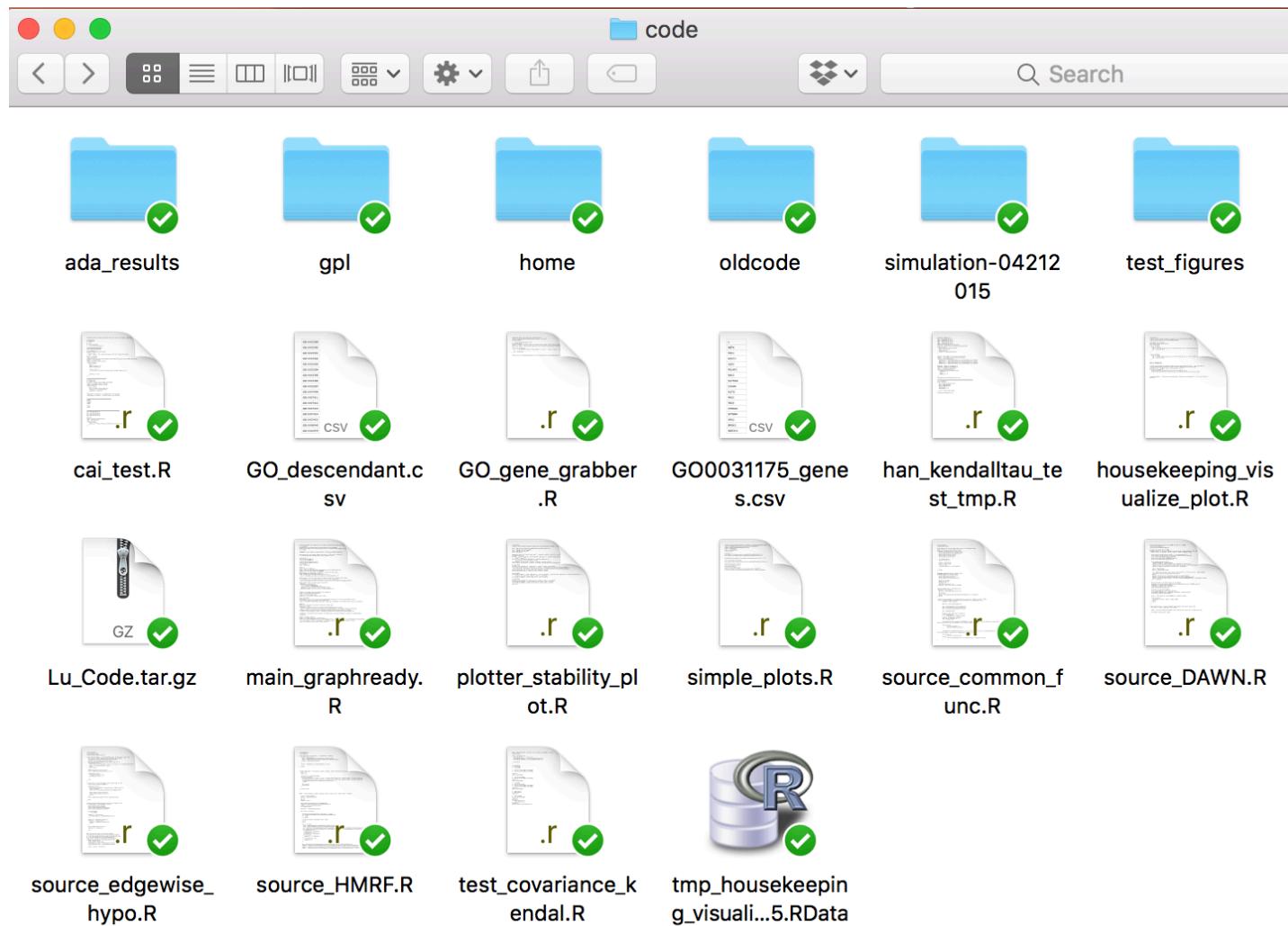


Coding Practices: R packages and other tools to solve problems

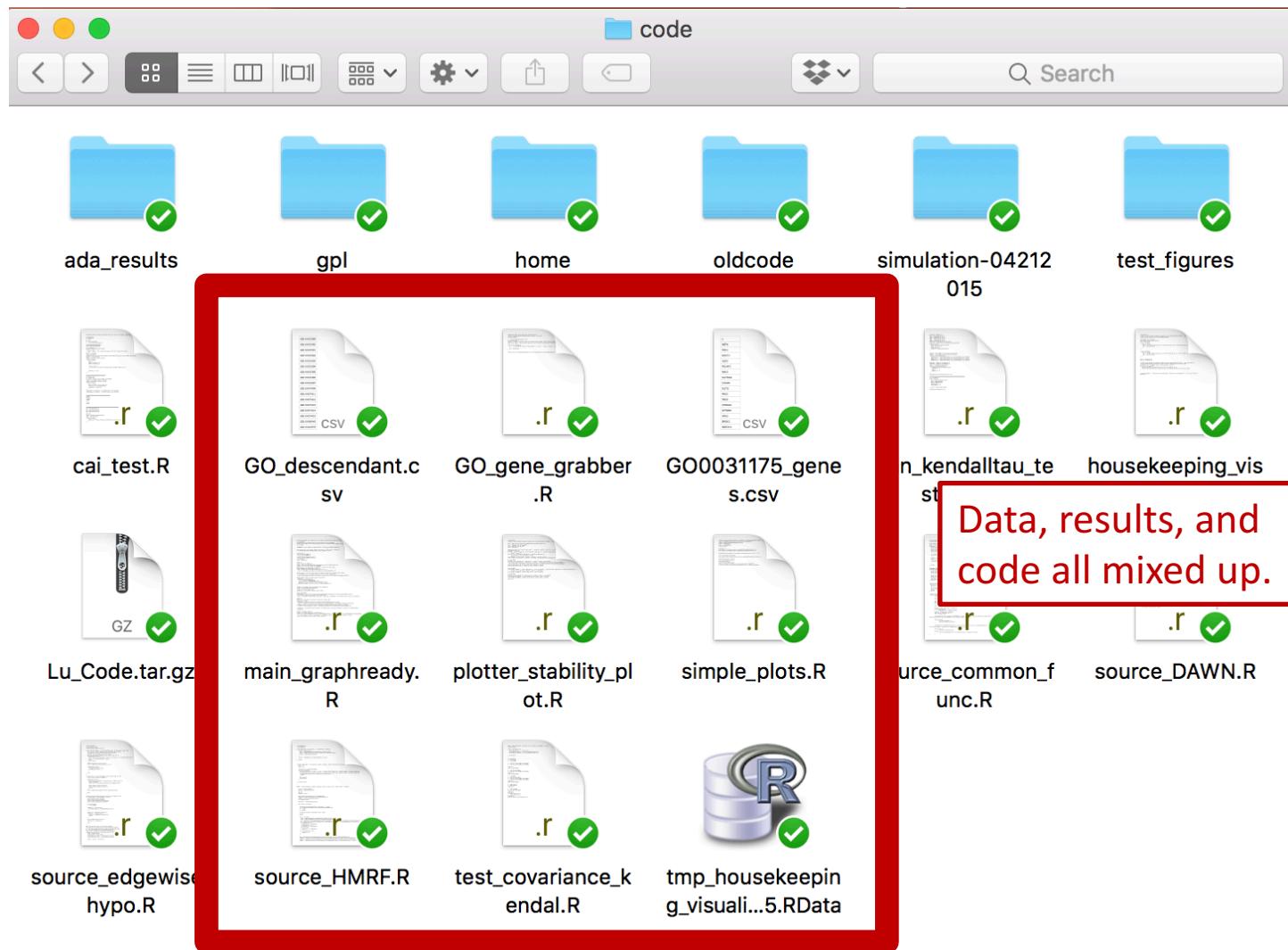
Kevin Lin

@linnylin92

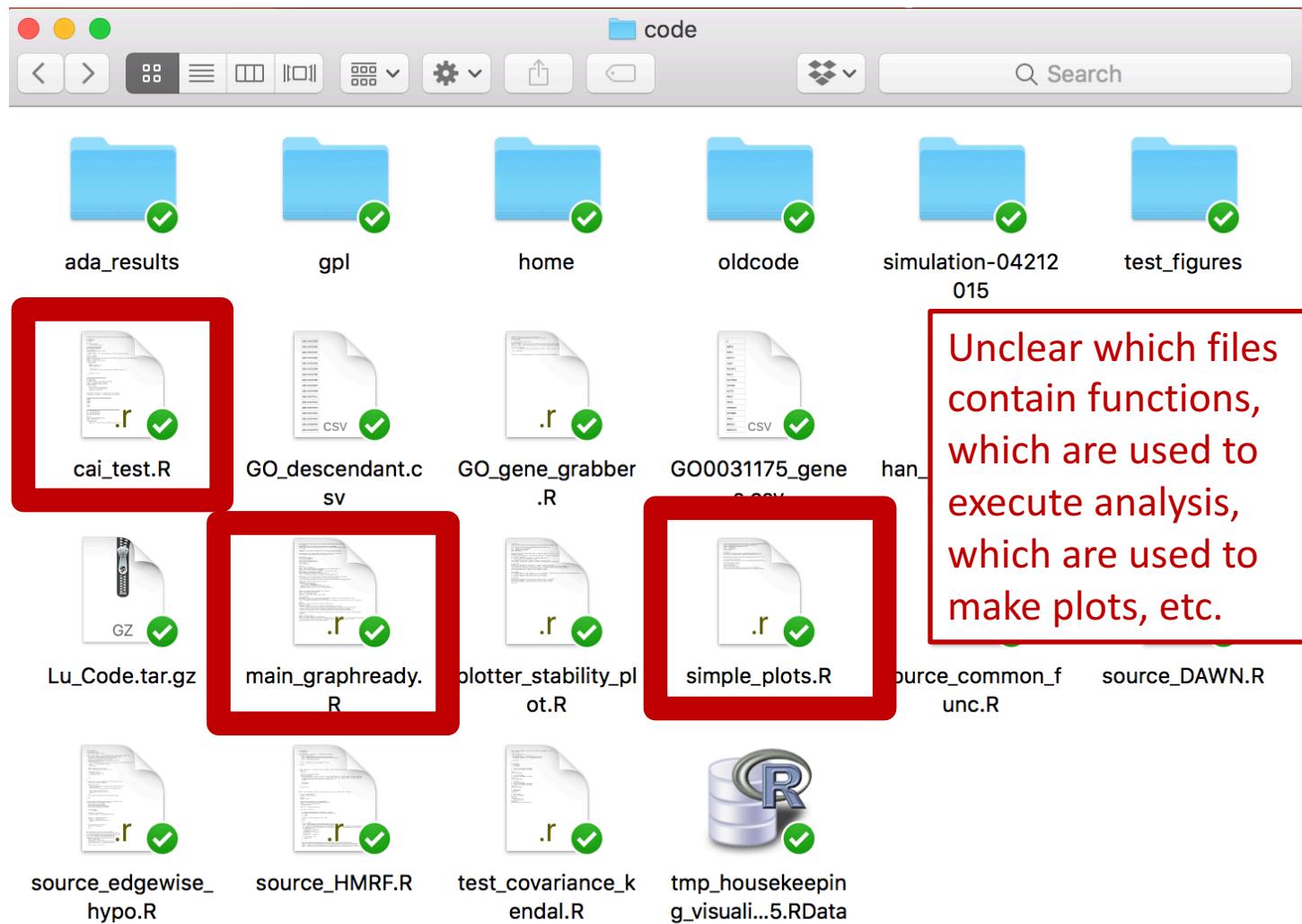
The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



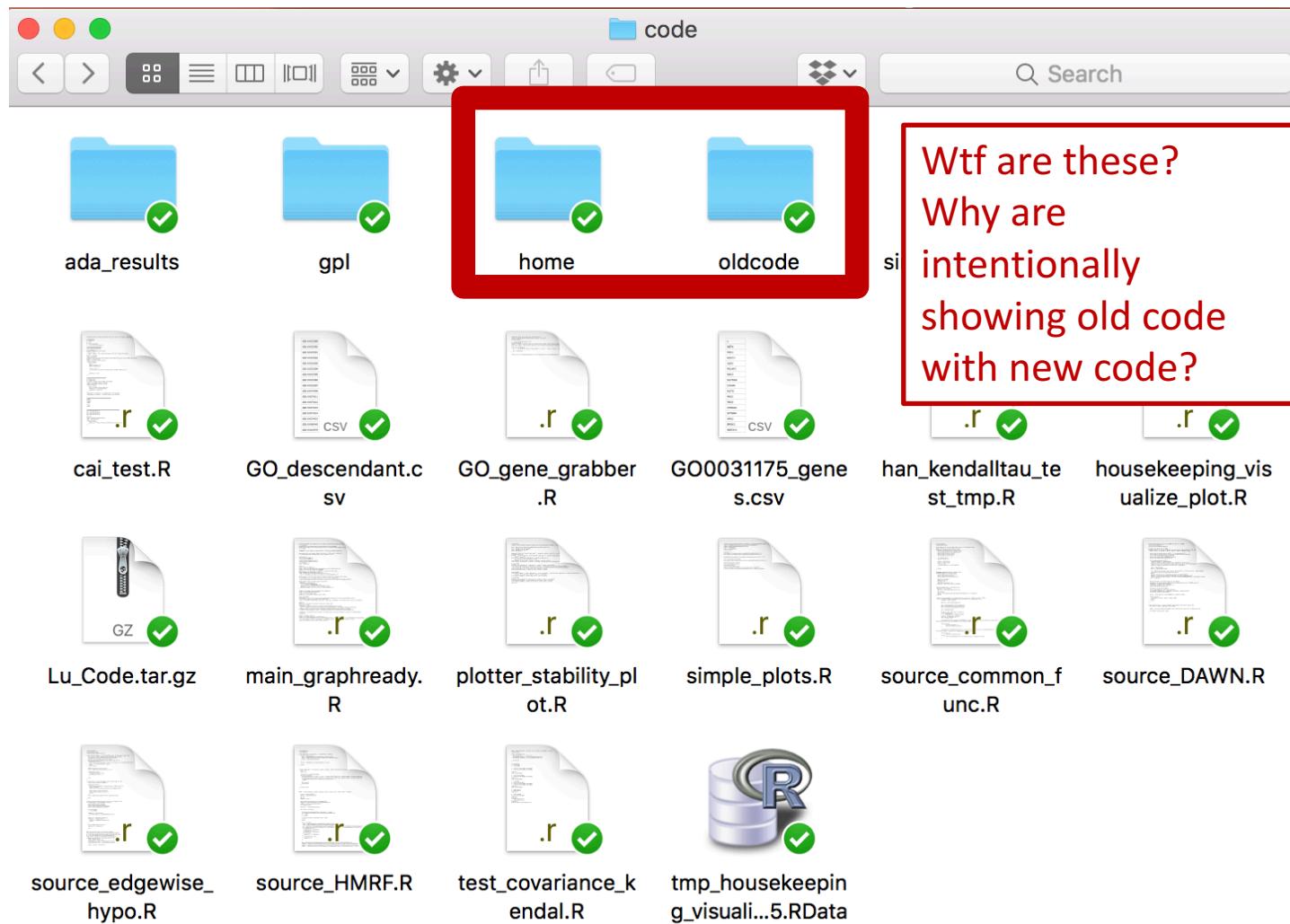
The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



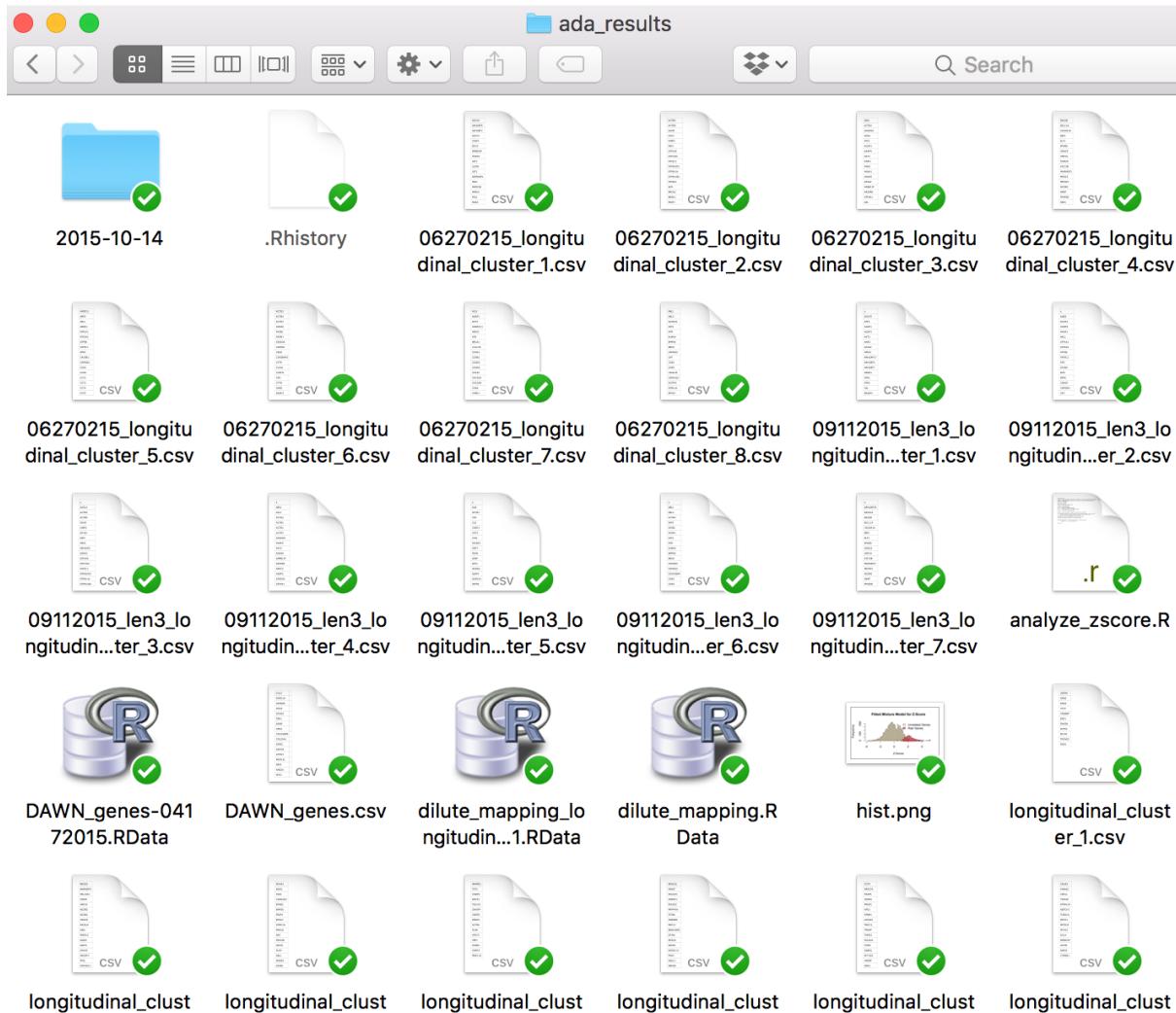
The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



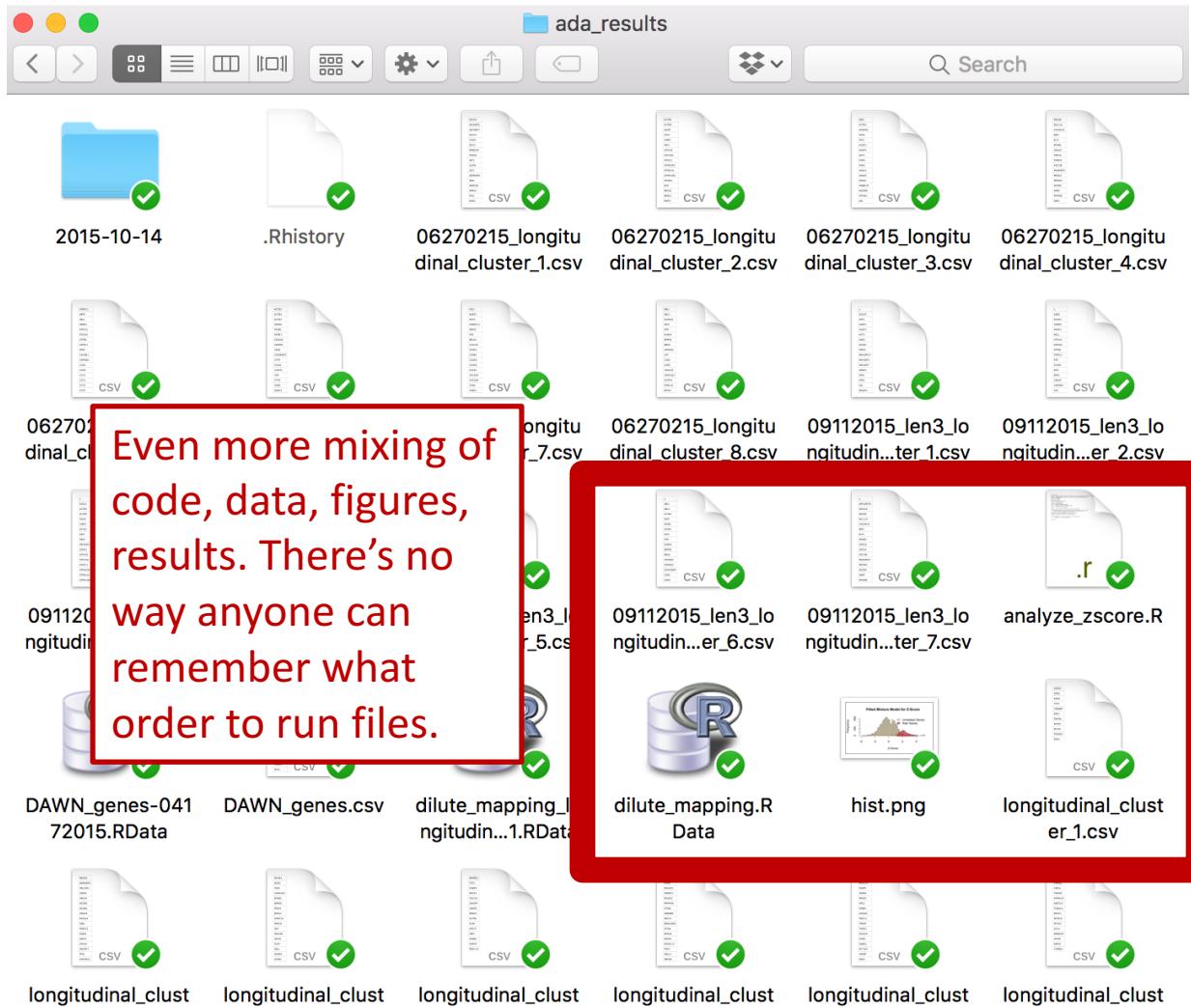
The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



