Sham's intro to Git/GitHub

Disclaimer:

This is non-exhaustive, and I'm not claiming any expertise.

I'm also assuming no one else is an expert.

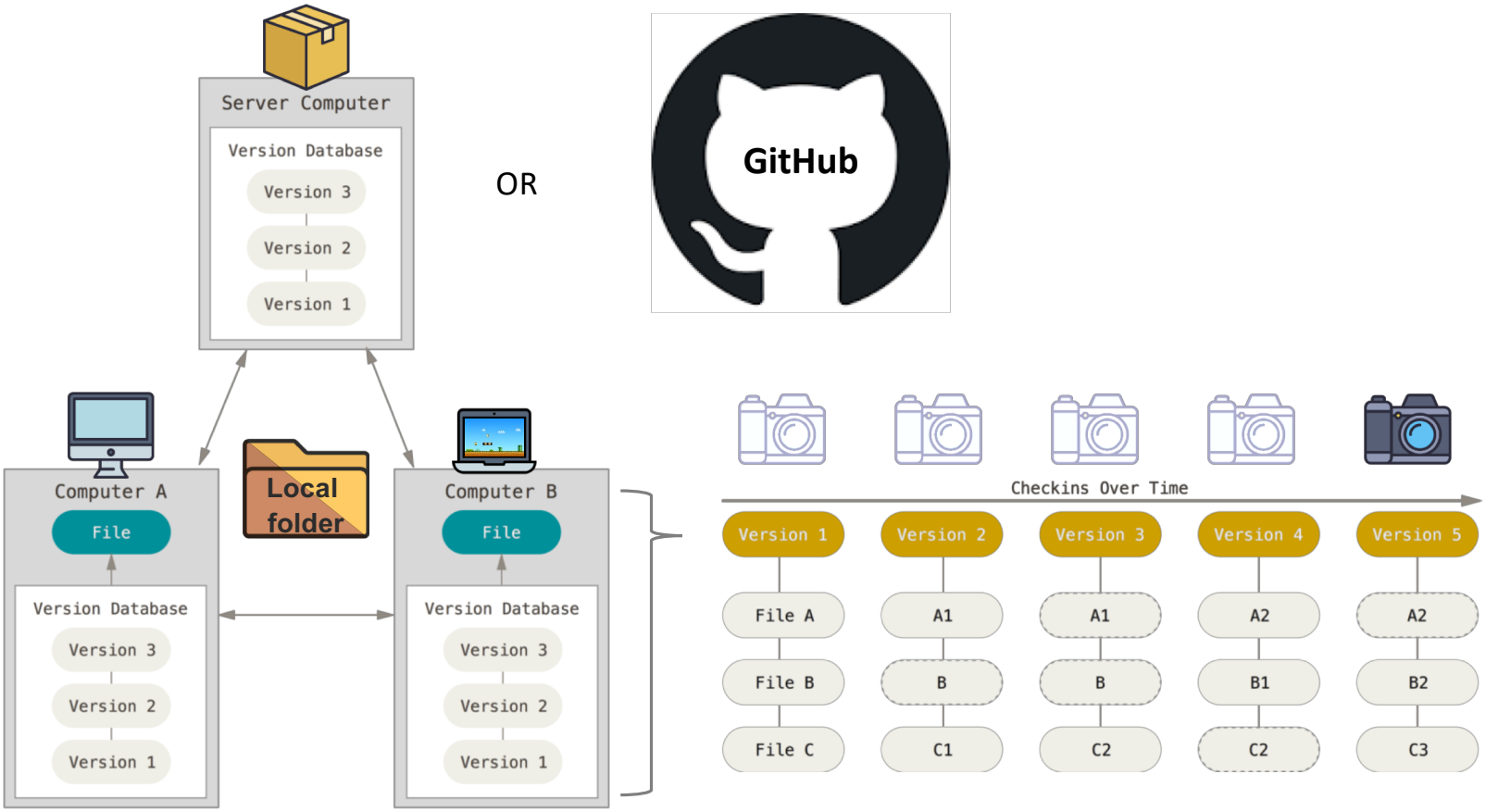
Git vs GitHub

- Version control system for recording changes and coordinating collaborative code
- Parses **text** files for changes – format is important
- Intentional saving – not automated backup
- Requires organization of projects into separate folders
- Macs already have git

