

Using Git, Part 3: Collaboration

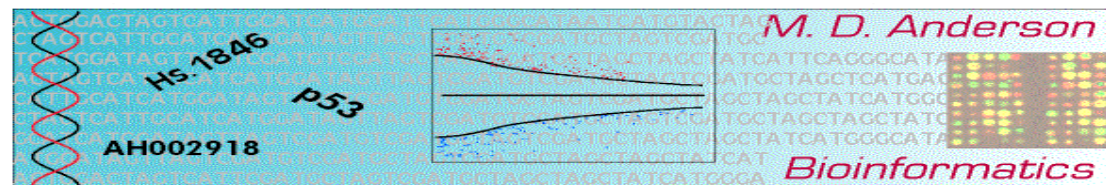
Keith A. Baggerly

Bioinformatics and Computational Biology

UT M. D. Anderson Cancer Center

kabagg@mdanderson.org

SISBID, July 18, 2017



Sharing vs Collaborating

A static webpage anyone can download from is sharing.

If multiple people interact on a final project, that's collaborating.

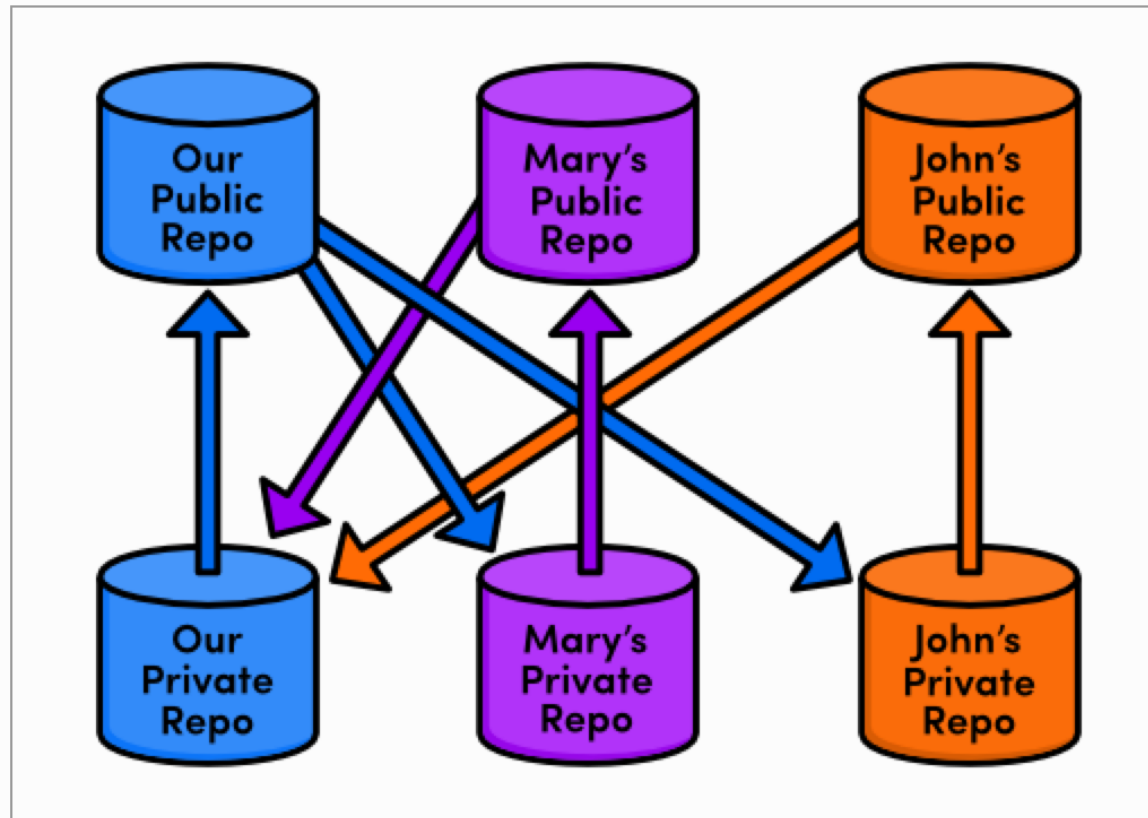
Your number of collaborators need not be large.

I often work with just one or two analysts on assembling reports, where after a few iterations, we're done.

Here, we can specify a few known collaborators who can all push to and pull from the repo (e.g., Karl and I are collaborators on the Module 3 repo).

It's possible to give or get feedback from others

Suggestions from Others



The integrator workflow with many developers

more from Ry's Git Tutorial, Distributed: Everyone can share!
Pull requests

How Other Changes Work

Let's say you're looking at someone else's repo because it relates to your work, and you notice a bug. If you see a fix, you can suggest this as follows.

fork the repo you're looking at. This creates a copy of the repo in your GitHub account which you own and can edit.

clone your repo copy from GitHub to your local machine.

