

Version control for reproducible science

https://bradduthie.github.io/version_control/vc_slides.pdf

Brad Duthie

16 January 2020

Focus of this afternoon

- ▶ Understand what version control is and how it can be integrated into your work flow

Focus of this afternoon

- ▶ Understand what version control is and how it can be integrated into your work flow
- ▶ Focus on practical skills for research
 - ▶ Learn and reinforce knowledge on how to use **key skills** effectively
 - ▶ Focus on [GitHub](#) and [GitKraken](#) software

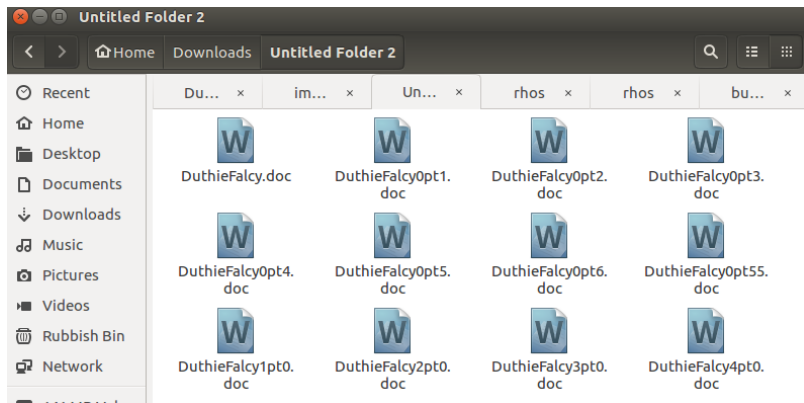
Focus of this afternoon

- ▶ Understand what version control is and how it can be integrated into your work flow
- ▶ Focus on practical skills for research
 - ▶ Learn and reinforce knowledge on how to use **key skills** effectively
 - ▶ Focus on [GitHub](#) and [GitKraken](#) software
- ▶ Hands-on practice setting up and using version control in your own work with [accompanying notes for guidance](#)

Rough outline of version control workshop

1. What is version control, and why use it?
2. Getting set up – good file management
3. The [GitKraken](#) interface and simple commits
4. Setting up [GitHub](#), pushing and pulling
5. Branching using [GitKraken](#)
6. Merging and merge conflicts
7. Independent work using version control

What is version control, and why use it?



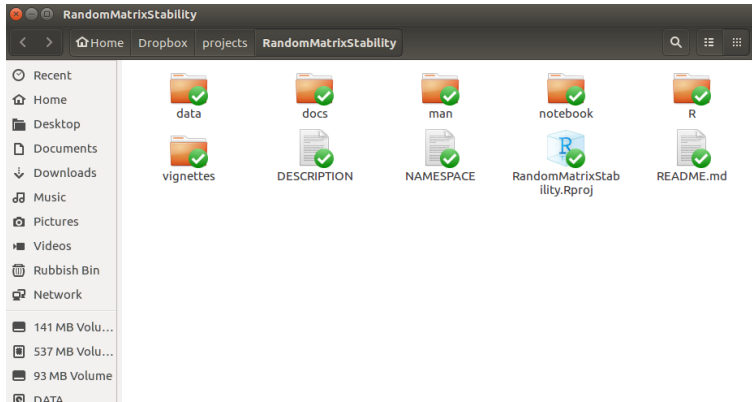
What version control software does

- ▶ Software that records changes you make to files over time
 - ▶ Manage different *versions* of files (no need to 'Save As...')
 - ▶ Recover old files, keep track of file changes
 - ▶ Collaborate with others on shared files

What version control software does

- ▶ Software that records changes you make to files over time
 - ▶ Manage different *versions* of files (no need to 'Save As...')
 - ▶ Recover old files, keep track of file changes
 - ▶ Collaborate with others on shared files
-
- ▶ **Put more intuitively**, version control takes a snapshot in time (called a '**commit**') of all the files in one of your folders (called '**repositories**')
 - ▶ Visualise changes to your files over time
 - ▶ Look at the differences between file versions
 - ▶ Record who changed files, and what they changed

Inside of a project on version control



Folders (a.k.a, 'repositories') include all data files, R code, notes, manuscript drafts, etc.