- Web host service that uses git version control systems
- Particularly useful for publicly sharing code
- Academic accounts (.edu) can be private

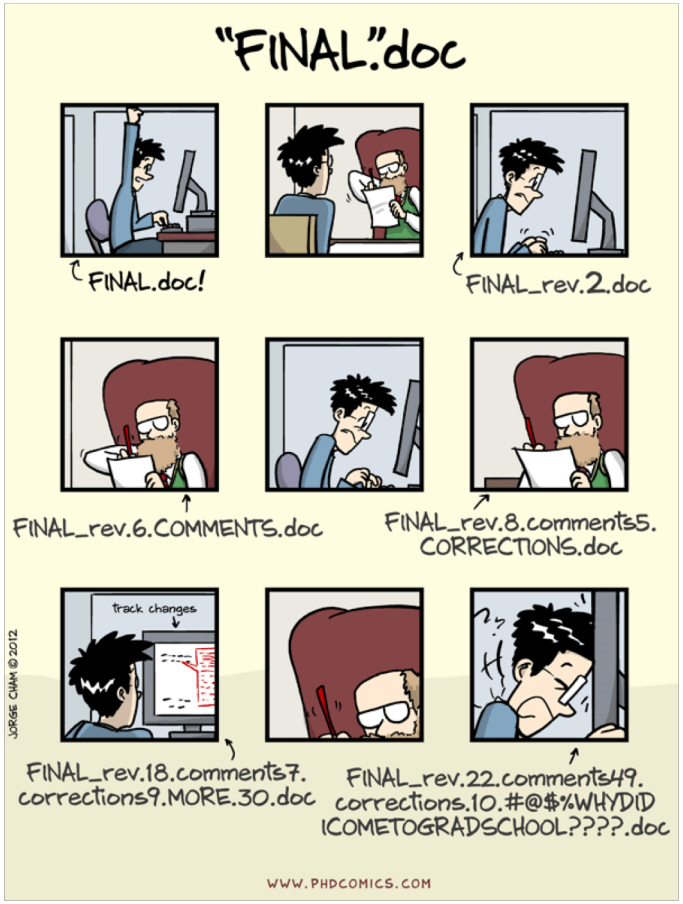
You can use Git without GitHub, but I've been using GitHub