Make your edits.

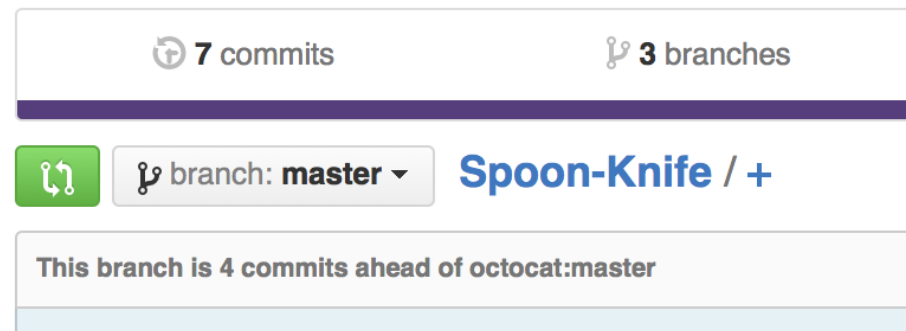
Push your edited version of the repo back to GitHub.

Send the original owner a pull request, asking them to pull the edits you've made into their repo.

Fork and Clone

Clicking the + sign next to the repo name will fork it for you.

This repo is for demonstration purposes only.



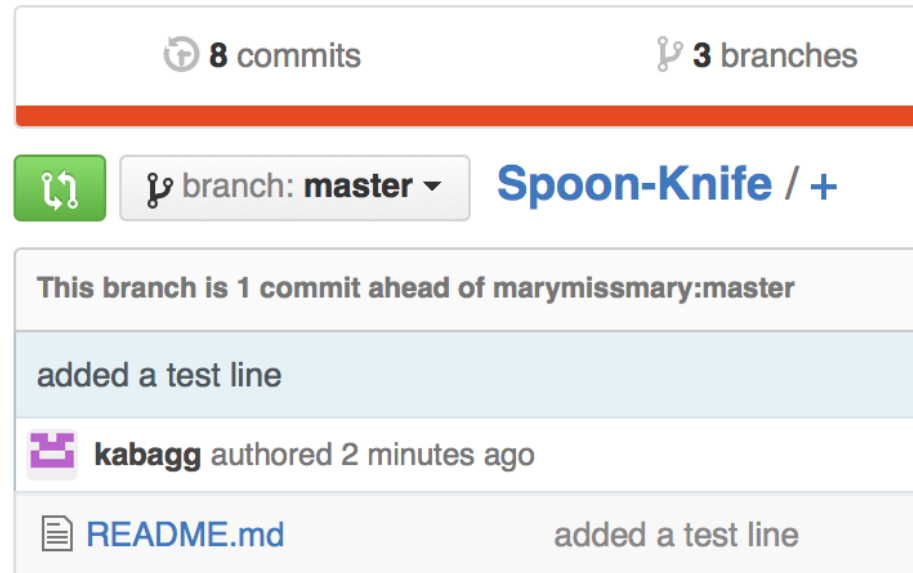
Now, within Rstudio, go to
File/New Project/Version Control/Project Setup
and give it the git url.

This will set up a local project repo for you (with a .Rproj file)

Edit and Push

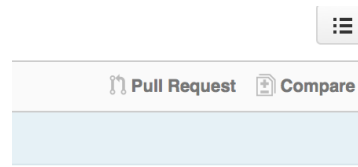
After making your edits, use the Rstudio commit popup to ignore the Rproj file and commit your desired changes.

