R Notebook Systems

Calla Martyn, Matt Johnson, and Miriam Goldman

Make pateint zscores for each person

Read in data

```
# find all the count table files
files <- list.files(pattern = "*.htseq.counts.gz", recursive = TRUE)
# read count tables into a list of tables
datalist <- lapply(files, function(x){read.table(file=x,header=FALSE,
                                                 col.names=c("gene", sub(".htseq.counts.gz", "",x)))})
# merge the individual count tables into a dataframe
m <- Reduce(function(...) merge(..., by=1, all = TRUE), datalist)
rownames(m) <- m[,1]
# get rid of the first few rows, they are summaries of the count tables
m <- m[6:nrow(m),-1]

# convert counts to z-score
m_scaled <- as.data.frame(t(scale(t(m))))

# read in the gene expression signature
signature9 <- read.csv('./sc_signatures/bxpc3_leiden9_logfoldchangeGT50pct_genes.csv',
                       stringsAsFactors = FALSE)
signature1 <- read.csv('./sc_signatures/bxpc3_leiden1_logfoldchangeGT50pct_genes.csv',
                       stringsAsFactors = FALSE)
express_table <- read.csv('./sc_signatures/BXPC3_pvals_monacle.csv',
                          stringsAsFactors = FALSE)
```

Trun into zscores and write out to pdata

```
# import expression table just to use for translating ensemble ids to gene names
gene_dict <- express_table %>% select(gene_ids, gene_short_name)

# strip suffix (isoform) from ensembl IDs
m_scaled$gene_ids <- unlist(lapply(rownames(m_scaled),
                                   function(x) unlist(strsplit(x,'\\.'))[[1]]))
# join genedict table with expression table by ensemble IDs, table now has gene name column
m_gene_names <- join(m_scaled, gene_dict, by = "gene_ids", type = "inner")
# change rownames from ensembl IDs to gene names for easier selection
rownames(m_gene_names) <- m_gene_names$gene_short_name
# select only the rows matching the gene signature rows
m_sig1 <- m_gene_names[signature1$names,]
m_sig9 <- m_gene_names[signature9$names,]

write.csv(m_sig1, './pData/clin_zscores_sig1.csv')
write.csv(m_sig9, './pData/clin_zscores_sig9.csv')
```

Make pateint metadata

All can be downloaded from http://www.cbioportal.org/study?id=paad_tcga&tab=clinicalData

```
import gzip
import csv
import io
import glob
import os
import pandas as pd
import numpy as np
files = glob.glob("./panc_expression/**/*.gz")
metalist = glob.glob("./panc_expression/**/*")
os.path.basename("./gdc_sample_sheet.2019-03-05.tsv")
name2tcga = "./gdc_sample_sheet.2019-03-05.tsv"
tcga = []
dirname=os.path.dirname
with open(name2tcga, 'r') as csvfile:
    spamreader = csv.reader(csvfile, delimiter='\t')
    for row in spamreader:
        for meta in metalist:
            if os.path.basename(meta) == row[1]:
                tcga.append([row[6], os.path.basename(dirname(meta))])
tcga2patientdata = "./paad_tcga_clinical_data.tsv"
tcga2patient = []
with open(tcga2patientdata, 'r') as csvfile:
    spamreader = csv.reader(csvfile, delimiter='\t')
    for row in spamreader:
        for tc in tcga:
            if tc[0][:-1] == row[2]:
                tmp = [tc[1], row[29], row[28], row[86]]
                #Disease Free (Months)  Disease \t Free Status \t sex
                tcga2patient.append(tmp)
df = pd.DataFrame(tcga2patient)
len(np.unique(df[2]))
df.to_csv("patient_metadata.tsv", sep='\t')
```