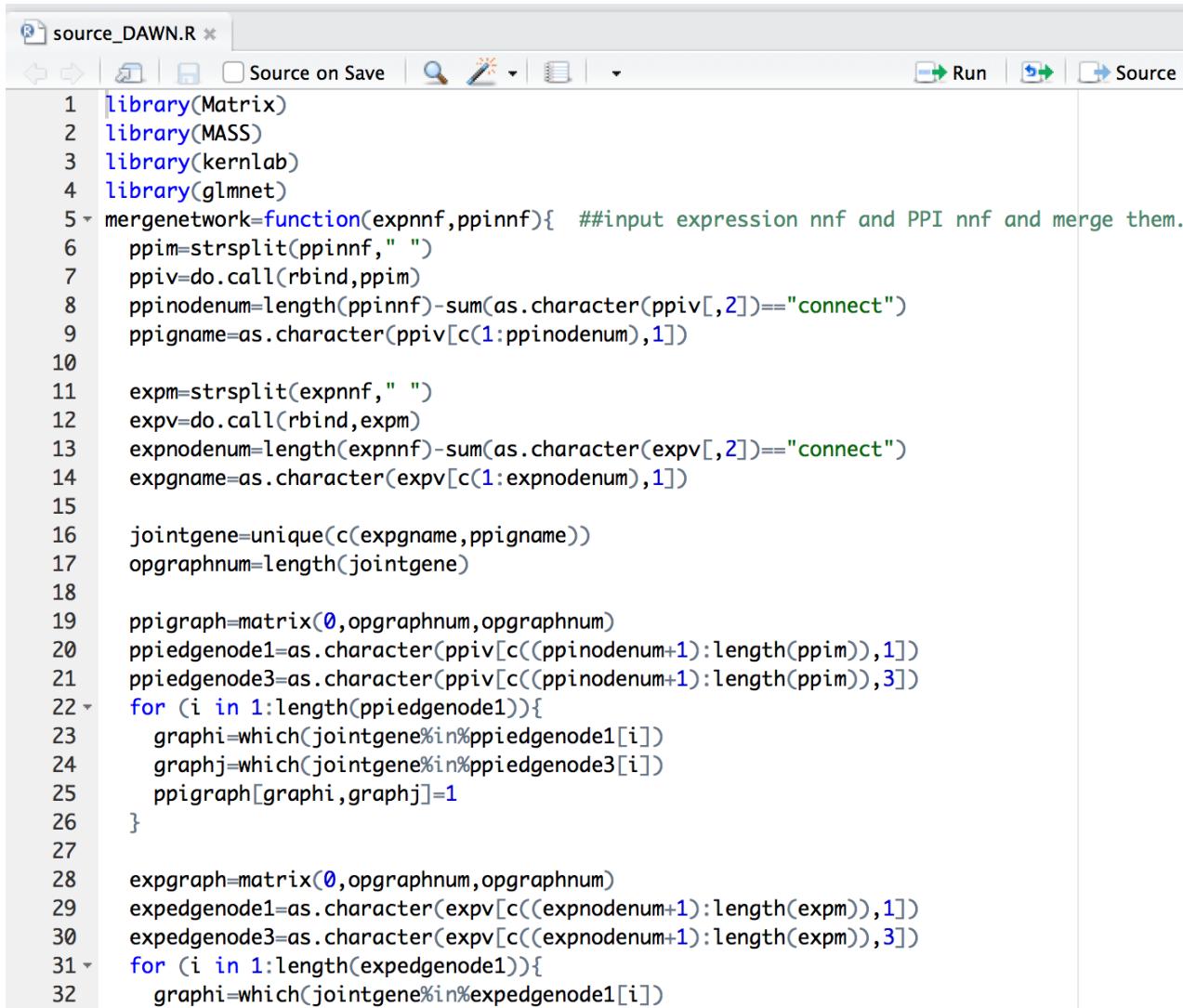
The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.

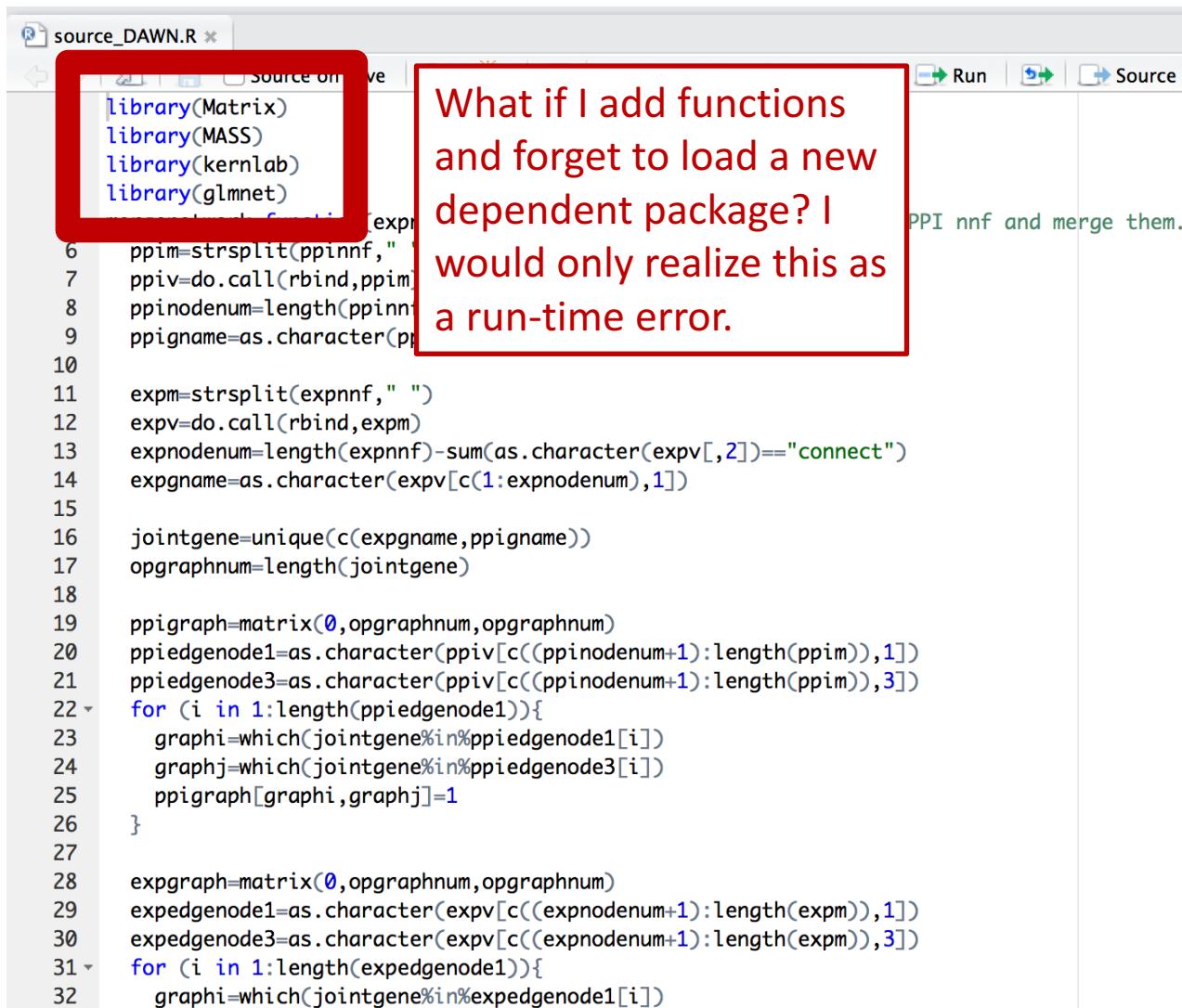


The screenshot shows an RStudio interface with the following details:

- Title Bar:** Shows the file name "source_DAWN.R".
- Toolbar:** Includes icons for back, forward, source control, and navigation.
- Status Bar:** Shows "Source on Save" and other status indicators.
- Run Bar:** Includes "Run", "Stop", and "Source" buttons.
- Code Area:** Displays the R code for the "mergenetwork" function.

```
1 library(Matrix)
2 library(MASS)
3 library(kernlab)
4 library(glmnet)
5 mergenetwork=function(expnnf,ppinnf){ ##input expression nnf and PPI nnf and merge them.
6   ppim=strsplit(ppinnf, " ")
7   ppiv=do.call(rbind,ppim)
8   ppinodenum=length(ppinnf)-sum(as.character(ppiv[,2])=="connect")
9   ppigname=as.character(ppiv[c(1:ppinodenum),1])
10
11  expm=strsplit(expnnf, " ")
12  expv=do.call(rbind,expm)
13  expnodenum=length(expnnf)-sum(as.character(expv[,2])=="connect")
14  expgname=as.character(expv[c(1:expnodenum),1])
15
16  jointgene=unique(c(expgname,ppigname))
17  opgraphnum=length(jointgene)
18
19  ppigraph=matrix(0,opgraphnum,opgraphnum)
20  ppiedgenode1=as.character(ppiv[c((ppinodenum+1):length(ppim)),1])
21  ppiedgenode3=as.character(ppiv[c((ppinodenum+1):length(ppim)),3])
22  for (i in 1:length(ppiedgenode1)){
23    graphi=which(jointgene%in%ppiedgenode1[i])
24    graphj=which(jointgene%in%ppiedgenode3[i])
25    ppigraph[graphi,graphj]=1
26  }
27
28  expgraph=matrix(0,opgraphnum,opgraphnum)
29  expedgenode1=as.character(expv[c((expnodenum+1):length(expm)),1])
30  expedgenode3=as.character(expv[c((expnodenum+1):length(expm)),3])
31  for (i in 1:length(expedgenode1)){
32    graphi=which(jointgene%in%expedgenode1[i])
```

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



A screenshot of the RStudio IDE showing an R script named "source_DAWN.R". The script contains several lines of R code, primarily dealing with package imports and data processing. A red box highlights the first four lines of code, which are package imports:

```
library(Matrix)
library(MASS)
library(kernlab)
library(glmnet)
```

Next to this red box is a callout box with the following text:

What if I add functions and forget to load a new dependent package? I would only realize this as a run-time error.

The rest of the code in the script is as follows:

```
6  ppim=strsplit(ppinnf, " ")
7  ppiv=do.call(rbind,ppim)
8  ppinodenumber=length(ppinnf)
9  ppigname=as.character(ppigname)

10 expm=strsplit(expnnf, " ")
11 expv=do.call(rbind,expm)
12 expnodenum=length(expnnf)-sum(as.character(expv[,2])=="connect")
13 expgname=as.character(expv[c(1:expnodenum),1])

14 jointgene=unique(c(expgname,ppigname))
15 opgraphnum=length(jointgene)

16 ppigraph=matrix(0,opgraphnum,opgraphnum)
17 ppiedgenode1=as.character(ppiv[c(ppinodenumber+1):length(ppim)],1])
18 ppiedgenode3=as.character(ppiv[c(ppinodenumber+1):length(ppim)],3])
19 for (i in 1:length(ppiedgenode1)){
20   graphi=which(jointgene%in%ppiedgenode1[i])
21   graphj=which(jointgene%in%ppiedgenode3[i])
22   ppigraph[graphi,graphj]=1
23 }
24
25 expgraph=matrix(0,opgraphnum,opgraphnum)
26 expedgenode1=as.character(expv[c((expnodenum+1):length(expm)),1])
27 expedgenode3=as.character(expv[c((expnodenum+1):length(expm)),3])
28 for (i in 1:length(expedgenode1)){
29   graphi=which(jointgene%in%expedgenode1[i])
```

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.

```
R source_DAWN.R ×
Source on Save | Run | Source
1 library(Matrix)
2 library(MASS)
4 library(glmnet)
5 mergenetwork=function(expnnf,ppinnf){ ##input expression nnf and PPI nnf and merge them.
6   ppim=strsplit(ppinnf, " ")
7   ppiv=do.call(rbind,ppim)
8   ppinodenumber=length(ppinnf)-sum(as.character(ppiv[,2])=="connect")
9   ppigname=as.character(ppiv[c(1:ppinodenumber),1])
10
11  expm=strsplit(expnnf, " ")
12  expv=do.call(rbind,expm)
13  expnodenum=length(expnnf)-sum(as.character(expv[,1])=="connect")
14  expgname=as.character(expv[c(1:expnodenum),1])
15
16  jointgene=unique(c(expgname,ppigname))
17  opgraphnum=length(jointgene)
18
19  ppigraph=matrix(0,opgraphnum,opgraphnum)
20  ppiedgenode1=as.character(ppiv[c((ppinodenumber+1):length(ppim)),1])
21  ppiedgenode3=as.character(ppiv[c((ppinodenumber+1):length(ppim)),3])
22  for (i in 1:length(ppiedgenode1)){
23    graphi=which(jointgene%in%ppiedgenode1[i])
24    graphj=which(jointgene%in%ppiedgenode3[i])
25    ppigraph[graphi,graphj]=1
26  }
27
28  expgraph=matrix(0,opgraphnum,opgraphnum)
29  expedgenode1=as.character(expv[c((expnodenum+1):length(expm)),1])
30  expedgenode3=as.character(expv[c((expnodenum+1):length(expm)),3])
31  for (i in 1:length(expedgenode1)){
32    graphi=which(jointgene%in%expedgenode1[i])
```

Wait, so what is the function doing again?
How are expnnf and ppinnf represented?

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.

```
source_DAWN.R
library(Matrix)
library(MASS)
library(kernlab)
library(glmnet)

ppim=strsplit(ppinnf, " ")
ppiv=do.call(rbind,ppim)
ppinodenumber=length(ppinnf)-sum(as.character(ppiv[,2])=="connect")
ppigname=as.character(ppiv[c(1:ppinodenumber),1])

expm=strsplit(expnnf, " ")
expv=do.call(rbind,expm)
expnodenumber=length(expnnf)-sum(as.character(expv[,2])=="connect")
expgname=as.character(expv[c(1:expnodenumber),1])

jointgene=unique(c(expgname,ppigname))
opgraphnum=length(jointgene)

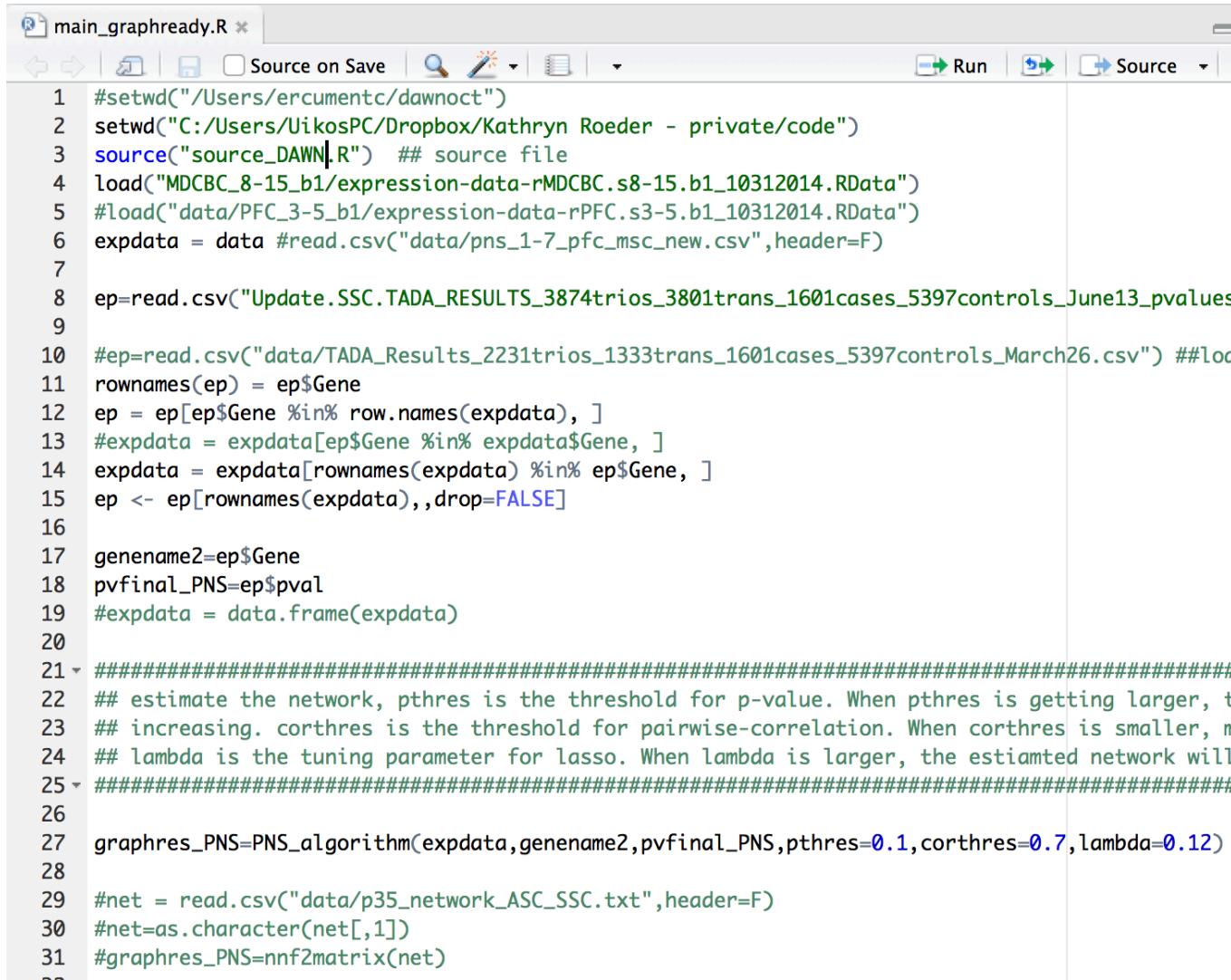
ppigraph=matrix(0,opgraphnum,opgraphnum)
ppiedgenode1=as.character(ppiv[c(ppinodenumber+1):length(ppim)],1])
ppiedgenode3=as.character(ppiv[c(ppinodenumber+1):length(ppim)],3])
for (i in 1:length(ppiedgenode1)){
  graphi=which(jointgene%in%ppiedgenode1[i])
  graphj=which(jointgene%in%ppiedgenode3[i])
  ppigraph[graphi,graphj]=1
}

expgraph=matrix(0,opgraphnum,opgraphnum)
expedgenode1=as.character(expv[c(expnodenumber+1):length(expm)],1])
expedgenode3=as.character(expv[c(expnodenumber+1):length(expm)],3])
for (i in 1:length(expedgenode1)){
  graphi=which(jointgene%in%expedgenode1[i])
  graphj=which(jointgene%in%expedgenode3[i])
  expgraph[graphi,graphj]=1
}
```

There's something aesthetically unsettling about this function that makes me not want to read it.

This function goes on for 200 lines...

