Research Progress Report

Xueyi Fan

July 20, 2016

Predicting Biomarker Genes of Pancreatic Ductal Adenocarcinoma (PDAC)

Abstract

Pancreatic Ductal Adenocarcinoma (PDAC) is an aggressive cancer and all cancerous tumors are made up of a combination of normal and cancerous cells. Although some previous work has found some biomarker candidates, few of them are legally introduced into clinical practice. In this research, I accessed two GEO datasets (GSE15471 and GSE71989) containing gene expression data of PDAC common and tumor cells and combined them using meta-analysis method. I identified 1249 genes significantly differentially expressed in tumor cells. 715 genes are up-regulated and 534 genes are down-regulated. Further fold-50 cross-validation analysis based on SVM-RFE identified 45 genes with the best diagnostic accuracy to distinguish the normal and tumor cells.

Introduction

Pancreatic cancer starts from the pancreas and its tumor cells perform a strong invasive ability. Pancreatic Ductal Adenocarcinoma (PDAC) is the most common

subtype of pancreatic cancer which is the fourth most common cause of global cancer-related deaths[1]. The overall 5-years survival rate is 5% and the localized, potential curable tumors can only be found in less than 20% PDAC patients[2]. PDAC tumors are complex mixtures of normal cell and low proportion of cancerous cells which hampers the analysis of PDAC. It is an urgent requirement to find predictive, prognostic, diagnostic biomarkers. A biomarker is a distinct substance produced by a tumor that helps doctors diagnose cancer and determine how a patient will respond to different kinds of treatment. They can indicate the presence, severity, or type of cancer[3]. For example, 50% to 60% of people with melanoma are found BRAF mutated and NRAS mutation cause about 20% of melanoma. A number of gene expression signatures can be used to estimate prognosis in breast cancer[4]. Extensive studies have been conducted to identify biomarkers of PDAC. In 2011, Nigel conformed REMARK (Reporting recommendations for tumor MARKer prognostic studies) criteria and predicting eight tissue biomarkers associated with PDAC[5]. And Peng's lab used meta-analysis and found candidate microRNA biomarkers of PDCA[6]. Although many markers have been routinely introduced into clinical practice, the only biomarker approved by FDA is CA 19-9 with several limitations[7].

Microarrays have developed as a popular tool for comparing gene expression profile in a high throughput. Many data from different studies are collected and available in Gene Expression Omnibus (GEO). However, there is a limitation of individual study that they often come out with opposite conclusions because of the effect of small sample size and data analysis method. Meta-analysis is an ideal approach to overcome this problem by combining multiple studies applying

identical statistical analysis in order to obtain a more precise estimate and more reliable results[8].

Machine learning is a subfield of computer science and is used to devise complex models and algorithms that lend themselves to prediction. Machine learning methods have been applied to a broad range of areas of omics. Support Vector Machines (SVMs) is a well-known method to generate prediction models and discover an informative pattern. In 2002, an advanced SVM method, Support Vector Machine based on Recursive Feature Elimination (RFE) was used to solve the small subset of gene selection problem[9]. This method could identify a smaller range of genes compared with traditional gene filter methods and improve the accuracy rate. In this study, I integrated two PDAC studies and identify 1249 genes with differential expression pattern between normal and PDAC cells. Further SVM-RFE analysis filter 45 genes associated with PDAC.

Code with Documentation

1. Datasets

Two datasets were got from GEO database (Table 1). The first dataset GSE15471 performed 78 GeneChip hybridization involving 36 pairs of normal and tumor tissue samples. Three of the 36 pairs were carried out as replications. The second dataset GSE71989 detect 8 normal tissues and 14 PDAC tissues.

In this study, all analysis was used R packages and stored in a R-mark file (see supplement file).

Dataset Accession ID	Platform	Sample		Reference (PMID)
		Normal	PDAC	
GSE15471	Affymetrix Human Genome U133 Plus 2.0 Array	36	36	19260470
GSE71989	Affymetrix Human Genome U133 Plus 2.0 Array	8	14	NA

Table 1. List of PDAC datasets used in this study

2. Pre-processing data

Two microarray data were preprocessed applying the Robust Multichip Average (RMA) method in R package LIMMA (Figure 1). RMA performed background correction, data normalization and summarization of multiple probes based on genes. PLIER and RMA are both useful methods for normalization microarray data. Comparing PLIER, RMA is a better method which the preprocessed data have a similar distribution and smaller intra-group variance. The results are shown in supplement part (Figure 7).

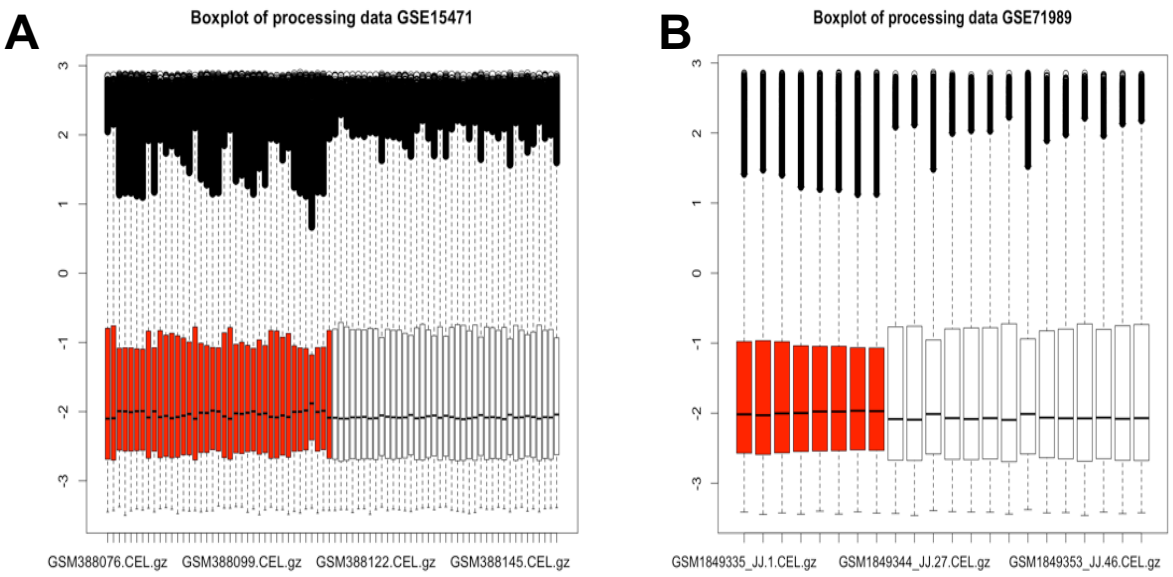

Figure 1. The pre-processing data of GSE15471 and GSE71989

3. Filtering differential genes of each dataset

I compare the samples by fitting a linear model and empirical Bayes to calculate the expression fold change between the control group and the umor group. Then using two filters which the absolute fold change >2 and multiple tests corrected P-value <0.05 get significantly differentially expressed genes. For individual genes, I choose the genes passing the one-sample test ($p<0.05$) (Figure 2).

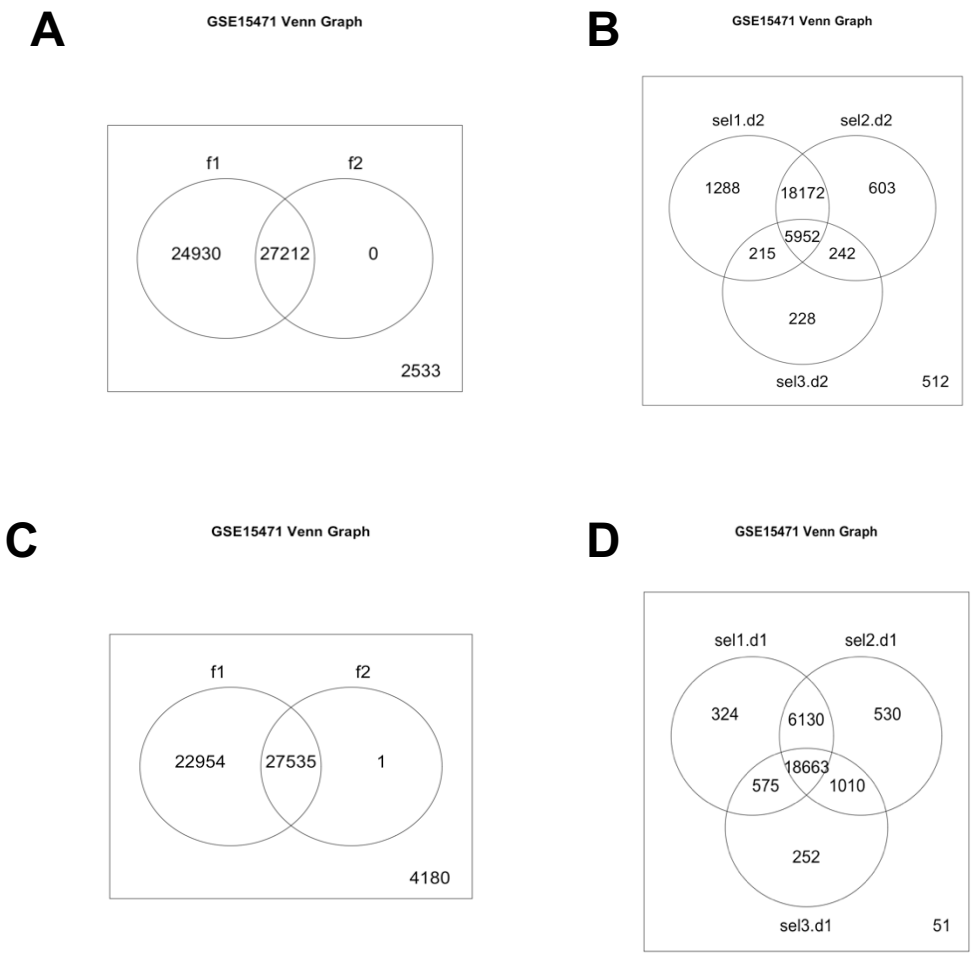

Figure 2. The Venn graph of two database

Finally, I filter the 3482 genes that occurred in both datasets which were differential expressed.