Full annotated timeline of folder changes (GitHub)

Commits on Mar 1, 2019

Some work on the SI

 bradduthle committed on 1 Mar 2019 ✓



5d29331



Major restructure and revision of the Discussion to compare to Gibbs ...

 bradduthle committed on 1 Mar 2019 ✓

..et al.



a40ede5



Commits on Feb 27, 2019

Edit the manuscript up to Reviewer 2 specific comment 3 -- these are ...

 bradduthle committed on 27 Feb 2019 ✓

..next on the list, specifically the new Discussion paragraph



d411fd5



Commits on Feb 22, 2019

First pass, fix some notation


 bradduthle committed on 22 Feb 2019 ✓



39f854b



Add some to the abstract to emphasise finite systems

 bradduthle committed on 22 Feb 2019 ✓



612aa4b



Full annotated timeline of folder changes (GitKraken)

The screenshot displays the GitKraken application interface. The top bar shows the repository name 'RandomMatrixStability' and the current branch 'master'. The left sidebar contains a file explorer with 'LOCAL' and 'REMOTE' sections. The 'LOCAL' section shows 'gh-pages' and 'master' branches. The 'REMOTE' section shows 'gitlab' with 'gh-pages', 'master', and 'origin' branches. Below the sidebar, there are sections for 'PULL REQUESTS' (0), 'TAGS' (1/1, v0.1.0), and 'SUBMODULES' (0). The main area shows a commit history timeline for the 'gh-pages' branch. The timeline consists of a vertical column of commit icons (blue circles with a white 'X') and a list of commit messages to the right. The messages are: 'Update for submission to Scientific Reports', 'Minor edits; arxiv ready', 'Tweaks', 'Add data for Gibbs table', 'Update the SI with new figures showing difference between my model and Gibbs et al', 'Prep the manuscript for both Scientific Reports and arxiv (SI still needs to be fixed in the arxiv text)', 'Revisions, need to add some still to the SI', 'Some work on the SI', 'Major restructure and revision of the Discussion to compare to Gibbs et al.', 'Edit the manuscript up to Reviewer 2 specific comment 3 -- these are next on the list, specifically the new Discu...', 'First pass, fix some notation', 'Add some to the abstract to emphasise finite systems', 'Change the wording', 'Update review', 'More setup before writing', 'Respond to Reviewer 2 specific comment 3', 'Respond to Reviewer 2 specific comment 4', 'Add code examples for Reviewer 2', 'Add more comments to Reviewer 2', 'Respond to Reviewer 2 specific comment 2', 'Respond to Reviewer 1 specific comment 1', 'Add one Reviewer 1 comment', 'Address the last of Reviewer 3's comments', 'Some initial responses to Reviewer 3', and 'Add structure to reviewer comments to begin addressing them'. The timeline is annotated with dates: '9 months ago', '10 months ago', '11 months ago', and 'a year ago'.

RandomMatrixStability - master

Undo Redo Pull Push Branch Stash Pop

search commits

Viewing 7/7 Show All
Filter (Ctrl + Alt + F)

gh-pages

LOCAL 2/2

- gh-pages
- master

REMOTE 4/4

- gitlab
 - gh-pages
 - master
- origin
 - gh-pages
 - master

PULL REQUESTS 0

TAGS 1/1

- v0.1.0

SUBMODULES 0

Commit history timeline for gh-pages:

- Update for submission to Scientific Reports (9 months ago)
- Minor edits; arxiv ready
- Tweaks
- Add data for Gibbs table
- Update the SI with new figures showing difference between my model and Gibbs et al
- Prep the manuscript for both Scientific Reports and arxiv (SI still needs to be fixed in the arxiv text)
- Revisions, need to add some still to the SI
- Some work on the SI
- Major restructure and revision of the Discussion to compare to Gibbs et al.
- Edit the manuscript up to Reviewer 2 specific comment 3 -- these are next on the list, specifically the new Discu... (10 months ago)
- First pass, fix some notation
- Add some to the abstract to emphasise finite systems
- Change the wording
- Update review
- More setup before writing (11 months ago)
- Respond to Reviewer 2 specific comment 3
- Respond to Reviewer 2 specific comment 4
- Add code examples for Reviewer 2
- Add more comments to Reviewer 2
- Respond to Reviewer 2 specific comment 2
- Respond to Reviewer 1 specific comment 1
- Add one Reviewer 1 comment
- Address the last of Reviewer 3's comments
- Some initial responses to Reviewer 3
- Add structure to reviewer comments to begin addressing them (a year ago)

Parallel versions ('branches') of a folder (GitKraken)

The screenshot displays the GitKraken application interface for a repository named 'version_control'. The left sidebar shows the repository structure with a 'LOCAL' section containing 'gh-pages' (checked) and 'list_A', 'list_B', 'list_C', 'list_D', 'master', and 'restructure'. The 'REMOTE' section shows 'version_control' with branches 'avocado', 'gh-pages', 'master', and 'restructure'. The 'PULL REQUESTS' section shows one pull request for 'version_control' titled '#1 Change to avocado ...'. The 'TAGS', 'SUBMODULES', and 'GITHUB ACTIONS' sections are empty.