Read in data from above

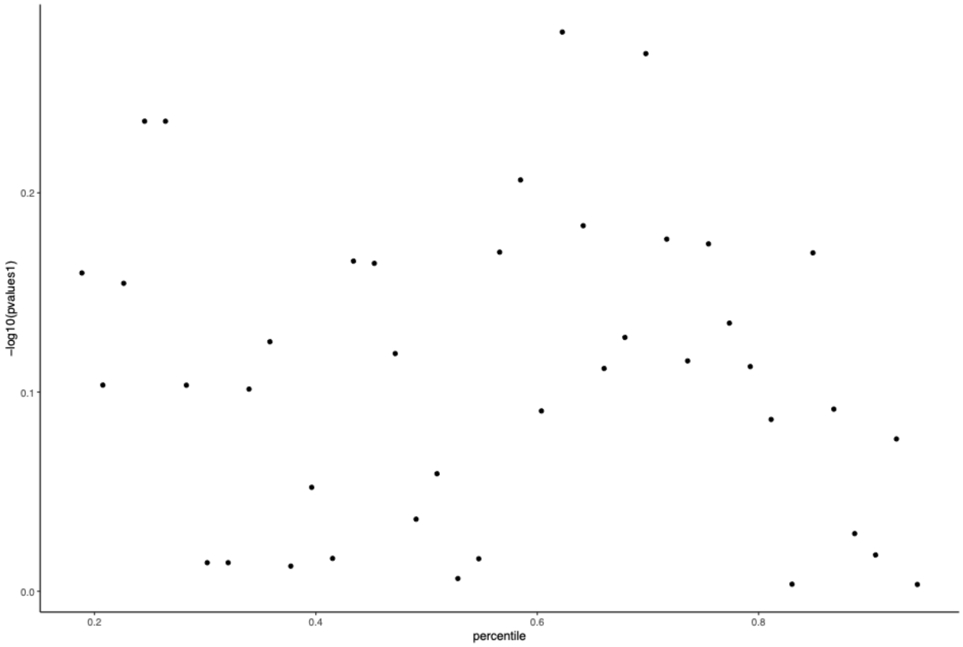
```
output<-read.csv('./pData/patient_metadata.tsv',sep='\t')
signature9<-read.csv('./sc_signatures/bxpc3_leiden9_logfoldchangeGT50pct_genes.csv')
zscore9<-read.csv('./pData/clin_zscores_sig9.csv')
signature1<-read.csv('./sc_signatures/bxpc3_leiden1_logfoldchangeGT50pct_genes.csv')
zscore1<-read.csv('./pData/clin_zscores_sig1.csv')
```

Survival Analysis

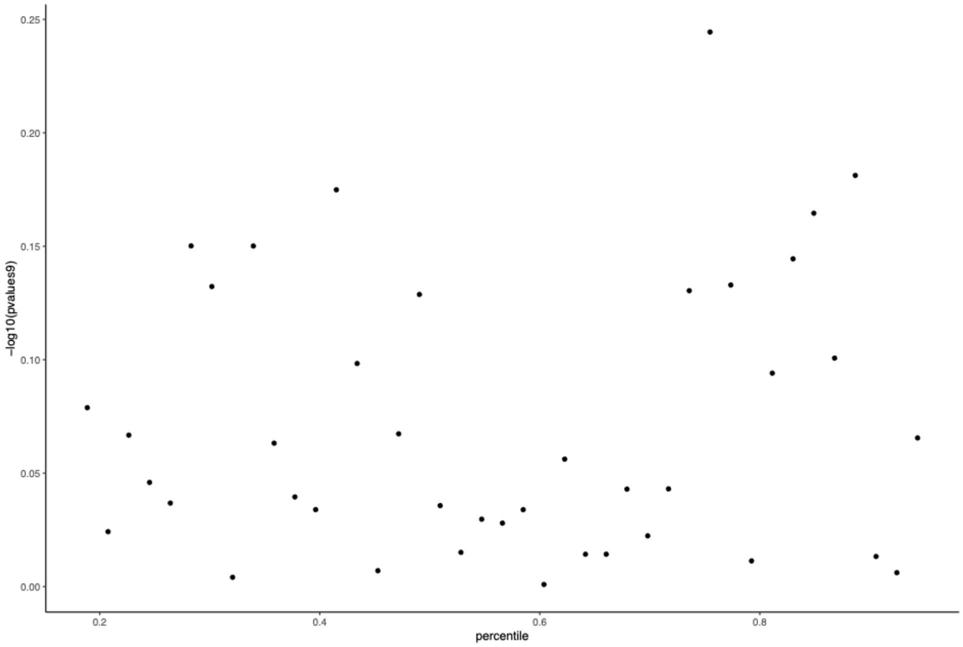
```
zscore9<-inner_join(zscore9,signature9,by=c("gene_short_name"="names"))
p_9<-zscore9 %>% select(contains("count_table")) #pull out patinet vectors
```

```
cors9<-apply(p_9,2,function(y) cor(zscore9$scores,as.numeric(y)))
#turn all columns numeric and get the corralation between the two
cors9<-data.frame("X0"=colnames(p_9),cors9)
# set X0 to be patient names
output$y<-ifelse(output$X1=="DiseaseFree",0,1)
# if Disease Free 0 else 1
output<-output %>% filter(!is.na(X1))
# filter out nas
#combine on directory name
output$dir_name<-unlist(lapply(output$X0,function(x) unlist(str_split(x,'-'))[5]))
cors9$dir_name<-unlist(lapply(cors9$X0,function(x) unlist(str_split(x,'\\.'))[6]))


corrlation9<-inner_join(cors9,output,by="dir_name")
cox<-coxph(Surv(corrlation9$X2,corrlation9$y)~corrlation9$cors9)
cox_fix<-survfit(cox)

cors1<-inner_join(zscore1,signature1,by=c("gene_short_name"="names"))
p_1<-cors1%>% select(contains("count_table"))

cors1<-apply(p_1,2,function(y) cor(cors1$scores,as.numeric(y)))
cors1<-data.frame("X0"=colnames(p_1),cors1)
cors1$dir_name<-unlist(lapply(cors1$X0,function(x) unlist(str_split(x,'\\.'))[6]))
corrlation1<-inner_join(cors1,output,by="dir_name")
cox1<-coxph(Surv(corrlation1$X2,corrlation1$y)~corrlation1$cors)
cox_fix1<-survfit(cox1)
```