Then push the changes over to GitHub



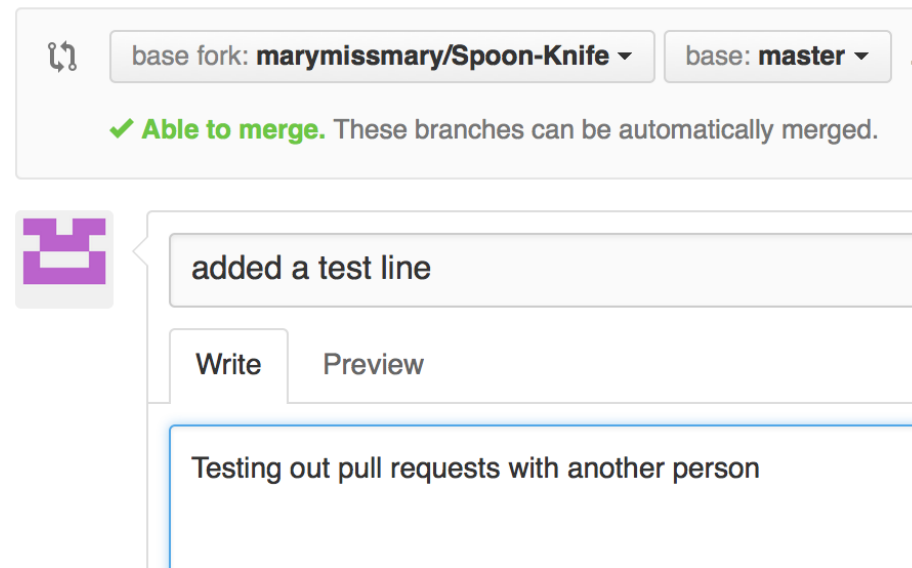
Submit the Pull Request

Assemble the pull request (PR), along with a brief comment about what you're trying to do.



Open a pull request

Create a new pull request by comparing changes across two branches.




You Get Some GitHub Feedback


Even before you submit the PR, GitHub will indicate whether including the suggested changes is easy.



Comparing changes


Choose two branches to see what's changed or to start a new pull request



 base fork: **marymissmary/Spoon-Knife** ▼ base: **master** ▼ ...

✓ **Able to merge.** These branches can be automatically merged.

 **Create pull request** Discuss and review the changes in this comparison

 **1 commit**  **1 file changed**

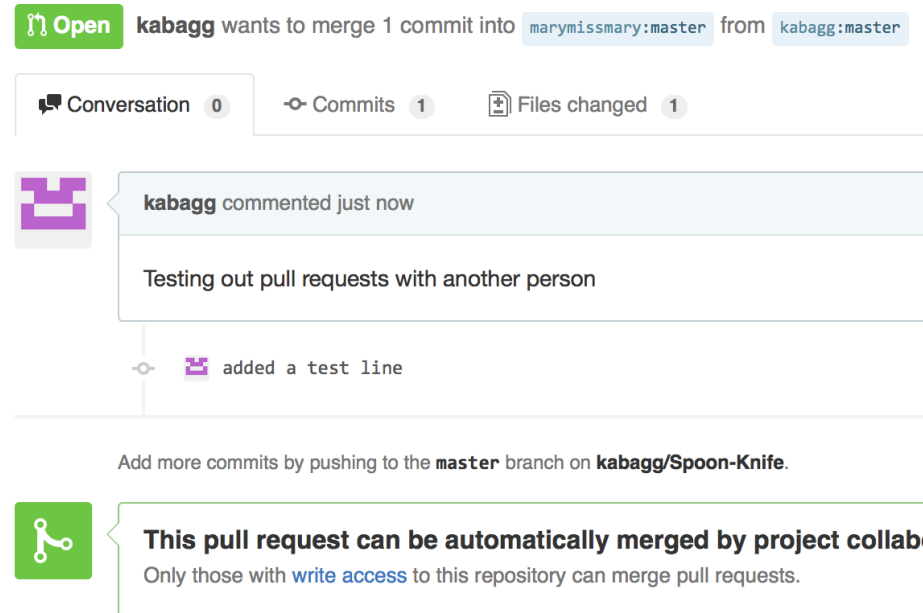
 Commits on Jul 08, 2015

  **kabagg** added a test line

Welcome to the Conversation

Submitting the PR starts a “conversation” with the owner (initially by email)

added a test line #1



The screenshot displays a GitHub pull request interface. At the top, a green button with a key icon and the word "Open" is visible. Below it, a status bar indicates "kabagg wants to merge 1 commit into marymismary:master from kabagg:master". A navigation bar shows "Conversation 0", "Commits 1", and "Files changed 1". The main content area features a conversation thread. A comment from "kabagg" is shown, stating "kabagg commented just now" and "Testing out pull requests with another person". Below the comment, a commit from "kabagg" is listed, titled "added a test line". At the bottom, a green box with a key icon contains the text: "This pull request can be automatically merged by project collaborator. Only those with write access to this repository can merge pull requests."

Open kabagg wants to merge 1 commit into `marymismary:master` from `kabagg:master`

Conversation 0 Commits 1 Files changed 1

kabagg commented just now

Testing out pull requests with another person

kabagg added a test line

Add more commits by pushing to the **master** branch on **kabagg/Spoon-Knife**.

This pull request can be automatically merged by project collaborator.
Only those with [write access](#) to this repository can merge pull requests.

