

# Git, GitHub, and Git LFS

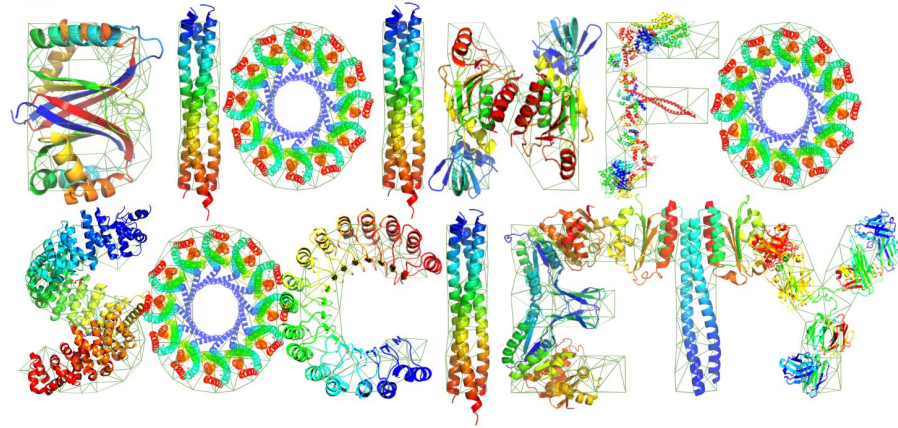
( Large File Storage )



@UoMBioinfoSoc

[uombio.info/join](https://uombio.info/join)

schedule  
contact



Louis Maddox









@biochemistries



Immx

# In today's workshop we will:


-  outline what git is, why it's so useful, and [vaguely] how it works
-  introduce GitHub and how to get student freebies
-  create a local *repository*, *commit* to it, and *push* to a *remote* copy
-  create a remote repo with the GitHub web interface, *clone* this repo locally, then *push* some *committed* changes back to the website
-  take a look at bioinformatics code on GitHub
-  introduce the Git Large File Storage (LFS) system

(*these words* will make sense by the end!)

GitHub's glossary: [help.github.com/articles/github-glossary](https://help.github.com/articles/github-glossary)

# To install Git:

## Windows

- [git-for-windows.github.io](https://git-for-windows.github.io) Git tool, mimics a Bash  command line
- [desktop.github.com](https://desktop.github.com) Desktop client buggy: command line preferable

## Mac OS X

- run `git` in Terminal for an Xcode install prompt OS X 10.9 Mavericks or later
- [git-scm.com/download/mac](https://git-scm.com/download/mac) Download a binary
- [git-scm.com/downloads/guis](https://git-scm.com/downloads/guis) Various GUI clients as above: CLI preferable

