 |
```

Various styles
can be mixed



Markdown syntax

1 **Code blocks**

```
2 ```python
3 import os
4 print("Hello world!")
5 ```
```

6
7 **Links**

8
9 [Text](link)

10

11 [Today's page](https://github.com/ceem-vienna/ceem-hackathon/tree/master/hackatons/2014-04-24)

12 |

Code blocks

```
import os
print("Hello world!")
```

Links

Text

Today's page

Github.com

Browsable, searchable, shareable projects (repositories)

Facilitates collaboration

Increases visibility





14 minutes ago

sp00nman pushed to **master** at **cemm-vienna/cemm-hackathon****55d6d00** fixed links for images

17 minutes ago

sp00nman pushed to **master** at **cemm-vienna/cemm-hackathon****143ab78** added images to description

3 days ago

davharris opened issue **tdsmith/aRrgh#19**

indexing gotcha

**lindenb** starred **common-workflow-language/common-workflow-language** 3 days ago**sp00nman** starred **jtleek/tspreg** 3 days ago

4 days ago

sp00nman pushed to **master** at **cemm-vienna/cemm-hackathon****6b6eb65** fixed some typos

4 days ago

sp00nman pushed to **master** at **cemm-vienna/cemm-hackathon****c2c408d** Added resource links & remote access information

4 days ago

mkschuster pushed to **master** at **mkschuster/bsfpython****ed40e43** Moved functions relevant for splitting the cohort into samples into...**0396d5d** Redefined DRMS.name values as class variables, which allows c...[37 more commits »](#)

5 days ago

fhalbritter pushed to **master** at **ComputationalEpigenetics/methBench****a974283** updated all supplementary, minor other changes

Repositories you contribute to

Computational... /labDocs	0 ★
Computational... /chipmentation	0 ★
Computational... /pypiper	0 ★
lomereiter/sambamba	81 ★

Your repositories 21

[+ New repository](#)

Find a repository...

[All](#) [Public](#) [Private](#) [Sources](#) [Forks](#) **cll-patients** **dotfiles** **zf-epigenomics** **setdb2** **chipseq-pipelines** **afrendeiro.github.io** **screenImage-pipeline** **sheffien/LOLA** **master-thesis** **spp** **divideAndSlurm** **chipmentation** **epiMine** **drugAnnotation** **annotateFromGFF** **nucleoHMM**



nucleosome calling using ATAC-seq

32 commits

2 branches

4 releases

1 contributor



branch: master

NucleoATAC / +



updated version 1 increment



AliciaSchep authored 4 days ago

latest commit b37e8c0d38

bin	removed duplicated input flag	11 days ago
example	Added files to examples, updated tests for changes to variable names	10 days ago
nucleoatac	Added a max queue size to write processes to limit memory use	5 days ago
pyatac	fixed other problem with get_nucleotide	5 days ago
tests	Added a max queue size to write processes to limit memory use	5 days ago
.gitignore	Reorganization of code. Addition of new pyatac command-line utility. ...	11 days ago
LICENSE	Initial commit to public repo	18 days ago
MANIFEST.in	Added pwm text files to MANIFEST.in	11 days ago
README.md	Update README.md	10 days ago
setup.cfg	changed setup	11 days ago
setup.py	updated version 1 increment	4 days ago

README.md

NucleoATAC

Python package for calling nucleosomes using ATAC-Seq data.

Versions:

version 0 represents code as used for initial characterization of NucleoATAC method (e.g. as

<> Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

SSH clone URL

git@github.com:Greenlea

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). ?

Download ZIP

issues



This repository Search

Explore Gist Blog Help



afrendeiro



relipmoc / skewer

Watch 3

Unstar 14

Fork 2

Issues

Pull requests

Labels

Milestones

Filters

is:issue is:open

New Issue

8 Open 2 Closed

Author

Labels

Milestones

Assignee

Sort

quality encoding auto detection fails on PE files merged with SeqPrep

#12 opened 21 days ago by andreas-wilm

2

Empty FASTQ sequences on input produce corrupt FASTQ output

#11 opened on 24 Feb by wltrimbl

0

Compilation issues

#7 opened on 21 Nov 2014 by StuntsPT

2

Input from/output to BAM file

#6 opened on 9 Oct 2014 by jdidion

0

Output trimmed and excluded reads to separate output files

#5 opened on 9 Oct 2014 by jdidion

0

polyA trimming enhancement

#4 opened on 4 Aug 2014 by roryk

2

question from SEQanswers here :) question

#3 opened on 3 Aug 2014 by MichaelFokinNZ

1

Galaxy wrapper? help wanted

#1 opened on 23 Jul 2014 by bwleng

1



ProTip! Notify someone on an issue with a mention, like: @afrendeiro.

issues



This repository Search

Explore Gist Blog Help



afrendeiro



relipmoc / skewer

Watch 3

Unstar 14

Fork 2

Galaxy wrapper? #1

New Issue

Open

bwlang opened this issue on 23 Jul 2014 · 1 comment



bwlang commented on 23 Jul 2014

I think a galaxy wrapper for this tool might lead to more usage.