Welcome to the Conversation (2)

The owner can choose to include your suggestion (or not)

added a test line #1

Merged maryissmary merged 1 commit into `maryissmary:master` from `kabagg:master` a day ago

Conversation 1

Commits 1

Files changed 1



kabagg commented a day ago

Testing out pull requests with another person

added a test line



maryissmary commented a day ago

I think these are great additions to this project! I'll merge them into the master branch




maryissmary merged commit **fa19650** into `maryissmary:master` a day ago

Including Suggestions from Others

Of course, they can comment on your code as well ;).


Embellished description of plotCircle #1

 **Open** marymismary wants to merge 2 commits into kabagg:master from marymismary:master

Conversation 0


Commits 2


Files changed 1




marymismary commented a day ago


Expanded description in plotCircle to briefly describe what the function does.


 Rohrdanz,Mary A and others added some commits a day ago


 Embellished description of plotCircle ... 5bd4c15

 making a merge conflict ... fe03da

Add more commits by pushing to the master branch on marymismary/toyPackage.

 **This pull request can be automatically merged.**
You can also merge branches on the [command line](#).

 **Merge pull request**



Write

Preview

Markdown supported Edit in fullscreen

pull = fetch + merge

When you pull down data from a remote repo, you're actually doing two things:

- (1) fetching the data (as a new branch), and
- (2) trying to merge that branch with what you've got in place.

Sometimes you may want to split these steps apart, so you can “fetch” the data and edit it before trying to merge it. This may let you avoid merge conflicts, or simply reduce their number.

Sending Yourself a Pull Request

Not all pull requests need to be to others; you can send them to yourself as well.

Why?

If you're actively interacting with others, this may be a way of getting input from them before making a change

If, in the request, you mention them by name with TheirGitHubID, they'll be specifically pinged (don't do this to folks you don't know)

Sometimes I have Issues

Sometimes I notice a problem with (or a potential improvement to) a package I or someone else has developed, but I don't know how to fix it (yet).

In this case, I can raise it as an Issue with the repository

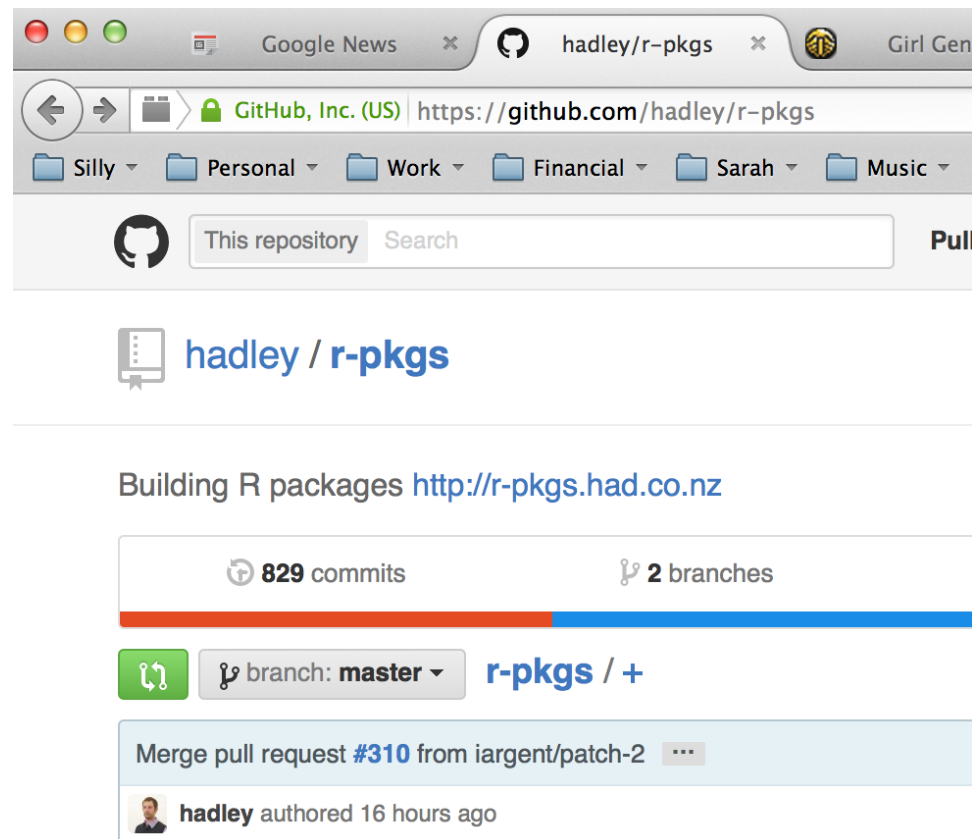
Not all issues are for problems! You can use issues as part of a “to do” list for the next release of your package

Issues get assigned numbers.

