Classification

In this lab, you will analyze an Affymetrix lung cancer data set that I have already preselected genes/probes. There are 3 different tissue types that gene expression information was generated for: small cell lung carcinoma (SCLC), adenocarcinoma, and normal healthy control. Like many microarray analysis studies in cancer, the investigators are interested in identifying transcripts that are both differentially expressed between different cancer types and normal tissue and can be subsequently used to classify unknown tissue into the appropriate cancer type.

For the analysis, you will calculate one of the classic classifier algorithms - linear discriminant analysis (LDA). The objective is to train a model on a subset of the arrays, then test the predictability of this model on the remaining arrays, for identifying the correct cancer (and normal) class. This type of classification problem is common in microarray analysis, when attempting to design a diagnostic or prognostic test based on transcripts.

1.) Obtain the GEO lung cancer data set from the course website and read in the data. The column headers explain what each sample type is. (Link #23 or #22)

```
>FilePath <- "C:/Users/fermi/Documents/Fall 2020/Lab 9/lung_cancer.txt"
>dat<- read.table(FilePath, header=T, row.names=1)
```

2.) Load the MASS library and create a variable that has the class names for the 3 classes in the data. This variable should be of length 24. Next, bind the variable that you just created to a transposition of the data matrix using the data.frame() function. Note: the dimensions of this final data matrix should be 24 x 3014.

```
> library(MASS)
```

```
>class=c("adeno","adeno","adeno","adeno","adeno","adeno","adeno","ad
eno", "adeno","adeno","SCLC","SCLC","SCLC","SCLC","SCLC",
"SCLC","SCLC","SCLC","SCLC","Normal","Normal","Normal","Nor
mal","Normal")
```

```
> clas<-names(dat)
> datx<-as.data.frame(t(dat))
>datx<-data.frame(class,datx)
```

3.) Now create 2 separate data matrices from the matrix created in #2 above – a training set and a test set. The training set should include the following number of samples from each class: first 6 adenocarcinomas, first 6 SCLC, and first 3

normals. The test set should include the remaining samples. Put the first column of the test matrix that you just created into a new variable, since this will be the actual sample classes. Then remove the first column from the test set.

>traindat<-datx[1:6,]
>traindat<-rbind(traindat, datx[11:16,])
>traindat<-rbind(traindat, datx[20:22,])

>testdat<-datx[7:10,]
>testdat<-rbind(testdat, datx[17:19,])
>testdat<-rbind(testdat, datx[23:24,])

> testclass<-testdat[,1]
> testdat<-testdat[,-1]

4.) Now we want to run a classifier to see if we can predict the lung cancer types and discriminate them from both each other and the normal samples. We will train the model on the training set and use the test set as our model accuracy assessment.

Use the lda() function and train the model using the training set, but ONLY use the first 2 genes. Predict the test set sample using the predict() function. This function only requires 2 arguments – the variable that you saved the model construction in and the test set. Make sure to only select the first 2 genes (i.e. columns) in the test set, when predicting the sample classes. Use the table() function to see the confusion matrix. Hint: if you saved your output from the predict function in the variable "out" and you called the variable created in #3 "lab", then you use table like this:

> table(out$class,lab)

How many total samples are misclassified?

#5 samples were misclassified

> traindat.lda<-lda(clas~X1007_s_at + X1053_at,traindat)
> testdat.pred<-predict(traindat.lda, testdat[,1:2])

$class
[1] adeno SCLC SCLC SCLC SCLC SCLC adeno adeno Normal

vs:
[1] "adeno" "adeno" "adeno" "adeno" "SCLC" "SCLC" "SCLC" "Normal" "Normal"

#5 samples were misclassified

5.) Now plot the first 2 discriminant functions versus each other in an xy plot. These vectors are saved in the object "x" from your predict() output. For example, if you used the following code to predict the samples:

> lda.test <- predict(lda.train,test[,1:2])

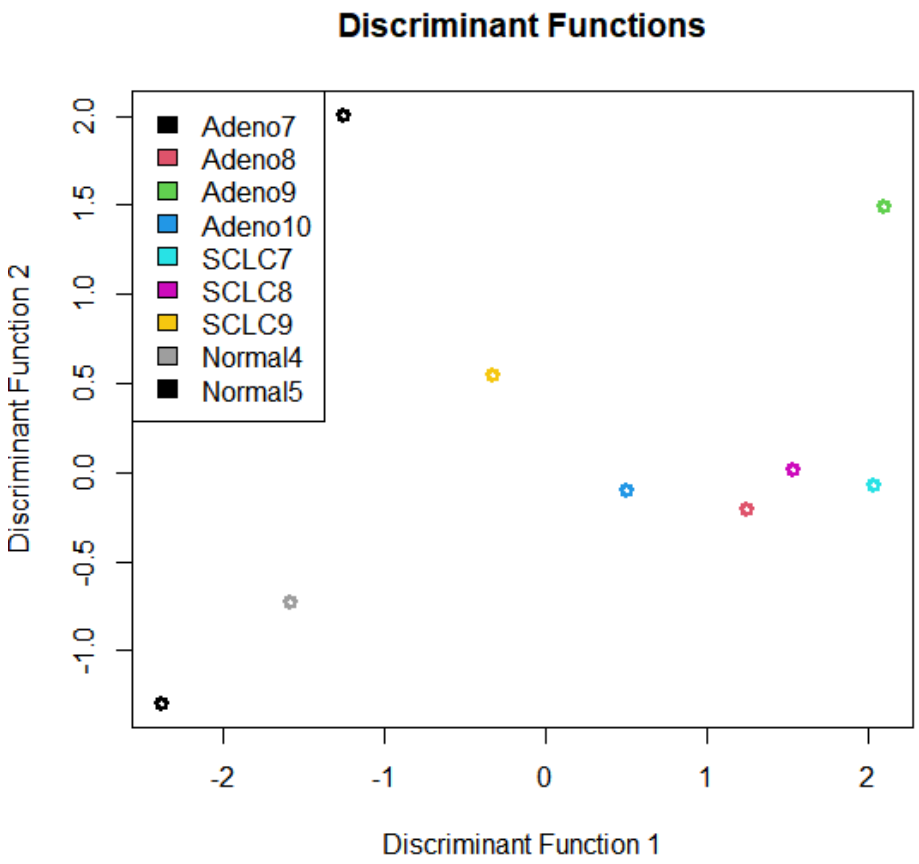
you can access the 2 vectors using:

> lda.test$x

Make sure to title the plot accordingly, color the points their appropriate cancer/normal type, and put a legend on the plot.

> plot(testdat.pred$x[,1], testdat.pred$x[,2], main='Discriminant Functions', xlab='Discriminant Function 1', ylab='Discriminant Function 2', col=1:length(c(rownames(testdat.pred$x))), lwd=3)

> legend("topleft",legend=c(rownames(testdat.pred$x)), fill=1:length(c(rownames(testdat.pred$x))))



6 and 7.) Now repeat #4 and #5 using all of the genes in the matrix as opposed to the first 2.

```
> traindat.all.lda<-lda(clas~.,traindat)
> testdat.all.pred<-predict(traindat.all.lda, testdat)
```

#0 misclassifications using all genes!!

```
> plot(testdat.all.pred$x[,1], testdat.all.pred$x[,2], main='Discriminant
Functions', xlab='Discriminant Function 1', ylab='Discriminant Function 2',
col=1:length(c(rownames(testdat.all.pred$x))), lwd=3)

> legend(0,0,legend=c(rownames(testdat.all.pred$x)),
fill=1:length(c(rownames(testdat.all.pred$x))))
```