relipmoc commented on 24 Jul 2014

Owner

@bwlang Thank you for the suggestion!
Who can help me to do this? I'm not familiar with the galaxy API.



relipmoc added the **help wanted** label on 25 Aug 2014

Labels

help wanted

Milestone

No milestone

Assignee

No one assigned

Notifications

Subscribe

You're not receiving notifications from this thread.

2 participants



Write

Preview

Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

private repos

Free with an
institutional
account



0 Quick Start

Alicia Schep edited this page 10 days ago · 8 revisions

Installation:

For global install of latest version:

```
git clone https://github.com/GreenleafLab/NucleoATAC.git
cd NucleoATAC
sudo pip install .
```

For more details, alternate installation schemes, or trouble-shooting, see [Installation](#)

Calling nucleosomes:

Prior to using nucleoatac to call nucleosomes, you will need at least three types of files:

- 1) Bam file with aligned reads. These generally will be filtered for reads not mapping to mitochondria & reads with high mapping quality.
- 2) Fasta file with genome used in alignment. This file must be indexed (using [samtools faidx](#))
- 3) Sorted bed file with regions for which nucleosome analysis is to be performed. These regions will generally be broad open-chromatin regions (i.e. regions called by MACS2 with the `--broad` flag). It is potentially advisable to extend these regions a bit (e.g. using `bedtools slop`).

For calling nucleosomes with mostly default parameters, use

Pages 8

[Home](#)

[0 Quick Start](#)

[1 Dependencies](#)

[2 Installation](#)

[3 Command Line Utilities](#)

[4 pyatac](#)

[5 nucleoatac](#)

[9 Issues?](#)

Clone this wiki locally

<https://github.com/GreenleafLab/>



gists

GitHub Gist

All Gists



afrendeiro



afrendeiro / [getChrSizesFromFasta.py](#)

Last active on 5 Feb 2014



Edit

Delete

★ Star

0

Make tab-delimited chromosome size file from fasta genome

[getChrSizesFromFasta.py](#)

Raw

```
1 import csv
2 from Bio import SeqIO
3
4 fastagenome = "data/oikopleura/assembly/Oikopleura_reference_unmasked_v3.0.fa"
5 output = "data/oikopleura/assembly/Oikopleura_reference_chrSizes.tsv"
6
7
8 myfile = open(output, "wb")
9 spamwriter = csv.writer(myfile, delimiter='\t', quoting=csv.QUOTE_MINIMAL)
10 for seq_record in SeqIO.parse(fastagenome, "fasta"):
11     spamwriter.writerow([str(seq_record.id) , str(len(seq_record))])
12
13 myfile.close()
```

<> Code

Revisions

3

Embed URL



HTTPS clone URL



You can clone with [HTTPS](#) or [SSH](#).

Download Gist



Write

Preview





Parsed as Markdown

Edit in fullscreen

Leave a comment

Comment

Github pages

 This repository Search Explore Gist Blog Help  afrendeiro +   
















 **cboettig / labnotebook**

Watch 8 Star 75 Fork 47

Source code and version history for my online lab notebook <http://www.carlboettiger.info>

2,734 commits 4 branches 0 releases 26 contributors

 branch: master labnotebook / + 

updated vistors notes		
 cboettig authored on 15 Jan	latest commit d59b62b917	
 _includes	fix typo	5 months ago
 _layouts	fixing a few validation errors	8 months ago
 _plugins	update sha and Dockerfile	5 months ago
 _posts	updated vistors notes	2 months ago
 assets	commit figures	3 months ago
 .dockerignore	update Dockerfile to test out with servr	3 months ago
 .gitignore	debugging build.R	3 months ago
 .htaccess	keep a local copy of the .htaccess file	2 years ago
 .travis.yml	toggle off travis deploy in favor of local deploy (gets sha hashes co...	3 months ago
 404.html	remove manual timestamp tag in favor of git modified based timestamp	2 years ago
 COPYING	Modified language refering to CC0 declaration, #110	a year ago
 Dockerfile	tell servr to only build	3 months ago
 Gemfile	update	7 months ago
 Gemfile.lock	experimenting with revised Dockerfile build	3 months ago

 Code

 Issues 13

 Pull requests 1

 Wiki

 Pulse

 Graphs

SSH clone URL

git@github.com:cboettig

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). 

 Download ZIP

Github pages

[Home](#)[Vita](#)[Research](#)[Teaching](#)[Community](#)[Lab Notebook](#)

Carl Boettiger

Theoretical Ecology and Evolution

I'm [Carl Boettiger](#), a post-doctoral researcher with [Marc Mangel](#) and [Steve Munch](#) in the [Center for Stock Assessment Research](#), at [UC Santa Cruz](#), working on regime shifts in ecology and evolution. Such shifts mark the most dramatic events in complex systems. What causes these transitions to occur? Can we detect or forecast these changes? How do we robustly manage ecosystems that experience such tipping points? Understanding and forecasting these events challenges the capacity of both our statistical methods and the data available. My research aims to extend and quantify the limits of understanding through nonparametric Bayesian inference, high performance computing and big data sets. I am also a founding member of [rOpenSci](#) and keep an open [lab notebook](#), as I describe in this [introduction](#).