How git stores your files



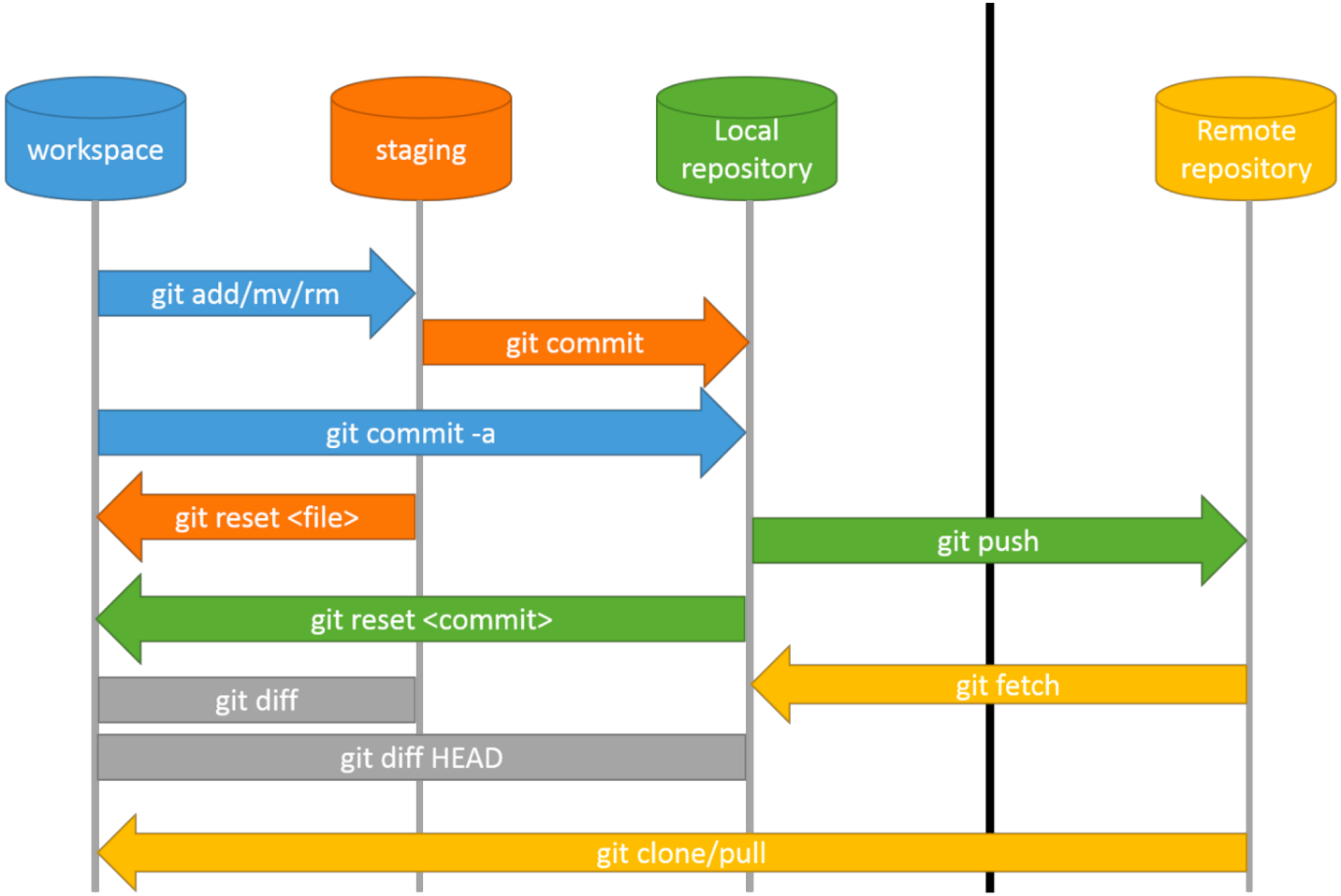
https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Why? To avoid this…..



