

Git

2018-09-25

1 Git

- noun
 1. British Slang. a foolish or contemptible person.
- "goddamn idiotic truckload of shit" – source code readme file
- track changes for anything

2 Take Home

2.1 Why use version control

2.2 4 Essentials (9 commands)

have typed in and understood

```
git init    / git clone      # get
git add     # modify
git commit -m "change note" # modify
git push    / git pull      # interact
git status  / git log / git diff # inspect
```

2.3 Github

- get/have one
- access <https://github.com/LabNeuroCogDevel/allscripts>
 - useful
- fork and pull

3 Getting Started

3.1 Why

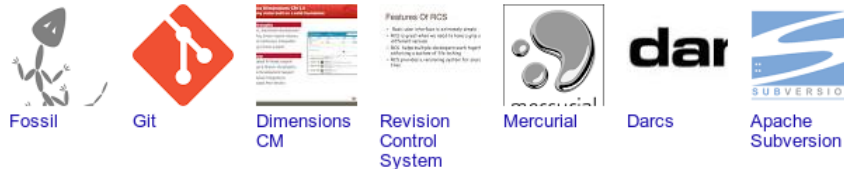
1. It was working!?! What happened! (git diff/log)
 2. Collaborate, identify conflicts/changes (git diff/log)
 3. Share. Reproduce. Reuse. (github, git clone)
 4. At home and now the scripts are 3 hops away (github)
 5. backup (git push)
 6. fast search (git grep)
- useful for any development
 - version control in shared storage is less exciting i.e. better for between labs than within a lab.

3.2 What

(distributed) Version Control

3.2.1 options: git, mercurial, fossil, darcs [centralized: svn, csv]

version control software



3.2.2 stores history of changes in a smart way

1. locally

```
#16:34:19 rhea:/Volumes/Phillips/mMR_PETDA/scripts
git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working di

    modified:   02_ppmprage.bash
    modified:   05_glm.bash
    modified:   07_extractDTBZBP.bash
    modified:   07b_extractBeta.bash
    modified:   10_extractR2p.bash
    modified:   11_extractR2p.bash
    modified:   12_resampleR2p.bash
    modified:   99_warp_to_18_09c.bash
    modified:   MT/runme.bash
    modified:   MTtimequick.bash
    modified:   Makefile
    modified:   checkStatus.sh
    modified:   cronenv.bash
    modified:   extract_pet_metrics.bash
```