4. Code: R-mark file

All codes used in this project are written in R-mark file named "Project_Code.Rmd". All codes are shown below:

```
---
title: 'ML_Project_Predicting Biomarker Genes of Pancreatic Ductal
Adenocarcinoma
  (PDAC) '
author: "Xueyi Fan"
date: "July 20, 2016"
output: word_document
---

DSCS6030 Machine Learning
#Download the data
```[10]
#download the data from GEO
library("GEOquery")
setwd("/Users/fanxueyi/Documents/NEU
Bioinformatics/DSCS6030_Intro_Data_Mining:Machine_Learing/Project/DATA")
GEO_id_list <- c("GSE71989","GSE15471")
for (i in GEO_id_list[1]){
  getGEOSuppFiles(i)
}

#system call to uncompress the data
try(system("ls"))
for (i in GEO_id_list[1]){
  call <- paste("tar zxvf ./", i, "/",i,"_RAW.tar -C ./",i, sep="")
  try(system(call))
}

```


#Pre-processing the affymetrix data of GSE15471
```{r}
```
```

```
library("affy")
library("limma")
library("genefilter")
library("hgu133plus2.db")
library("rgl")

#GSE15471 deatset
#Microdata need to do background correct and normalization, here I use RMA
to normalized each dataset
pathway <- paste("/Users/fanxueyi/Documents/NEU
Bioinformatics/DSCS6030_Intro_Data_Mining:Machine_Learing/Project/DATA/","GS
E15471",sep="")
setwd(pathway)
data2 <- ReadAffy()
eset.d2<- exprs(data2)
annotation(data2)



#log-scale transform data
qx <- as.numeric(quantile(eset.d2, c(0., 0.25, 0.5, 0.75, 0.99, 1.0),
na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0) ||
  (qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)
if (LogC) { eset.d2[which(eset.d2 <= 0)] <- NaN
exprs(data2) <- log2(eset.d2) }
data2.rma <- rma(data2)

#factor: 0 stands for normal tissue, 1 stands for tumor
gsms.d2 <- paste0("00000000000000000000000000000000000011111111111",
        "11111111111111111111111111")
sml.d2 <- c()
for (i in 1:nchar(gsms.d2)) { sml.d2[i] <- substr(gsms.d2,i,i) }
```

```
fl2 <- as.factor(sml.d2)

#reorganize the expression data
fl2 <- as.factor(c(rep(0,39),rep(1,39)))
eset2.d2<- exprs(data2.rma)
head(eset2.d2)
normal.d2<- eset2.d2[,which(fl2==0)]
tumor.d2 <- eset2.d2[,which(fl2==1)]
eset3.d2 <- data.frame(normal.d2, tumor.d2)

boxplot(eset3.d2, main=paste("Boxplot of processing data GSE15471"),
col=c(rep(2,39), rep(0,39)))

#get fold change data
fl2 <- factor(fl2, levels = c(0,1),labels=c("control","tumor"))
design.ma <- model.matrix(~fl2.d2)
fit2<-lmFit(eset3.d2, design.ma)
fit2 <- eBayes(fit2.d2)
cont.ma <- makeContrasts(control-tumor, levels=fl2.d2)
fit2_cont <- contrasts.fit(fit2,cont.ma)
fit2_cont <- eBayes(fit2_cont)
d2.fc <- topTable(fit2_cont,number= 54675,adjust.method = "fdr")
head(d2.fc)

#filter different expression genes
f1.d2 <- d2.fc$adj.P.Val <0.05
f2.d2 <- abs(d2.fc$logFC) >=2
f3.d2 <- f1.d2&f2.d2
dataset.2.selected <- eset3.d2[f3.d2,]

f4.d2 <- function(x) {shapiro.test(x)$p.value > 0.05}
f5.d2 <- function(x) {(sqrt(10)* abs(mean(x))/sd(x) > qt(0.975,9))}
sel1.d2 <- genefilter(dataset.2.selected[, fl2=="tumor"], f5.d2)
```

```
sel2.d2 <- genefilter(dataset.2.selected[, fl2=="control"], f5.d2)
sel3.d2 <- genefilter(dataset.2.selected, f4.d2)
sel4.d2 <- sel1.d2&sel2.d2&sel3.d2
dataset.2.selected <- dataset.2.selected[sel4.d2,]
dim(dataset.2.selected)


#venn diagram
x1.d2 <- apply(cbind(f1.d2,f2.d2), 2, as.integer)
vc1.d2 <- vennCounts(x1.d2,include="both")
vennDiagram(vc1.d2, main="GSE15471 Venn Graph")


x2.d2 <- apply(cbind(sel1.d2,sel2.d2,sel3.d2), 2, as.integer)
vc2.d2 <- vennCounts(x2.d2,include="both")
vennDiagram(vc2.d2, main="GSE15471 Venn Graph")

#PCA and its plot
d2.selected.pca <- princomp(dataset.2.selected, cor=T, scores = T)
plot(d2.selected.pca, type="l", main="GSE15471 PCA Plot")
plot3d(d2.selected.pca$loadings[,1:3], col=as.numeric(fl2.d2))
#biplot(d2.selected.pca)

```
```

#Pre-processing the affymetrix data of GSE71989
```{r}

#GSE71989 deatset
#Microdata need to do background correct and normalization, here I use RMA to normalized each dataset
```
```

```
pathway <- paste("/Users/fanxueyi/Documents/NEU
Bioinformatics/DSCS6030_Intro_Data_Mining:Machine_Learing/Project/DATA/","GS
E71989",sep="")
setwd(pathway)
data1 <- ReadAffy()
eset<- exprs(data1)
annotation(data1)

#log-scale transform data
qx <- as.numeric(quantile(eset, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))
LogC <- (qx[5] > 100) ||
  (qx[6]-qx[1] > 50 && qx[2] > 0) ||
  (qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)
if (LogC) { eset[which(eset <= 0)] <- NaN
exprs(data1) <- log2(eset) }
data1.rma <- rma(data1)



head(exprs(data2))
#factor: 0 stands for normal tissue, 1 stands for tumor
gsms <- paste0("00000000111111111111111")
sml <- c()
for (i in 1:nchar(gsms)) { sml[i] <- substr(gsms,i,i) }
fl1 <- as.factor(sml)

#reorganize the expression data
fl1 <- as.factor(c(rep(0,8),rep(1,14)))
eset2<- exprs(data1.rma)
head(eset2)
normal<- eset2[,which(fl1==0)]
tumor <- eset2[,which(fl1==1)]
eset3 <- data.frame(normal, tumor)
```

```
boxplot(eset3, main=paste("Boxplot of processing data GSE71989"),
col=c(rep(2,8), rep(0,14)))

#get fold change data
fl1.d1 <- factor(fl1, levels = c(0,1),labels=c("control","tumor"))
design.ma <- model.matrix(~fl1.d1)
fit1 <-lmFit(eset2, design.ma)
fit1 <- eBayes(fit1)
cont.ma <- makeContrasts(control-tumor, levels=fl1.d1)
fit1_cont <- contrasts.fit(fit1,cont.ma)
fit1_cont <- eBayes(fit1_cont)
d1.fc <- topTable(fit1_cont,number= 54675,adjust.method = "fdr")
head(d1.fc)

#filter different expression genes
f1 <- d1.fc$adj.P.Val <0.05
f2 <- abs(d1.fc$logFC) >=2
f3 <- f1&f2
dataset.1.selected <- eset3[f3,]


f4.d1 <- function(x) {shapiro.test(x)$p.value > 0.05}
f5.d1 <- function(x) {(sqrt(10)* abs(mean(x))/sd(x) > qt(0.975,9))}
sel1.d1 <- genefilter(dataset.1.selected[, fl1.d1=="tumor"], f5.d1)
sel2.d1 <- genefilter(dataset.1.selected[, fl1.d1=="control"], f5.d1)
sel3.d1 <- genefilter(dataset.1.selected, f4.d1)
sel4.d1 <- sel1.d1&sel2.d1&sel3.d1
dataset.1.selected <- dataset.1.selected[sel4.d1,]
dim(dataset.1.selected)

#venn diagram
x <- apply(cbind(f1,f2), 2, as.integer)
vc <- vennCounts(x,include="both")
```

```
vennDiagram(vc, main="GSE15471 Venn Graph")

x2.d1 <- apply(cbind(sel1.d1,sel2.d1,sel3.d1), 2, as.integer)
vc2.d1 <- vennCounts(x2.d1,include="both")
vennDiagram(vc2.d1, main="GSE15471 Venn Graph")

#PCA and its plot
d1.selected.pca <- princomp(dataset.1.selected, cor=T, scores = T)
plot(d1.selected.pca, type="l",main="GSE71989 PCA Plot")
plot3d(d1.selected.pca$loadings[,1:3], col=as.numeric(fl1.d1))

#venn diagram
x <- apply(cbind(sel1,sel2,sel3), 2, as.integer)
vc <- vennCounts(x,include="both")
vennDiagram(vc)