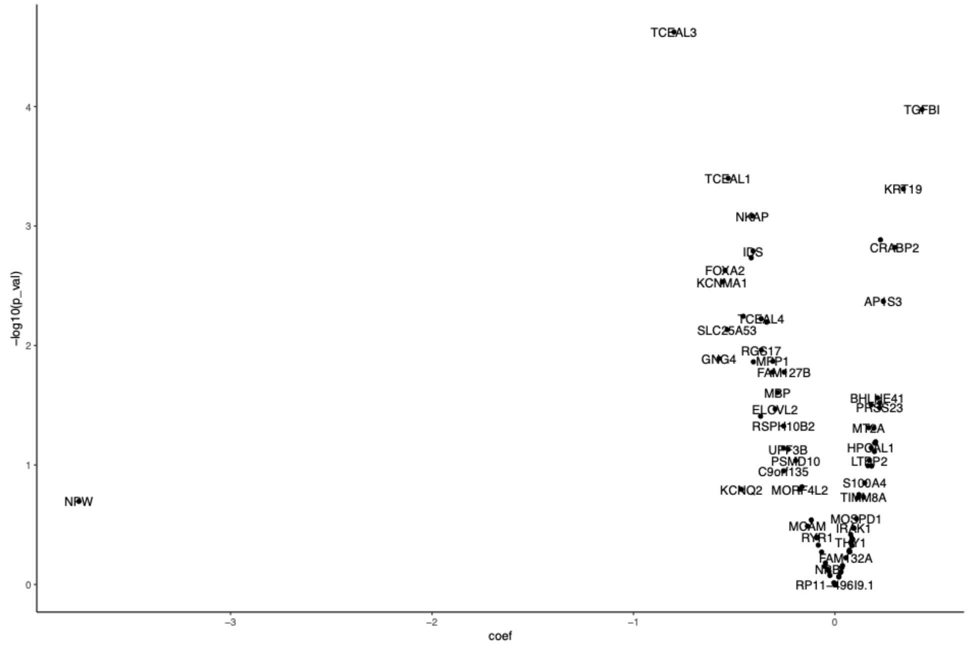
```
pvalues1<-unlist(lapply(sort(corrlation1$cors1)[10:50],
function(x)
  summary(coxph(Surv(corrlation1$X2,corrlation1$y)~ifelse(corrlation1$cors1>x,1,0)))$coef[5]))
ps<-data.frame(percentile=10:50/53,pvalues1,cors=sort(corrlation1$cors1)[10:50])
ggplot(ps) + geom_point(aes(percentile,-log10(pvalues1)))
```



```
pvalues9<-unlist(lapply(sort(corrlation9$cors9)[10:50],function(x)
  summary(coxph(Surv(corrlation9$X2,corrlation9$y)~ifelse(corrlation9$cors9>x,1,0)))$coef[5]))
ps9<-data.frame(percentile=10:50/53,pvalues9,cors=corrlation9$cors9[10:50])
ggplot(ps9) + geom_point(aes(percentile,-log10(pvalues9)))
```