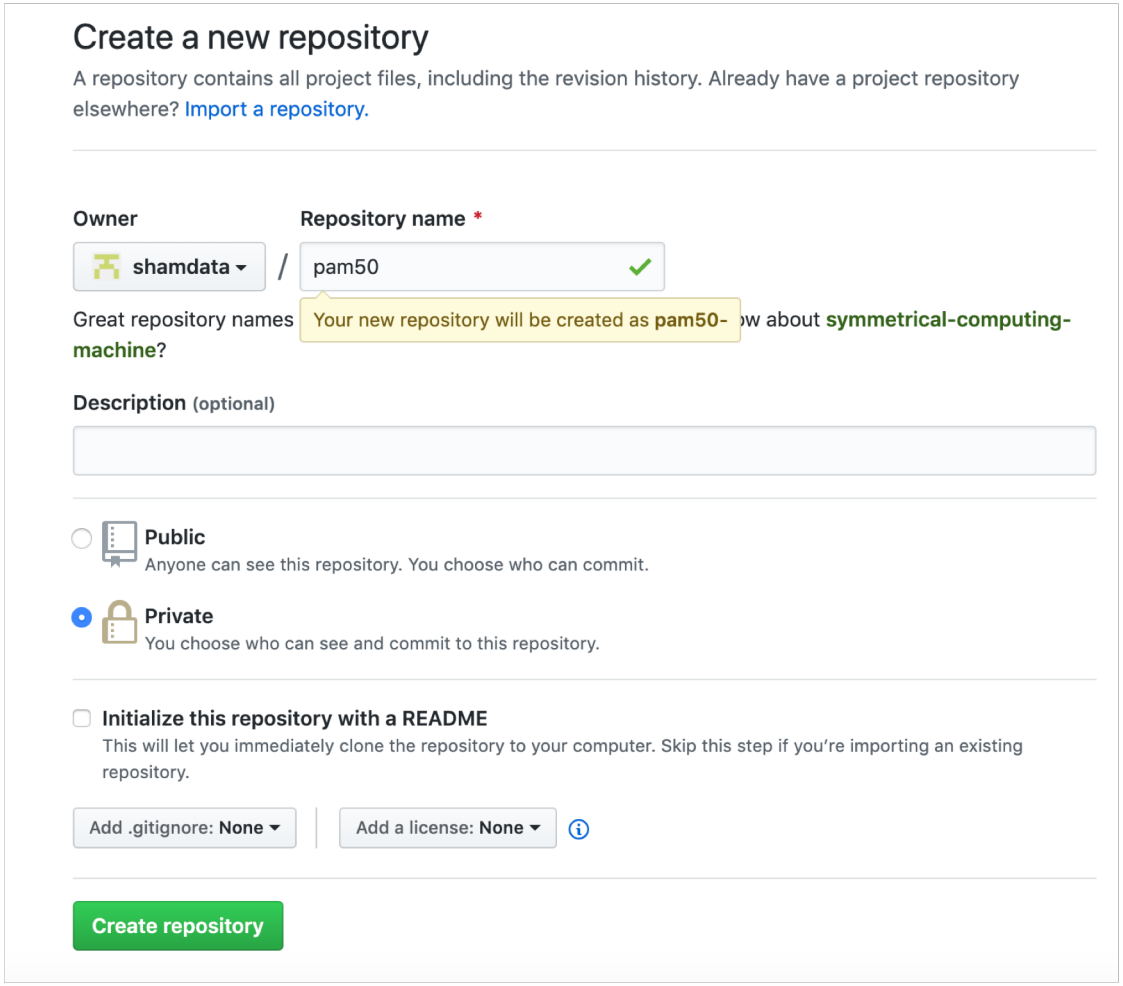
01_18_19_dataset7.R	R Source File	Apr 12, 2019 at 11:21 AM
04_08_19_TCGA_LCIS.R	R Source File	--
04_23_BCC_splitanalysis.R	R Source File	--
5geneValidation.R	R Source File	--
2018_09_19_BCC2_2_RNAseq.R	R Source File	Oct 3, 2018 at 1:56 PM
2018_09_20_PAM50_qPCR_BCC2.R	R Source File	Jan 9, 2019 at 11:47 AM
2018_09_27_deconvolution.R	R Source File	Jan 4, 2019 at 11:02 AM
2018_10_02_BCC5.R	R Source File	Oct 3, 2018 at 1:55 PM
2018_10_12_GenesDylan.R	R Source File	Oct 13, 2018 at 1:32 PM
2019_01_04_PAM50_cluster.R	R Source File	Apr 18, 2019 at 3:52 PM
2019_01_09_BCC_TCGA_cluster_PAM50.R	R Source File	Jan 10, 2019 at 11:42 AM
2019_01_10_BCC_hetgenes.R	R Source File	Feb 28, 2019 at 7:50 AM
2019_01_13_ercc.R	R Source File	Jan 15, 2019 at 4:21 PM
2019_01_16_ercc.R	R Source File	--
2019_01_16_TCGA_NMF.R	R Source File	--
2019_01_19_BCC_PAM50_resampling.R	R Source File	Apr 16, 2019 at 10:48 AM
2019_02_28_TPM_Hetgenes.R	R Source File	Apr 11, 2019 at 8:53 AM
2019_04_11_scde_overdispersion.R	R Source File	Apr 11, 2019 at 12:06 PM
al4ng-dplyr-hw.R	R Source File	--
al4ng-ggplot2-hw.R	R Source File	--
al4ng-stats-hw.R	R Source File	--
alignment_rate.R	R Source File	Nov 5, 2018 at 5:33 PM
alignment_rate.R	R Source File	Sep 26, 2018 at 1:07 PM