```

#Choose the same gene set of these two datasets
```{r}
head(dataset.1.selected)
head(dataset.2.selected)

name.d1 <- row.names(dataset.1.selected)
name.d2 <- row.names(dataset.2.selected)

same.d1<- name.d1 %in% name.d2
same.d2<- name.d2 %in% name.d1

table(same.d1)
table(same.d2)

d1.same <- dataset.1.selected[which(same.d1==TRUE),]
```

```
d2.same <- dataset.2.selected[which(same.d2==TRUE),]

library(Biobase)
library(MergeMaid)
library("MAMA")
library(RankProd)

#create MetaArray object
d1.spl <- data.frame(metastasis=fl1)
row.names(d1.spl) <- colnames(d1.same)
d2.spl <- data.frame(metastasis=fl2)
row.names(d2.spl) <- colnames(d2.same)
merged <- new("MetaArray", GEDM= list(d1.same,d2.same), clinical =
list(d1.spl,d2.spl),datanames=c("data1", "data2"))



#use method combine p-values
pval<- metaMA(merged,varname="metastasis", which="pval")
length(pval$Meta)

#use RankProb method
rp<- RankProduct(merged, varname= "metastasis", plot=T, rand=123,
cutoff=0.05, num.perm=100, gene.names = rownames(GEDM(merged))[[1]])
up <- rp$Table1
down <- rp$Table2

head(up)
head(down)

#get final differential expression genes

up.d1.exprs <- d1.same[up[,1],]
up.d2.exprs<- d2.same[up[,1],]
```

```
down.d1.exprs <- d1.same[down[,1],]
down.d2.exprs <- d2.same[down[,1],]

#add new column (symbol) to expression data

get_gene_name <- function(x) {
  if (is.character(get(x,env=hgu133plus2SYMBOL))){
    return(get(x,env=hgu133plus2SYMBOL))
    }
  else{
    return(NA)
  }
}
up.gene.name <- lapply(rownames(up),get_gene_name)
down.gene.name <- lapply(rownames(down),get_gene_name)

up.gene.name <- unlist(up.gene.name)
down.gene.name <- unlist(down.gene.name)
up.d1.exprs$symbol <- up.gene.name
up.d2.exprs$symbol <- up.gene.name
down.d1.exprs$symbol <-down.gene.name
down.d2.exprs$symbol <-down.gene.name

#plot heatmap with top 100 up-regulated genes and top 100 down-regulated
genes based on
up.100 <- head(up[order(up[,3]),],100)
down.100 <- head(down[order(down[,3],decreasing=T),],100)
up.d1.100.exprs <- d1.same[up.100[,1],]
up.d2.100.exprs<- d2.same[up.100[,1],]
down.d1.100.exprs <- d1.same[down.100[,1],]
down.d2.100.exprs <- d2.same[down.100[,1],]

up.gene.100.name <- lapply(rownames(up.100),get_gene_name)
```

```
down.gene.100.name <- lapply(rownames(down.100),get_gene_name)

up.gene.100.name <- unlist(up.gene.100.name)
down.gene.100.name <- unlist(down.gene.100.name)
up.d1.100.exprs$symbol <- up.gene.100.name
up.d2.100.exprs$symbol <- up.gene.100.name
down.d1.100.exprs$symbol <-down.gene.100.name
down.d2.100.exprs$symbol <-down.gene.100.name

d1.100.data <- rbind(up.d1.100.exprs,down.d1.100.exprs)
d2.100.data <- rbind(up.d2.100.exprs,down.d2.100.exprs)

gene.100.list<- c(up.gene.100.name,down.gene.100.name)

#get fold change data of two datasets
fl1.d1 <- factor(fl1, levels = c(0,1),labels=c("control","tumor"))
design.ma <- model.matrix(~fl1.d1)
fit1 <-lmFit(d1.100.data[,-23], design.ma)
fit1 <- eBayes(fit1)
cont.ma <- makeContrasts(control-tumor, levels=fl1.d1)
fit1_cont <- contrasts.fit(fit1,cont.ma)
fit1_cont <- eBayes(fit1_cont)
d1.fc <- topTable(fit1_cont, number=200, adjust.method = "fdr")


fl2.d2 <- fl2
design.ma <- model.matrix(~fl2.d2)
dim(design.ma)
dim(d2.100.data)
fit2 <-lmFit(d2.100.data[,-79], design.ma)
fit2 <- eBayes(fit2)
cont.ma2 <- makeContrasts(control-tumor, levels=fl2.d2)
fit2_cont <- contrasts.fit(fit2,cont.ma2)
```

```
fit2_cont <- eBayes(fit2_cont)
d2.fc <- topTable(fit2_cont, number=200, adjust.method = "fdr")

d1.fc$rowname<- rownames(d1.fc)
d2.fc$rowname<- rownames(d2.fc)

d1.fc.new <- data.frame(d1.fc[,1], probe_name=d1.fc$rowname)
d2.fc.new <- data.frame(d2.fc[,1], probe_name=d2.fc$rowname)
merge.d1.d2.fc <- merge(d1.fc.new,d2.fc.new, by="probe_name")
heatmap_data <- data.frame(merge.d1.d2.fc[,2:3])
rownames(heatmap_data) <- merge.d1.d2.fc$rowname
heatmap_data_gene_name <- lapply(rownames(up.100),get_gene_name)
heatmap_data$symbol <- unlist(heatmap_data_gene_name)



plot_data <- as.matrix(heatmap_data[,1:2])
plot_data <- apply(plot_data,2,as.numeric)

library("gplots")
heatmap.2(plot_data, col=redgreen(75), scale= "none", cexRow = 0.5,cexCol =
1,labRow=gene.100.list, key=T, keysize=1.5, key.title = "color key",
symkey=F, symbreaks = T, density.info="none", trace="none",dendrogram =
"none", labCol=c( "GSE71989","GSE15471"))

```


#DATA Mining
```{r}
#analysis the difficial expression genes
#combine d1, d2 expression data

up.d1.exprs$rowname <- rownames(up)
```

```
up.d2.exprs$rowname <- rownames(up)
all.up <- merge(up.d1.exprs[,-23], up.d2.exprs[,-79], by="rowname")

down.d1.exprs$rowname <- rownames(down)
down.d2.exprs$rowname <- rownames(down)
all.down <- merge(down.d1.exprs[,-23], down.d2.exprs[,-79], by="rowname")
all<- rbind.data.frame(all.up,all.down)
head(all)
dim(all)
all.gene.name <- lapply(all$rowname, get_gene_name)
all$symbol<- all.gene.name
all.factor <- factor(c(fl1.d1, fl2.d2), labels = c("control", "tumor"))


#cluster the differential expression data

#Hierarchical clustering
all.hclust <- hclust(d=dist(t(all[,c(-1,-102)])),method="single")
all.hclust.2<- cutree(all.hclust,2)
length(all.hclust.2)
length(all.factor)
cm <- table(all.hclust.2, all.factor)
cm
plot(cm, main="Hierarchical Clustering")
plot(all.hclust, label=F)

#K-medoids clustering
library("cluster")
all.pam <- pam(t(all[,c(-1,-102)]),2 )
cm2 <- table(all.pam$clustering, all.factor)
cm2
plot(cm2, main="2-medoids Clstering")
```

```

#classification using SVM-RFE
```{r}
library(e1071)
library(caret)

################################################
# Feature Ranking with SVM-RFE
################################################
svmrfeFeatureRanking = function(x,y){
    n = ncol(x)

    survivingFeaturesIndexes = seq(1:n)
    featureRankedList = vector(length=n)
    rankedFeatureIndex = n

    while(length(survivingFeaturesIndexes)>0){
        #train the support vector machine
        svmModel = svm(x[, survivingFeaturesIndexes], y, cost = 10,
cachesize=500,  scale=F, type="C-classification", kernel="linear" )

