



Big Data Ecology

Version Control



Christian König

Ecology and Macroecology Lab Institute for Biochemistry and Biology University of Potsdam

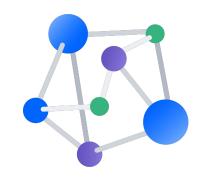
Source: https://git-scm.com/

What is version control?

- A system for managing changes to files (usually source code) that:
 - Allows multiple developers to work on the same source files
 - Stores a complete change history of all tracked files
 - Provides tools to safely experiment with your code without breaking it
 - Enables transparent and collaborative code development across the world
- Different architectures (centralized vs. distributed), implementations (e.g.: SubVersion, CVS, Git) and online hosters (e.g.: Bitbucket, GitLab, GitHub)



- Most powerful and widely used version control system available
- Distributed architecture → your local repository is self-contained and includes the full development history
- Elegant branching and merging logic
- Free web hosting with powerful features, e.g. on GitHub or GitLab





git Basics - File organization

Working directory / working tree

Your root project folder

Staging area

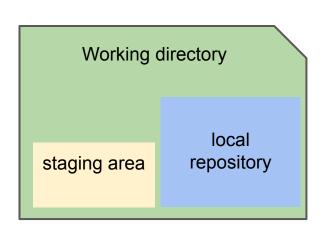
Snapshots of files that want to put under git version control

3. Local repository

Local files under git version control

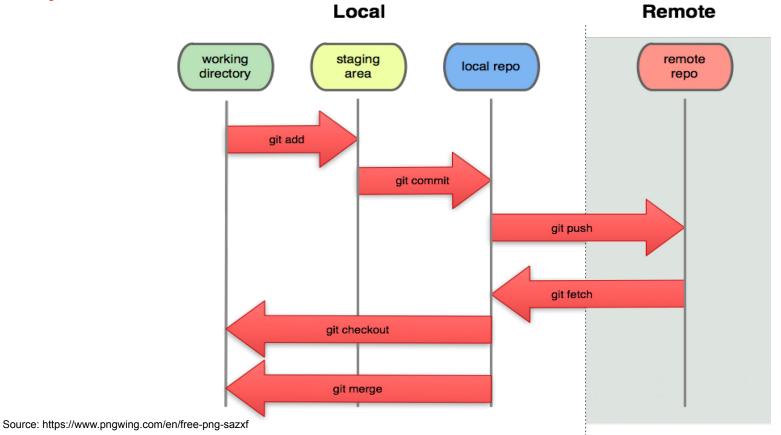
(Remote repository)

Remote repository of the same project, but with potentially different contents

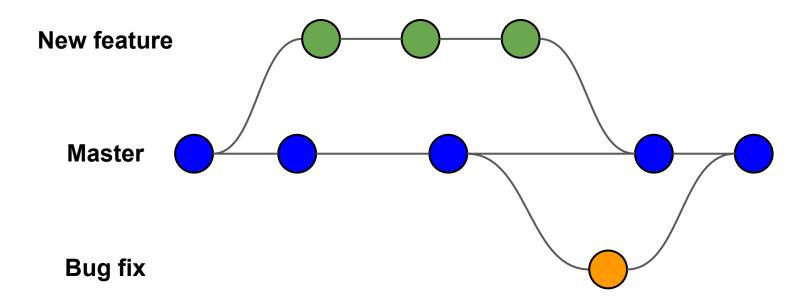


remote repository

git Basics - Workflow overview







git Web Hosting

- Web platforms can host remote repository
- They offer a number of additional powerful features such as:
 - o collaboration features, e.g. issue tracking, forks, pull requests
 - visualization tools, e.g. activity graphs, diffing tools
 - o continuous integration, i.e. automated compilation and unit testing of new code
- The most popular git web hosting platform is <u>Github</u>
- The University of Potsdam runs its own GitLab-based versioning service: Git.UP

git IDE integration

Most modern IDEs have integrated Git support

RStudio has a nice graphical interface for the most common git

operations **RStudio: Review Changes** Changes History master - G Stage • Revert Signore 🦊 Pull 🛮 👚 Push Staged Status A Path 10 Pacific invaders - upstream/ Commit message spatial_thin_log.txt Edit Code View Plots Session Build Profile Tools Help Debug Go to file/function - Addins -Diff "5modelfitting.R" Ctrl+Alt+D 5 model fitting.R > Log of "5modelfitting.R" Source on Save
 Source
 Sou → Ru Revert "5modelfitting.R"... A Packages blockCV, gbm, and 2 others required but are not 1 library(tidyverse) View "5modelfitting.R" on GitHub Amend previous commit Commit 2 library(parallel) Blame "5modelfitting.R" on GitHub library(doParallel) Show