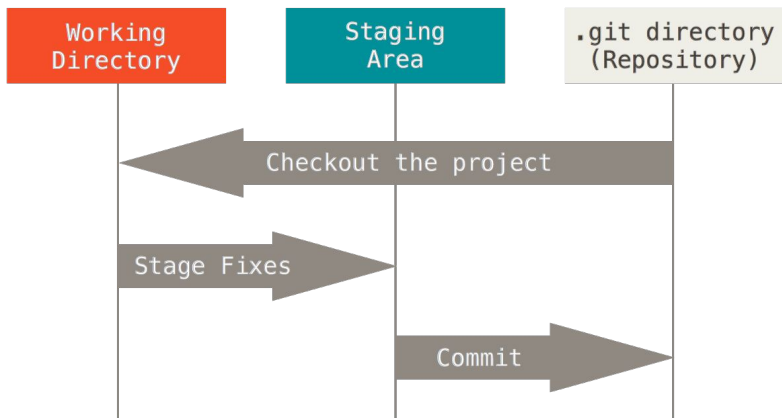
## Linux

`sudo yum install git`

or

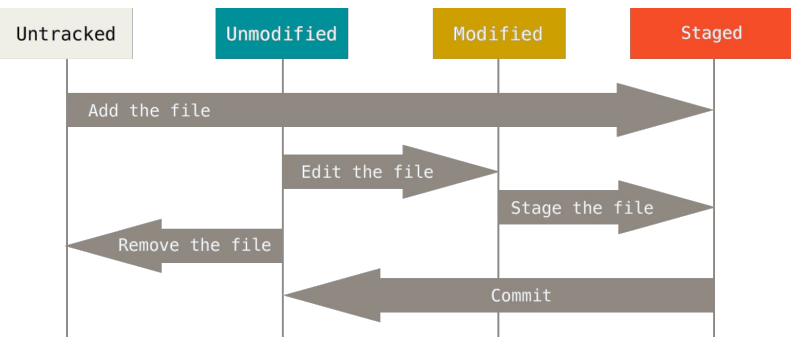
`sudo apt-get install git`

# Git is a file versioning system with 3 file states



a “commit” is a snapshot of code

- ↪ a **committed** file has been stored safely in your local database
- ↪ commits have descriptive messages, e.g. “*I fixed a bug in the program*” – annotates a file’s history



**modified** files get **staged** marked “to commit”

- ↪ `git add <file>` stages all changes in <file>
- ↪ `git commit -m “Fixed bug in <file>”`

# GitHub is an online Git repo hosting service

⇒ [GitHub.com](https://github.com)



- free bonuses for students



**GitHub**  
Student Developer Pack

- ‘Micro plan’ (5 private repos) for free : request discount [here](#)

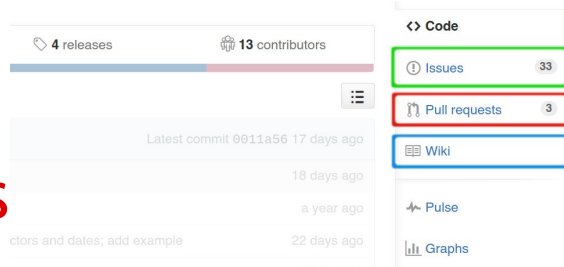
- doesn't require command line

- web interface, show history clearly, and even mobile apps
  - simplifies things like *pull requests* (“request” to merge code into repo, e.g. a *patch*)

- *Issues* for bug tracking

- In-browser handling of *pull requests*

- *Project wikis* for technical documentation (sometimes used)



# Git workflow to **send local files to remote** *e.g. GitHub*

Unix commands:

1. Initialise a local repo

```
git init
```

2. Add files (track/stage for commit)

```
git add .
```

3. Commit files

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

4. Set remote origin

```
git remote add origin <remote URL>
```

Create a new repo on GitHub to get this URL.  
Use HTTPS unless you have already set up SSH keys?

5. Push to remote

```
git push -u origin master
```

# Git workflow to **edit files in a remote repo** *e.g. GitHub*

*Note: if using this example, “fork” the repository – make a copy under your own account*

→ [github.com/UoMBioinfoSoc/cfl1-git-workshop](https://github.com/UoMBioinfoSoc/cfl1-git-workshop) → Click  in the top right)

Unix commands:

## 1. Clone the repo

`git clone https://github.com/{your-username}/cfl1-git-workshop.git`

↪ `repo URL` with `.git` on the end

## 2. Edit the downloaded files

## 3. Stage edited files

`git add .`

## 4. Commit files

`git commit -m "Commit message here"`

## 5. Push to remote

`git push -u origin master`

The remote origin is already set as where you ‘cloned’ it from.