        #compute the weight vector
        w = t(svmModel$coefs)%*%svmModel$SV

        #compute ranking criteria
        rankingCriteria = w * w

        #rank the features
        ranking = sort(rankingCriteria, index.return = TRUE)$ix

        #update feature ranked list
```
```

```
        featureRankedList[rankedFeatureIndex] =
survivingFeaturesIndexes[ranking[1]]
        rankedFeatureIndex = rankedFeatureIndex - 1

        #eliminate the feature with smallest ranking criterion
        (survivingFeaturesIndexes = survivingFeaturesIndexes[-ranking[1]])

    }

    return (featureRankedList)
}

featureRankedList <- svmrfeFeatureRanking(t(all[,c(-1,-102)]),all.factor)

#train a SVM with different N most relevant features (N=50,500,1000)

ranklist.50 <- featureRankedList[1:50]
ranklist.500 <- featureRankedList[1:500]
ranklist.1000 <- featureRankedList[1:1000]

#using 50 fold Cross-validation for ranklist.30
all.t <- t(all[,c(-1,-102)])
all.50 <- all.t[,ranklist.50]
n<- dim(all.50)[1]
index <- 1:n
K<-50
flds <- createFolds(index, k=K)
mcr.cv.raw <- rep(NA, K)
sen.cv.raw <- rep(NA,K)
spe.cv.raw <- rep(NA,K)
for (i in (1:K)){
  testID <- flds[[i]]
  data.train <- all.50[-testID,]
```

```
  data.test <- all.50[testID,]
  data.svm <- svm(data.train, all.factor[-testID], kernel="linear")
  data.pred <- predict(data.svm, newdata=data.test)
  mcr.cv.raw[i] <- mean(data.pred != all.factor[testID])
  sen.cv.raw[i] <- sum(data.pred == "tumor" &
all.factor[testID]=="tumor")/sum(all.factor[testID]=="tumor")
  spe.cv.raw[i] <-  sum(data.pred == "control" &
all.factor[testID]=="control")/sum(all.factor[testID]=="control")
}

mcr.cv.50 <- mean(mcr.cv.raw)
sen.cv.50 <- mean(na.omit(sen.cv.raw))
spe.cv.50 <- mean(na.omit(spe.cv.raw))

#using 50 fold Cross-validation for ranklist.500
all.t <- t(all[,c(-1,-102)])
all.500 <- all.t[,ranklist.500]
n<- dim(all.500)[1]
index <- 1:n
K<-50
flds <- createFolds(index, k=K)
mcr.cv.raw <- rep(NA, K)
sen.cv.raw <- rep(NA,K)
spe.cv.raw <- rep(NA,K)
for (i in (1:K)){
  testID <- flds[[i]]
  data.train <- all.500[-testID,]
  data.test <- all.500[testID,]
  data.svm <- svm(data.train, all.factor[-testID], cost=10, kernel="linear")
  data.pred <- predict(data.svm, newdata=data.test)
  mcr.cv.raw[i] <- mean(data.pred != all.factor[testID])
  sen.cv.raw[i] <- sum(data.pred == "tumor" &
all.factor[testID]=="tumor")/sum(all.factor[testID]=="tumor")
```

```
  spe.cv.raw[i] <-  sum(data.pred == "control" &
all.factor[testID]=="control")/sum(all.factor[testID]=="control")
}
mcr.cv.500 <- mean(mcr.cv.raw)
sen.cv.500 <- mean(na.omit(sen.cv.raw))
spe.cv.500 <- mean(na.omit(spe.cv.raw))


#using 50 fold Cross-validation for ranklist.1000
all.t <- t(all[,c(-1,-102)])
all.1000 <- all.t[,ranklist.1000]
n<- dim(all.1000)[1]
index <- 1:n
K<-50
flds <- createFolds(index, k=K)
mcr.cv.raw <- rep(NA, K)
sen.cv.raw <- rep(NA,K)
spe.cv.raw <- rep(NA,K)
for (i in (1:K)){
  testID <- flds[[i]]
  data.train <- all.1000[-testID,]
  data.test <- all.1000[testID,]
  data.svm <- svm(data.train, all.factor[-testID], cost=10, kernel="linear")
  data.pred <- predict(data.svm, newdata=data.test)
  mcr.cv.raw[i] <- mean(data.pred != all.factor[testID])
  sen.cv.raw[i] <- sum(data.pred == "tumor" &
all.factor[testID]=="tumor")/sum(all.factor[testID]=="tumor")
  spe.cv.raw[i] <-  sum(data.pred == "control" &
all.factor[testID]=="control")/sum(all.factor[testID]=="control")
}

mcr.cv.1000 <- mean(mcr.cv.raw)
sen.cv.1000 <- mean(na.omit(sen.cv.raw))
```

```
spe.cv.1000 <- mean(na.omit(spe.cv.raw))

mcr.cv.50
sen.cv.50
spe.cv.50

mcr.cv.500
sen.cv.500
spe.cv.500

mcr.cv.1000
sen.cv.1000
spe.cv.1000

dim(all)
ranklist.50
final_genes <- data.frame(probe_name = unlist(all$rowname[ranklist.50]),
Gene_name = unlist(all.gene.name[ranklist.50]))
final_genes <- na.omit(final_genes)
is_up <- final_genes$probe_name %in% all.up$rowname
down_index <- which(is_up == F)
final_up <- final_genes[-down_index,]
final_down <- final_genes[down_index,]
```

Finnally I identify 50 genes which is related to PDCA.

```{r}
#detect the pathway where the genes act functions
library(SPIA)
```

```
library("KEGG.db")

get_pathway <- function(x){
  if (is.character(get(x,env=hgu133plus2PATH))){
    return(get(x,env=hgu133plus2PATH))
  }
  else{
    return(NA)
  }
}
length(test)
for (i in test){
  print(get_pathway(i))
}

pathID<- lapply(as.vector(final_genes$probe_name),get_pathway)
pathID <- as.matrix(table(unlist(pathID)))
pathID <- rownames(pathID)
pathID
getPathName <- function(x){
  get(x, env=KEGGPATHID2NAME)
}
pathName <- unlist(lapply(pathID, getPathName))
pathname
```
########################End of Code#######################################

Results

1. Principle Component Analysis (PCA) of Two datasets

Princomp function in R was used here to do Principle Component Analysis. Then plot the variance of top 10 components of each dataset as well as 3D plot.