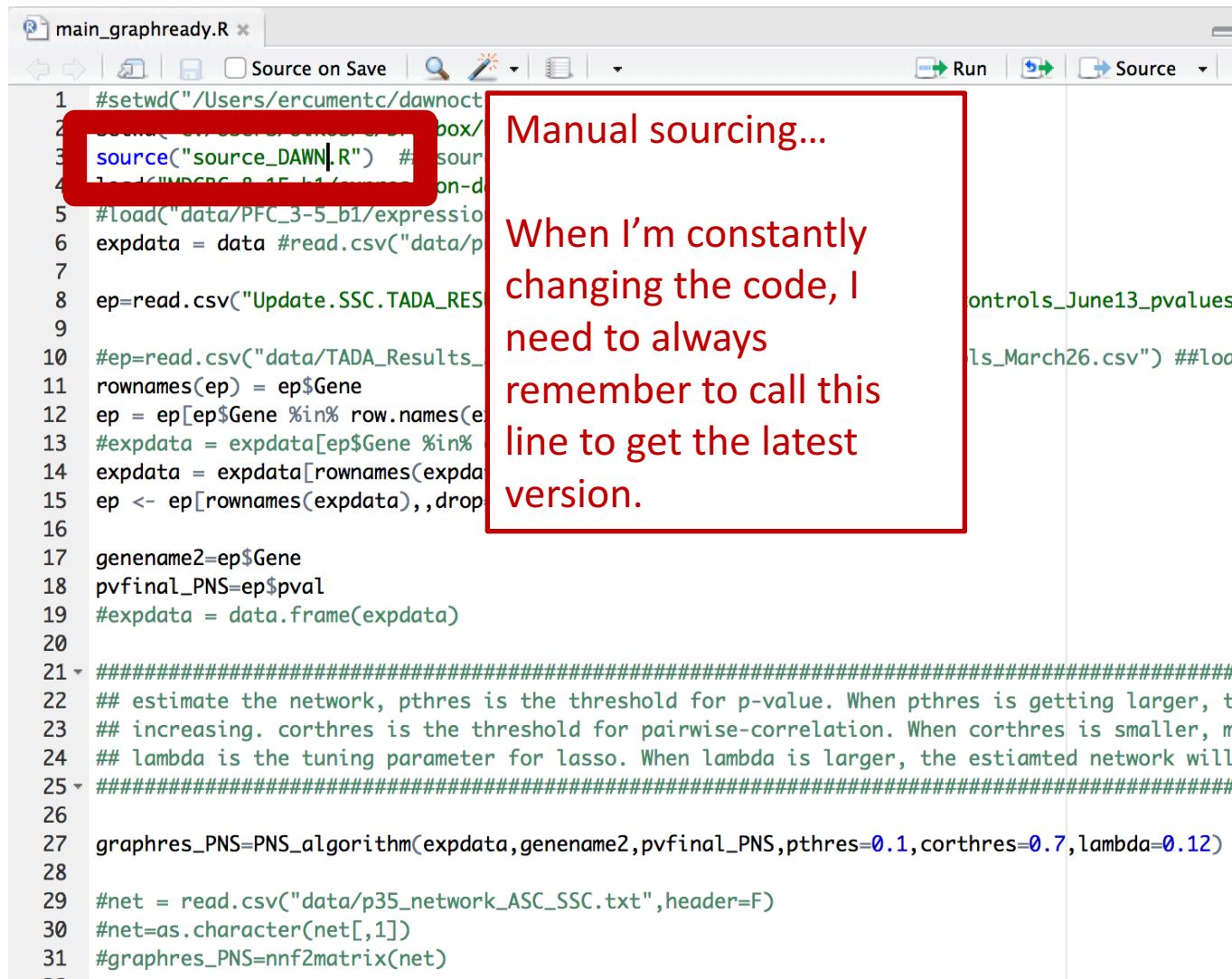
The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



The screenshot shows an RStudio interface with the file "main_graphready.R" open. The code is written in R and performs several steps to prepare data for network analysis. It starts by setting the working directory, loading data from various sources, and filtering the data to keep only genes present in both datasets. It then estimates a network using the PNS algorithm, specifying thresholds for p-values and correlation coefficients, and tuning parameters for Lasso regression. Finally, it reads a network file and converts it into a matrix format.

```
1 #setwd("/Users/ercumentc/dawnoct")
2 setwd("C:/Users/UikosPC/Dropbox/Kathryn Roeder - private/code")
3 source("source_DAWN.R") ## source file
4 load("MDCBC_8-15_b1/expression-data-rMDCBC.s8-15.b1_10312014.RData")
5 #load("data/PFC_3-5_b1/expression-data-rPFC.s3-5.b1_10312014.RData")
6 expdata = data #read.csv("data/pns_1-7_pfc_msc_new.csv",header=F)
7
8 ep=read.csv("Update.SSC.TADA_RESULTS_3874trios_3801trans_1601cases_5397controls_June13_pvalues
9
10 #ep=read.csv("data/TADA_Results_2231trios_1333trans_1601cases_5397controls_March26.csv") ##loa
11 rownames(ep) = ep$Gene
12 ep = ep[ep$Gene %in% row.names(expdata), ]
13 #expdata = expdata[ep$Gene %in% expdata$Gene, ]
14 expdata = expdata[rownames(expdata) %in% ep$Gene, ]
15 ep <- ep[rownames(expdata), ,drop=FALSE]
16
17 genename2=ep$Gene
18 pfinal_PNS=ep$pval
19 #expdata = data.frame(expdata)
20
21 #####
22 ## estimate the network, pthres is the threshold for p-value. When pthres is getting larger, t
23 ## increasing. corthres is the threshold for pairwise-correlation. When corthres is smaller, m
24 ## lambda is the tuning parameter for lasso. When lambda is larger, the estiamted network will
25 #####
26
27 graphres_PNS=PNS_algorithm(expdata,genename2,pfinal_PNS,pthres=0.1,corthres=0.7,lambda=0.12)
28
29 #net = read.csv("data/p35_network_ASC_SSC.txt",header=F)
30 #net=as.character(net[,1])
31 #graphres_PNS=nnf2matrix(net)
```

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.



The screenshot shows an RStudio interface with a code editor window titled "main_graphready.R". The code is written in R and performs various data manipulations and network analysis. A red rectangular box highlights the line of code `source("source_DAWN.R")`. To the right of this box, there are two red text boxes containing annotations:

- A top red box contains the text "Manual sourcing...".
- A larger bottom red box contains the text "When I'm constantly changing the code, I need to always remember to call this line to get the latest version."

```
#setwd("/Users/ercumentc/dawnoct")
source("source_DAWN.R") # sourc
#load("data/PFC_3-5_b1/expressio
expdata = data #read.csv("data/p
ep=read.csv("Update.SSC.TADA.RES
#ep=read.csv("data/TADA_Results_
rownames(ep) = ep$Gene
ep = ep[ep$Gene %in% row.names(ep
#expdata = expdata[ep$Gene %in%
expdata = expdata[rownames(expda
ep <- ep[rownames(expdata),,drop
genename2=ep$Gene
pvfinal_PNS=ep$pval
#expdata = data.frame(expdata)
#####
## estimate the network, pthres is the threshold for p-value. When pthres is getting larger, t
## increasing. corthres is the threshold for pairwise-correlation. When corthres is smaller, m
## lambda is the tuning parameter for lasso. When lambda is larger, the estiamted network will
#####
graphres_PNS=PNS_algorithm(expdata,genename2,pvfinal_PNS,pthres=0.1,corthres=0.7,lambda=0.12)
#net = read.csv("data/p35_network_ASC_SSC.txt",header=F)
#net=as.character(net[,1])
#graphres_PNS=nnf2matrix(net)
```

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.

```
main_graphready.R
Source on Save | Run | Source | ▾
```

```
1 #setwd("/Users/ercumentc/dawnoct")
2 setwd("C:/Users/UikosPC/Dropbox/Kathryn Roeder - private/code")
3 source("source_DAWN.R") ## source file
4 load("MDCBC_8-15_b1/expression-data-rMDCBC.s8-15.b1_10312014.RData")
5 #load("data/PFC_3-5_b1/expression-data-rPFC.s3-5.b1_10312014.RData")
6 expdata = data #read.csv("data/pns_1-7_pfc_msc_new.csv",header=F)
7
8 ep=read.csv("Update.SSC.TADA_RESULTS_3874trios_3801trans_1601cases_5397controls_June13_pvalues
9
10 #ep=read.csv("data/TADA_Results_2231trios_1333trans_1601cases_5397controls_March26.csv") ##loa
11 rownames(ep) = ep$Gene
12 ep = ep[ep$Gene %in% row.names(expdata), ]
13 #expdata = expdata[ep$Gene %in% expdata$Gene, ]
14 expdata = expdata[rownames(expdata) %in% ep$Gene, ]
15
16 Why are some lines
17 mysteriously commented out?
18
19
20
21
22
23
24
25
26
27 graphres_PNS-PNS algorithm(expdata,genename?,nvfinal_PNS,pthres=0.1,corthres=0.7,lambda=0.12)
28
29 #net = read.csv("data/p35_network_ASC-SSC.txt",header=F)
30 #net=as.character(net[,1])
31 #graphres_PNS=nnf2matrix(net)
```

Why are some lines mysteriously commented out?

Was I testing something at the time, or did I mean to delete them?

for p-value. When pthres is getting larger, t
wise-correlation. When corthres is smaller, m
n lambda is larger, the estiamted network will
#####

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.

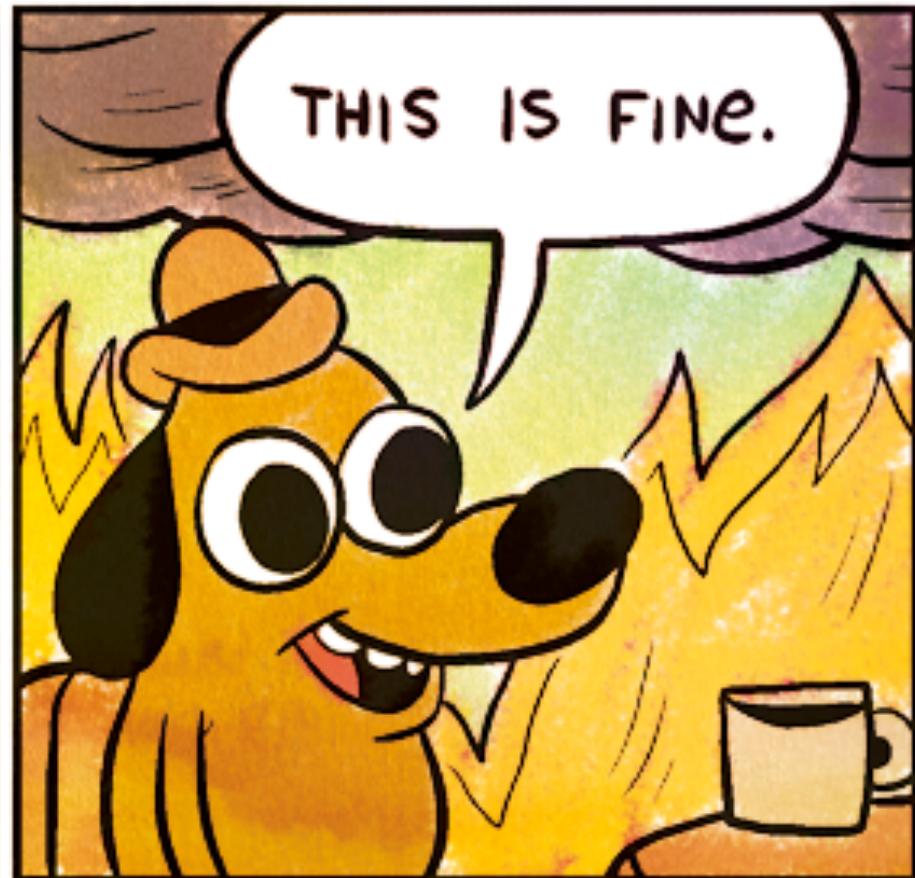
```
main_graphready.R *
```

```
1 #setwd("/Users/ercumentc/dawnoct")
2 setwd("C:/Users/UikosPC/Dropbox/Kathryn Roeder - private/code")
3
4 load("MDCBC_8-15_b1/expression-data-rMDCBC.s8-15.b1_10312014.RData")
5 #> [1] "C:/Users/UikosPC/Dropbox/Kathryn Roeder - private/code/MDCBC_8-15_b1_10312014.RData"
6 expdata = data #read.csv("data/pns_1-7_pfc_msc_new.csv",header=F)
7
8 ep=read.csv("Update.SSC.TADA_RESULTS.csv")
9
10 #ep=read.csv("data/TADA_Results.csv")
11 rownames(ep) = ep$Gene
12 ep = ep[ep$Gene %in% row.names(expdata), ]
13 #expdata = expdata[ep$Gene %in% expdata$Gene, ]
14 expdata = expdata[rownames(expdata) %in% ep$Gene, ]
15 ep <- ep[rownames(expdata), ,drop=FALSE]
16
17 genename2=ep$Gene
18 pfinal_PNS=ep$pval
19 #expdata = data.frame(expdata)
20
21 #####
22 ## estimate the network, pthres is the threshold for p-value. When pthres is getting larger, t
23 ## increasing. corthres is the threshold for pairwise-correlation. When corthres is smaller, m
24 ## lambda is the tuning parameter for lasso. When lambda is larger, the estiamted network will
25 #####
26
27 graphres_PNS=PNS_algorithm(expdata,genename2,pfinal_PNS,pthres=0.1,corthres=0.7,lambda=0.12)
28
29 #net = read.csv("data/p35_network_ASC_SSC.txt",header=F)
30 #net=as.character(net[,1])
31 #graphres_PNS=nnf2matrix(net)
```

Annoying necessity to load data and csv's from other places.

The best way to think about “good” coding practice is to understand what obstacles you’re trying to solve/avoid.

```
main_graphready.R
1 #setwd("/Users/ercumentc/dawnoct")
2 setwd("C:/Users/UikosPC/Dropbox/Kathryn Roeder - private/code")
3 source("source_DAWN.R") ## source file
4 load("MDCBC_8-15_b1/expression-data-rMDCBC.s8-15.b1_10312014.RData")
5 #load("data/PFC_3-5_b1/expression-data-rPFC.s3-5.b1_10312014.RData")
6 expdata = data #read.csv("data/pns_1-7_pfc_msc_new.csv",header=F)
7
8 ep=read.csv("Update.SSC.TADA_RESULTS_3874trios_3801trans_1601cases_5397controls_June13_pvalues")
9
10 #ep=read.csv("data/TADA_Results_2231trios_1333trios")
11 rownames(ep) = ep$Gene
12 ep = ep[ep$Gene %in% row.names(expdata), ]
13 #expdata = expdata[ep$Gene %in% expdata$Gene, ]
14 expdata = expdata[rownames(expdata) %in% ep$Gene]
15 ep <- ep[rownames(expdata),,drop=FALSE]
16
17 genename2=ep$Gene
18 pvfinal_PNS=ep$pval
19 #expdata = data.frame(expdata)
20
21 #####
22 ## estimate the network, pthres is the threshold
23 ## increasing. corthres is the threshold for pair
24 ## lambda is the tuning parameter for lasso. When lambda is larger, the estimated network will
25
26
27 graphres_PNS=PNS_algorithm(expdata,genename2,pvfinal_PNS,pthres=0.1,corthres=0.7,lambda=0.12)
28
29 #net = read.csv("data/pns_NETWORK_ASC_SSC.csv",header=1)
30 #net=as.character(net[,1])
31 #graphres_PNS=nnf2matrix(net)
```



You should be worried if your code resembled what was previously shown in any way.

- No clear organizing, hard to reproduce results.

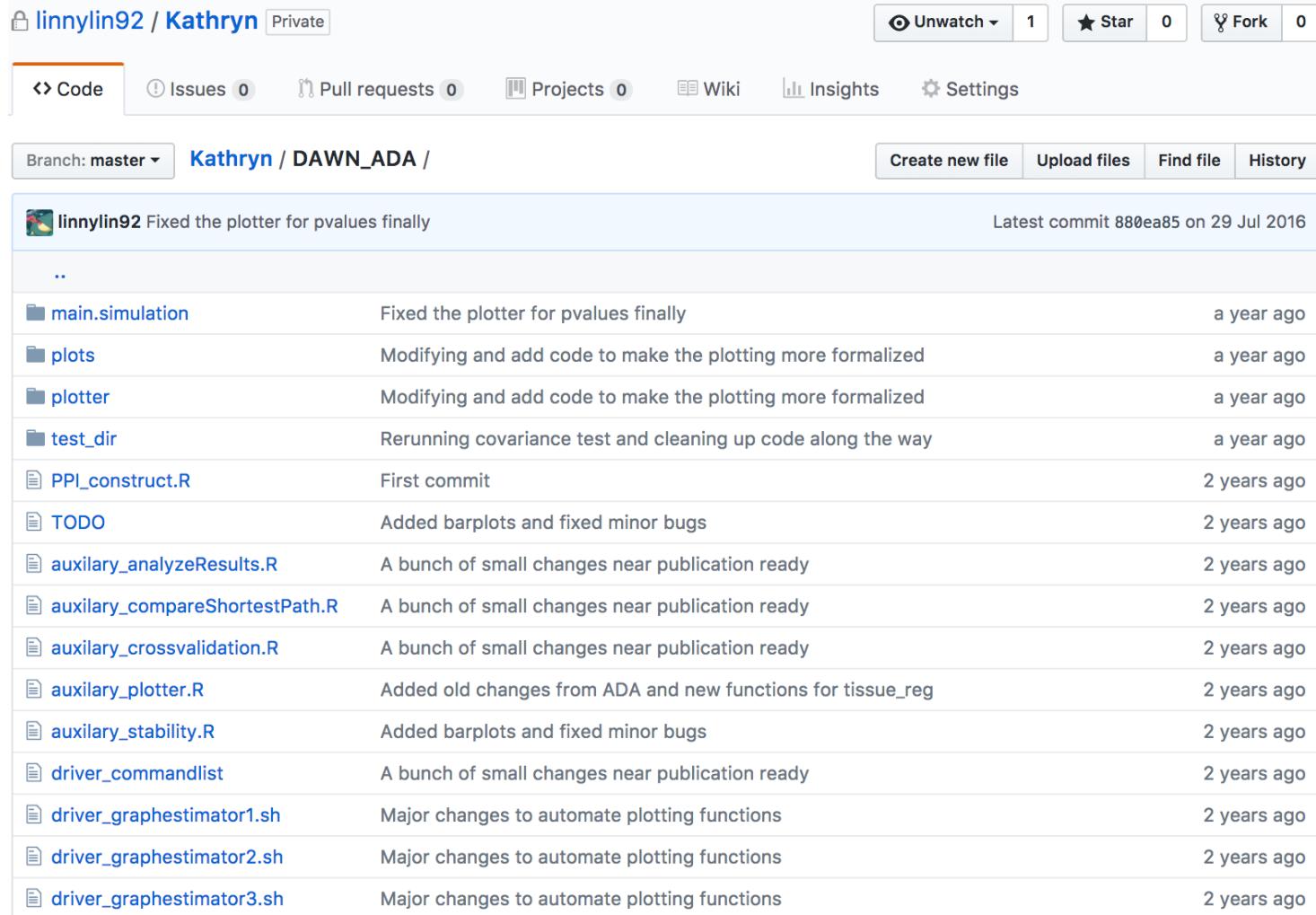
You should be worried if your code resembled what was previously shown in any way.

- No clear organizing, hard to reproduce results.
- No clear documentation, hard to know what functions do what, take as input, return as output.
- Manual sourcing and loading of files, hard to edit code on the fly.

You should be worried if your code resembled what was previously shown in any way.

- No clear organizing, hard to reproduce results.
- No clear documentation, hard to know what functions do what, take as input, return as output.
- Manual sourcing and loading of files, hard to edit code on the fly.
- No repository, hard to know what was being tested or meant to be deleted.
- No testing, hard to know if anything broke if I "fixed a bug."

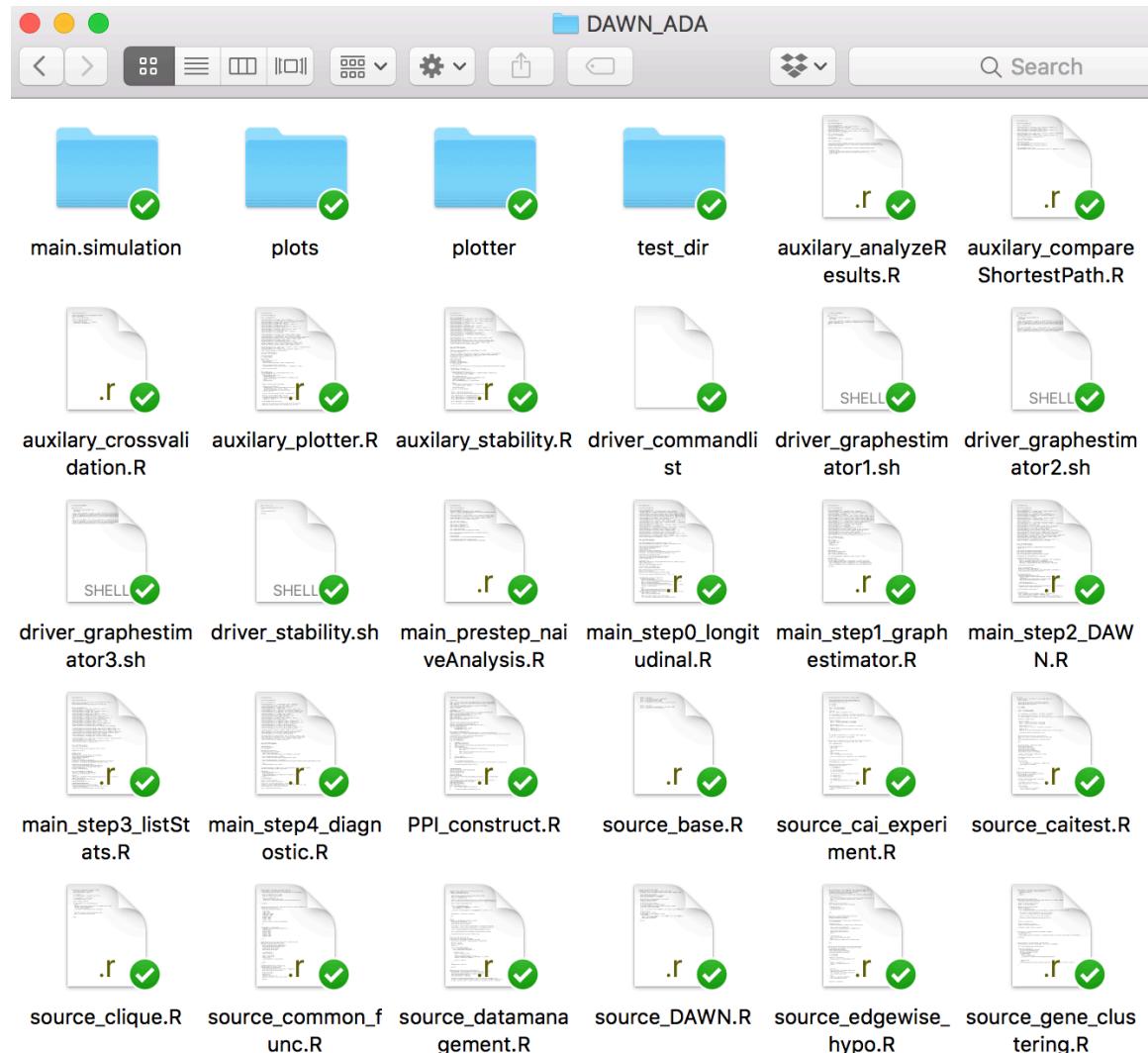
I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.