2. online github diff

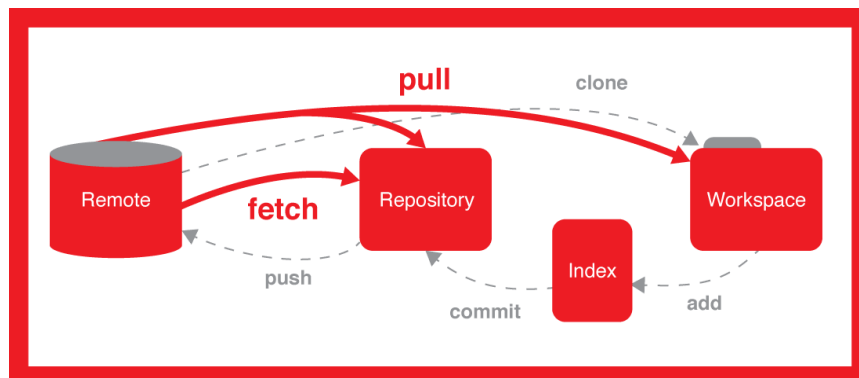
```

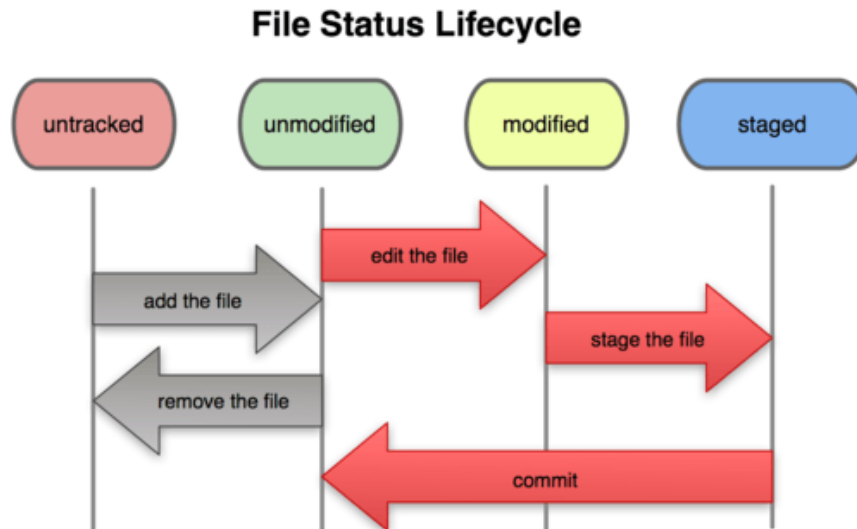
33 ... -1,19 +1,7 @@
1
2
3
4 - #3dLME -overwrite -prefix ext_ME_cue_Ecor_20180828 \
5 - # -mask ../cov_mask_thresholded_999_20180614.nii.gz \
6 - # -model 'Ext_t+ageatvisit+visit+reward+meanmotion' \
7 - # -qVars "Ext_t,ageatvisit,meanmotion" \
8 - # -ranEff '~1' \
9 - # -jobs 8 \
10 - # -SS_type 3 \
11 - # -num_glt 1 \
12 - # -gltLabel 1 'ext' -gltCode 1 'Ext_t : ' \
13 - # -dataTable @dataTable_3dLME_ECOR_20180823_cue.txt
14
15
16 - 3dLME -overwrite -prefix ext_ME_prep_Ecor_20180905 \
17 -mask ../cov_mask_thresholded_999_20180614.nii.gz \
18 -model 'Ext_t+ageatvisit+visit+reward+meanmotion' \
19 -qVars "Ext_t,ageatvisit,meanmotion" \
20
21 @@ -22,10 +10,10 @@
22 -SS_type 3 \
23 -num_glt 1 \
24 -gltLabel 1 'ext' -gltCode 1 'Ext_t : ' \
25 -dataTable @dataTable_3dLME_ECOR_20180823_prep.txt
26
27
28 - 3dLME -overwrite -prefix ext_ME_resp_cor_20180905 \
29 -mask ../cov_mask_thresholded_999_20180614.nii.gz \
30 -model 'Ext_t+ageatvisit+visit+reward+meanmotion' \
31 -qVars "Ext_t,ageatvisit,meanmotion" \
32
33 @@ -34,4 +22,17 @@
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

3.3 How

3.3.1 Workflow - take home part 2





3.3.2 Git software (clients)

- git (command line interface)
- <https://git-scm.com/download/gui/windows>

3.4 Setup

If you don't do this now, git will demand you do it later.

```
git config --global user.name "First Last"
git config --global user.email "email@domain.com"
```

4 Make our own

4.1 overview

1. open terminal
2. make a new directory and init a git repo
3. create
4. stage (add) a file
5. commit

6. change, add, commit

7. view log

4.2 looks like

```
test -r ~/src/git_demo && rm -rf $_ # just for me
mkdir -p ~/src/git_demo # be creative here
cd $_
# run git status after each command
git init # start git repo

# round one - a new file
echo "a\nb" > my_file # modify a file
git add my_file # stage file
git commit # stage to HEAD

# round two - change a file
sed -i -e 's/a/c/' my_file # modify a file
git diff # whats changed
git add my_file # stage file
git commit # stage to HEAD