General overview of how it works



Start a web repository:

1. On GitHub's website:



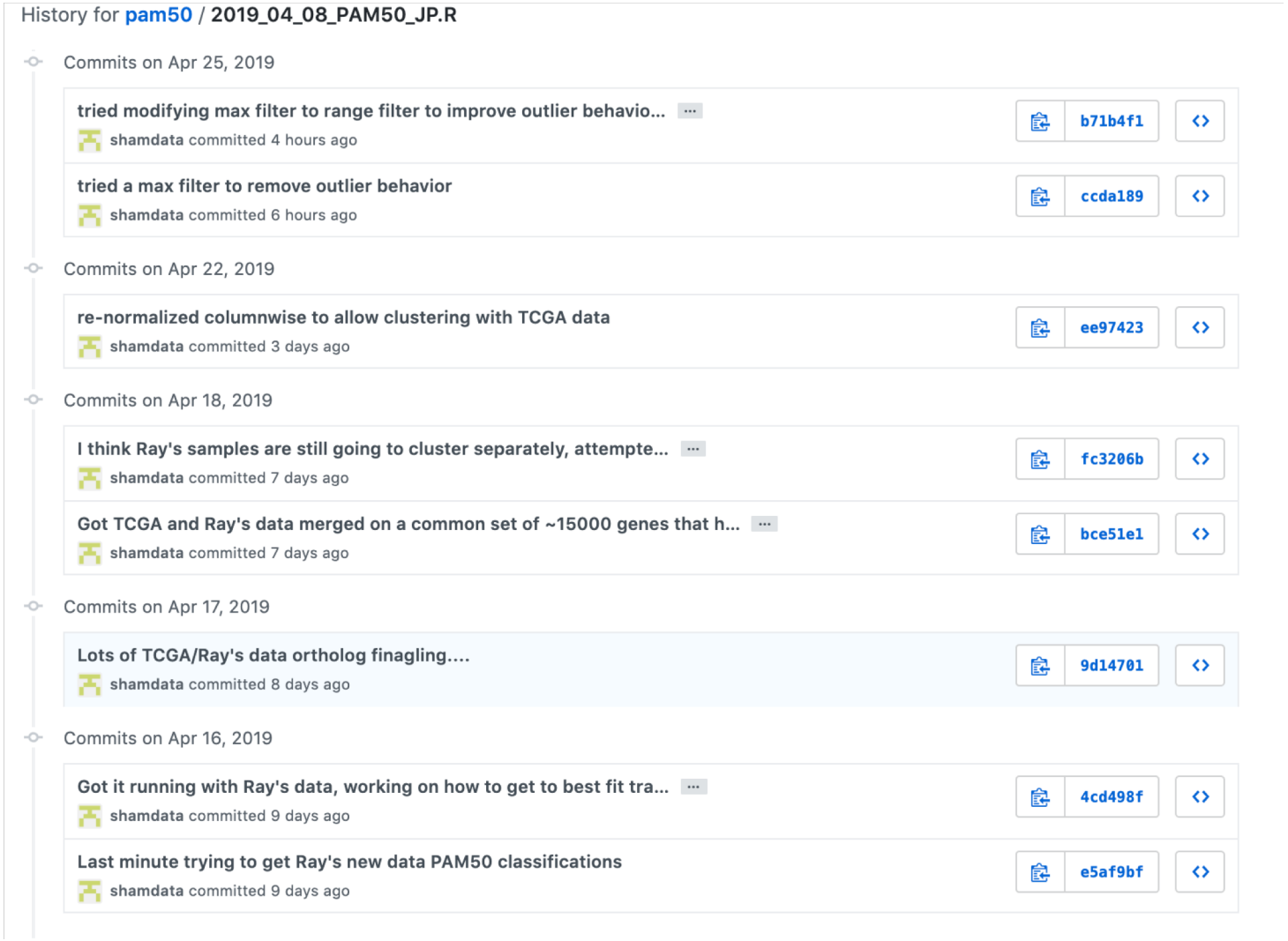
Initiate local version control workflow

NOTE: you can add EVERY file in the directory if you want or the script of interest like in this example