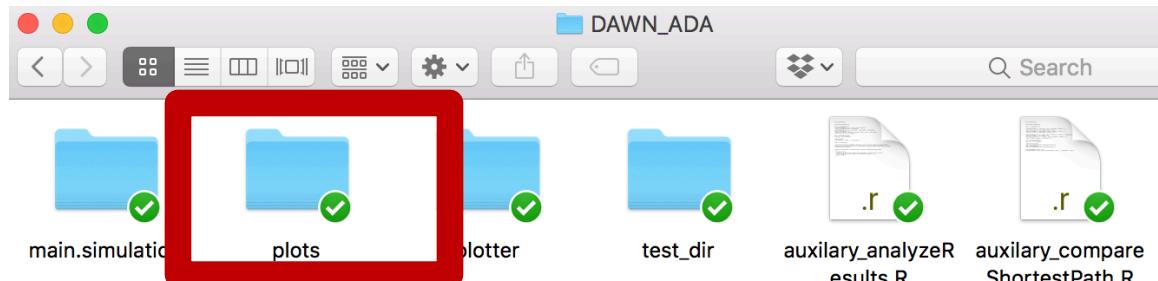
A screenshot of a GitHub repository page for 'linnylin92 / Kathryn'. The repository is private. The commit history shows 14 commits from 'linnylin92' over the past year. The commits are as follows:

Commit	Message	Date
Fixed the plotter for pvalues finally	Fixed the plotter for pvalues finally	29 Jul 2016
..		
main.simulation	Fixed the plotter for pvalues finally	a year ago
plots	Modifying and add code to make the plotting more formalized	a year ago
plotter	Modifying and add code to make the plotting more formalized	a year ago
test_dir	Rerunning covariance test and cleaning up code along the way	a year ago
PPI_construct.R	First commit	2 years ago
TODO	Added barplots and fixed minor bugs	2 years ago
auxiliary_analyzeResults.R	A bunch of small changes near publication ready	2 years ago
auxiliary_compareShortestPath.R	A bunch of small changes near publication ready	2 years ago
auxiliary_crossvalidation.R	A bunch of small changes near publication ready	2 years ago
auxiliary_plotter.R	Added old changes from ADA and new functions for tissue_reg	2 years ago
auxiliary_stability.R	Added barplots and fixed minor bugs	2 years ago
driver_commandlist	A bunch of small changes near publication ready	2 years ago
driver_graphestimator1.sh	Major changes to automate plotting functions	2 years ago
driver_graphestimator2.sh	Major changes to automate plotting functions	2 years ago
driver_graphestimator3.sh	Major changes to automate plotting functions	2 years ago

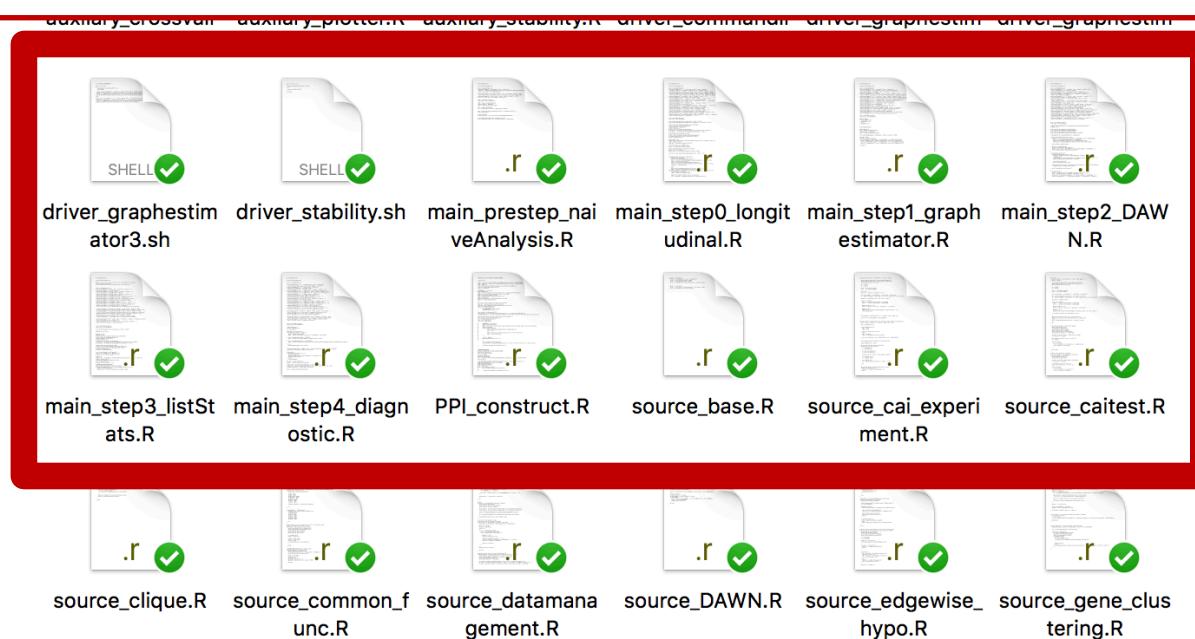
I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.



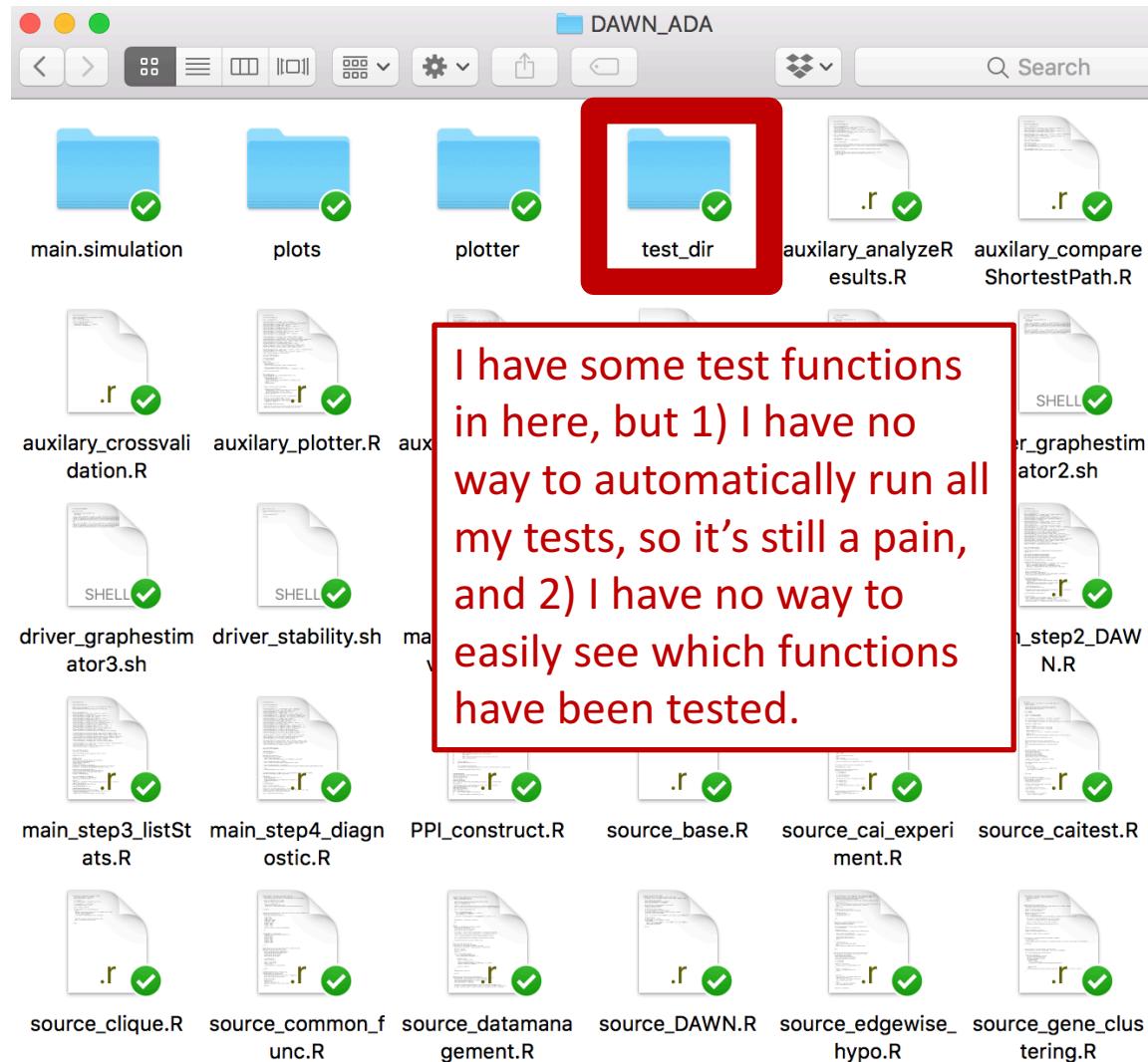
I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.



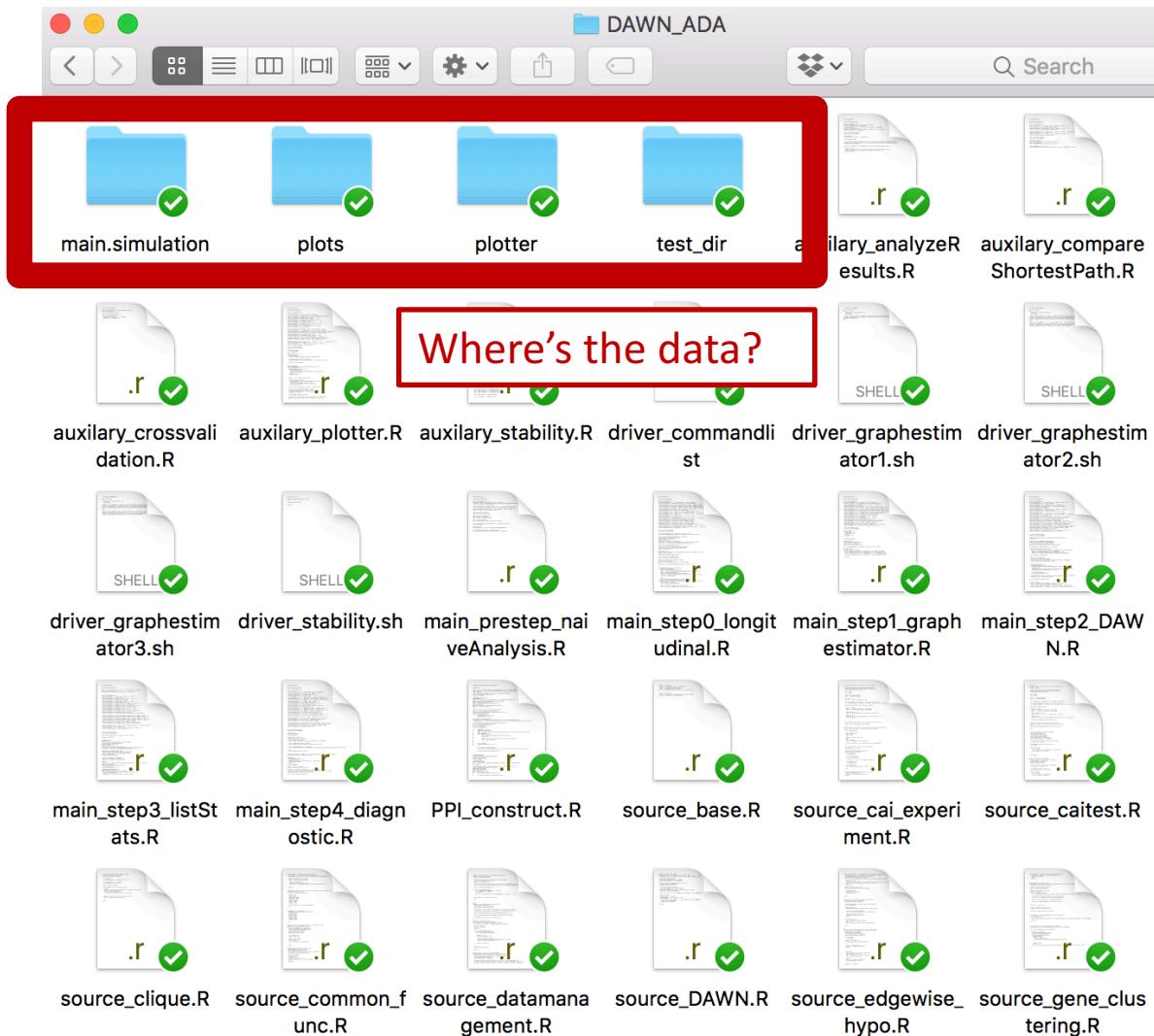
I have “main” and “source” files, separate from plots, so we’re doing slightly better...



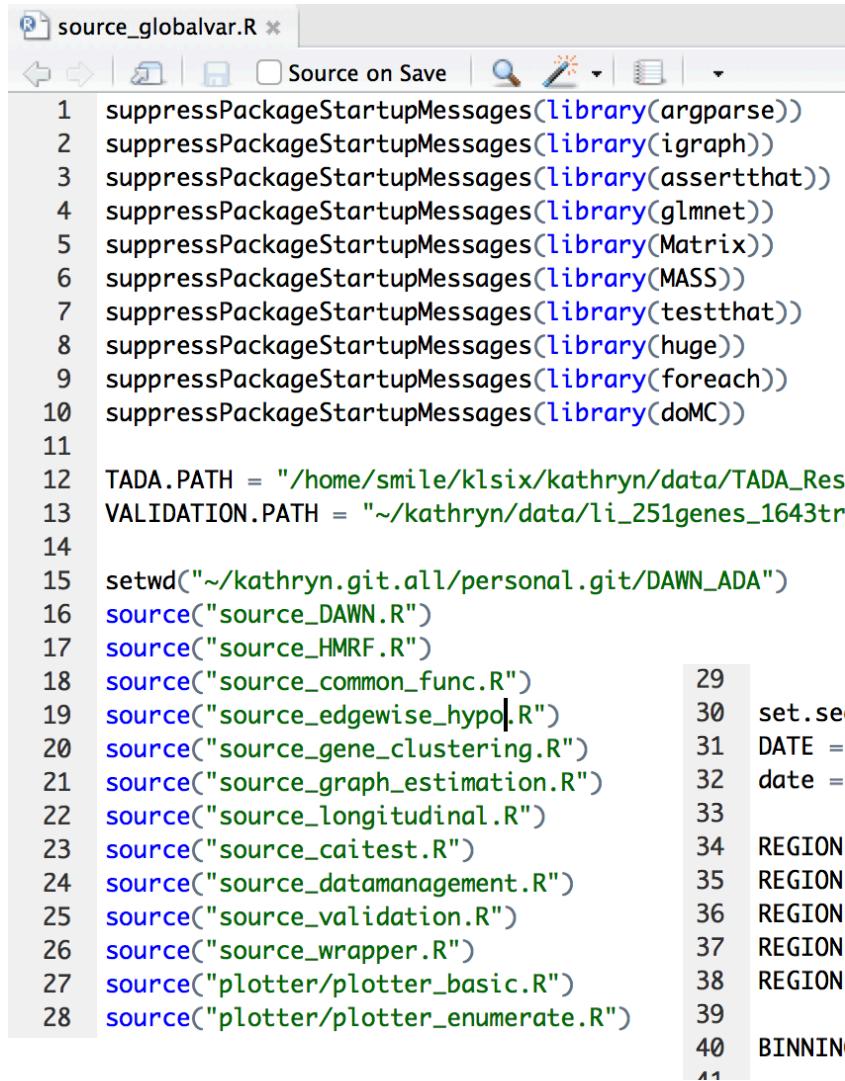
I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.



I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.



I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.



The screenshot shows an RStudio interface with a code editor containing the file "source_globalvar.R". The code is a series of R commands used to suppress startup messages from various packages and set up project paths and working directories. It includes calls to `suppressPackageStartupMessages` for packages like `argparse`, `igraph`, `assertthat`, `glmnet`, `Matrix`, `MASS`, `testthat`, `huge`, `foreach`, and `doMC`. It also sets `TADA.PATH` and `VALIDATION.PATH` to specific local and validation data directories. The code then sources several files from a local Git repository at `~/kathryn.git.all/personal.git/DAWN_ADA`. These sourced files include common functions, edge-wise hypothesis testing, gene clustering, graph estimation, longitudinal analysis, caitest, data management, validation, and a wrapper function. The code concludes by setting a seed, defining a date object, creating a list of subregions, and defining a binning matrix.

```
source_globalvar.R *
 1 suppressPackageStartupMessages(library(argparse))
 2 suppressPackageStartupMessages(library(igraph))
 3 suppressPackageStartupMessages(library(assertthat))
 4 suppressPackageStartupMessages(library(glmnet))
 5 suppressPackageStartupMessages(library(Matrix))
 6 suppressPackageStartupMessages(library(MASS))
 7 suppressPackageStartupMessages(library(testthat))
 8 suppressPackageStartupMessages(library(huge))
 9 suppressPackageStartupMessages(library(foreach))
10 suppressPackageStartupMessages(library(doMC))
11
12 TADA.PATH = "/home/smile/klsix/kathryn/data/TADA_Resu
13 VALIDATION.PATH = "~/kathryn/data/li_251genes_1643tri
14
15 setwd("~/kathryn.git.all/personal.git/DAWN_ADA")
16 source("source_DAWN.R")
17 source("source_HMRF.R")
18 source("source_common_func.R")
19 source("source_edgewise_hypo.R")
20 source("source_gene_clustering.R")
21 source("source_graph_estimation.R")
22 source("source_longitudinal.R")
23 source("source_caitest.R")
24 source("source_datamanagement.R")
25 source("source_validation.R")
26 source("source_wrapper.R")
27 source("plotter/plotter_basic.R")
28 source("plotter/plotter_enumerate.R")
29
30 set.seed(10)
31 DATE = Sys.Date()
32 date = DATE
33
34 REGION.SUBREGION = list()
35 REGION.SUBREGION[["PFC"]] = c("MFC", "DFC", "OFC", "MSC", "VFC", "M1C", "S1C")
36 REGION.SUBREGION[["VIIAS"]] = c("V1C", "ITC", "IPC", "A1C", "STC")
37 REGION.SUBREGION[["SHA"]] = c("STR", "HIP", "AMY")
38 REGION.SUBREGION[["MDCBC"]] = c("MD", "CBC")
39
40 BINNING = matrix(c(1,2,3,5,6,8,9,15), ncol = 2, byrow = TRUE)
```

I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.

The screenshot shows an RStudio interface with a script file named "source_globalvar.R". The code in the script is as follows:

```
1 suppressPackageStartupMessages(library(argparse))
2 suppressPackageStartupMessages(library(igraph))
3 suppressPackageStartupMessages(library(assertthat))
4 suppressPackageStartupMessages(library(glmnet))
5 suppressPackageStartupMessages(library(Matrix))
6 suppressPackageStartupMessages(library(MASS))
7 suppressPackageStartupMessages(library(testthat))
8 suppressPackageStartupMessages(library(huge))
9 suppressPackageStartupMessages(library(foreach))
10 suppressPackageStartupMessages(library(doMC))
11
12 TADA.PATH = "/home/smile/klsix/kathryn/data/TADA_Results"
13 VALIDATION.PATH = "~/kathryn/data/li_251genes_1643tri"
14
15 setwd("~/kathryn.git.all/personal.git/DAWN_ADA")
16 source("source_DAWN.R")
17 source("source_HMRF.R")
18 source("source_common_func.R")
19 source("source_edgewise_hypo.R")
20 source("source_gene_clustering.R")
21 source("source_graph_estimation.R")
22 source("source_longitudinal.R")
23 source("source_caitest.R")
24 source("source_datamanagement.R")
25 source("source_validation.R")
26 source("source_wrapper.R")
27 source("plotter/plotter_basic.R")
28 source("plotter/plotter_enumerate.R")
```

A large red box highlights the first 28 lines of code, which are used to suppress startup messages from various R packages. A smaller red box highlights the last two lines of code, which define paths and set the working directory.

This is clearly a well-intended, poorly-executed hack.

You would only realize you forgot to source something during runtime.

```
29
30 set.seed(10)
31 DATE = Sys.Date()
32 date = DATE
33
34 REGION.SUBREGION = list()
35 REGION.SUBREGION[["PFC"]] = c("MFC", "DFC", "OFC", "MSC", "VFC", "M1C", "S1C")
36 REGION.SUBREGION[["VIAS"]] = c("V1C", "ITC", "IPC", "A1C", "STC")
37 REGION.SUBREGION[["SHA"]] = c("STR", "HIP", "AMY")
38 REGION.SUBREGION[["MDCBC"]] = c("MD", "CBC")
39
matrix(c(1,2,3,5,6,8,9,15), ncol = 2, byrow = TRUE)
```

I rewrote the code after I took Stat Computing, but before I thought about R packages. The results are mixed.

```
source_globalvar.R * 
Source on Save | 
1 suppressPackageStartupMessages(library(argparse))
2 suppressPackageStartupMessages(library(igraph))
3 suppressPackageStartupMessages(library(assertthat))
4 suppressPackageStartupMessages(library(glmnet))
5 suppressPackageStartupMessages(library(Matrix))
6 suppressPackageStartupMessages(library(MASS))
7 suppressPackageStartupMessages(library(testthat))
8 suppressPackageStartupMessages(library(huge))
9 suppressPackageStartupMessages(library(foreach))
10 suppressPackageStartupMessages(library(doMC))
11
12 TADA.PATH = "/home/smile/klsix/kathryn/data/TADA_Results"
13 VALIDATION.PATH = "~/kathryn/data/li_251genes_1643tri"
14
15 setwd("~/kathryn.git.all/personal.git/DAWN_ADA")
16 source("source_DAWN.R")
17 source("source_HMRF.R")
18 source("source_common_func.R")
19 source("source_edgewise_hypo.R")
20 source("source_gene_clustering.R")
21 source("source_graph_estimation.R")
22 source("source_longitudinal.R")
23 source("source_caitest.R")
24 source("source_datamanagement.R")
25 source("source_validation.R")
26 source("source_wrapper.R")
27 source("plotter/plotter_basic.R")
28 source("plotter/plotter_enumerate.R")
```

```
29
30 set.seed(10)
31 DATE = Sys.Date()
32 date = DATE
33
34 REGION.SUBREGION = list()
35 REGION.SUBREGION[["PFC"]] = c("MFC", "DFC", "OFC", "MSC", "VFC", "M1C", "S1C")
36 REGION.SUBREGION[["VIIAS"]] = c("V1C", "ITC", "IPC", "A1C", "STC")
37 REGION.SUBREGION[["SHA"]] = c("STR", "HIP", "AMY")
38 REGION.SUBREGION[["MDCBC"]] = c("MD", "CBC")
39
40 BINNING = matrix(c(1, 2, 3, 5, 6, 8, 9, 15), ncol = 2, byrow = TRUE)
```

I'm defining some global variable here, but now we're mixing data with code.