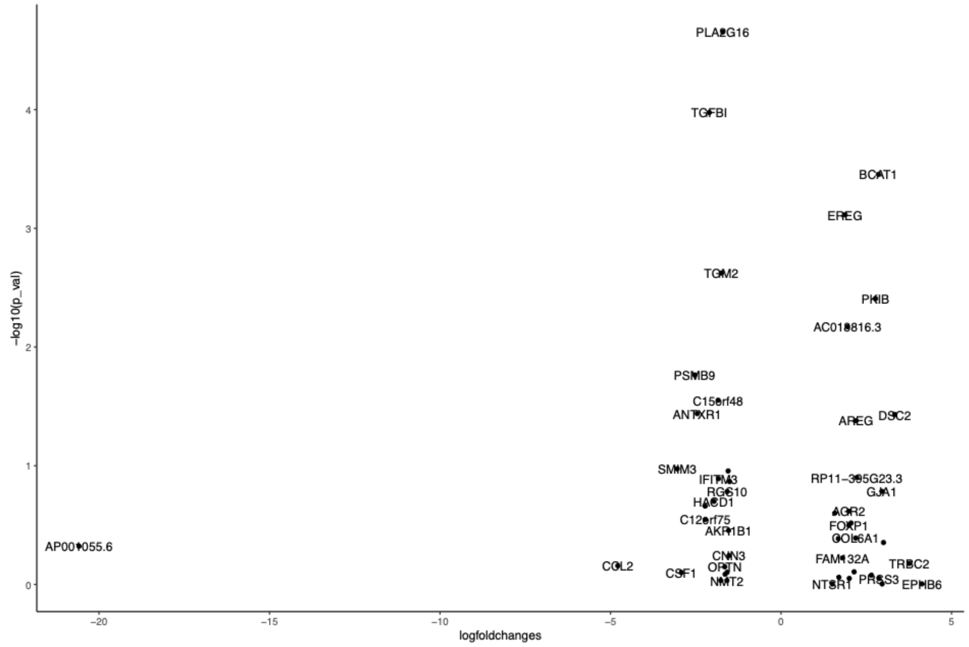


```
gene_9<-p_9 %>% t() %>% as.data.frame()
colnames(gene_9)<-signature9$names
gene_9$dir_name<-unlist(lapply(rownames(gene_9),function(x) unlist(str_split(x,'\\.'))[6]))
genes9<-inner_join(gene_9,output,by="dir_name") %>% select(-dir_name,-X,-X0,-X1)
gene_p_9<-apply(genes9[,1:83],2,function(x)
  summary(coxph(Surv(genes9$X2,genes9$y)~x))$coef[5]) %>% data.frame()
gene_coef_9<-apply(genes9[,1:83],2,function(x)
  summary(coxph(Surv(genes9$X2,genes9$y)~x))$coef[1]) %>% data.frame()
gene_p_9$names<-rownames(gene_p_9)
colnames(gene_p_9)<-c("p_val","names")
gene_coef_9$names<-rownames(gene_coef_9)
colnames(gene_coef_9)<-c("coef","names")

surv_genes_9<-inner_join(gene_p_9,signature9,by="names")
surv_genes_9<-inner_join(gene_coef_9,surv_genes_9,by="names")
ggplot(surv_genes_9,aes(coef,-log10(p_val),label=names))+geom_point()+geom_text(check_overlap = TRUE)
```

