Assignment 3

Philip Harmuth

2018-01-26

Exploratory questions

First we load the data and packages.

```
library(msgl)
```

```
## Loading required package: Matrix

## Loading required package: sglOptim

## Loading required package: foreach

## Loading required package: doParallel

## Loading required package: iterators

## Loading required package: parallel
```

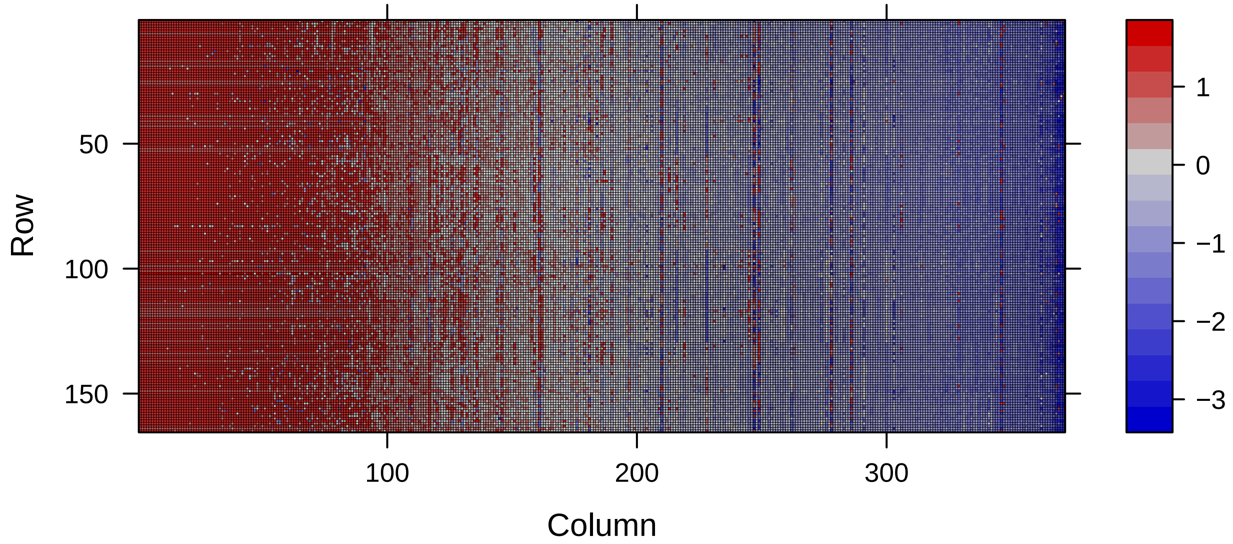
```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:Matrix':
##
##     expand
```

```
data(PrimaryCancers)
ord_class <- order(classes)
ord_mir <- order(colMeans(x), decreasing = TRUE)
x <- x[ord_class, ord_mir]
classes <- classes[ord_class]
image(Matrix(x))
```



Dimensions: 165 x 371

And plot the matrix.

Class means

First we calculate the class means for each column

```r
class.col.means <- data.frame(class = classes) %>%
  bind_cols(as.data.frame(x)) %>%
  group_by(class) %>%
  summarise_all(mean)
```

and plots the column means for each class

```r
plot.cols <- class.col.means %>%
  gather(col,means,-class) %>%
  group_by(class) %>%
  mutate(col.num = row_number()) %>%
  arrange(class,col.num)

ggplot(plot.cols,aes(x=col.num,y=means)) +
  geom_line() + facet_wrap(~class)
```

We see generally declining means over the columns for each class with some noise, which is expected since column of X are ordered by declining means.