If you think “Only I need to understand my code, who cares?”, think again.

If you think “Only I need to understand my code, who cares?”, think again.

Help with Nature Neuroscience Code Review Project 

  to 

Hello.

My name is [REDACTED] and I'm an [REDACTED] at Nature Neuroscience. I'm writing to see if you might be able to help me seed a project.

We are looking into starting up a pilot project to implement code review as part of the peer review process. The process is going to be very close to the following:

- 1) At the time their paper is accepted, authors are expected to upload code to a sharing site like GitHub.
- 2) It would be up to the author as to what they share, we'd advise them that we'd like them to provide the code that generates the figures in the paper.
- 3) We will then go to a code referee and ask them to try to run the code. They will be asked to submit a very brief report that centers on whether the shared code suite is accessible, executable, and accurate (it produces the figure in the paper). If not, which functions etc. are missing?
- 4) Code reviewers will not be asked to find bugs in the code.
- 5) No paper will be rejected for failing this review. Authors will be given one opportunity to fix their code deposition.
- 6) We would expect a turnaround time of 1 week.

If you think “Only I need to understand my code, who cares?”, think again.

Help with Nature Neuroscience Code Review Project 

 to 

Hello.

My name is [REDACTED] and I'm an [REDACTED] at Nature Neuroscience. I'm writing to see if you might be able to help me seed a project.

We are looking into starting up a pilot project to implement code review as part of the peer review process. The process is going to be very close to the following:

- 1) At the time their paper is accepted, authors are expected to upload code to a sharing site like GitHub.
- 2) It would be up to the author as to what they share, we'd advise them that we'd like them to provide the code that generates the figures in the paper.
- 3) We will then go to a code referee and ask them to try to run the code. They will be asked to submit a very brief report that centers on whether the shared code suite is accessible, executable, and accurate (it produces the figure in the paper). If not, which functions etc. are missing?

[REDACTED]

- 5) No paper will be rejected for failing this review. Authors will be given one opportunity to fix their code deposition.
- 6) We would expect a turnaround time of 1 week.

If you think “Only I need to understand my code, who cares?”, think again.

JASA Applications and Case Studies Reproducibility Initiative - Author Guidelines:

This section outlines the journal's expectations for authors with regards to submission of code and data to allow for reproducibility of the work. During the review process, the reviewers and editors will assess the potential for the work to be reproduced based on the code and data provided. However, as with other aspects of a manuscript, it is ultimately the authors' responsibility to ensure that the code and data are of high quality and that the work is reproducible. As the focal point for reproducibility information, authors are required to include an Author Contributions Checklist (ACC) Form with their submission. Please see the [ACC Form](#) for further details, including suggestions for good practices regarding reproducibility.

- The code should be in a form that can be used and understood by others, including being readable at a line-by-line level in terms of syntax and comments.
- There should be a clear, documented workflow to reproduce the key results. For workflows involving more than a single script, there should be a master script, Makefile or other mechanism available such that it is clear what each component does, in what order to run the components, and what are the inputs to and outputs from the different components.
- Once a manuscript is conditionally accepted, the code and data (unless an exception is granted for the latter) must be made available as detailed in Appendix A. Manuscripts will not be accepted and should not be claimed as such by the authors until this occurs.

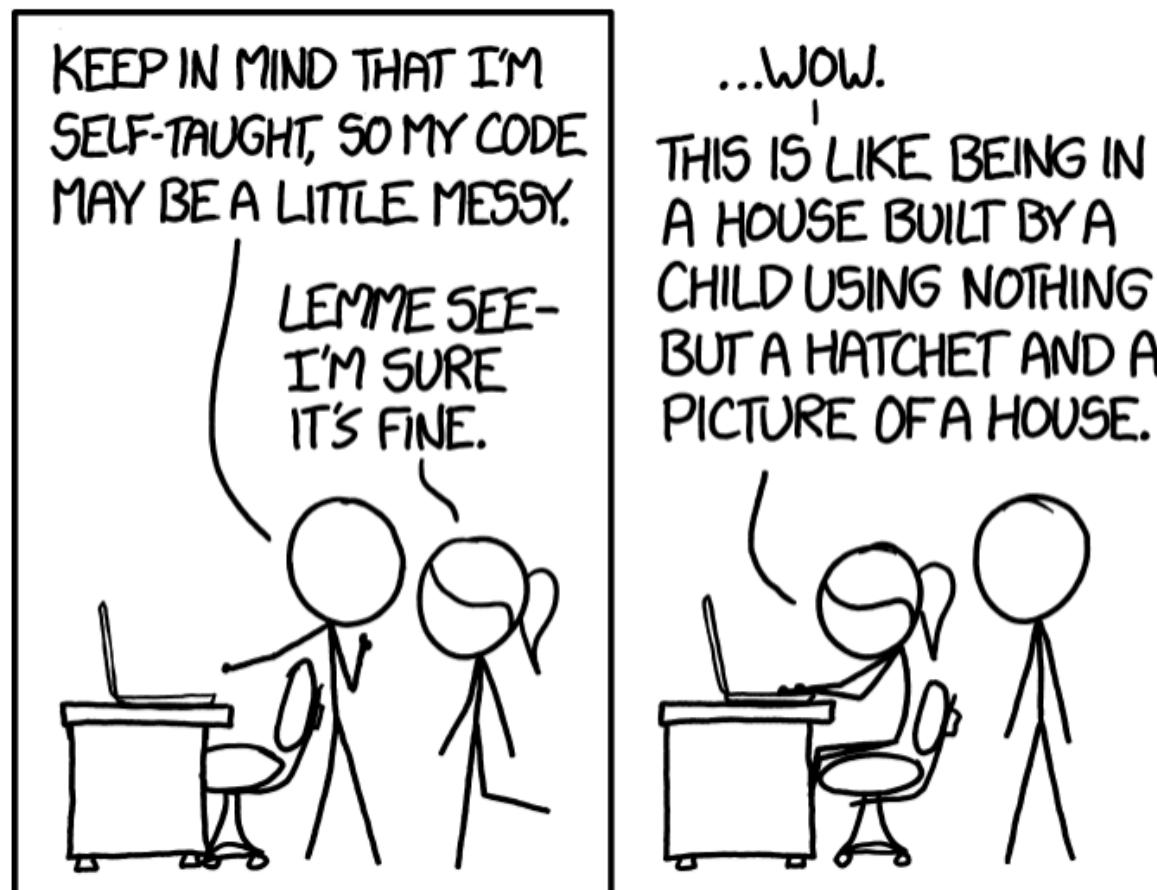
If you think “Only I need to understand my code, who cares?”, think again.

JASA Applications and Case Studies Reproducibility Initiative - Author Guidelines:

This section outlines the journal's expectations for authors with regards to submission of code and data to allow for reproducibility of the work. During the review process, the reviewers and editors will assess the potential for the work to be reproduced based on the code and data provided. However, as with other aspects of a manuscript, it is ultimately the authors' responsibility to ensure that the code and data are of high quality and that the work is reproducible. As the focal point for reproducibility information, authors are required to include an Author Contributions Checklist (ACC) Form with their submission. Please see the [ACC Form](#) for further details, including suggestions for good practices regarding reproducibility.

- The code should be in a form that can be used and understood by others, including being readable at a line-by-line level in terms of syntax and comments.
- There should be a clear, documented workflow to reproduce the key results. For workflows involving more than a single script, there should be a master script, Makefile or other mechanism available such that it is clear what each component does, in what order to run the components, and what are the inputs to and outputs from the different components.
- Once a manuscript is conditionally accepted, the code and data (unless an exception is granted for the latter) must be made available as detailed in Appendix A. Manuscripts will not be accepted and should not be claimed as such by the authors until this occurs.

If you think “Only I need to understand my code, who cares?”, think again.



Interlude

(Next: How R packages can
solve these problems)

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Organization**)

The screenshot shows a GitHub repository page for 'longitudinalGM'. At the top, there's a navigation bar with links for GitHub, Inc. [US] and https://github.com/linnylin92/longitudinalGM. Below the navigation bar, there are tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The main content area is titled 'Longitudinal Gaussian Graphical Model: JASA Applications and Case Study' and includes an 'Edit' button and a 'Add topics' link. A summary bar at the top provides metrics: 298 commits, 5 branches, 0 releases, and 1 contributor. Below this, there are buttons for Branch: master, New pull request, Create new file, Upload files, Find file, and Clone or download. The main part of the page displays a list of commits from user 'linnylin92'. Each commit includes a small icon, the author's name, a brief description, and the time since the commit. The commits are as follows:

Author	Commit Description	Date
linnylin92	Made figure 12	Latest commit 062e0b8 a day ago
data	Add remaining datasets	5 months ago
experiment	Changed how screening works	2 days ago
figures	Made figure 12	a day ago
longitudinalGM	Fixed step3 to account for the screening	2 days ago
main	Updated simulatino for analysis_overalpha	a day ago
results	Fixed largest_clique to prioritize the first num_in genes	a month ago
simulation	Updating figure 5 calculations	2 days ago
.DS_Store	Fixed figure 4	2 days ago
.gitattributes	Add BrainSpan dataset using git lfs	5 months ago
.gitignore	Changed up the simulation setting	4 days ago
.travis.yml	I think I fixed travis to work with bioconductor	5 months ago
README.md	Initial commit	a year ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Organization**)

Longitudinal Gaussian Graphical Model: JASA Applications and Case Study

Add topics

298 commits 5 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Separate folders

- large data files,
- miscellaneous experiments
- figures
- R package
- analysis
- results
- synthetic simulations

File	Message	Date
.DS_Store		5 months ago
.gitattributes		2 days ago
.gitignore	Changed up the simulation setting	a day ago
.travis.yml	I think I fixed travis to work with bioconductor	2 days ago
README.md	Initial commit	a year ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Organization**)

Branch: master	longitudinalGM / longitudinalGM /	Create new file	Upload files	Find file	History
 linnylin92	Fixed step3 to account for the screening			Latest commit f5df949	2 days ago
..					
 R	Changed how screening works				2 days ago
 data-raw	Reupdated the gene synonym list				24 days ago
 data	Reupdated the gene synonym list				24 days ago
 man	Changed how screening works				2 days ago
 tests	Fixed step3 to account for the screening				2 days ago
 .DS_Store	Reupdated the gene synonym list				24 days ago
 .Rapp.history	Tested graphicalModel_store				5 months ago
 .Rbuildignore	Added some documentation to regroup				5 months ago
 .gitignore	Initial commit				a year ago
 DESCRIPTION	Changed travis and not sure if it crashes travis?				5 months ago
 LICENSE	Changed the parameter vector in the simulation				11 months ago
 NAMESPACE	Added documentation and updated simulation				9 days ago
 longitudinalGM.Rproj	Initial commit				a year ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Organization**)

Branch: master [longitudinalGM / longitudinalGM /](#)

Create new file Upload files Find file History

linnylin92 Fixed step

Latest commit f5df949 2 days ago

R		2 days ago
data-raw		24 days ago
data		24 days ago
man		2 days ago
tests		2 days ago
.DS_Store		24 days ago
.Rapp.history		5 months ago
.Rbuildignore		5 months ago
.gitignore	Initial commit	a year ago
DESCRIPTION	Changed travis and not sure if it crashes travis?	5 months ago
LICENSE	Changed the parameter vector in the simulation	11 months ago
NAMESPACE	Added documentation and updated simulation	9 days ago
longitudinalGM.Rproj	Initial commit	a year ago

Within the R package, we have files for:

- the R functions,
- the files to create/prepare the datasets,
- the RData files that our analysis relies on,
- the documentation files
- the tests

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Organization**)

Branch: master [longitudinalGM / longitudinalGM /](#)

Create new file Upload files Find file History

linnylin92 Fixed step3 to account for the screening Latest commit f5df949 2 days ago

..

📁 R	Changed how screening works	2 days ago
📁 data-raw	Reupdated the gene synonym list	24 days ago
📁 data	Reupdated the gene synonym list	24 days ago
📁 man	Changed how screening works	2 days ago
📁 tests	Fixed step3 to account for the screening	2 days ago
.DS_Store		24 days ago
.Rapp.history		5 months ago
.Rbuildignore		5 months ago
DESCRIPTION	This file dictates which other packages your package depends on.	a year ago
LICENSE	If set up, all your dependencies will be loaded whenever you load your package.	5 months ago
NAMESPACE		11 months ago
longitudinalGM.Rproj		9 days ago
	uses travis?	a year ago
	simulation	
	simulation	

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Dependency**)

Branch: master ▾ [longitudinalGM / longitudinalGM / DESCRIPTION](#) Find file Copy path

 linnylin92 Changed travis and not sure if it crashes travis? b3fa93b on 28 Jun

1 contributor

25 lines (24 sloc) | 487 Bytes Raw Blame History

```
1 Package: longitudinalGM
2 Title: What the Package Does (one line, title case)
3 Version: 0.0.0.9000
4 Authors@R: person("First", "Last", email = "first.last@example.com", role = c("aut", "cre"))
5 Description: What the package does (one paragraph).
6 Depends:
7   R (>= 3.3.0)
8 License: MIT +file LICENSE
9 Encoding: UTF-8
10 LazyData: true
11 Imports:
12   igraph,
13   glmnet,
14   Matrix,
15   MASS,
16   huge,
17   plyr,
18   foreach,
19   doMC,
20   hash,
21   biomaRt
22 Suggests:
23   testthat
24 RoxygenNote: 6.0.1
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Dependency**)

Branch: master ▾ [longitudinalGM / longitudinalGM / DESCRIPTION](#) Find file Copy path

 linnylin92 Changed travis and not sure if it crashes travis? b3fa93b on 28 Jun

1 contributor

25 lines (24 sloc) | 487 Bytes Raw Blame History

```
1 Package: longitudinalGM
2 Title: What the Package Does (one line, title case)
3 Version: 0.0.0.9000
4 Authors@R: person("First", "Last", email = "first.last@example.com", role = c("aut", "cre"))
5 Description: What the package does (one paragraph).
6 Depends:
7   R (>= 3.3.0)
8 License: MIT +file LICENSE
9 Encoding: UTF-8
10 Imports:
11   igraph,
12   glmnet,
13   Matrix,
14   MASS,
15   huge,
16   plyr,
17   foreach,
18   doMC,
19   hash,
20   biomaRt
21 Suggests:
22   testthat
23   RoxygenNote: 6.0.1
```

Here are my dependent packages.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Dependency**)

The screenshot shows a PDF viewer interface with the following details:

- Title:** devtools.pdf
- Page Number:** 36 / 64
- Content:**
 - Function Definition:** `load_all` *Load complete package.*
 - Description:** `load_all` loads a package. It roughly simulates what happens when a package is installed and loaded with `library`.
 - Usage:** `load_all(pkg = ".", reset = TRUE, recompile = FALSE, export_all = TRUE, quiet = FALSE, create = NA)`

Thoughtful organization, Git, and R packages can solve most of our previous problems. ([Documentation](#))

Branch: master ▾

[longitudinalGM](#) / [longitudinalGM](#) / R / [graphicalModel_form.R](#)

[Find file](#) [Copy path](#)



[linnylin92](#) Fixed warnings from R check

8ddab7d on 8 Oct

1 contributor

74 lines (63 sloc) | 2.43 KB

Raw Blame History

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

Branch: master ▾

[longitudinalGM](#) / [longitudinalGM](#) / R / graphicalModel_form.R[Find file](#) [Copy path](#)

linnylin92 Fixed warnings from R check

8ddab7d on 8 Oct

1 contributor

74 lines (63 sloc) | 2.43 KB

[Raw](#)[Blame](#)[History](#)

```
#' Graphical model estimate
#'
#' After a call of \code{graphicalModel_store}, \code{graphicalmodel_form}
#' will take the saved glmnet objects (dictated by \code{filename_func})
#' to form the precision matrix
#'
#' @param dat the nxn matrix
#' @param filename_func a function to create the file name. it takes in
#' \code{i}, the dimension index, as a parameter
#' @param lambda lambda value
#' @param num_primary integer from 1 to \code{ncol(dat)} that denotes
#' the number of genes to consider
#' @param tol numeric
#' @param cores number of cores
#' @param verbose boolean for verbose
#'
#' @return a precision matrix estimate, dxd
#' @export
```

l(dat),
tol = 1e-6, cores = 1, verbose = F){

20

This is ROxygen. It's a template that you fill out so R can later convert into proper documentation easily.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

Branch: master ▾

[longitudinalGM](#) / [longitudinalGM](#) / R / graphicalModel_form.R[Find file](#) [Copy path](#)

linnylin92 Fixed warnings from R check

8ddab7d on 8 Oct

1 contributor

74 lines (63 sloc) | 2.43 KB

[Raw](#) [Blame](#) [History](#)

```
1  #' Graphical model estimate
2  #
3  #' After a call of \code{graphicalModel_store}, \code{graphicalmodel_form}
4  #' will take the saved glmnet objects (dictated by \code{filename_func})
5  #' to form the precision matrix

6  #' @param dat the nxn matrix
7  #' @param filename_func a function to create the file name. it takes in
8  #' \code{i}, the dimension index, as a parameter
9  #' @param lambda lambda value
10 #' @param num_primary integer from 1 to \code{ncol(dat)} that denotes
11 #' the number of genes to consider
12 #' @param tol numeric
13 #' @param cores number of cores
14 #' @param verbose boolean for verbose

15
16
17 #' @return a precision matrix estimate, dxd
18 #' @export
19 graphicalModel_form <- function(dat, filename_func, lambda, num_primary = ncol(dat),
20                               tol = 1e-6, cores = 1, verbose = F){
```

R will check you put a description next to each argument to the function.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

Branch: master ▾

[longitudinalGM](#) / [longitudinalGM](#) / R / graphicalModel_form.R[Find file](#) [Copy path](#)

linnylin92 Fixed warnings from R check

8ddab7d on 8 Oct

1 contributor

74 lines (63 sloc) | 2.43 KB

[Raw](#) [Blame](#) [History](#)

```
1  #' Graphical model estimate
2  #
3  #' After a call of \code{graphicalModel_store}, \code{graphicalmodel_form}
4  #' will take the saved glmnet objects (dictated by \code{filename_func})
5  #' to form the precision matrix
6  #
7  #' @param dat the nxn matrix
8  #' @param filename_func a function to create the file name. it takes in
9  #' \code{i}, the dimension index, as a parameter
10 #' @param lambda lambda value
11 #' @param num_primary integer from 1 to \code{ncol(dat)} that denotes
12 #' the number of genes to consider
13 #' @param tol numeric
14 #' @param cores number of cores
15 #' @param verbose boolean for verbose
16 #
17 #' @return a precision matrix estimate, dxd
18 #' @export
19 graphicalModel_form <- function(dat, filename_func, lambda, num_primary = ncol(dat),
20                               tol = 1e-6, cores = 1, verbose = F){
```

R will check you put a description next to the output.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

Branch: master ▾

longitudinalGM / longitudinalGM / R / graphicalModel_form.R

Find file Copy path



linnylin92 Fixed warnings from R check

8ddab7d on 8 Oct

1 contributor

74 lines (63 sloc) | 2.43 KB

Raw

Blame

History