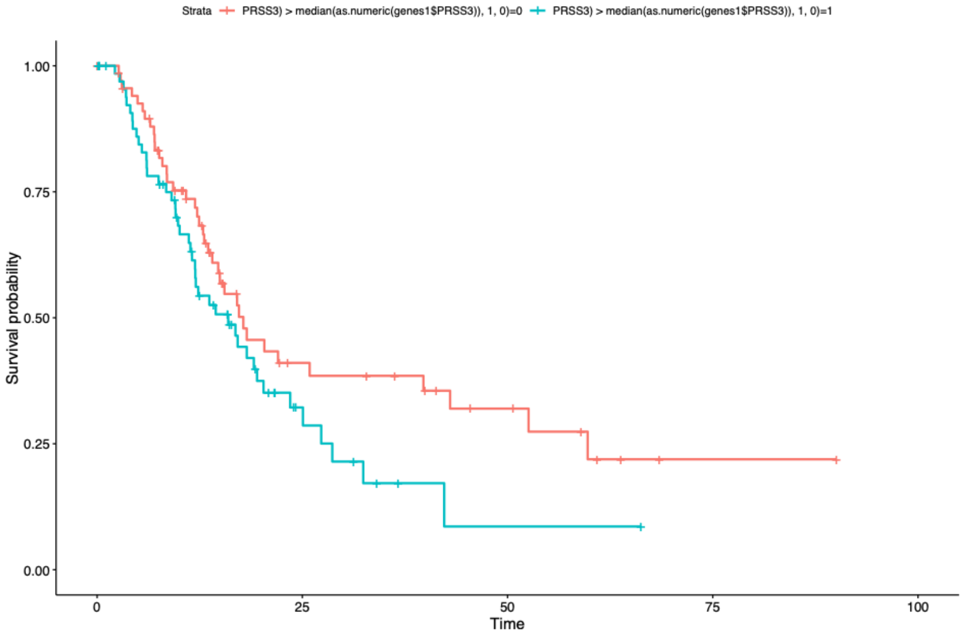


```
gene_1<-p_1 %>% t() %>% as.data.frame()
colnames(gene_1)<-signature1$names
gene_1$dir_name<-unlist(lapply(rownames(gene_1)
                               ,function(x) unlist(str_split(x,'\\.'))[6]))
genes1<-inner_join(gene_1,output,by="dir_name") %>% select(-dir_name,-X,-X0,-X1)
gene_p_1<-apply(genes1[,1:49],2,function(x) summary(coxph(Surv(genes1$X2,genes1$y)~as.numeric(x)))$coef
gene_coef_1<-apply(genes1[,1:49],2,
                   function(x) summary(coxph(Surv(genes1$X2,genes1$y)~x))$coef[1]) %>% data.frame()

gene_p_1$names<-rownames(gene_p_1)
colnames(gene_p_1)<-c("p_val","names")

gene_coef_1$names<-rownames(gene_coef_1)
colnames(gene_coef_1)<-c("coef","names")

surv_genes_1<-inner_join(gene_p_1,signature1,by="names")
surv_genes_1<-inner_join(gene_coef_1,surv_genes_1,by="names")
```

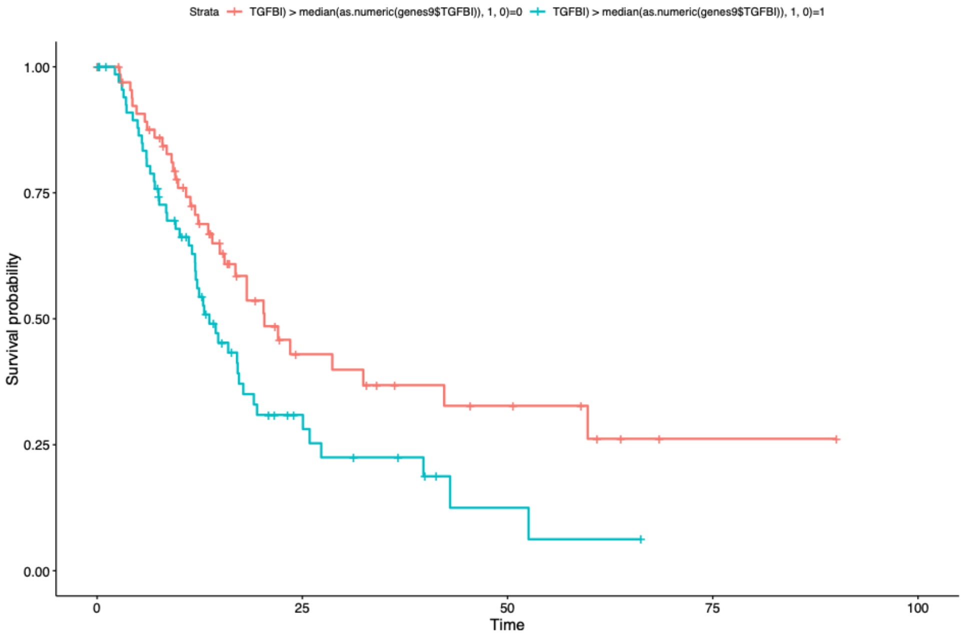
```
ggplot(surv_genes_1,aes(logfoldchanges,-log10(p_val),label=names))+geom_point()+geom_text(check_overlap
```



```r
ggsurvplot(survfit(Surv(genes1$X2,genes1$y)~
                      ifelse(as.numeric(genes1$PRSS3)>median(as.numeric(genes1$PRSS3)),1,0),
                   data=genes1))
```



```
ggsurvplot(survfit(Surv(genes9$X2,genes9$y)~
                     ifelse(as.numeric(genes9$TGFBI)>median(as.numeric(genes9$TGFBI)),1,0),
                   data=genes9))
```


Strata
TGFBI) > median(as.numeric(genes9$TGFBI)), 1, 0)=0
TGFBI) > median(as.numeric(genes9$TGFBI)), 1, 0)=1
Survival probability
1.00
0.75
0.50
0.25
0.00
0
25
50
75
100
Time