Near-optimal RNA-Seq quantification <https://pachterlab.github.io/kallisto>

416 commits 13 branches 6 releases 4 contributors

Branch: master - kallisto / +

Merged Merge branch 'master' of https://github.com/pachterlab/kallisto Latest commit f0678a2 9 hours ago

- exitcatch add catch as an external project 9 months ago
- src Merge branch 'master' of https://github.com/pachterlab/kallisto 8 hours ago
- test add snakemake and readme to test 26 days ago
- unit\_tests Adding test 6 months ago
- gffignore Reformatting using astyle 9 months ago
- gffmodules remove Catch as a submodule 9 months ago
- ycom\_extra\_conf.py first cut of EM... not normalized 9 months ago
- CMakeLists.txt rearrange pthread to dynamically link it 5 months ago
- INSTALL.md remove note about installing tests in INSTALL 4 months ago
- README.md typo fix in sleuth URL 2 months ago
- astyle.txt Reformatting using astyle 9 months ago
- gen\_release.sh add state to gen\_release 5 months ago
- license.txt format license to 80 chars per line 6 months ago

Download ZIP

Code

Issues 13

Pull requests 1

Wiki

Pulse

Graphs

SSH clone URL

git@github.com:pr

You can clone with HTTPS, SSH, or Subversion

Download ZIP

## kallisto

**kallisto** is a program for quantifying abundances of transcripts from RNA-Seq data, or more generally of target sequences using high-throughput sequencing reads. It is based on the novel idea of *pseudalignment* for rapidly determining the compatibility of reads with targets, without the need for alignment. On benchmarks with standard RNA-Seq data, **kallisto** can quantify 30 million human reads in less than 3 minutes on a Mac desktop computer using only the read sequences and a transcriptome index that itself takes less than 10 minutes to build. Pseudalignment of reads preserves the key information needed for quantification, and **kallisto** is therefore not only fast, but also as accurate than existing quantification tools. In fact, because the pseudalignment procedure is robust to errors in the reads, in many benchmarks **kallisto** significantly outperforms existing tools.

**kallisto** quantified RNA-Seq can be analyzed with **sleuth**.

## Manual

Please visit <http://pachterlab.github.io/kallisto/manual.html> for the manual.

## License

Please read the license before using kallisto. The license is distributed with **kallisto** in the file `license.txt`, also viewable [here](#).

## Announcements

There is a low traffic Google Group, [kallisto-sleuth-announcements](#) where we make announcements about new releases. This is a read-only mailing list.

## Getting help

For help running **kallisto**, please post to the [kallisto-sleuth-users](#) Google Group.

## Reporting bugs

Please report bugs to the [GitHub issues page](#)

## Development and pull requests

# Bioinformatics on GitHub

omarwagih / siftr

9 commits 1 branch 0 releases 0 contributors

Branch: master - siftr / +

Merged Omar Wagih added median lc to output Latest commit 62384d9 on 31 Aug

- R added median lc to output 2 months ago
- build added median lc to output 2 months ago
- inst first commit 2 months ago
- man added median lc to output 2 months ago
- DS\_Store added median lc to output 2 months ago
- DESCRIPTION first commit 2 months ago
- NAMESPACE first commit 2 months ago
- README.md readme update 2 months ago

README.md

ATOMIC  
NATURAL  
KILLS

## siftr

### Predicting the impact of mutations on protein function

### Installation

To install siftr, first make sure your R version is at least R 3.0. You can check by typing the following into your R console:

```
R.Version()$Major
```

Install and load devtools package:

```
install.packages("devtools")
library("devtools")
```

Download and install the **siftr** package from github:

```
install_github("omarwagih/siftr")
```

Load the **siftr** package

```
library("siftr")
```

### Running siftr on sample data

```
# Get the path to the sample amino acid alignment
```

chr1swallace / GUESSFM

66 commits 1 branch 1 release 1 contributor

Branch: master - GUESSFM / +

chris docs Latest commit dec3bd8 on 31 Jul

- R added options to cond.best to better control initial model choice for... 3 months ago
- inst meet R CMD check 7 months ago
- man docs 3 months ago
- tests lots of changes, new vignettes, new qc functions (which need a vignette) 7 months ago
- vignettes meet R CMD check 7 months ago
- \_Rbuildignore meet R CMD check 7 months ago
- DESCRIPTION minor changes for check() 7 months ago
- NAMESPACE docs 3 months ago
- README.md Update README.md 7 months ago

README.md

## GUESSFM

R package for fine mapping genetic associations in imputed GWAS data, detailed in <http://biorxiv.org/content/early/2015/02/12/015164>.