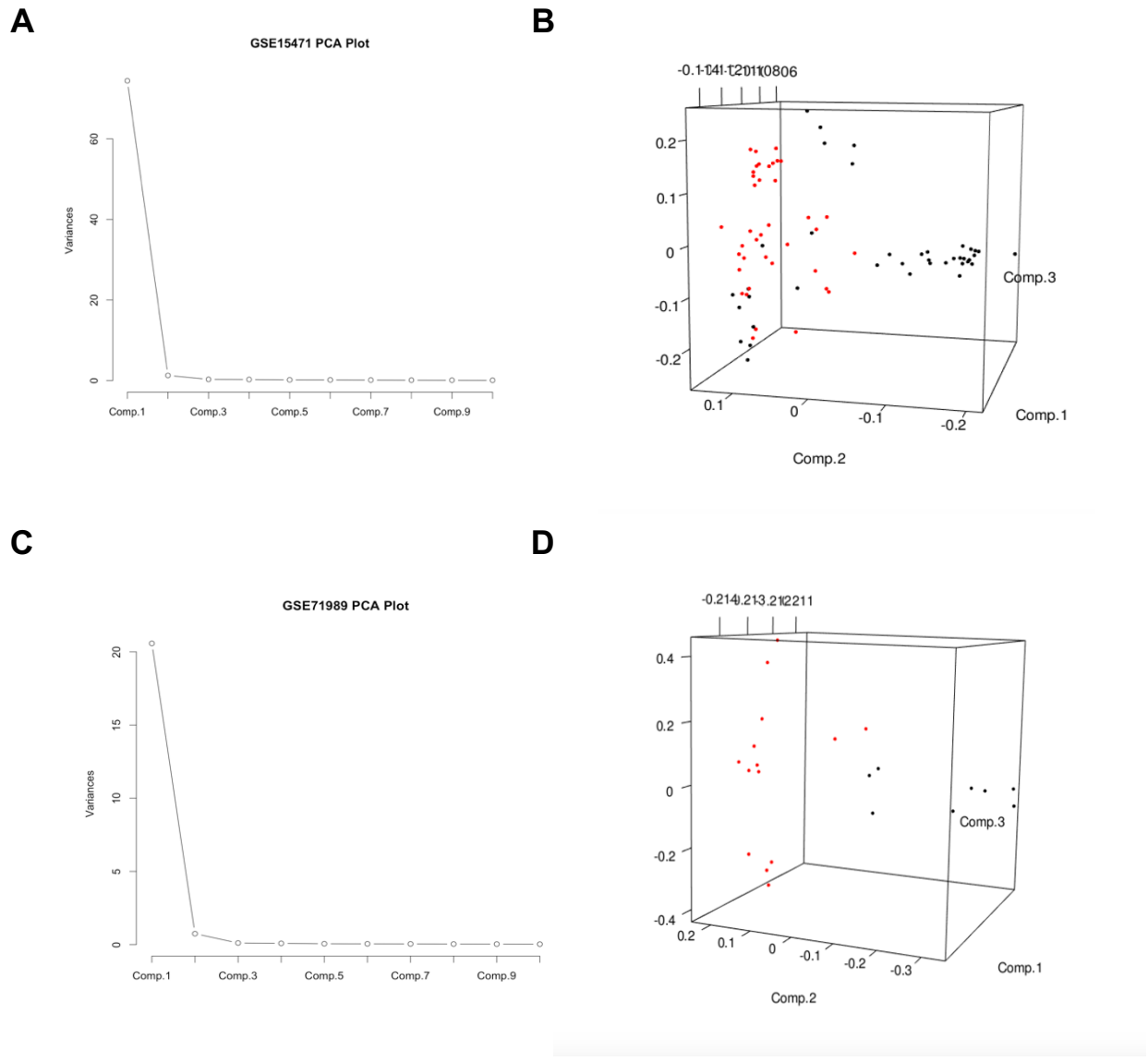

Figure 3. PCA analysis of two datasets

2. Meta-analysis to get differential genes set

Rank based meta analysis were performed here to identify the differentially expressed genes for the two datasets using the R package “MAMA”. I got 715 genes are up-regulated and 534 genes are down-regulated. The false-positive predictions were restricted to less than 5% (FDR <=0.05), based on 1000 class labels-based random permutations. Further visualization of the gene expression data was shown as heatmap. I chose 100 top regulated genes from both up regulated gene set and down regulated gene set based on the fold change (Fc) (Figure 4).

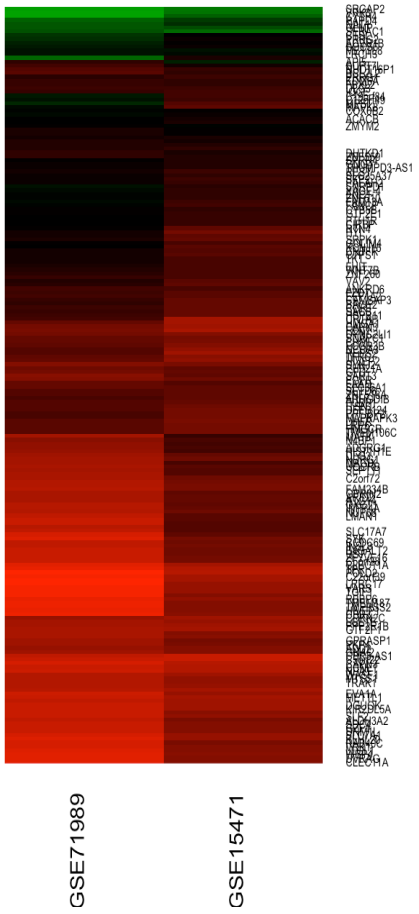

Figure 4. Heatmap of ford change (Fc) of top 100 up-regulated genes and 100 down-regulated genes in both datasets

3. Clustering Analysis (Unsupervised Analysis)

Compared both Partition Around Mediods algorithm and Hierarchical Clustering analysis using chosen gene expression profile (Figure 5&6).

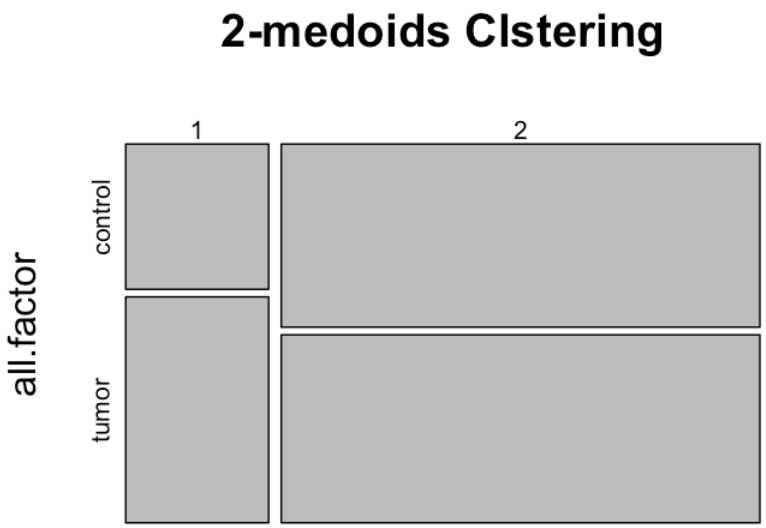

Figure 5. Contingency table of Partition Around Mediods Analysis

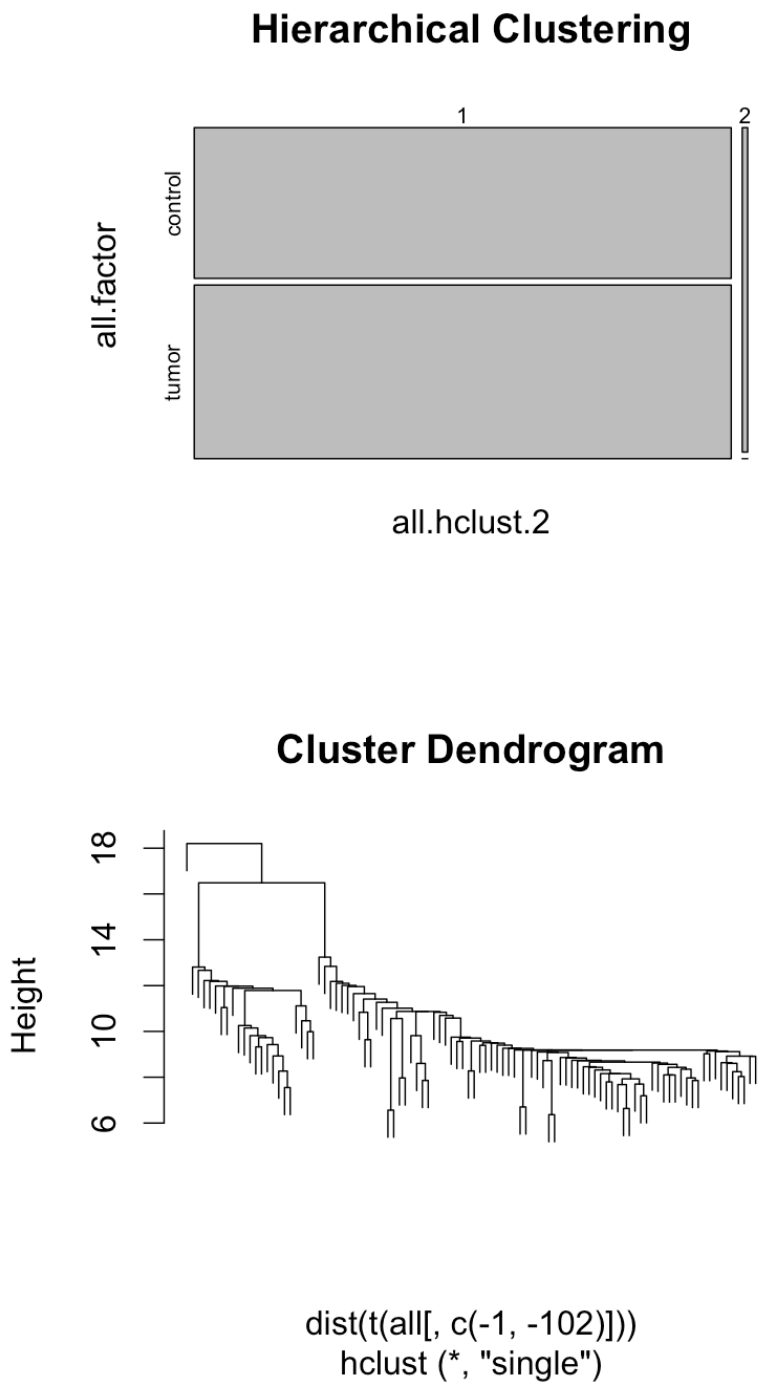

Figure 6. Hierarchical Clustering analysis to explore the inner relationship of chosen genes.

4. SVM-RFE analysis

All codes are based on the algorithms described in by Guyon using the package "e1071". 50 fold Cross-validation are used here to compare the accuracy of the results of SVM-RFE analysis with different size of gene sets (Table 2).

Number of Gene	Misclassification Rate	Sensitivity	Specificity
50	0	1	1
500	0.02	1	0.9487179
1000	0.06	0.9605263	0.9571429

Table 2. Evaluating the performance of different size of gene set got from SVM-RFE analysis.

5. Annotation of the final genes

According to the annotation database, I got all the chosen genes and divided them into to up expression group and down expression group (Table 3). Further mapping was performed in KEGG pathway database (Table 4).