Install/see if you have git	git --version
Your local project folder	cd /directory
Initialize repo	git init
Set web repository as master	git remote add origin https://github.com/shamdata/pam50
Add all files/changing	git add “Code_of_interest.R”
Commit snapshot	git commit –m “Initial commit”
Push snapshot to the master	git push

Your local project folder where you store this code

History on GitHub:



Showing 1 changed file with 26 additions and 8 deletions.

Unified | Split

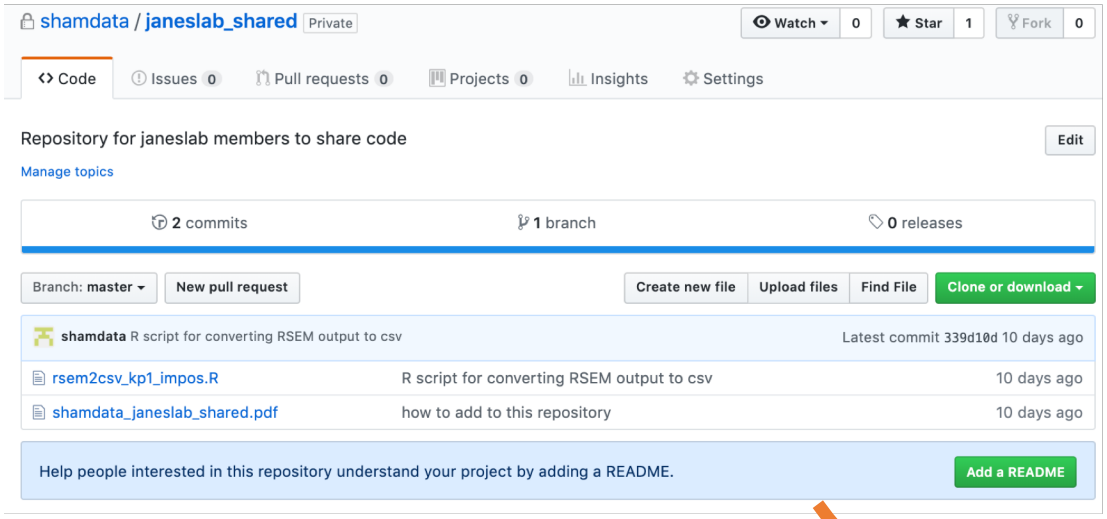
34 2019_04_08_PAM50_JP.R

```
@@ -19,38 +19,54 @@ data <- read.delim('gene_tpm_all_samples.tsv',sep='\t',stringsAsFactors = F)
 row.names(data) <- data$Gene_ID
 ensgtoid <- read.delim('ENSMUSG_ID2Name.txt',sep='\t',header=F,stringsAsFactors = F)
 row.names(ensgtoid) <- ensgtoid$V1
-data.id <- cbind(data,ensgtoid)

 #log2 transform
 data_norm <- data.id
 data_norm[,index] <- log2(data_norm[,index]+1)

 #genelist
 pam50 <- read.csv("/Users/ss4cf/Desktop/Lab/Data/TCGA/PAM50/pam50.csv",header=TRUE,stringsAsFactors = F)
 row.names(pam50) <- pam50$x

 orthologs <- read.delim("mart_export.txt",sep='\t',stringsAsFactors = F)
 pam50mouse <- (orthologs[which(orthologs$Gene.name %in% pam50$x),c(3,5)])
 unique_human <- unique(pam50mouse$Gene.name)
 unique_mouse <- unique(pam50mouse$Mouse.gene.name)
 unique_mouse <- unique_mouse[-c(23,25,26,39)]
 pam50mouse <- data.frame(cbind(unique_human,unique_mouse))
 row.names(pam50mouse) <- pam50mouse$unique_mouse

-#NOW, get PAM50 expression in ALL TCGA samples and correct with this median