git log # history

git grep b # search files for 'b'
```

4.3 explore

```
git log
git diff HEAD~1
git diff $checksum
```

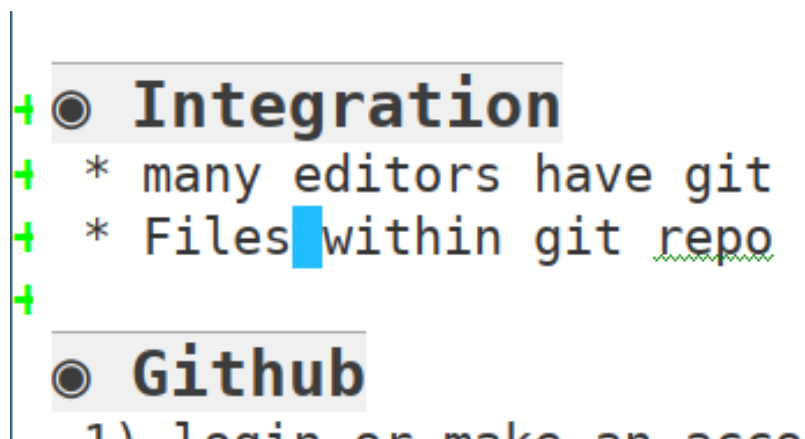
4.4 locally, thats it

- on Zeus, Hera, and Phillips, that's all there is.
- be a code custodian
 - commit in meaningful units with useful messages
 - git log will help everyone figure out whats going on
 - git diff will explain changes

5 Integration

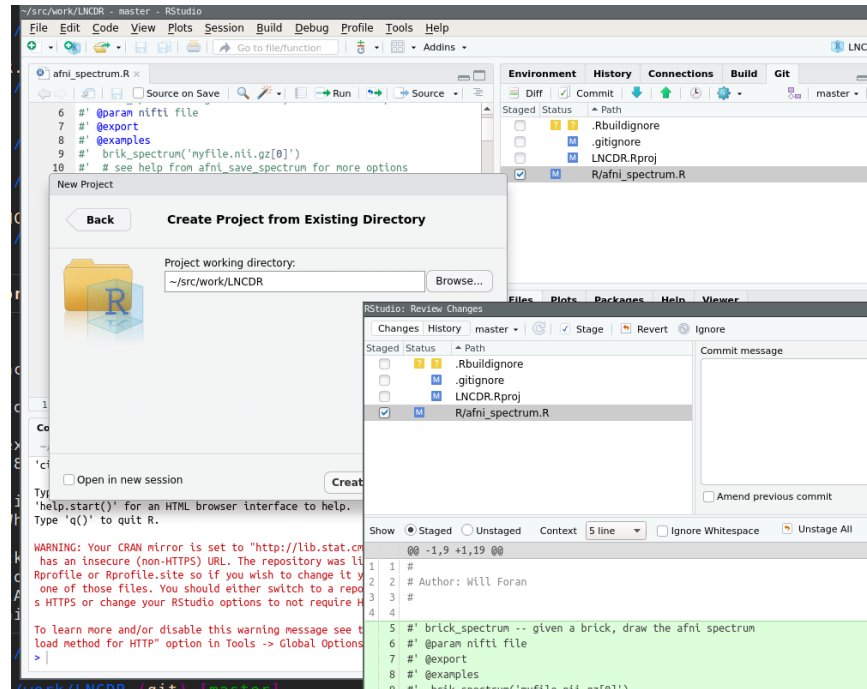
5.1 Generic

- many editors have git support baked in (matlab, rstudio, vim, ...)
- see atom
- Files within git repo have visual indications.



5.2 RStudio

file -> new project : see git tab tutorial



6 Github

6.1 Your very own repo

1. login or make an account on github.com
2. confirm or request access to the group github.com/LabNeuroCogDevel
3. create a new repo (for your demo above, creative naming still encouraged)
4. follow instructions to sync github to your code