Probe Name	Gene Name	Up-/Down- Regulation
201373_at	PLEC	Up-Expression
204028_s_at	RABGAP1	
204927_at	RASSF7	
213260_at	FOXC1	
231838_at	PABPC1L	

202071_at	SDC4	
1556277_a_at	PAPD4	
204793_at	GPRASP1	
203313_s_at	TGIF1	
1557065_at	YLPM1	
204270_at	SKI	
209215_at	MFSD10	
203491_s_at	CEP57	
205250_s_at	CEP290	
1555858_at	THUMPD3-AS1	
203354_s_at	PSD3	
205171_at	PTPN4	
204273_at	EDNRB	
214850_at	SMA4	
203062_s_at	MDC1	
201340_s_at	ENC1	
212684_at	ZNF3	
205904_at	MICA	
220319_s_at	MYLIP	
203705_s_at	FZD7	
1559096_x_at	FBXO9	
200632_s_at	NDRG1	
201654_s_at	HSPG2	
225391_at	LOC93622	

216641_s_at	LAD1	
214436_at	FBXL2	
213484_at	ADD2	Down expression
1554335_at	CYTH4	
1566785_x_at	NSF	
205203_at	PLD1	
211237_s_at	FGFR4	
205723_at	CNTFR	
1554894_a_at	PCBD2	
1555294_a_at	ERC1	
210963_s_at	GYG2	
1556601_a_at	SPATA13	
227828_s_at	EVA1A	
219399_at	LIN7C	
205514_at	ZNF415	
201883_s_at	B4GALT1	

Table 3. Annotation the probe to genes

Genes	**KEGG Pathway ID**	**Name**	**Related Field**	**Up-/Down-Regulation**
PABPC1L	03013	RNA transport	RNA	Up

	03015	mRNA surveillance pathway		
	03018	RNA degradation		
NSF	04962	Vasopressin-regulated water reabsorption		Down
PLD1	00564	Glycerophospholipid metabolism	Metabolism	Down
	00565	Ether lipid metabolism		
	01100	Metabolic pathways		
	04144	Endocytosis		
	04666	Fc gamma R-mediated phagocytosis		
	04912	GnRH signaling pathway	Cancer related	
	05200	Pathways in cancer		
	05212	Pancreatic cancer		
FGFR4	04010	MAPK signaling pathway		Down
	04144	Endocytosis		
	04810	Regulation of actin cytoskeleton		

CNTFR	04060	Cytokine-cytokine receptor interaction		Down
	04630	Jak-STAT signaling pathway		
PSD3	04144	Endocytosis		Up
EDNRB	04020	Calcium signaling pathway		Up
	04080	Cytokine-cytokine receptor interaction		
	04916	Neuroactive ligand-receptor interaction		
FZD7	04310	Wnt signaling pathway	Cancer related	Up
	04916	Melanogenesis		
	05200	Pathways in cancer		
	05217	Basal cell carcinoma		
HSPG2	04512	ECM-receptor interaction		Up
B4GALT1	00052	Galactose metabolism	Metabolism	Down
	00510	N-Glycan biosynthesis		

	00514	Other types of O-glycan biosynthesis		
	00533	Glycosaminoglycan biosynthesis - keratan sulfate		
	00601	Glycosphingolipid biosynthesis - lacto and neolacto series		
	01100	Metabolic pathways		

Table 4. Mapping all significant differential expression genes on KEGG pathway database.

6. Robustness Analysis of algorithms

Randomly choosing 1% of the data from each dataset and add +/- 5% change of the value and applying the algorithms mentioned above to test whether these algorithms have a good performance to deal with noise. 50-fold Cross-validation are used here to compare the accuracy of the results of SVM-RFE analysis with different size of gene sets. The results are shown in supplement section (Figure 8-11, Table 5-6).

Discussion

After Processing microarray data and filter genes based on four criteria, 5952 genes were chosen from GSE15471 and 18663 genes were chosen from GSE71989. Principle Component Analysis was performed on both selected datasets, and the first three components showed a good separation of control samples and tumor samples on both results. Then I selected 3482 genes that existed in both datasets and transform the data to prepare further meta-analysis.

Rank-based meta-analysis based on rank products (RP) algorithm was used to explore the common differential expressed genes. RP compares different studies according to their p-values ranking and identify upregulated/downregulated gene sets. Here, I obtained 1294 genes. 715 genes are significantly upregulated expressed in PDAC tumor cells and normal cells among these two datasets and 534 genes are downregulated. The top 100 genes of up-/down-regulated genes were displayed as a heatmap using the fold change data in their own dataset. Many of these genes performed an up-regulated pattern.

In order to figure out the inner data structure of my data, PAM and Hierarchical Clustering analysis were used in this study. Compared with Hierarchical clustering results, PAM had a better performance to distinguish control group and test group. In hierarchical cluster analysis, it couldn't separate normal cells and tumor cells well. The results of PAM could cluster most tumor samples, but couldn't identify control samples. The unsupervised analysis didn't have a good performance using my current gene data, my gene data don't have obvious heterogeneity.

However, the situation changed when I use supervised machine learning method SVM-RFE. A better performance to get biologically relevant gene to cancer shows an advantage of this method. After recursive feature elimination (RFE), I got a ranked list of my filter genes based on the coefficient score of each recursion. A further selection of top 50 genes, 500 genes, 1000 genes combined with downstream SVM analysis to get the best gene set predicting the PDAC. Considering the limitation of the sample size, I trained and tested SVM models and calculate the accuracy of the three subsets using 50-fold cross validation. The top 50 genes had the best performance showing a low misclassification rate and high sensitivity and specificity.

Further annotating these significantly expressed genes and mapping them in KEGG pathway database. PLD1 which is down-regulated in cancerous cells has been reported in many cancers like melanoma, breast cancer as well as pancreatic cancer. B4GALT1 and PLD1 involving in metabolism show down-regulation. In order to verify the association of these genes with PDAC, a series downstream experiments must be applied for example: RT-PCR, construct cell lines or mice knocking out each gene.

Robustness analysis of algorithms shown the effect of the noises. After low-level filtering of the significant expression genes, the number of genes I got from here is different from the genes got from data without noises. Using the data with +/- 5% change values, Finally, I got 1251 genes, 712 genes are up-regulated and 539 genes are down-regulated. Within my expectation, the performances of

unsupervised machine learning algorithms (PAM and Hierarchical Clustering analysis) are worse. However, the supervised machine learning algorithm (SVM-RFE analysis) can deal with noise data and keep its stability. In final results, Most of the genes selected are same as the data without noises, like FZD7, B4GALT1.

Obviously, there are some limitations of this study. The more datasets are collected, the more precise results I would get. Nowadays, data coming from Next Generation Sequencing (NGS) like microRNA seq, epigenetic seq data, are available in public databases. Combining more data in this study could give a better explanation of the mechanism of gene regulation.