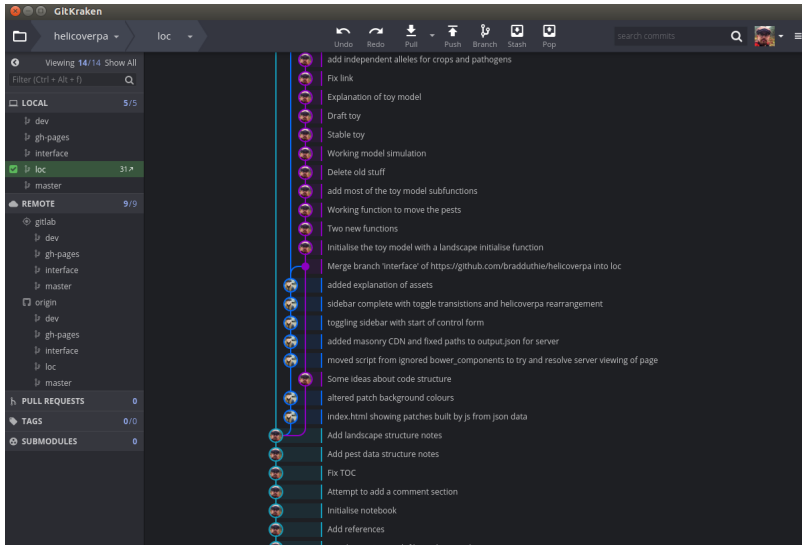
The main area shows a commit history graph with a vertical timeline of commits. The 'gh-pages' branch is highlighted in green. The commit history includes:

- Discuss branching in GitHub
- Change to avocado and start a new ...
- Merged branch master into gh-pages
- Fix links and minor edits
- Merged branch master into gh-pages
- Fix spelling
- Merged master into gh-pages
- Typo and cheet sheets.
- First draft of notes added
- Nearing end of GitHub stuff
- How do you link them apples?
- More notes
- Merge branch 'list_C'
- Add the notes I just wrote
- Change Apples to Bananas
- Change Apples to Pears
- CLI merge conflict
- Change to Lettuce
- Another paragraph on merge conflict
- Merge branch 'list_B'
- Merged branch list_A into master
- Paragraph explaining merge conflict
- Change Lettuce to Spinach
- Change Lettuce to Cucumber on list
- Commit the first change with the list
- Start merge conflict section

The right sidebar shows the commit details for commit '5ad7ea'. The commit message is 'Merge branch 'master' into gh-pages'. The commit was authored by Brad Duthie on 3/12/2019 at 21:00. The commit includes 4 modified files and 5 added files:

- RStudio_and_git.html
- RStudio_and_git.Rmd
- vc_notes.html
- vc_notes.Rmd
- vc_presentation.html
- vc_presentation.pdf
- vc_presentation.Rmd
- vc_slides.pdf
- vc_slides.Rmd

Collaborative history or a shared folder (GitKraken)



Clear breakdown of what has changed (GitKraken)

Repository: RandomMatrixStability x +

branch: master

Undo Redo Pull Push Branch Stash Pop

notebook/ms.Rmd

Edit in working directory File View Diff View Blame History

22 file changes in working directory View changes

commit: d411fd

Edit the manuscript up to Reviewer 2 specific comment 3 -- these are next on the list, specifically the new Discussion paragraph