Staged

Unstaged

Context

Slines ▼

Ignore Whitespace library(wrswoR) # fast integer sampling Commit... Ctrl+Alt+M @@ -0,0 +1,370 @@ Pull Branches library(raster) 1 ********************** library(sf) Push Branch Beginning Spatial Thinning. library(mecofun) Script Started at: Mon Apr 26 11:50:11 2021 library(dismo) History ************ library(mgcv) Beginning Spatial Thinning. Project Setup... library(gbm) Script Started at: Mon Apr 26 11:51:19 2021 library(randomForest) ************* library(blockCV) Beginning Spatial Thinning. 14 Script Started at: Mon Apr 26 11:51:37 2021 rm(list = ls())setwd("~/PacificInvadersSDM/") source("scripts/utils.R") 18



GIT BASICS		

git init <directory></directory>	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository.	git commit amend	Replace the last commit with the staged changes and last commit combined. Use with nothing staged to edit the last commit's message.
git clone <repo></repo>	Clone repo located at <repo> onto local machine. Original repo can be located on the local filesystem or on a remote machine via HTTP or SSH.</repo>	git rebase <base/>	Rebase the current branch onto dase>. dase> can be a commit ID, branch name, a tag, or a relative reference to HEAD.
git config user.name <name></name>	Define author name to be used for all commits in current repo. Devs commonly use —global flag to set config options for current user.	git reflog	Show a log of changes to the local repository's HEAD. Add —relative-date flag to show date info orall to show all refs.
git add <directory></directory>	Stage all changes in <directory> for the next commit. Replace <directory> with a <file> to change a specific file.</file></directory></directory>	GIT BRANCHES	
git commit -m " <message>"</message>	Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit message.</message>	git branch	List all of the branches in your repo. Add a <branch> argument to create a new branch with the name <branch>.</branch></branch>
git status	List which files are staged, unstaged, and untracked.	git checkout -b branch>	Create and check out a new branch named <branch>. Drop the -b flag to checkout an existing branch.</branch>
git log	Display the entire commit history using the default format. For customization see additional options.	git merge <branch></branch>	Merge <branch> into the current branch.</branch>
git diff	Show unstaged changes between your index and working directory.	REMOTE REPOSITORIES	

REWRITING GIT HISTORY

UNDOING CHANGES

UNDOING CHANGES		Chames (dil)	you can use thams as a shortcactor talls in other communities.
git revert <commit></commit>	Create new commit that undoes all of the changes made in <commit>, then apply it to the current branch.</commit>	git fetch <remote> <branch></branch></remote>	Fetches a specific <branch>, from the repo. Leave off <branch> to fetch all remote refs.</branch></branch>
git reset <file></file>	Remove <file> from the staging area, but leave the working directory unchanged. This unstages a file without overwriting any changes.</file>	git pull <remote></remote>	Fetch the specified remote's copy of current branch and immediately merge it into the local copy.
git clean -n	Shows which files would be removed from working directory. Use the -f flag in place of the -n flag to execute the clean.	git push <remote> <branch></branch></remote>	Push the branch to <remote>, along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.</remote>

git remote add

chames curls



Create a new connection to a remote repo. After adding a remote,

you can use chames as a shortcut for curls in other commands

Practical

Work through the R practical version_control.Rmd (30 min)

Further readings

Tutorials, Videos & Documentation:

https://git-scm.com/doc

https://www.atlassian.com/git

https://www.youtube.com/watch?v=2sjqTHE0zok

Git & RStudio:

https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

Github:

https://guides.github.com/activities/hello-world/

Troubleshooting:

https://stackoverflow.com/questions/tagged/git?sort=MostVotes&edited=true