Supplements

1. **Cooperation of normalization methods: PLIER vs. RMA**

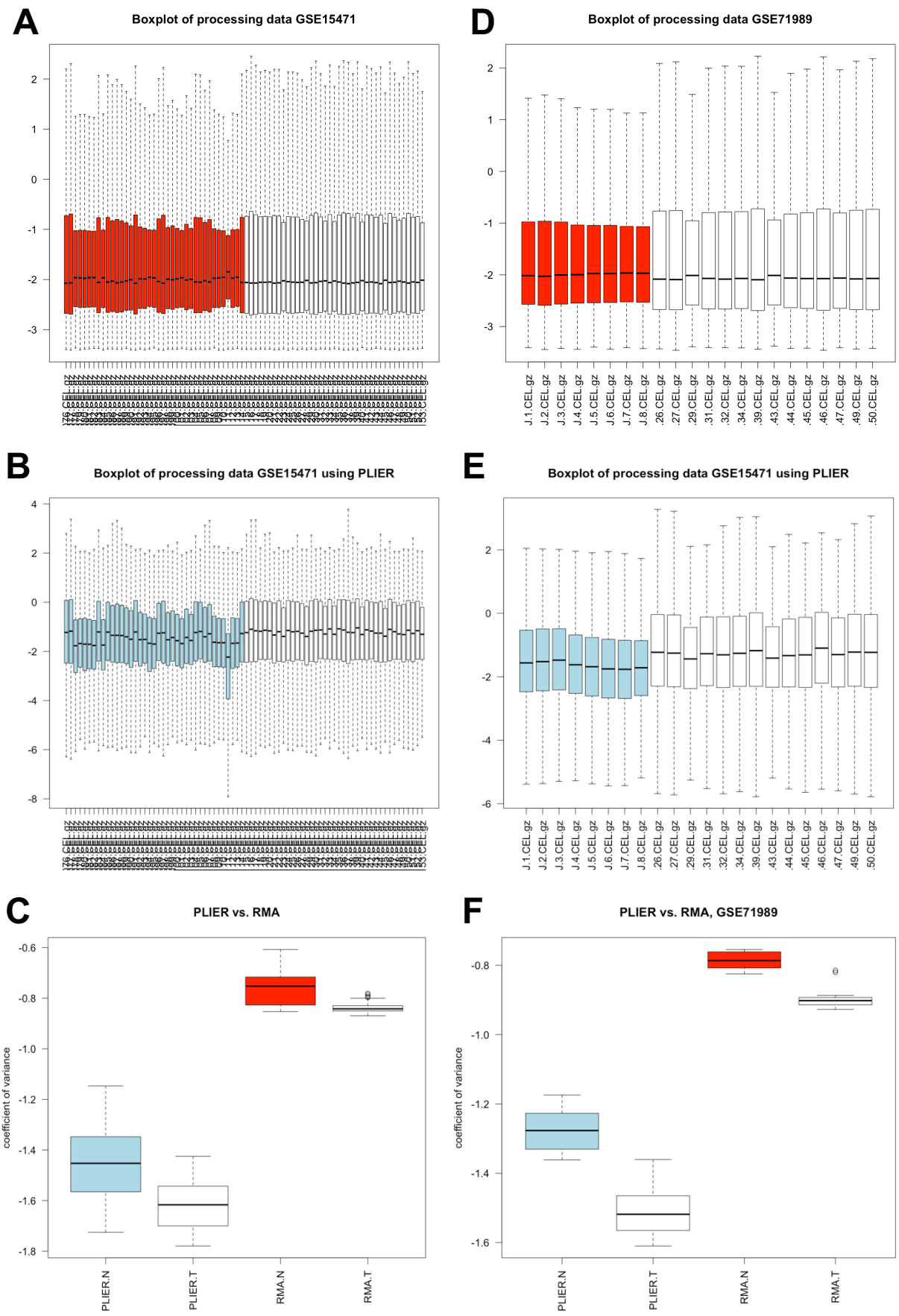
A
Boxplot of processing data GSE15471
B
Boxplot of processing data GSE15471 using PLIER
C
PLIER vs. RMA
coefficient of variance
PLIER.N
PLIER.T
RMA.N
RMA.T
D
Boxplot of processing data GSE71989
J.1.CEL.gz
J.2.CEL.gz
J.3.CEL.gz
J.4.CEL.gz
J.5.CEL.gz
J.6.CEL.gz
J.7.CEL.gz
J.8.CEL.gz
.26.CEL.gz
.27.CEL.gz
.29.CEL.gz
.31.CEL.gz
.32.CEL.gz
.34.CEL.gz
.39.CEL.gz
.43.CEL.gz
.44.CEL.gz
.45.CEL.gz
.46.CEL.gz
.47.CEL.gz
.49.CEL.gz
.50.CEL.gz
E
Boxplot of processing data GSE15471 using PLIER
F
PLIER vs. RMA, GSE71989

Figure 7. Cooperation of PLIER and RMA normalization method. GSE15471 data (A-C) and GSE71989 (E-F) are used to compare PLIER and RMA methods.

2. Robustness of noise

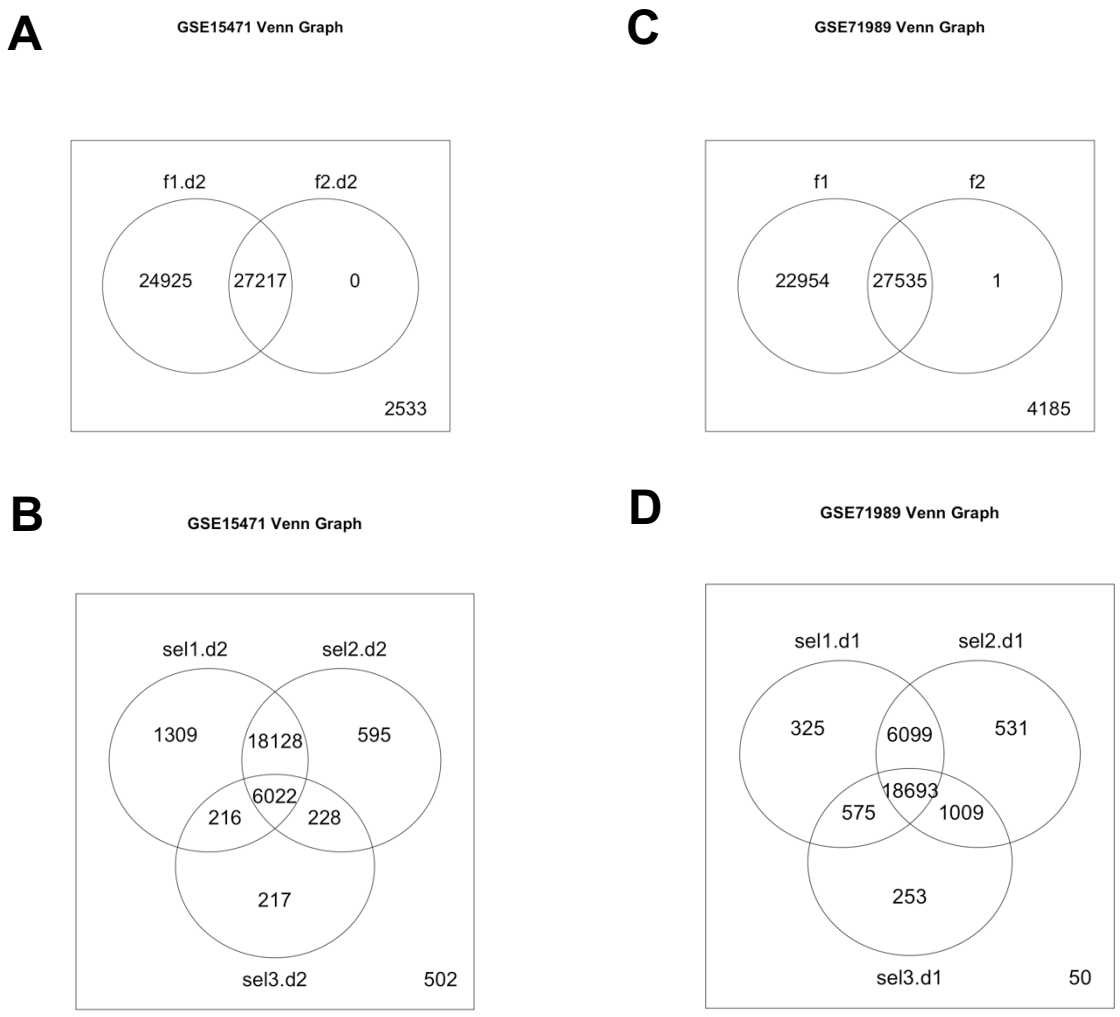

Figure 8. The Venn graph of two datasets with data adding noises. GSE15471 data (A, B) and GSE71989 (C, D) are used.