## Installation

Depends on GUESS available at <http://www.bgx.org.uk/software/guess.html> and described in the paper <http://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1003657>

To install, you may first need some R package dependencies, the packages VGAM, reshape, ggplot2, grid, ggbio, snpStats, parallel, R2GUESS. Eg, if you don't have ggbio, from inside R, do

```
install.packages("ggbio")
```



rcsb / **symmetry** Watch 10

Detect, analyze, and visualize protein symmetry

1,340 commits 2 branches 4 releases 6 contributors

Branch: master symmetry / +

laftia Merge branch 'master' of https://github.com/rcsb/symmetry.git Latest commit 9eac401 2 days ago

docu/img	Change images in docu	2 months ago
symmetry-benchmark	Update symmetry-benchmark README	2 months ago
symmetry-core	Introduce a verbose option in CLI symmetry	4 days ago
symmetry-tools	Merge branch 'master' of https://github.com/rcsb/symmetry.git	2 days ago
.gitignore	Merge symmetry-tools repository	3 months ago
.travis.yml	Updating travis.yml for their container-based infrastructure	2 months ago
README.md	Update symmetry project README	2 months ago
pom.xml	Bumping version to 1.1.2-SNAPSHOT to distinguish from legacy branch	3 months ago

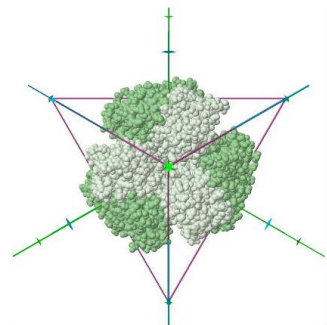
README.md

## Symmetry in Biomolecular Structures

Detect, analyze, and visualize **protein symmetry** in biological assemblies (**quaternary symmetry**) and inside single chains or domains (**internal symmetry**). The results can be visualized in Jmol.

### Quaternary Symmetry

The code to analyze and visualize the symmetry in biological assemblies was ported to the [biojava repository](#).



View the results when using this code base to [analyze all of the PDB](#).

### Internal Symmetry

The stable code to analyze and visualize the internal symmetry of biomolecular structures was also

# Bioinformatics on GitHub

lh3 / **bwa**

Watch 71 Star 231 Fork 146

Burrow-Wheeler Aligner for pairwise alignment between DNA sequences

829 commits 10 branches 28 releases 9 contributors

Branch: master bwa / +

lh3 Merge branch 'master' of github.com:lh3/bwa Latest commit 0911122 on 27 Aug

bwakit	fixed a bug in reverse-complement	2 months ago
.gitignore	Added Makefile.bak and bwamem-lite to .gitignore	3 years ago
.travis.yml	removed coverity	a year ago
COPYING	Imported from my local bwa repository, the master repository.	5 years ago
ChangeLog	Update to the latest modification 0.5.9rc1-2. Update ChangeLog	5 years ago
Makefile	added maxk: max unique k-mer at each position	9 months ago
NEWS.md	Released 0.7.12	10 months ago
QSufSort.c	removed a few unused variables	3 years ago
QSufSort.h	move bwt_gen/* to the root directory	4 years ago
README-alt.md	changed download link: 0.7.11 => 0.7.12	10 months ago
README.md	more FAQs on ALT mapping	11 months ago
bamlite.c	r837: removed BAM support	a year ago
bamlite.h	r837: removed BAM support	a year ago
bntseq.c	r896: more flexible ALT reading	a year ago

<> Code

Pull requests 22

Pulse

Graphs

SSH clone URL

git@github.com:lh3

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

Download ZIP



⚠ NOT PEER-REVIEWED. This is a rapid communication before peer review - learn more about preprints.