```
1  #' Graphical model estimate
2  #
3  #' After a call of \code{graphicalModel_form} or \code{graphicalModel_e_func}
4  #' will take the saved information and use it to form the precision matrix.
5  #' to form the precision matrix. This function is not exported.
6  #
7  #' @param dat the nxn matrix to be used in the estimation
8  #' @param filename_func the name of the file containing the function to be used
9  #' \code{lambda}, the dimension of the precision matrix
10 #' @param lambda lambda value
11 #' @param num_primary the number of generic eigenvalues to be used
12 #' @param tol numerical tolerance
13 #' @param cores number of cores to be used
14 #' @param verbose boolean indicating whether to print progress messages
15 #' @param verbose boolean indicating whether to print progress messages
16 #
17 #'
18 #'
19 #' @export
20 graphicalModel_form <- function(dat, filename_func, lambda, num_primary = ncol(dat),
21                                tol = 1e-6, cores = 1, verbose = F){
```

You can control whether or not you want users to be able to use this function.

Functions that are not exported are understood to be “helper” functions, i.e., functions the users don’t need to know even exist.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

Branch: master ▾ [longitudinalGM / longitudinalGM / man / graphicalModel_form.Rd](#) Find file Copy path

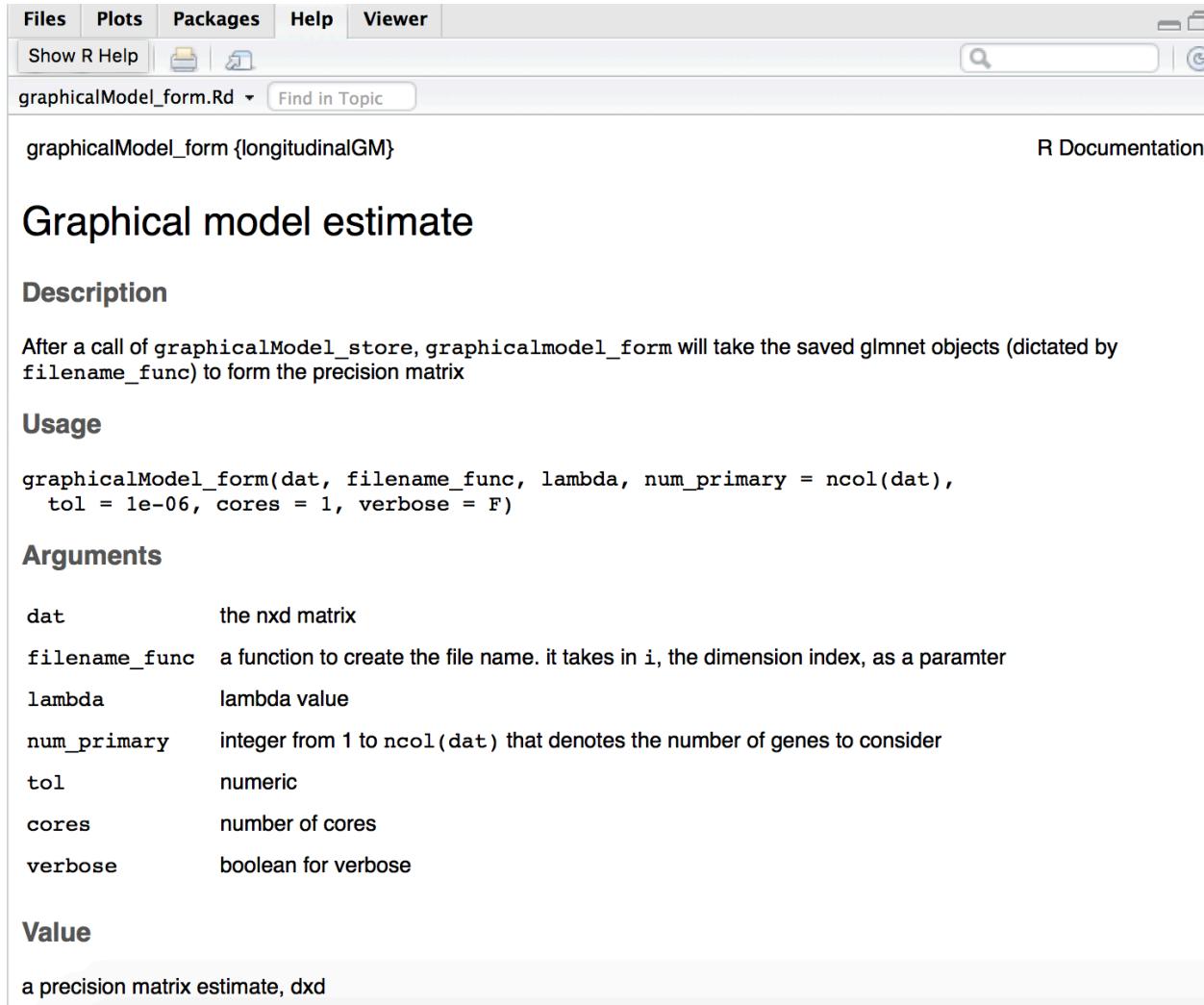
 linnylin92 Fixed warnings from R check 8ddab7d on 8 Oct

1 contributor

35 lines (28 sloc) | 916 Bytes Raw Blame History

```
1 % Generated by roxygen2: do not edit by hand
2 % Please edit documentation in R/graphicalModel_form.R
3 \name{graphicalModel_form}
4 \alias{graphicalModel_form}
5 \title{Graphical model estimate}
6 \usage{
7   graphicalModel_form(dat, filename_func, lambda, num_primary = ncol(dat),
8     tol = 1e-06, cores = 1, verbose = F)
9 }
10 \arguments{
11   \item{dat}{the nxd matrix}
12 
13   \item{filename_func}{a function to create the file name. it takes in
14     \code{i}, the dimension index, as a parameter}
15 
16   \item{lambda}{lambda value}
17 
18   \item{num_primary}{integer from 1 to \code{ncol(dat)} that denotes
19     the number of genes to consider}
20 
21   \item{tol}{numeric}
22 
23   \item{cores}{number of cores}
24 
25   \item{verbose}{boolean for verbose}
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)



The screenshot shows the R Help Viewer interface. The title bar includes tabs for Files, Plots, Packages, Help, and Viewer, with 'Viewer' being the active tab. Below the title bar, there are buttons for Show R Help, Print, and Copy. The search bar contains 'graphicalModel_form.Rd'. The main content area displays the R documentation for the 'graphicalModel_form' function from the 'longitudinalGM' package. The title is 'graphicalModel_form {longitudinalGM}'. The documentation is titled 'Graphical model estimate'. It includes sections for 'Description', 'Usage', 'Arguments', and 'Value'. The 'Description' section states: 'After a call of graphicalModel_store, graphicalmodel_form will take the saved glmnet objects (dictated by filename_func) to form the precision matrix'. The 'Usage' section provides the R code: 'graphicalModel_form(dat, filename_func, lambda, num_primary = ncol(dat), tol = 1e-06, cores = 1, verbose = F)'. The 'Arguments' section lists the parameters: dat (nxd matrix), filename_func (function to create file name), lambda (lambda value), num_primary (integer from 1 to ncol(dat)), tol (numeric), cores (number of cores), and verbose (boolean for verbose). The 'Value' section states: 'a precision matrix estimate, dxd'.

graphicalModel_form {longitudinalGM}

R Documentation

Graphical model estimate

Description

After a call of `graphicalModel_store`, `graphicalmodel_form` will take the saved `glmnet` objects (dictated by `filename_func`) to form the precision matrix

Usage

```
graphicalModel_form(dat, filename_func, lambda, num_primary = ncol(dat),
  tol = 1e-06, cores = 1, verbose = F)
```

Arguments

<code>dat</code>	the nxd matrix
<code>filename_func</code>	a function to create the file name. it takes in <code>i</code> , the dimension index, as a parameter
<code>lambda</code>	lambda value
<code>num_primary</code>	integer from 1 to <code>ncol(dat)</code> that denotes the number of genes to consider
<code>tol</code>	numeric
<code>cores</code>	number of cores
<code>verbose</code>	boolean for verbose

Value

a precision matrix estimate, dxd

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

```
clique_selection <- function(g, threshold = 0.95, mode = "all",
                           target_idx = NA, prune = T,
                           verbose = F, time_limit = 3600,
                           max_length = 500000){
  stopifnot(mode %in% c("all", "or"), length(mode) == 1, class(g) == "igraph")

  d <- igraph::vcount(g)
  adj <- as.matrix(igraph::as_adjacency_matrix(g))

  clique_list <- lapply(igraph::maximal.cliques(g), function(x){sort(as.numeric(x))})
  if(any(!is.na(target_idx))) clique_list <- .prune_clique(clique_list, target_idx)
  if(length(clique_list) == 1) return(list(1:ncol(adj)))
  largest_clique <- list(sort(clique_list[[which.max(sapply(clique_list, length))]]))

  len <- length(clique_list)
  queue <- .initialize_queue(len)
  hash_history <- hash::hash()
  hash_children <- .initialize_children(clique_list)
  hash_unique <- .inititalize_unique(clique_list, d)
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

```
clique_selection <- function(g, threshold = 0.95, mode = "all",
                           target_idx = NA, prune = T,
                           verbose = F, time_limit = 3600,
                           max_length = 500000){

stopifnot(mode %in% c("all", "or"), length(mode) == 1, class(g) == "igraph")

d <- igraph::vcount(g)
adj <- as.matrix(igraph::as_adjacency_matrix(g))

clique_list <- lappl(igraph::maximal.cliques(g)) function(x){sort(as.numeric(x))})
if(any(!is.na(target_idx))) clique_list <- .prune_clique(clique_list, target_idx)
if(length(clique_list) == 1) return(list(1:ncol(adj)))

largest_clique <- list(sort(clique_list[[1]]))

len <- length(clique_list)
queue <- .initialize_queue(len)
hash_history <- hash::hash()
hash_children <- vector("list", len)
hash_unique <- .initialize_unique()
```

You can/should explicitly denote the originating packages whenever you use a non-base R function.

This helps you keep track of where you use each package.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

```
'#' Compute the bootstrap empirical covariance (numerator)
#
#' @param mat data matrix
#' @param noise_vec vector of noise
#' @param cov_mat covariance matrix
#' @param idx vector of the indices of the lower triangle
#' @param unscaled boolean for whether to scale
#
#' @return vector
.compute_bootSigma <- function(mat, noise_vec, cov_mat = NA, idx = NA,
                                unscaled = T){
  if(length(noise_vec) != nrow(mat)) stop("length(noise_vec) not equal to nrow(mat)")

  n <- nrow(mat)
  if(any(is.na(cov_mat))){
    cov_mat <- (n-1)/n*stats::cov(mat)
  } else {
    stopifnot(ncol(cov_mat) == ncol(mat))
  }

  if(unscaled) mat <- scale(mat, center = TRUE, scale = FALSE)

  res <- t(mat) %*% diag(noise_vec/n) %*% mat - (sum(noise_vec)/n)*cov_mat
  if(any(is.na(idx))){
    c(res[lower.tri(res, diag = T)])
  } else {
    c(res[idx])
  }
}
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

```
#' Compute the bootstrap empirical covariance (numerator)
#'
#' @param mat data matrix
#' @param noise_vec vector
#' @param cov_mat covariance matrix
#' @param idx vector of the indices
#' @param unscaled boolean
#'
#'
#' @return vector
.comPUTE_BOOTSIGMA <- function(mat, noise_vec, cov_mat, idx, unscaled) {
  if(length(noise_vec) != nrow(mat)) {
    stopifnot(length(noise_vec) == to_nrow(mat))
  }
  n <- nrow(mat)
  if(any(is.na(cov_mat))){
    cov_mat <- (n-1)/n*mat
  } else {
    stopifnot(ncol(cov_mat) == to_nrow(mat))
  }

  if(unscaled) mat <- scale(mat, center = TRUE, scale = FALSE)

  res <- t(mat) %*% diag(noise_vec/n) %*% mat - (sum(noise_vec)/n)*cov_mat
  if(any(is.na(idx))){
    c(res[lower.tri(res, diag = T)])
  } else {
    c(res[idx])
  }
}
```

This function is not exported. The documentation is strictly for my own benefit (as Roxygen will not actually touch this).

I use a “.” in front of the function name to further emphasize this.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Documentation**)

```
#' Compute the bootstrap empirical covariance (numerator)
#'
#> @return vector
#> @param mat matrix
#> @param noise_vec vector
#> @param cov_mat matrix
#> @param idx integer
#> @param unscaled logical
#>
#> @details This function computes the bootstrap empirical covariance
#> (numerator) for a given matrix mat, noise vector noise_vec, covariance
#> matrix cov_mat, index idx, and unscaled flag unscaled. It first checks if the
#> length of noise_vec is equal to the number of rows in mat. If not, it stops with an
#> error message. Then, it calculates the covariance matrix cov_mat from the matrix mat.
#> If cov_mat is NA, it is set to (n-1)/n times the sample covariance of mat. Otherwise,
#> it checks if the number of columns in cov_mat is equal to the number of columns in mat.
#> If not, it stops with an error message. Then, it scales the matrix mat if unscaled is
#> TRUE. Finally, it calculates the result res as the difference between the transpose of
#> mat and the product of noise_vec/n and cov_mat. If idx is NA, it returns the lower
#> triangle of res with diagonal elements set to zero. Otherwise, it returns the element at
#> index idx.
```

In general, keep functions short in length so it's easier to test it.

lower triangle
scale

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

linnylin92 / longitudinalGM

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master → longitudinalGM / longitudinalGM / tests / testthat / Create new file Upload files Find file History

linnylin92 Fixed step3 to account for the screening Latest commit f5df949 2 days ago

..

.DS_Store	Added binning function and step5	4 months ago
test_binning.R	Added binning function and step5	4 months ago
test_cai_test.R	Added a stepdown path method	9 days ago
test_cai_test_numericSimulation.R	Added a small numerical test for Cai covariance test	a year ago
test_clique_selection.R	Added one more test	11 days ago
test_decorrelated_test.R	Parallelized decorrelated test	2 months ago
test_demean.R	Added tests to demean a matrix	5 months ago
test_extractor.R	Added binning function and step5	4 months ago
test_filename.R	Added filename function	5 months ago
test_graph_functions.R	Added new way to do sample selection	a month ago
test_graphicalModel.R	Changed workings of graphicalModel	5 months ago
test_graphicalModel_form.R	Parallelized forming of graphical model	5 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

```
## .compute_bootSigma is correct

test_that(".compute_bootSigma works", {
  set.seed(10)
  d <- 5; n <- 10
  mat <- matrix(rnorm(n*d), n, d)
  noise_vec <- rnorm(n)
  res <- .compute_bootSigma(mat, noise_vec)

  expect_true(is.numeric(res))
  expect_true(length(res) == d + d*(d-1)/2)
})

test_that(".compute_bootSigma is computed correctly", {
  set.seed(10)
  d <- 5; n <- 10
  mat <- matrix(rnorm(n*d), n, d)
  noise_vec <- rnorm(n)
  res <- .compute_bootSigma(mat, noise_vec)

  cov_mat <- (n-1)/n*stats::cov(mat)
  mat <- scale(mat, center = T, scale = F)
  res2.mat <- matrix(0, d, d)
  for(i in 1:d){
    for(j in 1:i){
      res2.mat[i,j] <- sum(noise_vec * (mat[,i] * mat[,j] - cov_mat[i,j]))/n
    }
  }
})
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

```
## .compute_bootSigma is correct
```

```
test_that(".compute_bootSigma works", {  
  set.seed(10)  
  d <- 5; n <- 10  
  mat <- matrix(rnorm(n*d), n, d)  
  noise_vec <- rnorm(n)  
  res <- .compute_bootSigma(mat, noise_vec)  
  
  expect_true(is.numeric(res))  
  expect_true(length(res) == d + d*(d-1)/2)  
})
```

```
test_that(".compute_bootSigma is computed correct", {  
  set.seed(10)  
  d <- 5; n <- 10  
  mat <- matrix(rnorm(n*d), n, d)  
  noise_vec <- rnorm(n)  
  res <- .compute_bootSigma(mat, noise_vec)  
  
  cov_mat <- (n-1)/n*stats::cov(mat)  
  mat <- scale(mat, center = T, scale = F)  
  res2.mat <- matrix(0, d, d)  
  for(i in 1:d){  
    for(j in 1:i){  
      res2.mat[i,j] <- sum(noise_vec * (mat[,i] * mat[,j] - cov_mat[i,j]))/n  
    }  
  }
```

I include a “this function works” test typically. This is usually the first test I write for every function, and it’s essentially my most basic example of how to use the function.

This compensates for sparsely documented functions I might have.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

```
> devtools::test()  
Loading longitudinalGM  
Loading required package: testthat  
Testing longitudinalGM  
Test binning: ..  
Test Cai covariance test numerically: .  
Test Cai covariance test: .....  
Test clique selection: .....  
Test decorrelated score test: ..  
Test demeaning matrix: ..  
Test extractor: ..  
Test filename functions: ...  
Test graph functions: ....  
Test graphical model forming: .....  
Test graphical model storing: ..  
Test graphical model: ..  
Test hmrf: ...  
Test isCheck, type checkers: .....  
Test matching: .  
Test naming: .....  
Test pairwise test: ..  
Test pairwise test numerically: ..  
Test regroup: .....  
Test results: ...  
Test screening: ..  
Test stepdown path: .....  
Test stepdown: ..
```

DONE =====

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

```
> devtools::check()
Updating longitudinalGM documentation
Loading longitudinalGM

* checking whether the namespace can be loaded with stated dependencies ...
... OK
* checking whether the namespace can be unloaded cleanly ... OK
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd metadata ... OK
* checking Rd line widths ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd \usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking contents of 'data' directory ... OK
* checking data for non-ASCII characters ... OK
* checking data for ASCII and uncompressed saves ... OK
* checking examples ... NONE
* checking for unstated dependencies in 'tests' ... OK
* checking tests ...
  Running 'testthat.R' [32s/47s]
OK
* DONE

Status: OK

R CMD check results
0 errors | 0 warnings | 0 notes
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Data**)

Branch: master ▾	longitudinalGM / longitudinalGM / data-raw /	Create new file	Upload files	Find file	History
	linnylin92 Reupdated the gene synonym list				Latest commit b5db80b 24 days ago
	..				
	📄 .DS_Store	Minor adjustments			2 months ago
	📄 Brain_expression.txt	Adding the initial datasets we need (minus BrainSpan)			5 months ago
	📄 TADA_Results_2231trios_1333trans_1...	Adding the initial datasets we need (minus BrainSpan)			5 months ago
	📄 brain_expression.R	Add remaining datasets			5 months ago
	📄 brainseq.info.txt	Adding the initial datasets we need (minus BrainSpan)			5 months ago
	📄 brainspan_id.R	Add brainspan_id data			5 months ago
	📄 gene-tools-kevinl1-2016-03-29.RData	Changed travis and not sure if it crashes travis?			5 months ago
	📄 gene_synonym_list_bk-03-12-2015.txt	Reupdated the gene synonym list			24 days ago
	📄 iossifov.R	Added binning function and step5			4 months ago
	📄 li_251genes_1643trios_March2015.csv	Adding the initial datasets we need (minus BrainSpan)			5 months ago
	📄 partitioning.R	Changed the partitioning dataset			5 months ago
	📄 synonyms.R	Explicitly added the synonym formation			2 months ago
	📄 tada.R	Add remaining datasets			5 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Data**)

Branch: master ▾		longitudinalGM / longitudinalGM / data /	Create new file	Upload files	Find file	History
 linnylin92 Reupdated the gene synonym list			Latest commit b5db80b 24 days ago			
..						
 brain_expression.rda		Add remaining datasets	5 months ago			
 brainspan_id.rda		Add brainspan_id data	5 months ago			
 iossifov.rda		Added binning function and step5	4 months ago			
 region_subregion.rda		Changed the partitioning dataset	5 months ago			
 synonyms.rda		Reupdated the gene synonym list	24 days ago			
 tada.rda		Add remaining datasets	5 months ago			

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master ▾	longitudinalGM / main /	Create new file	Upload files	Find file	History
 linnylin92 Updated simulatino for analysis_overalpha					Latest commit 13002bc 2 days ago
..					
 .DS_Store	Added subject selection to select only PFC 35				a month ago
 analysis.R	Changing minor details on server				a month ago
 analysis_all.R	Changed minor things				a month ago
 analysis_overalpha.R	Updated simulatino for analysis_overalpha				2 days ago
 analysis_overlambda.R	Added subject selection to select only PFC 35				a month ago
 analysis_overlambda_collect.R	Fixed common typo				2 months ago
 analysis_overquantile_search.R	Added search for quantile.				24 days ago
 analysis_overselection.R	Updated analysis over selection				2 months ago
 analysis_overselection_collect.R	Exported Cai test, starting test for null distribution				2 months ago
 analysis_simultaenous.R	Updated simulatino for analysis_overalpha				2 days ago
 step0_header.R	Updated simulatino for analysis_overalpha				2 days ago
 step1_loading.R	The entire pipeline is up and running again				a month ago
 step2_subjectselection.R	Running analysis over alpha				2 days ago
 step2_subjectselection_alternative.R	Trying to set up the global simulation				10 days ago
 step2_subjectselection_pfc35.R	Trying to set up the global simulation				10 days ago
 step3_graph.R	Fixed step3 to account for the screening				2 days ago
 step4_hmrf.R	Trying to set up the global simulation				10 days ago
 step5_results.R	Preparing for re-running new analysis				11 days ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master ▾	longitudinalGM / main /	Create new file	Upload files	Find file	History
 linnylin92 Updated simulatino for analysis_overalpha					Latest commit 13002bc 2 days ago
..					
.DS_Store	Added subject selection to select only PFC 35				a month ago
analysis.R	Changing minor details on server				a month ago
analysis_all.R	Changed minor things				a month ago
analysis_overalpha.R	Updated simulatino for analysis_overalpha				2 days ago
analysis_overlambda.R	Added subject selection to select only PFC 35				a month ago
analysis_overlambda_collect.R	Fixed common typo				2 months ago
analysis_overquantile_search.R	Added search for quantile.				24 days ago
analysis_overselection.R	Updated analysis over selection				2 months ago
analysis_overselection_collect.R	Exported Cai test, starting test for null distribution				2 months ago
analysis_overselection_overalpha.R	Updated simulatino for analysis_overalpha				2 days ago
step0_header.R					2 days ago
step1_loading.R	Updated simulatino for analysis_overalpha				
step2_subjectselection.R					
step2_subjectselection_alternative.R	The				
step2_subjectselection_pfc35.R	Run				
step3_graph.R	Try				
step4_hmrf.R	Try				
step5_results.R	Fix				
	Trying to set up the global simulation				
	Preparing for re-running new analysis				