If a new commit to the main branch mentions “Fixes” IssueNumber, the issue is “closed” with a link to that commit.

Others' Files on GitHub

While we're here, let's look around. Exploring others' repos is a great way to learn. Here're some snapshots from Hadley's *R Packages book* repo



An Introduction

Zooming in on specific files shows the contents



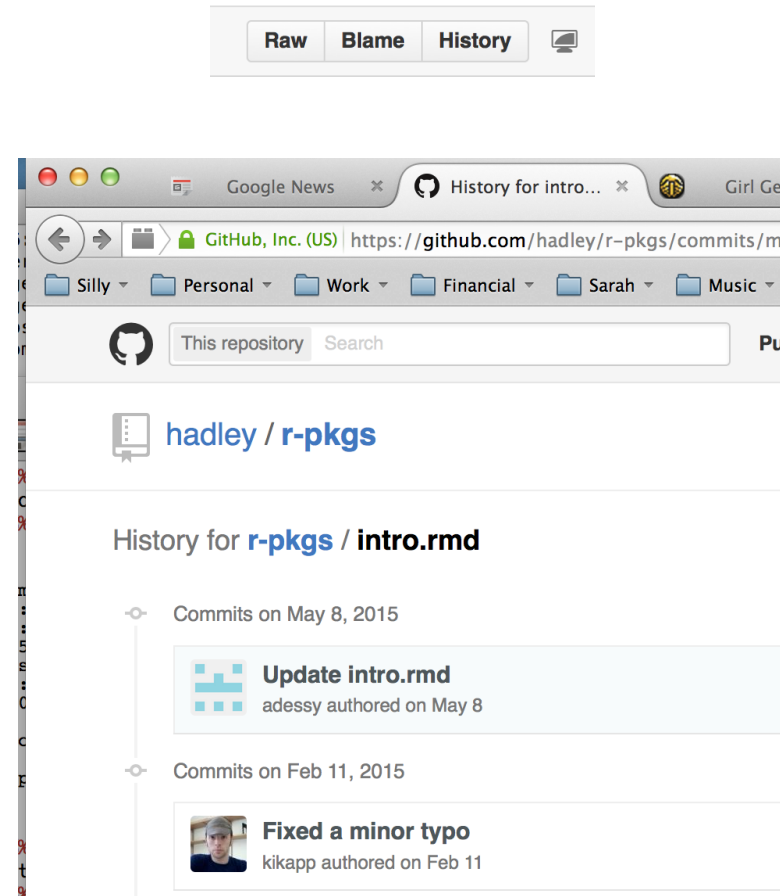
The screenshot shows a web browser window displaying the GitHub repository page for 'hadley / r-pkgs'. The browser's address bar shows the URL 'https://github.com/hadley/r-pkgs/blob/master/intro.rmd'. The repository name 'hadley / r-pkgs' is prominently displayed. Below this, a commit message 'adessy on May 8 Update intro.rmd' is visible, along with a row of 17 contributor avatars. The file statistics show '207 lines (146 sloc)' and '14.943 kB'. A table below the statistics lists the file's configuration:

layout	title	output
default	Introduction	bookdown::html_chapter

At the bottom of the page, the title 'Introduction {#intro-why}' is displayed in a large, bold font.

Checking the History

Checking the history shows when edits were made



Checking an Edit

Checking an edit shows the specific changes

Fixed a minor typo


 master (#251)



kikapp authored on Feb 11



Showing **1 changed file** with **1 addition** and **1 deletion**.

2  intro.rmd



@@ -91,7 +91,7 @@ You'll start by learning about

91	91	The final three chapters describe general best p
92	92	
93	93	* [Git and github](#git): mastering a version co
94		- colloborate with others, and is useful even fo
	94	+ collaborate with others, and is useful even fo
95	95	you to easily undo mistakes. In this chapter y
96	96	popular git and GitHub combo with RStudio.
97	97	



Improving Signal to Noise

Find a few folks who you think do good work, and explore their repositories to see what they do (and possibly why they do it).

Hadley Wickham

Yihui Xie

Jenny Bryan

RStudio folks

Jeff Leek

Learning More

Git will require practice, and more examples than we've provided here. Fortunately, there are many good examples on the web.

Scott Chacon's Pro Git, also in book form

Ry's Git Tutorial, also in e-book form

Jon Loeliger, Version Control with Git, 2e (Amazon)

GitHub Help

How We Can Collaborate with GitHub

So, what can you apply this to?

R packages

analyses for papers

grant applications (e.g., references, multiple drafts)

homework submissions (precise timestamps!)

editing/commenting on reports

and lots of other stuff!