✉ **Contact:** cboettig@gmail.com

📍 **Address:** Center for Stock Assessment Research, 110 Shaffer Rd,
Santa Cruz, CA 95050, USA

🆔 **Orcid ID:** [0000-0002-1642-628X](#)



Government like you've never imagined.

Collaborate on code, data, or policy, within your organization or with the public.



Upgrade your IT workflow

[Git](#) is an open-source version control system that tracks who made what change when across software projects.



Public engagement, private collaboration

Publish open source software on [GitHub.com](#) or collaborate privately and securely behind your agency's firewall.



Communicate more, manage less

GitHub's issue tracking tools give you day-to-day visibility into your project's success, without the need for lengthy meetings.



Enterprise controls, startup speed

Collaborate openly while still limiting access to a team, to select stakeholders, or to an entire organization.



In your datacenter or in the cloud

Run [GitHub Enterprise](#) in your data center as a secure virtual appliance, or let us host the code for you on [GitHub.com](#).



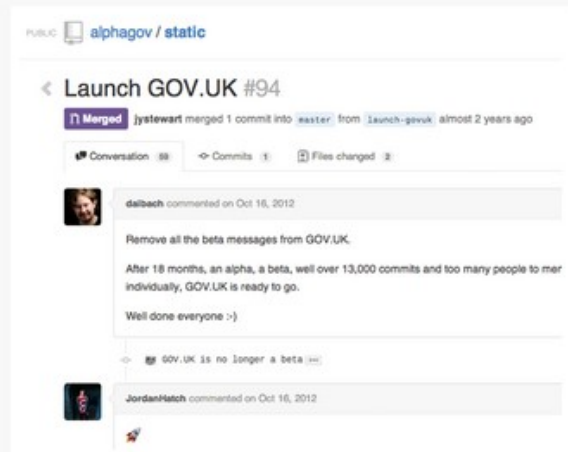
IT without the paperwork

GitHub.com's [terms of service](#) are approved for government use by the GSA, and [most plans](#) fall below the micro-purchase threshold.

More than just code.

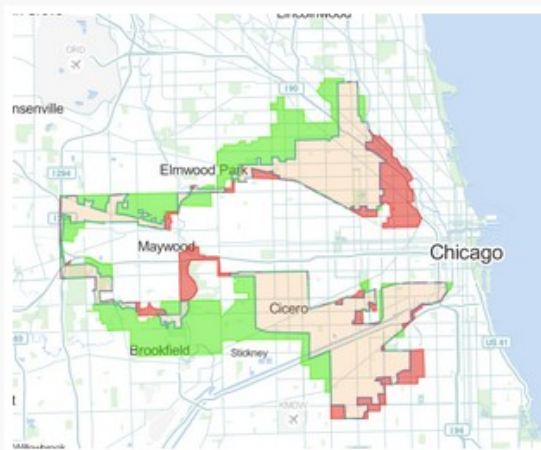
Agencies use GitHub to engage developers and collaborate with the public on open source, open data and open government efforts. GitHub even renders common formats like [text](#), [CSV](#), and [geospatial data](#).

Open Source



The UK Government launches its new site with a pull request.

Open Data



The City of Chicago release geographic city data.

Open Government



The US Government Open Data policy on GitHub.

[See more in the government showcase →](#)

Join the Revolution.

Join the **455+** government organizations around the world who collaborate on code, data, policy, and more. See

Hands on

- 1) Set up git
- 2) Create Github account + add keys
- 3) Join the CeMM team
- 4) Get a text editor

Hands on

- Write a markdown document, add to repository
- Contribute to a tutorial (*cemm-vienna/tutorials*) on:
 - "introduction to programming"
e.g. unix commands tips, project organization, bioinformatics file types
 - bioinformatic workflows
e.g. practical principles of read alignment, variant calling, transcriptome assembly, epigenome data processing
 - basic plotting tutorial
e.g. R introduction, ggplot, matplotlib
- Convert an existing project into a repository and upload it to Github
personal/private accounts for now

Git for research

- **Sample/experiment annotation**
track changes, searchable, indexed annotations - save it forever.
- **Analysis (computational or not)**
 - track changes, make reusable code, share it
 - display/share results (through md)
- **Paper writing**
changes tracked, collaborative writing/editing, comments supported

Git for CeMM

- Should we start using it systematically?
- How can we organize the repositories?

NOT SO LONG AGO.
IN A GALAXY CLOSE BY...

HEY GEORGE
WHAT'S UP?

I ACCIDENTILY
DELETED ANOTHER
PAGE OF MY
MANUSCRIPT...

NOT THIS STUPID
'SUN BATTLE' THING
AGAIN...

IT'S NOT STUPID!
- YOUR STUPID!

OH WELL...
YOU HAD IT ALL
UNDER VERSION
CONTROL RIGHT?

VERSION CON-WHAT?

UGH...