These functions are organized to the different segments of the analysis pipeline.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master ▾	longitudinalGM / main /	Create new file	Upload files	Find file	History
 linnylin92 Updated simulation for analysis_overalpha					Latest commit 13002bc 2 days ago
..					
 DS_Store	Added subject selection to select only PFC 35				a month ago
 analysis.R	Changing minor details on server				a month ago
 analysis_all.R	Changed minor things				a month ago
 analysis_overalpha.R	Updated simulation for analysis_overalpha				2 days ago
 analysis_overlambda.R	Added search for quantile.				a month ago
 analysis_overlambda_collect.R	Updated analysis over selection				2 months ago
 analysis_overquantile_search.R	Exported Cai test, starting test for null distribution				24 days ago
 analysis_overselection.R	Updated simulation for analysis_overalpha				2 months ago
 analysis_overselection_collect.R	Updated simulation for analysis_overalpha				2 days ago
 analysis_simultaenous.R	Updated simulation for analysis_overalpha				2 days ago
 step0_header.R	The entire pipeline is up and running again				a month ago
 step1_loading.R	Running analysis over alpha				2 days ago
 step2_subjectselection.R	Trying to set up the global simulation				10 days ago
 step2_subjectselection_alternative.R	Trying to set up the global simulation				10 days ago
 step2_subjectselection_pfc35.R	Fixed step3 to account for the screening				2 days ago
 step3_graph.R	Trying to set up the global simulation				10 days ago
 step4_hmr.R	Preparing for re-running new analysis				11 days ago

These functions pipe all the steps together in sequence.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master [longitudinalGM / main / analysis.R](#) Find file Copy path

 linnylin92 Changing minor details on server 4d473ea on 15 Oct

1 contributor

11 lines (8 sloc) | 235 Bytes Raw Blame History   

```
1 source("../main/step0_header.R")
2
3 lambda <- 0.16
4 source("../main/step1_loading.R")
5 source("../main/step2_subjectselection.R")
6 source("../main/step3_graph.R")
7 source("../main/step4_hmrdf.R")
8 source("../main/step5_results.R")
9
10 warnings()
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master [longitudinalGM / main / step1_loading.R](#) Find file Copy path

 linnylin92 The entire pipeline is up and running again 3ae5c29 on 13 Oct

1 contributor

59 lines (47 sloc) | 1.77 KB Raw Blame History

```
1 #format the brainspan dataset
2 load("/raid6/Kevin/newGenexp.RData")
3 rownames(genexp) <- genexp[,1]
4 genexp <- genexp[,-1]
5 genexp <- t(genexp)
6 genexp <- as.data.frame(genexp)
7
8 #determine brain-expressed genes
9 brain_expression <- longitudinalGM::brain_expression
10 brain_genes <- brain_expression$Gene[brain_expression$Brain_expressed != 'No']
11 idx <- which(colnames(genexp) %in% brain_genes)
12 genexp <- genexp[,idx]
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master [longitudinalGM / main / step1_loading.R](#) Find file Copy path

 linnylin92 The entire pipeline is up and running again 3ae5c29 on 13 Oct

1 contributor

59 lines (47 sloc) | 1.77 KB Raw Blame History

```
1 #format the brainspan dataset
2 load("/raid6/Kevin/newGenexp.RData")
3 rownames(genexp) <- genexp[,1]
4 genexp <- genexp[,-1]
5 genexp <- t(genexp)
6 genexp <- as.data.frame(genexp)
7
8
9 brain_expression <- longitudinalGM::brain_expression
10 brain_genes <- brain_expression$gene[brain_expression$brain_
11 idx <- which(colnames(genexp) %in% brain_genes)
12 genexp <- genexp[,idx]
```

No more ugly “read.csv” lines where I need to hope the filepath on someone else’s machine is properly set up.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master [longitudinalGM / main / step4_hmrf.R](#) [Find file](#) [Copy path](#)

linnylin92 Trying to set up the global simulation 28f8351 10 days ago

1 contributor

39 lines (32 sloc) | 1.63 KB [Raw](#) [Blame](#) [History](#)   

```
1 pthres <- 0.05
2 lambda <- 0.16
3 alpha_edge <- 0.05
4 num_target <- 214
5
6 if(verbose) print("Forming graph")
7 prec_mat <- longitudinalGM::graphicalModel_form(dat, filename_func, lambda = lambda,
8                                         num_primary = num_primary,
9                                         cores = cores, verbose = verbose)
10 if(verbose) print(paste0(Sys.time(), ": Finished forming graph"))
11 sigma_mat <- stats::cov(dat)
12 d <- ncol(dat)
13 tmp <- prec_mat; tmp[lower.tri(tmp, diag = T)] <- 0
14 edges <- which(abs(tmp) > 1e-6, arr.ind = T)

36 rm(list = c("sigma_mat", "d", "tmp", "edges", "idx", "pthres", "seedindex"))
37
38 save.image(file = paste0("/raid6/Kevin/longitudinal_results/step4_res", additional_name, ".RData"))
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Analysis**)

Branch: master [longitudinalGM / main / step4_hmrf.R](#) Find file Copy path

linnylin92 Trying to set up the global simulation 28f8351 10 days ago

1 contributor

39 lines (32 sloc) | 1.63 KB Raw Blame History

```
pthres <- 0.05
lambda <- 0.16
alpha_edge <- 0.05
num_target <- 214

if(verbose) print("Forming graph")
prec_mat <- longitudinalGM::graph

if(verbose) print(paste0("System matrix"))
sigma_mat <- stats::cov(dat)
d <- ncol(dat)
tmp <- prec_mat; tmp[lower.tri(tmp)] <- 0
edges <- which(abs(tmp) > 1e-6)
```

I set up all the important tuning parameters upfront.

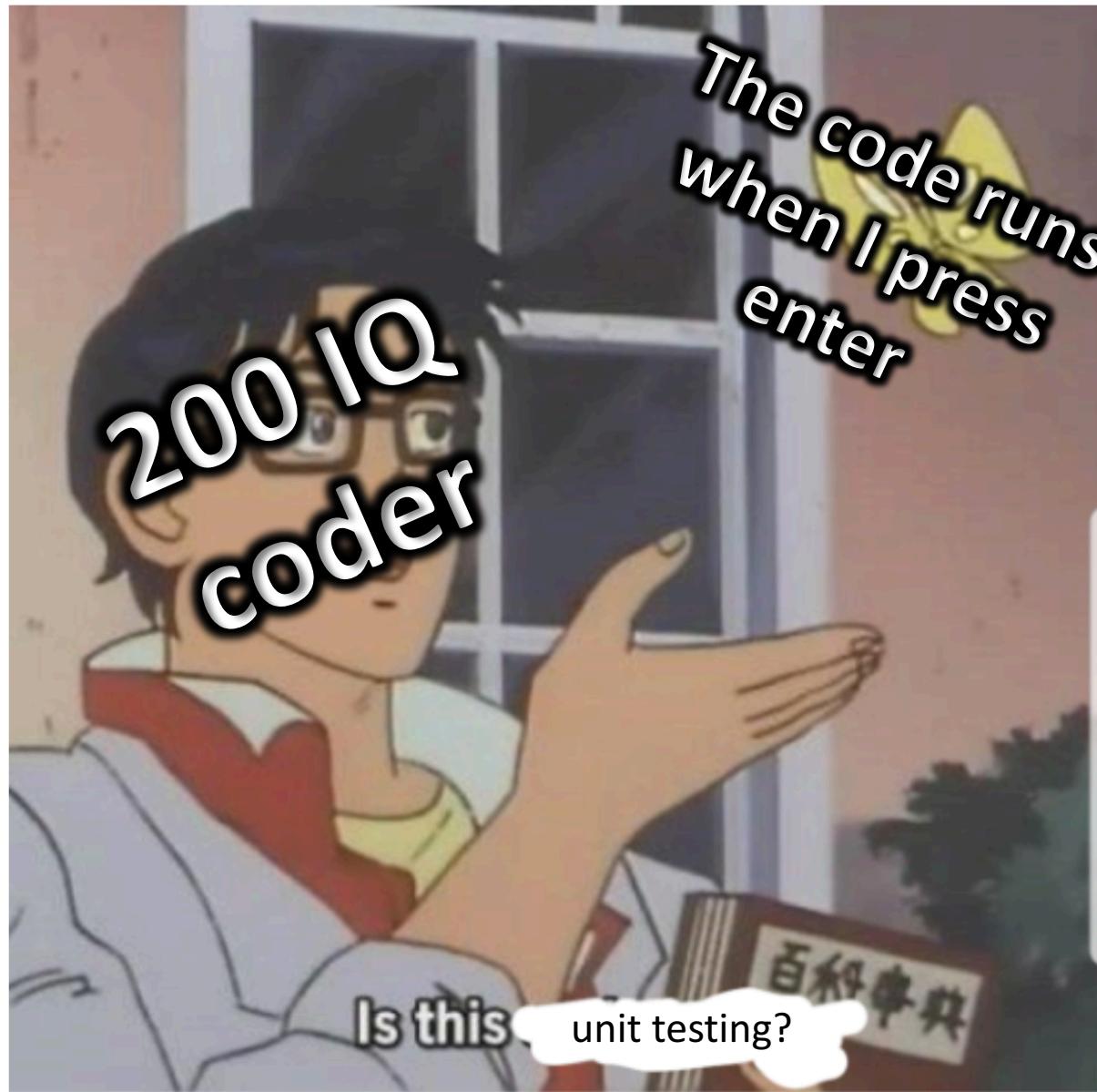
And then after some cleanup, I save the entire workspace so my parameter settings are tied to my results.

```
rm(list = c("sigma_mat", "d", "tmp", "edges", "idx", "pthres", "seedindex"))

save.image(file = paste0("/raid6/Kevin/longitudinal_results/step4_res", additional_name, ".RData"))
```

Interlude

(Next: A tale of unit testing)



Unit testing requires you to pay an upfront cost in terms of time...

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection

The “main novelty” in methods development for my ADA.

Let's see what functions it depends on

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection
 - .prune_clique
 - .initialize_queue / .initialize_unique
 - %while loop

- .post_fillin
- .prune_nodes

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection
 - .prune_clique
 - .initialize_queue / .initialize_unique

- %while loop

This function is essentially a huge while loop, each iteration popping things from a queue, and then adding more things to the queue.

The loop ends when there are no more things in the queue

- .post_fillin
- .prune_nodes

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection
 - .prune_clique
 - .initialize_queue / .initialize_unique
 - %while loop
 - % pop from a a queue
 - .check_nonunique
 - .pass_threshold
 - .add_to_queue
 - .post_fillin
 - .prune_nodes

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection
 - .prune_clique
 - .initialize_queue / .initialize_unique
 - .convert_pair_to_string
 - %while loop
 - % pop from a a queue
 - .check_nonunique
 - .convert_indices_to_binary
 - .pass_threshold
 - .add_to_queue
 - .check_nonunique
 - .check_superset
 - .check_pairs
 - .push_to_queue
 - .post_fillin
 - .prune_nodes

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- clique_selection
 - .prune_clique
 - .initialize_queue / .initialize_unique
 - .convert_pair_to_string
 - %while loop
 - % pop from a a queue
 - .check_nonunique
 - .convert_indices_to_binary
 - .pass_threshold
 - .add_to_queue
 - .check_nonunique
 - .check_superset
 - .check_pairs
 - .push_to_queue
 - .post_riffin
 - .prune_nodes

That's a **LOT** of functions.

How could you possibly ensure it's correct, given that it's sequential, and loops over tens of thousands of iterations?

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- An analogy of factories:

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- A (not very relevant) analogy of factories:

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- A (not very relevant) analogy of factories:
 - (coding)

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- A (not very relevant) analogy of factories:
 - (coding)
- Imagine you're the manager of a factory with hundreds of assembly lines
 - (developer of an analysis that uses many function)

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- A (not very relevant) analogy of factories:
 - (coding)
- Imagine you're the manager of a factory with hundreds of assembly lines
 - (developer of an analysis that uses many function)
- You can't realistically check literally every single stage of literally every single product
 - (literally every single type of input to every function)

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- A (not very relevant) analogy of factories:
 - (coding)
- Imagine you're the manager of a factory with hundreds of assembly lines
 - (developer of an analysis that uses many function)
- You can't realistically check literally every single stage of literally every single product
 - (literally every single type of input to every function)
- You can't expect to see problems by only superficially inspecting the final product
 - (inspecting the output of the main function)

Unit testing requires you to pay an upfront cost in terms of time, but will allow you to sleep better at night.

- A (not very relevant) analogy of factories:
 - (coding)
- Imagine you're the manager of a factory with hundreds of assembly lines
 - (developer of an analysis that uses many function)
- You can't realistically check literally every single stage of literally every single product
 - (literally every single type of input to every function)
- You can't expect to see problems by only superficially inspecting the final product
 - (inspecting the output of the main function)
- However, you can teach each worker a basic checklist to follow routinely (where alarms go off when the checklist is violated), and “usually” that's good enough.
 - (function unit tests that are routinely checked (where you're told when unit tests fail))

What kind of unit tests could I write?

```
## .convert_pair_to_string is correct

test_that(".convert_pair_to_string works", {
  res <- .convert_pair_to_string(c(4,10))
  expect_true(res == "4-10")
})
```

What kind of unit tests could I write?

```
## .convert_pair_to_string is correct

test_that(".convert_pair_to_string works", {
  res <- .convert_pair_to_string(c(4,10))
  expect_true(res == "4-10")
})
```

Seems innocuous. Simple tests to simply show they work on a simple input.

Utility: if these functions depend on lower-level function that are constantly being change

```
## .add_to_queue is correct

test_that(".add_to_queue works", {
  n <- 10
  combn_mat <- combn(n, 2)

  set.seed(10)
  edges <- combn_mat[, sample(1:ncol(combn_mat), floor(0.9*ncol(combn_mat)))]
  g <- igraph::graph.empty(n = n, directed = F)
  g <- igraph::add.edges(g, edges)

  queue <- .initialize_queue(n)
  hash_history <- hash::hash()
  clique_list <- lapply(igraph::maximal.cliques(g), function(x){sort(as.numeric(x))})
  hash_children <- .initialize_children(clique_list)
  hash_unique <- .inititalize_unique(clique_list, n)

  len <- length(clique_list)
  .add_to_queue(queue, len+1, sort(unique(unlist(clique_list[1:2]))), c(1,2),
                hash_children, hash_unique, hash_history, n)

  expect_true(length(queue) > len)
})
```

```
## .add_to_queue is correct

test_that ".add_to_queue works",
  n <- 10
  combn_mat <- combn(n, 2)

  set.seed(10)
  edges <- combn_mat[, sample(1:ncol(combn_mat), 10)]
  g <- igraph::graph.empty(n = n, directed = TRUE)
  g <- igraph::add.edges(g, edges)

  queue <- .initialize_queue(n)
  hash_history <- hash::hash()
  clique_list <- lapply(igraph::maximal.cliques(g), function(x){sort(as.numeric(x))})

  hash_children <- .initialize_children(clique_list)
  hash_unique <- .initialize_unique(clique_list, n)

  len <- length(clique_list)
  .add_to_queue(queue, len+1, sort(unique(unlist(clique_list[1:2]))), c(1,2),
                hash_children, hash_unique, hash_history, n)

  expect_true(length(queue) > len)
}
```

Sometimes “showing it works” requires showing that your outputs of functions can be accepted as inputs of other functions

Utility: This resembles how data is passed through your pipeline

What kind of unit tests could I write?

```
test_that(".pass_threshold can accept", {  
  adj_mat <- matrix(1, 10, 10)  
  res <- .pass_threshold(adj_mat, 0.95)  
  expect_true(res)  
})  
  
test_that(".pass_threshold can reject", {  
  set.seed(10)  
  adj_mat <- matrix(sample(c(0,1),100, replace = T), 10, 10)  
  adj_mat <- adj_mat * t(adj_mat)  
  
  res <- .pass_threshold(adj_mat, 0.95)  
  
  expect_true(!res)  
})
```

What kind of unit tests could I write?

```
test_that(".pass_threshold can accept", {  
    adj_mat <- matrix(rnorm(100), 10, 10)  
    res <- .pass_threshold(adj_mat, 0.95)  
    expect_true(res)  
})
```

```
test_that(".pass_threshold can reject", {  
    set.seed(10)  
    adj_mat <- matrix(sample(c(0,1), 100, replace = T), 10, 10)  
    adj_mat <- adj_mat * t(adj_mat)  
  
    res <- .pass_threshold(adj_mat, 0.95)  
  
    expect_true(!res)  
})
```

Making sure your function doesn't always return the same thing

Utility: Basic test to ensure that it's doing something meaningful

What kind of unit tests could I write?

```
test_that("clique_selection all pass threshold", {  
  trials <- 20  
  combn_mat <- combn(10, 2)  
  
  bool_vec <- rep(NA, trials)  
  for(i in 1:trials){  
    set.seed(20*i)  
    edges <- combn_mat[, sample(1:ncol(combn_mat), floor(0.9*ncol(combn_mat)))]  
  
    g <- igraph::graph.empty(n = 10, directed = F)  
    g <- igraph::add.edges(g, edges)  
  
    res <- clique_selection(g, threshold = 0.95)[[1]]  
  
    adj <- as.matrix(igraph::as_adjacency_matrix(g))  
    bool_vec[i] <- .pass_threshold(adj[res, res, drop = F], threshold = 0.95)  
  }  
  
  expect_true(all(bool_vec))  
})
```

What kind of unit tests could I write?

```
  _    _ pass threshold", {  
trials <- 20  
combn_mat <- combn(10, 2)  
  
bool_vec <- rep(NA, trials)  
for(i in 1:trials){  
  set.seed(20+i)  
  edges <- combn_mat[, sample(1:ncol(combn_mat), 2)]  
  
  g <- igraph::graph.empty(n = 10, directed = TRUE)  
  g <- igraph::add.edges(g, edges)  
  
  res <- clique_selection(g, threshold = 0.95)[[1]]  
  
  adj <- as.matrix(igraph::as_adjacency_matrix(g))  
  bool_vec[i] <- .pass_threshold(adj[res, res, drop = F], threshold = 0.95)  
}  
  
expect_true(all(bool_vec))
```

Simple simulation test to ensure “it works for many settings”

Utility: At least it’s better than testing just one specific case

What kind of unit tests could I write?

```
test_that(".convert_pair_to_string is reversible", {  
  vec <- c(27,67)  
  vec2 <- .convert_string_to_pair(.convert_pair_to_string(vec))  
  
  expect_true(all(vec == vec2))  
})
```

What kind of unit tests could I write?

```
test_that(".convert_pair_to_string is reversible", {  
  vec <- c(27,67)  
  vec2 <- .convert_string_to_pair(.convert_pair_to_string(vec))  
  
  expect_true(all(vec == vec2))  
})
```

Simple properties that should be true. A form of property testing

Utility: Basic checks of correctness

```
test_that(".gradient_vec satisfies the gradient definition", {
  trials <- 100

  bool_vec <- sapply(1:trials, function(x){
    set.seed(x)
    dat <- abs(matrix(rnorm(40), nrow = 10, ncol = 4))
    u_vec <- abs(rnorm(2))
    u_vec2 <- abs(rnorm(2))
    v_mat <- -abs(matrix(rnorm(8), nrow = 4, ncol = 2))

    i <- sample(1:10, 1)
    dat_vec <- dat[i,]
    class(dat_vec) <- c("exponential", class(dat_vec)[length(class(dat_vec))])
    grad <- .gradient_vec(dat_vec, u_vec, v_mat)

    res <- .evaluate_objective_single(dat_vec, u_vec, v_mat)
    res2 <- .evaluate_objective_single(dat_vec, u_vec2, v_mat)

    res2 >= res + as.numeric(grad %*% (u_vec2 - u_vec)) - 1e-6
  })

  expect_true(all(bool_vec))
})
```

```
test_that(".gradient_vec satisfies the gradient definition", {  
  trials = 100  
})
```

```
bool_vec <- sapply(1:trials, function(x){  
  set.seed(x)  
  dat <- abs(matrix(rnorm(40), nrow = 10, ncol = 4))  
  u_vec <- abs(rnorm(2))  
  u_vec2 <- abs(rnorm(2))  
  v_mat <- -abs(matrix(rnorm(8), nrow = 4, ncol = 2))  
  
  i <- sample(1:10, 1)  
  dat_vec <- dat[i,]  
  class(dat_vec) <- c("exponential", class(dat_vec))  
  grad <- .gradient_vec(dat_vec, u_vec, v_mat)  
  
  res <- .evaluate_objective_single(dat_vec, u_vec, v_mat)  
  res2 <- .evaluate_objective_single(dat_vec, u_vec2, v_mat)  
  
  res2 >= res + as.numeric(grad %*% (u_vec2 - u_vec)) - 1e-6  
})  
  
expect_true(all(bool_vec))  
})
```

Knowing theoretical properties that need to be satisfied are worth “rigorously” checking.

These can be math, optimization, computational, statistical theorems.

Utility: A **MUCH** stronger guarantee of correctness

What kind of unit tests could I write?

```
## .initialize_children is correct

test_that(".initialize_children works", {
  load("../assets/clique_selection1.RData")

  len <- 25
  edges <- combn(25, 2)[,lis[[1]]]
  g <- igraph::graph.empty(n = len, directed = F)
  g <- igraph::add_edges(g, edges)

  lis <- lapply(igraph::maximal.cliques(g), as.numeric)

  res <- .initialize_children(lis)

  expect_true(length(res) == length(lis))
  class_vec <- sapply(1:length(lis), function(x){
    class(res[[as.character(x)]]))
  })
  expect_true(all(class_vec == "node"))
})
```

What kind of unit tests could I write?