Brad Duthie authored 27/2/2019 @ 18:00 parent: 39f854

8 modified + 1 added

Expand All notebook ms_files 4 ms.pdf ms.Rmd ms.tex PLoS_Compu_reviews.html PLoS_Compu_reviews.md

```
251 251
252 252 Randomly assembled complex systems can be represented as large square matrices ( $M$ ) with  $S$  components (e.g.
253 253
254 254 May's [May1972; Allesina2012] stability criterion  $\sigma(\sqrt{SC}) < 1$  assumes that the expected response rates ( $\gamma$ )
255 255
256 256 May's [May1972; Allesina2012] stability criterion  $\sigma(\sqrt{SC}) < 1$  assumes that the expected response rates ( $\gamma$ )
257 257

360 360
361 361 It is important to emphasise that variation in component response rate is not stabilising per se; that is, adding variation in comp
362 362
363 363 -- Also important to emphasise Gibbs result -- I'm doing this for finite systems, and I deliberate stressed the system complexi
364 364
365 365 Also important to emphasise Gibbs result -- I'm doing this for finite systems, and I deliberate stressed the system complexity to
366 366
367 367 But Gibbs was more interested in first assuming a stable matrix and then showing that the vector of abundances would not cha
368 368
369 369
370 370

364 371
365 372 The potential importance of component response rate variation was most evident from the results of simulations in which the
366 373

387 394
388 395  $\frac{d\mathbf{M}(\mathbf{v})(t)}{dt} = \mathbf{M}(\mathbf{v})(t) \mathbf{A} \mathbf{v}(t)$ 
389 396
390 396
391 397 In the above,  $\mathbf{M}(\mathbf{v})(t)$  is a diagonal matrix in which elements correspond to individual component response rates. T
392 398
393 399 **Genetic algorithm**. Ideally, to investigate the potential of  $\mathbf{S}(\mathbf{v})$  for increasing the proportion of stable complex s
394 400
```

Version control makes science easier

- ▶ **Organises files** by avoiding 'save as' multiple versions
 - ▶ analysis_1.R
 - ▶ analysis_2.R
 - ▶ analysis_FINAL.R
 - ▶ analysis_FINAL_no_really_this_time.R

Version control makes science easier

- ▶ **Organises files** by avoiding 'save as' multiple versions
 - ▶ analysis_1.R
 - ▶ analysis_2.R
 - ▶ analysis_FINAL.R
 - ▶ analysis_FINAL_no_really_this_time.R
- ▶ **Provides a clear history** of what you have done, when, and why (through commit comments)

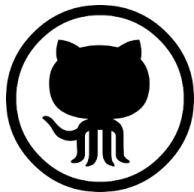
Version control makes science easier

- ▶ **Organises files** by avoiding 'save as' multiple versions
 - ▶ analysis_1.R
 - ▶ analysis_2.R
 - ▶ analysis_FINAL.R
 - ▶ analysis_FINAL_no_really_this_time.R
- ▶ **Provides a clear history** of what you have done, when, and why (through commit comments)
- ▶ **Saves time** by avoiding loss of data, analysis, or writing when integrating with [GitHub](#)

Version control makes science easier

- ▶ **Organises files** by avoiding 'save as' multiple versions
 - ▶ analysis_1.R
 - ▶ analysis_2.R
 - ▶ analysis_FINAL.R
 - ▶ analysis_FINAL_no_really_this_time.R
- ▶ **Provides a clear history** of what you have done, when, and why (through commit comments)
- ▶ **Saves time** by avoiding loss of data, analysis, or writing when integrating with [GitHub](#)
- ▶ **Gives peace of mind** to experiment by removing any fear of breaking something that you know works

Version control can help open science



- ▶ Transparent record of data collection, analysis, and writing
- ▶ Record publicly available on [GitHub](#), [Bitbucket](#), or [GitLab](#)
- ▶ GitHub repository can be copied, reproduced, and discussed
- ▶ [git](#) and GitHub can track individual contributions to a project

Most researchers use git (and GitHub)



- ▶ Free and open-source
- ▶ Separate from [GitHub](#)

Most researchers use git (and GitHub)



- ▶ Free and open-source
- ▶ Separate from [GitHub](#)
- ▶ Works across platforms
 - ▶ Windows
 - ▶ Linux
 - ▶ Mac
- ▶ Invented by [Linus Torvalds](#)

Why focus on using GitKraken?



- ▶ Free to download and use
- ▶ Easy GitHub integration
- ▶ Graphical user interface
- ▶ Visualisation of repository

Accompanying notes to these slides are available in the **version_control** repository, and include instructions for using the command line interface, and for editing directly in GitHub.

Objectives: using version control

Guided walkthrough of setting up a project in [GitHub](#) and [GitKraken](#) to manage a project with version control.

Slides:

https://bradduthie.github.com/version_control/vc_slides.pdf

Notes:

https://bradduthie.github.com/version_control/vc_notes.html

Discuss, share, and get additional help by [raising an issue](#) in the [version_control repository](#) on the [Stirling Coding Club](#).