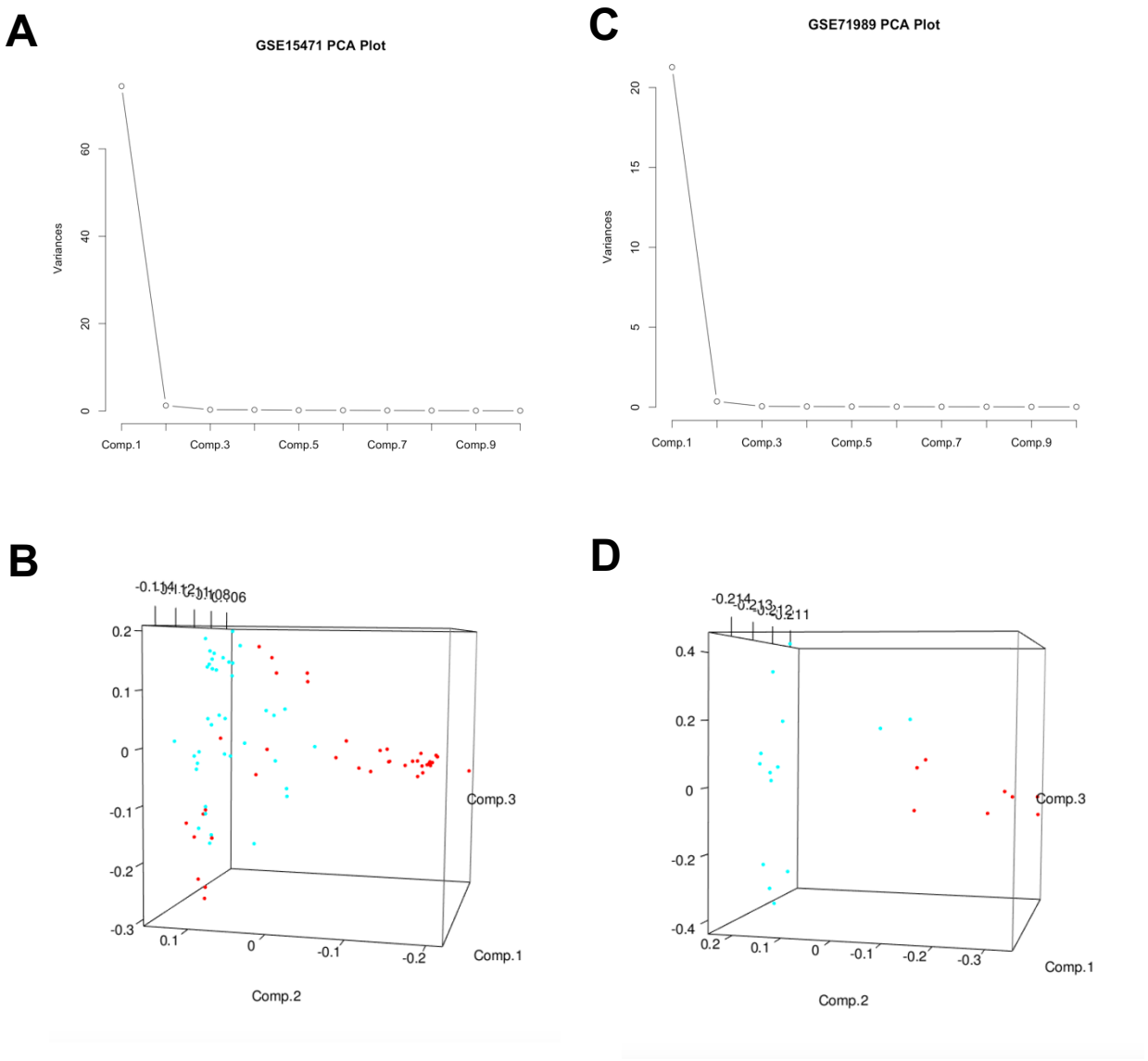

Figure 9. Figure 3. PCA analysis of two datasets. GSE15471 data (A, B) and GSE71989 (C, D) are used.

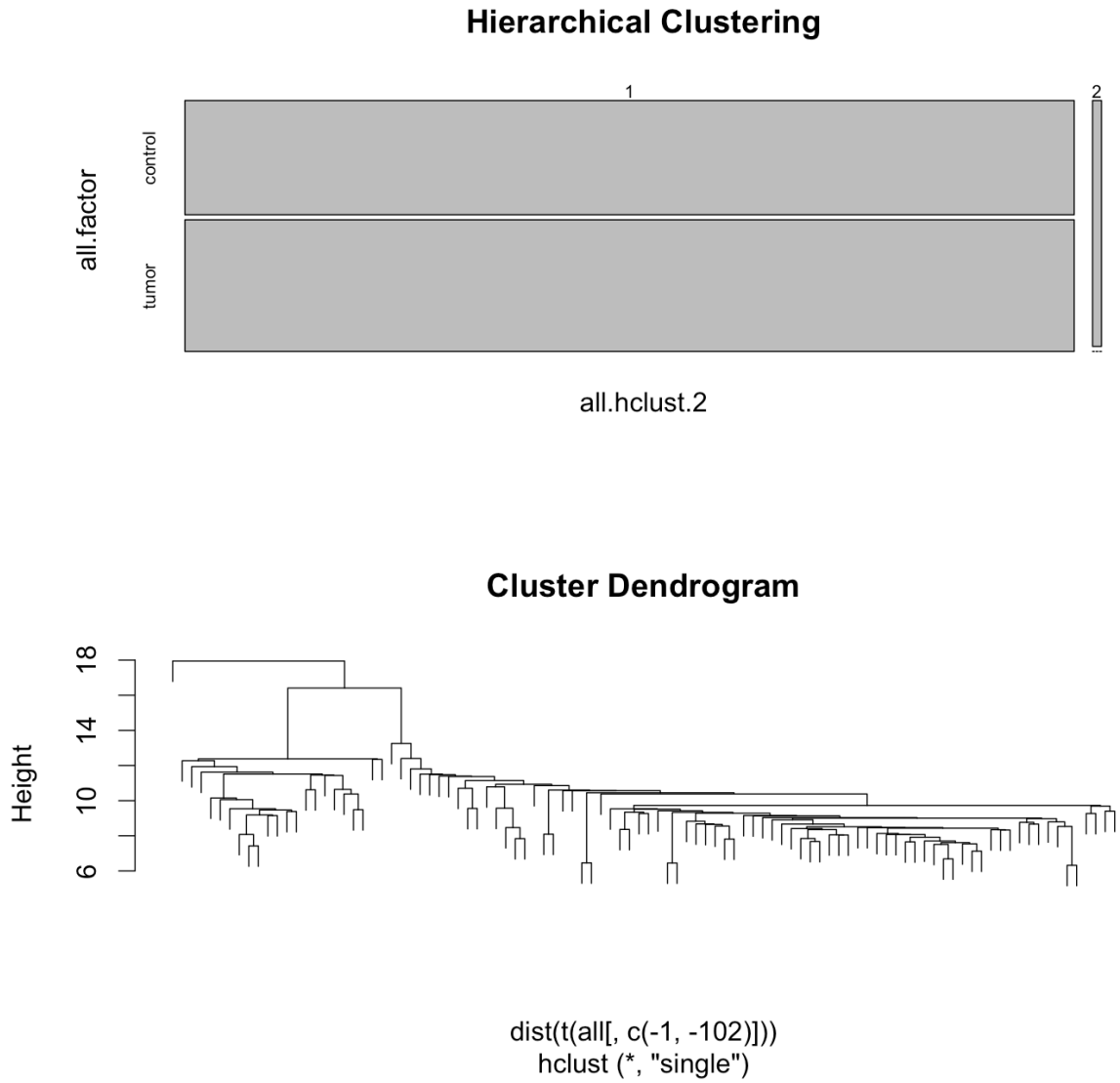

Figure 10. Hierarchical Clustering analysis to explore the inner relationship of chosen genes.

Figure 11. Contingency table of Partition Around Mediods Analysis

Number of Gene	**Misclassification Rate**	**Sensitivity**	**Specificity**
50	0	1	1
500	0.03	0.974359	0.9583333
1000	0.06	0.9512195	0.9473684

Table 5. Evaluating the performance of different size of gene set got from SVM-RFE analysis.

Probe Name	Gene Name	Up-/Down- Regulation
201373_at	PLEC	Up-Expression
214850_at	SMA4	
214052_x_at	PRRC2C	
204273_at	EDNRB	
204793_at	GPRASP1	
209215_at	MFSD10	
220319_s_at	MYLIP	
1556277_a_at	PAPD4	
202071_at	SDC4	
1552611_a_at	JAK1	
211034_s_at	HECTD4	
204270_at	SKI	
203705_s_at	FZD7	
203313_s_at	TGIF1	
203664_s_at	POLR2D	
201654_s_at	HSPG2	
205171_at	PTPN4	
203354_s_at	PSD3	
205904_at	MICA	
201461_s_at	MAPKAPK2	
204028_s_at	RABGAP1	
201340_s_at	ENC1	
203062_s_at	MDC1	

214436_at	FBXL2	
230499_at	BIRC3	
1558700_s_at	ZNF260	
205250_s_at	CEP290	
216641_s_at	LAD1	
200632_s_at	NDRG1	
213471_at	NPHP4	
1557065_at	YLPM1	
204273_at	EDNRB	
207069_s_at	SMAD6	
1598_g_at	GAS6	
1555294_a_at	ERC1	Down expression
1554335_at	CYTH4	
1554894_a_at	PCBD2	
1566785_x_at	NSF	
1556601_a_at	SPATA13	
201883_s_at	B4GALT1	
206572_x_at	ZNF85	
204333_s_at	AGA	
210963_s_at	GYG2	
227828_s_at	EVA1A	
213484_at	ADD2	

Table 6. Annotation the probe to genes

References

1. Hariharan, D., A. Saied, and H.M. Kocher, *Analysis of mortality rates for pancreatic cancer across the world.* HPB (Oxford), 2008. **10**(1): p. 58-62.
2. Le, N., et al., *Prognostic and predictive markers in pancreatic adenocarcinoma.* Dig Liver Dis, 2016. **48**(3): p. 223-30.
3. *Top Melanoma Biomarkers*. Available from: https://www.cancercommons.org/patients-caregivers/melanoma/top-melanoma-biomarkers/.
4. Paik, S., et al., *A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer.* N Engl J Med, 2004. **351**(27): p. 2817-26.
5. Jamieson, N.B., et al., *Tissue biomarkers for prognosis in pancreatic ductal adenocarcinoma: a systematic review and meta-analysis.* Clin Cancer Res, 2011. **17**(10): p. 3316-31.
6. Ma, M.Z., et al., *Candidate microRNA biomarkers of pancreatic ductal adenocarcinoma: meta-analysis, experimental validation and clinical significance.* J Exp Clin Cancer Res, 2013. **32**: p. 71.
7. Winter, J.M., C.J. Yeo, and J.R. Brody, *Diagnostic, prognostic, and predictive biomarkers in pancreatic cancer.* J Surg Oncol, 2013. **107**(1): p. 15-22.
8. Wang, J., et al., *Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells: a meta-analysis of three microarray studies.* Bioinformatics, 2004. **20**(17): p. 3166-78.
9. Guyon, I., et al., *Gene Selection for Cancer Classification Using Support Vector Machines.* Machine Learning, 2002. **46**(1/3): p. 389-422.