## Cross-platform normalization of microarray and RNA-seq data for machine learning applications

Bioinformatics Computational Biology Genomics

Jeffrey A Thompson<sup>1,2</sup>, Jie Tan<sup>1,3</sup>, Casey S Greene<sup>1,4,5,6</sup>

October 30, 2015

DOI: [10.7287/peerj.preprints.1460v1](https://doi.org/10.7287/peerj.preprints.1460v1)



greenelab / TDMresults

## TDMresults

DOI: [10.5281/zenodo.32851](https://doi.org/10.5281/zenodo.32851)



Scripts and data for re-creating TDM results.

Use `run_experiments.R` to regenerate the analyses

zenodo



# Bioinformatics on GitHub ...and Zenodo

Journal (or preprint server like *PeerJ Preprints* or *BioRxiv*) gives article a DOI,

...its associated dataset gets one that points to Zenodo.org

28 October 2015

Software Open access

## Training Distribution Matching (TDM) R Package

Jeffrey A. Thompson ; Casey S. Greene

(show affiliations)

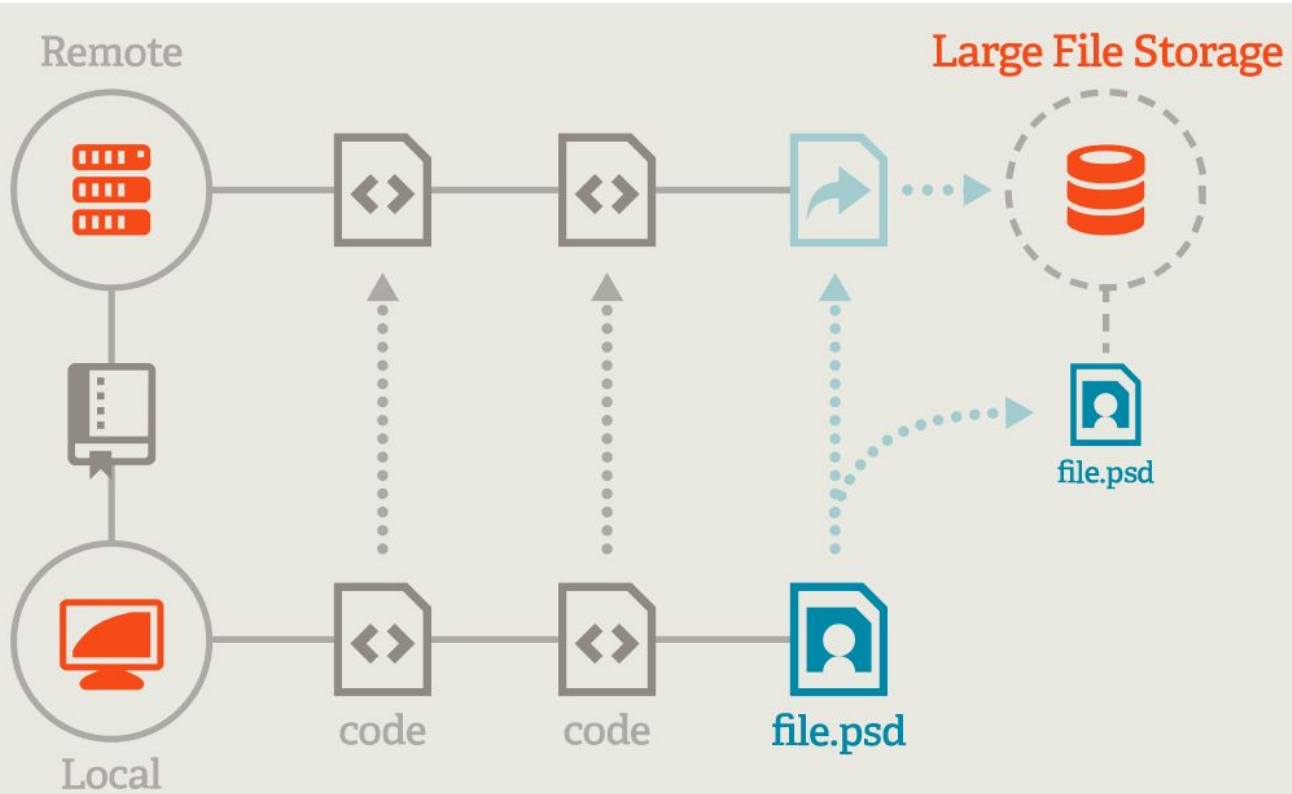
This R package implements Training Distribution Matching (TDM) as described in Thompson et al. (submitted). TDM is a normalization technique that allows machine learning models built for one platform for gene expression measurement (e.g. microarrays) to be applied to a different platform (e.g. RNA-sequencing). The companion repository (see "related") provides analytical code used to evaluate this method.