```
git remote add origin remote $GITHUB_URL
git push origin master
```


6.2 Collaborating with strangers

6.2.1 overview



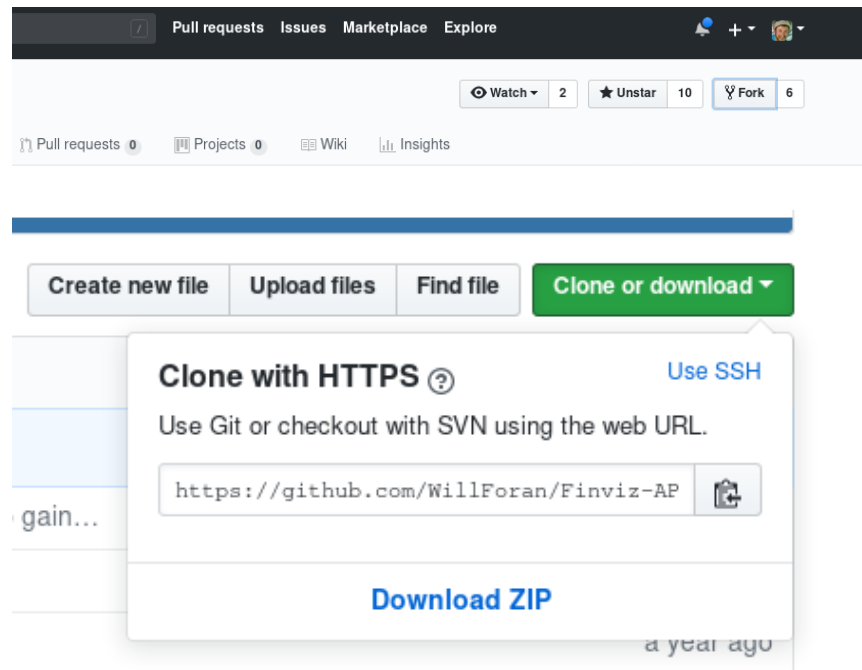
collaborating with others on github

1. fork repo to make it your own
2. clone your own repo
3. make changes, commit, push
4. make a pull request

6.2.2 Fork

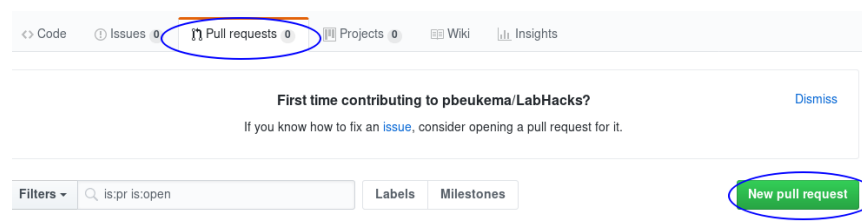
Awesome paper publishes code on github. There's a bug and we can fix it!

1. go to repo: `github.com/$USER/$REPO`
 - for now, get the repo from the person to your left
 - or fork `https://github.com/pbeukema/LabHacks`
 - or this: `https://github.com/WillForan/git_demo`
2. click fork (N.B. not going to contribute? just clone – no need to fork)
3. clone your forked copy (`git clone https://github.com/\protect\T1\textdollarUSER/\protect\T1\textdollarREPO.git`)



6.2.3 Pull Request

When you want to send the original author your changes, you make a "pull request".



6.2.4 Merge

You've shared useful code. Someone's made a change. You can merge it into your own.

1. pull requests -> accept

6.3 Hub

hub makes this easier

6.3.1 creating

```
hub create LabNeuroCogDevel/new_project
```

6.3.2 forking

```
hub clone LabNeuroCogDevel/LabHacks
cd LabHacks
hub fork
git add ... git commit
hub pull-request
```

7 Finer points

- Make commit messages useful: <https://gist.github.com/matthewhudson/1475276>
- git does branching very well. ideal for experimenting with radical code changes
- .git/ folder
- ssh keys for github

8 Resources

(see wiki)

- https://www.authorea.com/users/5990/articles/17489-a-quick-introduction-to-version-control-show_article
- <https://berkeley-stat159-f17.github.io/stat159-f17/lectures/01-git/Git-Tutorial..html>
- <https://try.github.io/>
- <https://blogs.mathworks.com/community/2014/10/20/matlab-and-git/>

9 Not git

9.1 Colophon

emacs

- org mode
- leuven theme
- emamux:

```
(defun send-line ()
  "Select the current line"
  (interactive)
  (move-beginning-of-line nil)
  (evil-visual-char)
  (move-end-of-line nil)
  (emamux:send-region (region-beginning) (region-end))
  (evil-exit-visual-state)
  (next-line 1)
)

(global-set-key (kbd "C-c s") 'send-line)

(defun org-show-current-heading-tidily ()
  (interactive) ;Interactive
  "Show next entry, keeping other entries closed."
  (if (save-excursion (end-of-line) (outline-invisible-p))
      (progn (org-show-entry) (show-children))
      (outline-back-to-heading)
      (unless (and (bolp) (org-on-heading-p))
        (org-up-heading-safe)
        (hide-subtree)
        (error "Boundary reached")))
      (org-overview)
      (org-reveal t)
      (org-show-entry)
      (show-children)))

(defun show-next ()
  "focus next header in org mode"
  (interactive)
  ;(call-interactively (org-next-visible-heading))
  (execute-kbd-macro (kbd "C-c C-n"))
  (org-show-current-heading-tidily)
  (org-show-current-heading-tidily))
```

```
(recenter-top-bottom 2)
)

(global-set-key "\M= " 'show-next)
```

9.2 Deja vu

file:///Volumes/L/bea_res/Luna/Journalmtg/2017-04-27_IT_WF.txt