 data.pam50 <- data_norm[which(data_norm$V2 %in% pam50mouse$unique_mouse),c(index,15)]
 rownames(data.pam50) <- data.pam50$V2 #NEED this, was erroring without
 row.names(data) <- data$Gene_ID
 ensgtoid <- read.delim('ENSMUSG_ID2Name.txt',sep='\t',header=F,stringsAsFactors = F)
 row.names(ensgtoid) <- ensgtoid$V1
+data.id <- merge(as.data.frame(data), as.data.frame(ensgtoid), by='row.names')
+data.id <- data.id[,-1]
+#quantile normalized
+library(edgeR)
+data_mat <- data.matrix(data.id[,index])
+normfactors <- calcNormFactors(data_mat,method="upperquartile")
+data_mat <- sweep(data_mat,2,normfactors,'/')

 #log2 transform
+data.id[,index] <- data_mat
 data_norm <- data.id
 data_norm[,index] <- log2(data_norm[,index]+1)

 #genelist
 pam50 <- read.csv("/Users/ss4cf/Desktop/Lab/Data/TCGA/PAM50/pam50.csv",header=TRUE,stringsAsFactors = F)
 row.names(pam50) <- pam50$x
+
+#got a list of human to mouse orthologs from ensbml biomart:
 orthologs <- read.delim("mart_export.txt",sep='\t',stringsAsFactors = F)
 pam50mouse <- (orthologs[which(orthologs$Gene.name %in% pam50$x),c(3,5)])
 unique_human <- unique(pam50mouse$Gene.name)
 unique_mouse <- unique(pam50mouse$Mouse.gene.name)
 unique_mouse <- unique_mouse[-c(23,25,26,39)]
 pam50mouse <- data.frame(cbind(unique_human,unique_mouse))
 row.names(pam50mouse) <- pam50mouse$unique_mouse
+#lots of finagling

+#NOW, get PAM50 expression in ALL  samples and correct with this median
 data.pam50 <- data_norm[which(data_norm$V2 %in% pam50mouse$unique_mouse),c(index,15)]
 rownames(data.pam50) <- data.pam50$V2 #NEED this, was erroring without
```