Preview

TDM-0.9.9.zip

greenelab-TDM-a342dd5	
.gitignore	18 Bytes
DESCRIPTION	400 Bytes
LICENSE	1.5 kB
NAMESPACE	302 Bytes
R	
package_loader.R	697 Bytes
tdm.R	16.3 kB
README.md	316 Bytes
data	
meta.RData	1.4 MB
tcga.RData	1.1 MB
man	
inv_log.Rd	322 Bytes

# Git Large File Storage



Storage (🗄️) can be GitHub.com, Amazon S3 “bucket” [etc.](#)

Files *appear to be* handled as normal



Same Git workflow

Work like you always do on Git—no need for additional commands, secondary storage systems, or toolsets.

# File types must be tracked

```
git lfs track *.extension
```

*e.g.*

```
git lfs track *.mp3
```

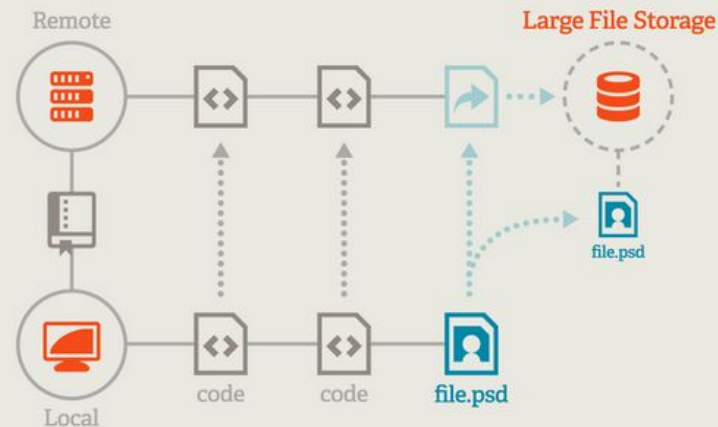
then just

```
git add .  
git commit -m "Added some mp3s"  
git push origin master
```

## An open source Git extension for versioning large files

Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise.

 **Install** v1.0.2 via PackageCloud (Linux)



*Git stores the full version of each file in "loose" format and uses compressed incremental diffs (originally based on xdiff) in packfiles (after "git gc") without distinguishing text vs binary in either case. The issue is that binary files are often compressed themselves (so a one-byte semantic change has nonlocal effect) or have positional references (like jump targets in an executable, causing small changes to cascade).*

*These factors explain the inefficient handling of binary files, but improving efficiency requires changing the semantics. LFS follows in the path of a few other tools (based on smudge/clean filters) that try to hide the semantic difference from the casual user, though that difference seems to bite people more frequently than we'd like.*

# Extras and links

- GitHub “cheat sheets” :
  - official : [training.github.com/kit/downloads/github-git-cheat-sheet.pdf](https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf)
  - long list of pro tips: [git.io/sheet](https://git.io/sheet)
- GitHub gists ([gist.github.com](https://gist.github.com))
  - Example using it as a to-do list: [bit.ly/1o76k6t](https://bit.ly/1o76k6t)
  - Ruby program to generate gists on the command line: [github.com/defunkt/gist](https://github.com/defunkt/gist)
- GitHub pages: site generator for repo named {url-prefix}.github.io
  - our site made with Jekyll + GitHub Pages
- [git.io](https://git.io): free short URL service for GitHub (including gists)
- GitHub now renders PDF, Word, Excel, PowerPoint