```
## .initialize_children is correct

test_that(". initialize_children works", {
  load("../assets/clique_selection1.RData")

  len <- 25
  edges <- combn(25, 2)[,lis[[1]]]
  g <- igraph::graph.empty(n = len, directed = TRUE)
  g <- igraph::add.edges(g, edges = edges)

  lis <- lapply(igraph::maximal.cliques(g), as.list)

  res <- .initialize_children(lis)

  expect_true(length(res) == length(lis))
  class_vec <- sapply(1:length(lis), function(x){
    class(res[[as.character(x)]])
  })
  expect_true(all(class_vec == "node"))
})
```

Aha! After using the code, I found that it unexpectedly crashed in a very particular configuration.

Utility: Make sure you can handle those “0.01% of the time” cases

Thank you!

View my GitHub page (username: linnylin92) to see specific details on how I set things up.

Email me if you have further questions.

Shameless plug: Follow me on twitter ([@linnylin92](https://twitter.com/linnylin92))
(or join me for zumba! M/Tues)

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾	selectiveModel / selectiveModel /	Create new file	Upload files	Find file	History
	linnylin92 Added a way for known sigma test to test something not a zero null mean			Latest commit df7222c on 26 Sep	
..					
📁 R	Added a way for known sigma test to test something not a zero null mean				a month ago
📁 man	Added a way for known sigma test to test something not a zero null mean				a month ago
📁 src	I think I fixed the bug				2 months ago
📁 tests	Added a way for known sigma test to test something not a zero null mean				a month ago
📄 .DS_Store	Adding the decluttering functions				a month ago
📄 .Rbuildignore	Was able to solve the jump s3 problem				2 months ago
📄 .extension	Modified the package to depend on Justin's bsfs				2 months ago
📄 DESCRIPTION	Fixed the documentation, added the capability to test one-sided				a month ago
📄 LICENSE	Nuked the default package				a year ago
📄 NAMESPACE	Modifying the clean version of master				2 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾ [selectiveModel](#) / [selectiveModel](#) /

Create new file Upload files Find file History

 linnynlin92 Added a way for known sigma test to test something not a zero null mean Latest commit df7222c on 26 Sep

..

 R

 src

 tests

 .DS_Store

 .Rbuildignore

 .extension

 DESCRIPTION

 LICENSE

 NAMESPACE

You can bundle other languages into an R package.

Especially useful if you want to write in lower-level for more than 100x boost

test to test something not a zero null mean a month ago

test to test something not a zero null mean a month ago

est to test something not a zero null mean a month ago

ns a month ago

problem a month ago

Modified the package to depend on Justin's bsfs 2 months ago

Fixed the documentation, added the capability to test one-sided a month ago

Nuked the default package a year ago

Modifying the clean version of master 2 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Screenshot of a GitHub repository page for `linnylin92 / selectiveModel`.

The repository has 1 unwatched star, 0 forks, and 3 issues.

The current branch is `master`. The commit history shows the following changes:

- linnylin92 I think I fixed the bug - Latest commit 4234578 on 8 Sep
- ..
- `.DS_Store` Removed the rejection sampler - 2 months ago
- `RcppExports.cpp` Added tests for `intersect_basis` - 2 months ago
- `interval.cpp` Fixed a bug that didn't respect having only one root of quadratic for... - 2 months ago
- `interval.h` Tested the way the intervals are formed - 2 months ago
- `plane.cpp` Realized a bug where I need to worry about NAs for `intersect_basis` - 2 months ago
- `plane.h` Added `basic_interval` functions - 2 months ago
- `radians.cpp` I think I fixed the bug - 2 months ago
- `radians.h` Added `point_on_plane` function - 2 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

linnylin92 / selectiveModel

Unwatch 1 Star 0 Fork 0

Code Issues 3 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master selectiveModel / selectiveModel / src / Create new file Upload files Find file History

linnylin92 I think I fixed the bug Latest commit 4234578 on 8 Sep

..

.DS_Store	Removed the rejection sampler	2 months ago
RcppExports.cpp	Added tests for intersect_basis	2 months ago
interval.cpp	Fixed a bug that didn't respect having only one root of quadratic for...	2 months ago
interval.h	Tested the way the intervals are formed	2 months ago
plane.cpp	about NAs for intersect_basis	2 months ago
plane.h		2 months ago
radians.cpp		2 months ago
radians.h	Added point_on_plane function	2 months ago

In this case, C++ files and their corresponding headers are here

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾

[selectiveModel](#) / [selectiveModel](#) / [src](#) / [radians.cpp](#)

[Find file](#) [Copy path](#)

 linnylin92 I think I fixed the bug

4234578 on 8 Sep

1 contributor

169 lines (128 sloc) | 4.04 KB

[Raw](#) [Blame](#) [History](#)   

```
1 #include "radians.h"
2
3 // [[Rcpp::export]]
4 Rcpp::NumericVector c_unique_sort_native(const Rcpp::NumericVector & x) {
5     Rcpp::NumericVector res = Rcpp::clone(x);
6     res.sort(false);
7
8     Rcpp::NumericVector::iterator it;
9     it = std::unique(res.begin(), res.end());
10
11    res.erase(it, res.end());
12
13    return res;
14 }
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾ [selectiveModel](#) / [selectiveModel](#) / sr / **radians.cpp** Find file Copy path

linnylin92 I think I fixed the bug 4234578 on 8 Sep

1 contributor

Raw Blame History

```
169 170 // (120 - 1) * 1.04 KB
171
172 #include "radians.h"
173
174 // [[Rcpp::export]]
175 Rcpp::NumericVector unique_
176 {
177     Rcpp::NumericVector res = R
178     res.sort(false);
179
180     Rcpp::NumericVector::iterator
181     it = std::unique (res.begin(),
182                      res.end());
183
184     res.erase(it, res.end());
185
186     return res;
187 }
```

The `[[Rcpp::export]]` function tells R Studio that it should automatically generate relevant files so your package will compile properly

(More on this later)

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

linnylin92 / selectiveModel

Code Issues 3 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master [selectiveModel](#) / [selectiveModel](#) / src / radians.h Find file Copy path

linnylin92 Added point_on_plane function 6e5db3c on 19 Aug

1 contributor

16 lines (12 sloc) | 659 Bytes Raw Blame History

```
1 #ifndef _RADIAN_H
2 #define _RADIAN_H
3
4 #include <Rcpp.h>
5
6 Rcpp::NumericVector c_unique_sort_native(const Rcpp::NumericVector & x);
7 Rcpp::NumericVector c_construct_midpoints(const Rcpp::NumericVector & x);
8 Rcpp::IntegerVector c_which_native(const Rcpp::LogicalVector & x);
9 Rcpp::IntegerMatrix c_consecutive_true(const Rcpp::LogicalVector & vec);
10 Rcpp::NumericVector c_unlist_native(const Rcpp::List & list);
11 Rcpp::LogicalVector c_theta_in_matrix(const double & x, const Rcpp::NumericMatrix & mat);
12 Rcpp::LogicalVector c_theta_in_all_matrix(const double & x, const Rcpp::List & list);
13 Rcpp::NumericMatrix c_intersect_intervals(const Rcpp::List & list);
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

linnylin92 / selectiveModel

Code Issues 3 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master **selectiveModel / selectiveModel / src / radians.h** Find file Copy path

linnylin92 Added point_on_plane function 6e5db3c on 19 Aug

1 contributor

16 lines (12 sloc) | 659 Bytes

```
1 #ifndef _RADIAN_H
2 #define _RADIAN_H
3
4 #include <Rcpp.h>
5
6 Rcpp::NumericVector c_unique_sort_native(const Rcpp::NumericVector & x);
7
8 Rcpp::IntegerVector c_which_native(const Rcpp::LogicalVector & x);
9 Rcpp::IntegerMatrix c_consecutive_true(const Rcpp::LogicalVector & vec);
10 Rcpp::NumericVector c_unlist_native(const Rcpp::List & list);
11 Rcpp::LogicalVector c_theta_in_matrix(const double & x, const Rcpp::NumericMatrix & mat);
12 Rcpp::LogicalVector c_theta_in_all_matrix(const double & x, const Rcpp::List & list);
13 Rcpp::NumericMatrix c_intersect_intervals(const Rcpp::List & list);
```

Remember, C++ is a “real” programming language.

You need to declare the inputs and outputs of a function, and variables can be passed by pointers.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Screenshot of a GitHub repository page for `linnylin92 / selectiveModel`.

The repository has 1 unwatched star, 0 forks, and 3 issues.

The current branch is `master`. The commit history shows the following changes:

- linnylin92 I think I fixed the bug - Latest commit 4234578 on 8 Sep
- ..
- `.DS_Store` Removed the rejection sampler - 2 months ago
- `RcppExports.cpp` Added tests for `intersect_basis` - 2 months ago
- `interval.cpp` Fixed a bug that didn't respect having only one root of quadratic for... - 2 months ago
- `interval.h` Tested the way the intervals are formed - 2 months ago
- `plane.cpp` Realized a bug where I need to worry about NAs for `intersect_basis` - 2 months ago
- `plane.h` Added `basic_interval` functions - 2 months ago
- `radians.cpp` I think I fixed the bug - 2 months ago
- `radians.h` Added `point_on_plane` function - 2 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

The screenshot shows a GitHub repository page for 'linnylin92 / selectiveModel'. The repository path is 'selectiveModel / selectiveModel / src /'. The commit history is visible, with the latest commit being '4234578 on 8 Sep'. A red box highlights the file 'RcppExports.cpp' in the list of files. A large red box contains the following explanatory text:

These are generated automatically. These tell R that there C++ functions and tell the corresponding R functions “where to point to” to find the compiled C++ function counterparts.

File	Commit Message	Time Ago
RcppExports.cpp	I think I fixed the bug	2 months ago
interval.cpp		2 months ago
interval.h		2 months ago
plane.cpp		2 months ago
plane.h		2 months ago
radians.cpp	I think I fixed the bug	2 months ago
radians.h	Added point_on_plane function	2 months ago

Thoughtful organization, Git, and R packages can solve most of our previous problems. (C++ Code)

Branch: master ▾ [selectiveModel](#) / [selectiveModel](#) / [src](#) / [RcppExports.cpp](#) [Find file](#) [Copy path](#)

 linnylin92 Added tests for intersect_basis 72e8839 on 29 Aug

1 contributor

288 lines (281 sloc) | 13.8 KB [Raw](#) [Blame](#) [History](#)   

```
1 // Generated by using Rcpp::compileAttributes() -> do not edit by hand
2 // Generator token: 10BE3573-1514-4C36-9D1C-5A225CD40393
3
4 #include <Rcpp.h>
5
6 using namespace Rcpp;
7
8 // c_initial_theta
9 double c_initial_theta(const Rcpp::NumericVector& y, const Rcpp::NumericVector& v, const Rcpp::NumericVector& w);
10 RcppExport SEXP _selectiveModel_c_initial_theta(SEXP ySEXP, SEXP vSEXP, SEXP wSEXP) {
11 BEGIN_RCPP
12     Rcpp::RObject rcpp_result_gen;
13     Rcpp::RNGScope rcpp_rngScope_gen;
14     Rcpp::traits::input_parameter< const Rcpp::NumericVector& >::type y(ySEXP);
15     Rcpp::traits::input_parameter< const Rcpp::NumericVector& >::type v(vSEXP);
16     Rcpp::traits::input_parameter< const Rcpp::NumericVector& >::type w(wSEXP);
17     rcpp_result_gen = Rcpp::wrap(c_initial_theta(y, v, w));
18     return rcpp_result_gen;
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (C++ Code)

Branch: master ▾ [selectiveModel](#) / selectiveModel / src **RcppExports.cpp** Find file Copy path

linnylin92 Added tests for intersect_basis 72e8839 on 29 Aug

1 contributor

288 lines (281 sloc) | 13.9 KB

Raw Blame History

```
// Generated by using Rcpp::compileAttributes() -> do not edit by hand
// Generator token: 10BE3573-1514-4C36-9D1C-5A225CD40393
```

#include <Rcpp.h>

using namespace Rcpp;

// c_initial_theta

double c_initial_theta(const Rcpp::NumericVector& y, const Rcpp::NumericVector& v, const Rcpp::NumericVector& w);

RcppExport SEXP _selectiveModel_c_initial_theta(SEXP ySEXP, SEXP vSEXP, SEXP wSEXP) {

BEGIN_RCPP

Rcpp::RObject rcpp_result_gen;

Rcpp::RNGScope rcpp_rngScope_gen;

Rcpp::traits::input_parameter< const Rcpp::NumericVector& >::type y(ySEXP);

Rcpp::traits::input_parameter< const Rcpp::NumericVector& >::type v(vSEXP);

Rcpp::traits::input_parameter< const Rcpp::NumericVector& >::type w(wSEXP);

rcpp_result_gen = Rcpp::wrap(c_initial_theta(y, v, w));

return rcpp_result_gen;

I wouldn't touch this file at all if I were you.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾ [selectiveModel](#) / [selectiveModel](#) / R / [RcppExports.R](#)

[Find file](#) [Copy path](#)

 linnlylin92 Clean push

61ae91d on 30 Aug

1 contributor

84 lines (62 sloc) | 2.31 KB

[Raw](#) [Blame](#) [History](#)   

```
1 # Generated by using Rcpp::compileAttributes() -> do not edit by hand
2 # Generator token: 10BE3573-1514-4C36-9D1C-5A225CD40393
3
4 .c_initial_theta <- function(y, v, w) {
5     .Call(`_selectiveModel_c_initial_theta`, y, v, w)
6 }
7
8 .c_basic_interval <- function(endpoints, theta) {
9     .Call(`_selectiveModel_c_basic_interval`, endpoints, theta)
10 }
11
12 .c_partition_interval <- function(interval) {
13     .Call(`_selectiveModel_c_partition_interval`, interval)
14 }
15
16 .c_interval <- function(endpoints, initial_theta) {
17     .Call(`_selectiveModel_c_interval`, endpoints, initial_theta)
18 }
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾ [selectiveModel](#) / selectiveModel / R / RcppExports.R

Find file Copy path

linnylin92 Clean push 61ae91d on 30 Aug

1 contributor

Raw Blame History

```
84 1
1 # Generated by using Rcpp::compileAttributes() -> do not edit by hand
2 # Generator token: 10BE3573-1514-4C36-9D1C-5A225CD40393
3
4 .c_initial_theta <- function(y, v, w) {
5   .Call(`_selectiveModel_c_initial_theta`, y, v, w)
6 }
7
8 .c_basic_interval <- function(endpoints, theta) {
9   .Call(`_selectiveModel_c_basic_interval`, endpoints, theta)
10}
11
12 .c_partition_interval <- function(interval) {
13   .Call(`_selectiveModel_c_partition_interval`, interval)
14}
15
16 .c_interval <- function(endpoints, initial_theta) {
17   .Call(`_selectiveModel_c_interval`, endpoints, initial_theta)
18}
```

There's an analogous R file. This one generates the R interface, one for each C++ function you exported.

Also wouldn't touch this if I were you.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾ [selectiveModel](#) / [selectiveModel](#) / **DESCRIPTION** [Find file](#) [Copy path](#)

 linnylin92 Fixed the documentation, added the capability to test one-sided 89dc564 on 24 Sep

1 contributor

25 lines (24 sloc) | 503 Bytes [Raw](#) [Blame](#) [History](#)   

```
1 Package: selectiveModel
2 Title: Selective Model Inference
3 Version: 0.0.0.9000
4 Authors@R: person("First", "Last", email = "first.last@example.com", role = c("aut", "cre"))
5 Description: What the package does (one paragraph).
6 Depends:
7   R (>= 3.3.0),
8   Rcpp (>= 0.12.18)
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

Branch: master ▾ [selectiveModel](#) / [selectiveModel](#) / **DESCRIPTION** Find file Copy path

 linnylin92 Fixed the documentation, added the capability to test one-sided 89dc564 on 24 Sep

1 contributor

25 lines (24 sloc) | 503 Bytes Raw Blame History

```
1 Package: selectiveModel
2 Title: Selective Model Inference
3 Version: 0.0.0.9000
4 Authors@R: person("First", "Last", email = "first.last@example.com", role = c("aut", "cre"))
Description: what the package does (one paragraph).
Depends:
  R (>= 3.3.0),
  Rcpp (>= 0.12.18)
```

You need to tell R that it needs to check to see that Rcpp is installed when loading your package.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

linnylin92 / selectiveModel

Code Issues 3 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master → selectiveModel / selectiveModel / NAMESPACE Find file Copy path

linnylin92 Modifying the clean version of master 648d17b on 30 Aug

1 contributor

5 lines (4 sloc) | 115 Bytes Raw Blame History

```
1 useDynLib(selectiveModel, .registration=TRUE)
2 exportPattern("^[[:alpha:]]+")
3 import(Rcpp)
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**C++ Code**)

The screenshot shows a GitHub repository page for 'linnylin92 / selectiveModel'. The 'Code' tab is selected. A red box highlights the 'NAMESPACE' file in the breadcrumb navigation. Another red box highlights the code content within the 'NAMESPACE' file.

linnylin92 / selectiveModel

Code Issues 3 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master **selectiveModel / selectiveModel / NAMESPACE**

Find file Copy path

linnylin92 Modifying the clean version of master 648d17b on 30 Aug

1 contributor

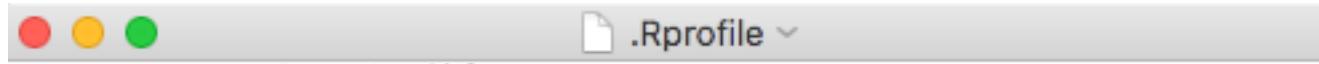
5 lines (4 sloc) | 115 Bytes

Raw Blame History

```
useDynLib(selectiveModel, .registration=TRUE)
exportPattern("^[[:alpha:]]+")
import(Rcpp)
```

You also need to tell R the way it should link the C++ functions to the R session once the package is loaded.

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)



The screenshot shows an RStudio interface with a code editor window titled ".Rprofile". The window contains R code for managing package documentation and testing. The code includes functions for checking files, loading data, and testing packages.

```
.CHECK_ALL <- function(){
  files <- list.files("man", full.names = T)
  file.remove(files)
  file.remove("NAMESPACE")
  devtools::document()
  devtools::check(document = F)

  invisible()
}

.LOAD_ALL <- function(nuke = FALSE){
  if(nuke) rm(list = ls(globalenv()), envir = globalenv())
  library(devtools)
  devtools::load_all()
  x <- list.files("data/raw", pattern = "*source*.R$", full.names = T)
  if(length(x) > 0){
    for(i in x){source(i)}
  }

  invisible()
}

.TEST_ALL <- function(){
  devtools::test()

  invisible()
}

options(stringsAsFactors=FALSE)
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

☰ README.md

[build](#) passing [codecov](#) 98%

Warning: On windows, to install the "doMC" package, use `install.packages("doMC", repos="http://R-Forge.R-project.org")` in R. See: <http://stackoverflow.com/questions/16453625/package-domc-not-available-for-r-version-3-0-0-warning-in-install-packages>

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

Branch: master ▾ [longitudinalGM / .travis.yml](#) Find file Copy path

 linnylin92 I think I fixed travis to work with bioconductor 865be15 on 28 Jun

1 contributor

25 lines (20 sloc) | 390 Bytes

Raw Blame History

```
language: r
before_install:
- Rscript -e 'source("https://bioconductor.org/biocLite.R"); biocLite("biomaRt")'
- cd longitudinalGM
cache: packages
r_binary_packages:
- igraph
- glmnet
- Matrix
- MASS
- huge
- plyr
- devtools
- foreach
- doMC
- covr
- hash
after_success:
- Rscript -e 'library(covr); codecov()'
warnings_are_errors: true
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

[linnylin92 / longitudinalGM](#)  build passing

Current

Branches

Build History

Pull Requests

More options



✓ master Made figure 12

→ #259 passed

 Restart build

→ Commit 062e0b8 ↗

⌚ Ran for 10 min 56 sec

↳ Compare fbc04d4..062e0b8 ↗

🕒 a day ago

↳ Branch master ↗

⌚ klin1234@gmail.com authored and committed

[Job log](#)

[View config](#)

X Remove log

↓ Raw log

```
▶ 1 Worker information
▶ 6 Build system information
402
403 removed '/etc/apt/sources.list.d/basho_riak.list'
404 Executing: /tmp/tmp.jwc3SCdDsS/gpg.1.sh --keyserver
405 hkp://keyserver.ubuntu.com:80
406 --recv
407 EA312927
408 gpg: requesting key EA312927 from hkp server keyserver.ubuntu.com
409 gpg: key EA312927: "MongoDB 3.2 Release Signing Key <packaging@mongodb.com>" 1 new signature
410 gpg: Total number processed: 1
411 gpg:      new signatures: 1
412 W: http://dl.hhvm.com/ubuntu/dists/trusty/InRelease: Signature by key
36AEF64D0207E7EEE352D4875A16E7281BE7A449 uses weak digest algorithm (SHA1)
413 W: http://ppa.launchpad.net/couchdb/stable/ubuntu/dists/trusty/Release.gpg: Signature by key
158668AFD9BCC4F3C1E0DFC7D69548E1C17EAB57 uses weak digest algorithm (SHA1)
414 R for Travis-CI is not officially supported, but is community maintained.
415 Please file any issues at https://github.com/travis-ci/travis-ci/issues
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

Branch: master ▾ [longitudinalGM](#) / [.travis.yml](#) Find file Copy path

 linnylin92 I think I fixed travis to work with bioconductor 865be15 on 28 Jun

1 contributor

25 lines (20 sloc) | 390 Bytes Raw Blame History

```
language: r
before_install:
- Rscript -e 'source("https://bioconductor.org/biocLite.R"); biocLite("biomaRt")'
- cd longitudinalGM
cache: packages
r_binary_packages:
- igraph
- glmnet
- Matrix
- MASS
- huge
- plyr
- devtools
- foreach
- doMC
- covr
- hash
after_success:
- Rscript -e 'library(covr); codecov()'
warnings_are_errors: true
```

Thoughtful organization, Git, and R packages can solve most of our previous problems. (**Testing**)