To view changes on your computer: git diff <source_branch> <target_branch>

Collaborative repositories/code

You can add to this today!!



1

Clone repo `git clone https://github.com/shamdata/janeslab_shared`

(optional) Branch code `git checkout –b YOU`

2 You edit or add code in *your folder*

Add all changes `git add .`

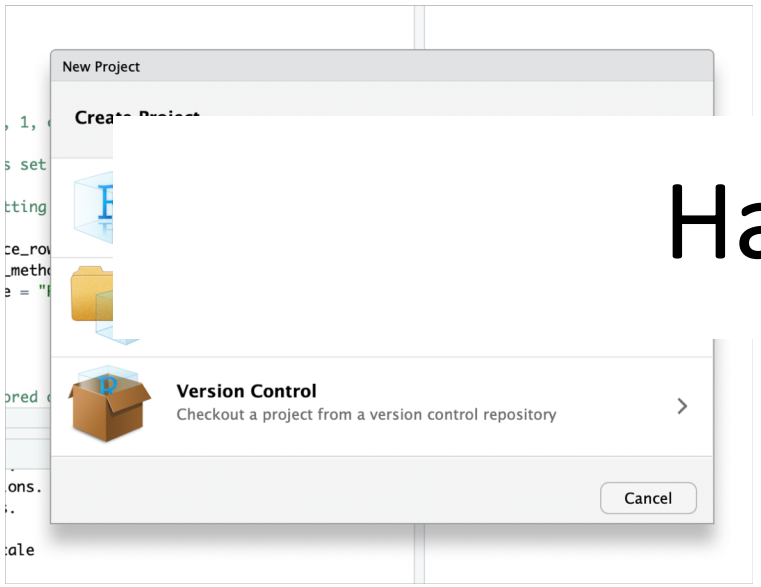
Commit snapshot `git commit –m "[]"`

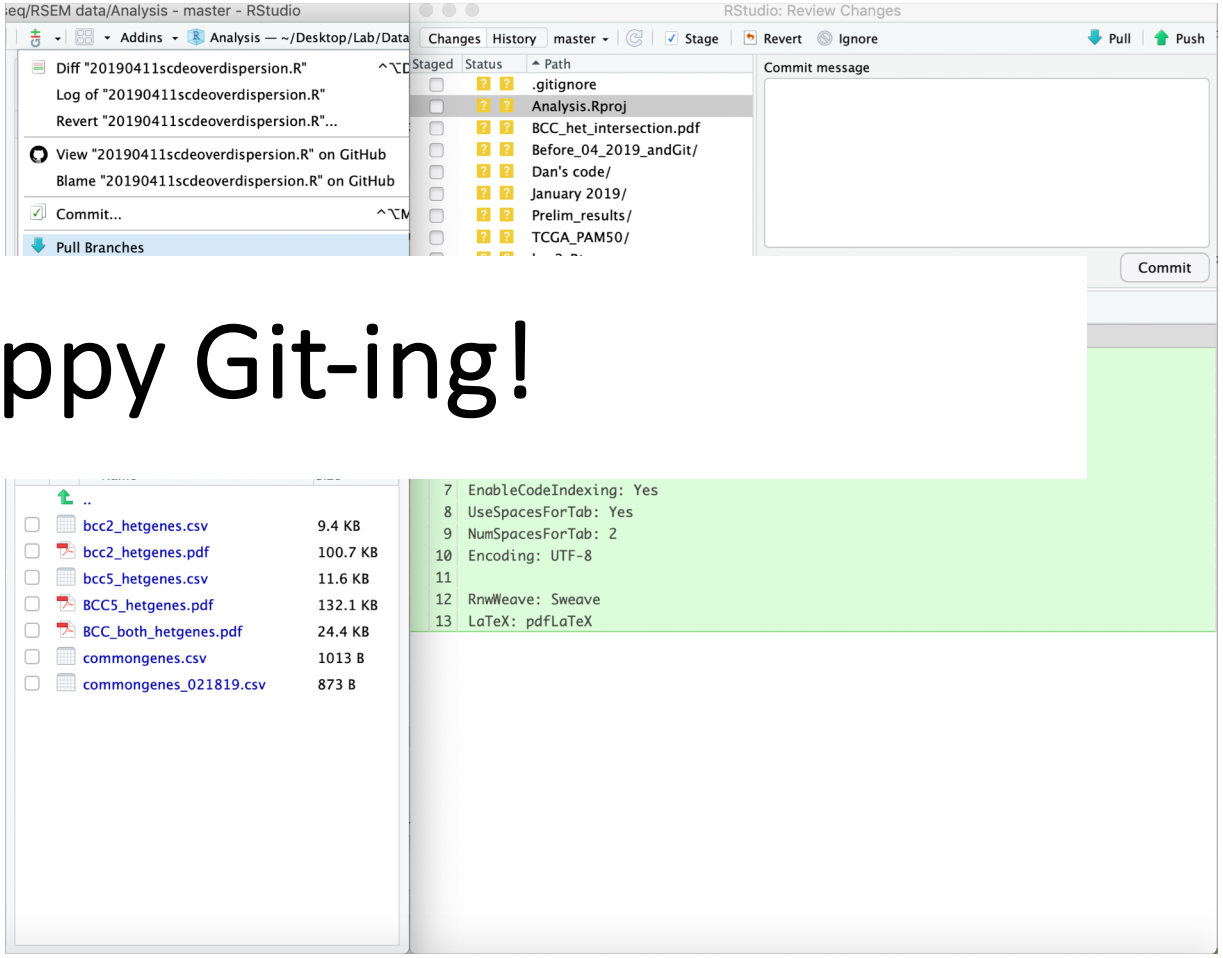
Repeat

Push changes to server `git push origin master`

(if branched) `git push origin YOU`

GIT and RStudio:





Happy Git-ing!