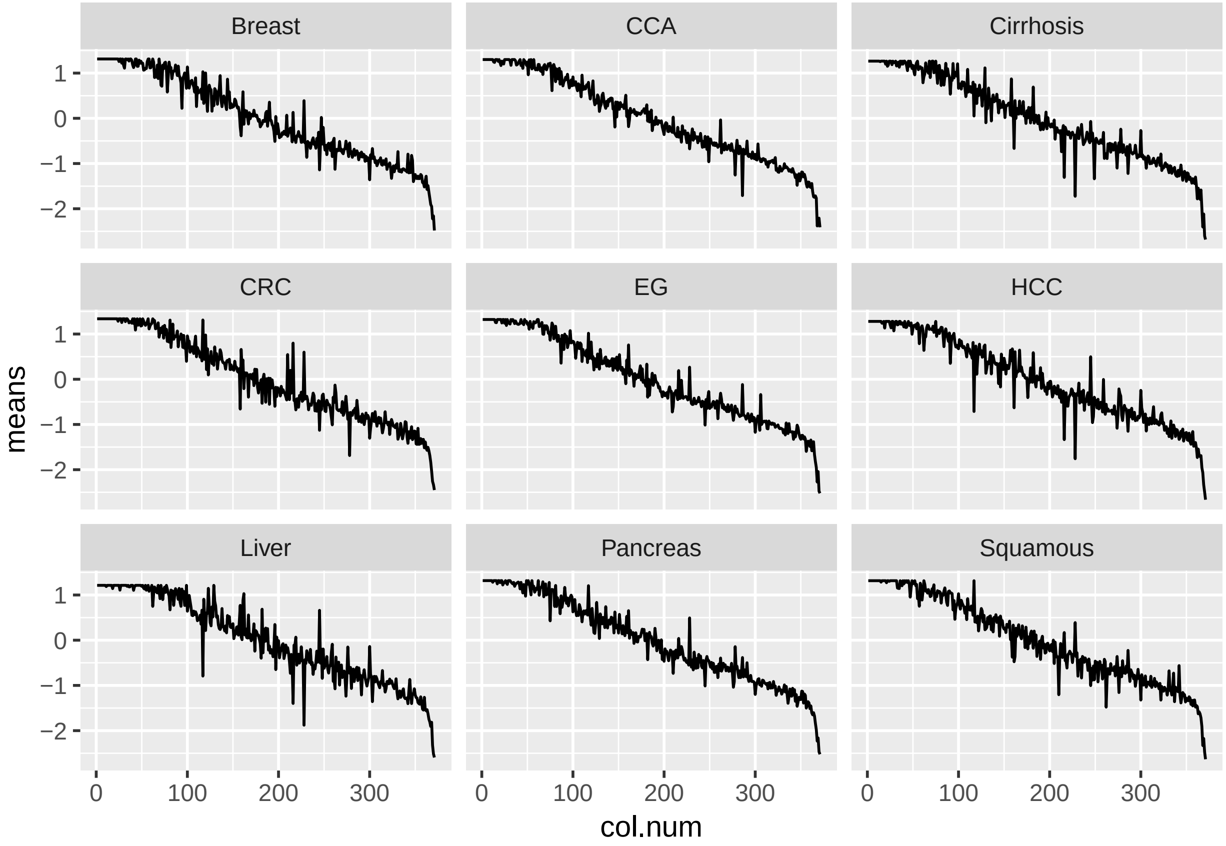


Figure 1: Columns means by class and column

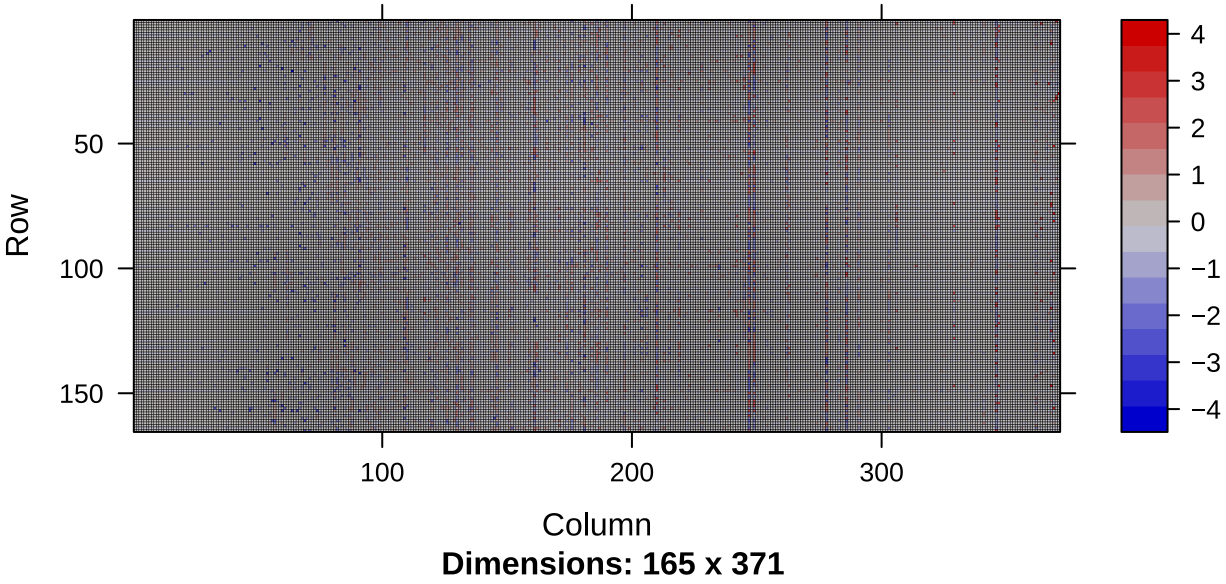


Figure 2: Image of residual matrix

PCA on the residual matrix

We now do PCA on the residual matrix by first subtracting the class means per columns from X and then using the singular value decomposition of the residual matrix so calculate the principal components.

```
X.mean.matrix <- data.frame(class = classes) %>%
  left_join(class.col.means) %>%
  select(-class)
```

```
## Joining, by = "class"
```

```
X.residuals <- x-X.mean.matrix
X.residuals.svd <- svd(X.residuals)
pc <- t(X.residuals.svd$d * t(X.residuals.svd$u)) %>%
  as.data.frame() %>%
  bind_cols(data.frame(class = classes))
image(Matrix(as.matrix(X.residuals)))
```

We now plot the first and second principal component against eachother with each observation colored according to class.

```
ggplot(pc, aes(V1,V2,col=class)) +
  geom_point() +
  ylab("2nd PC") +
  xlab("1st PC")
```

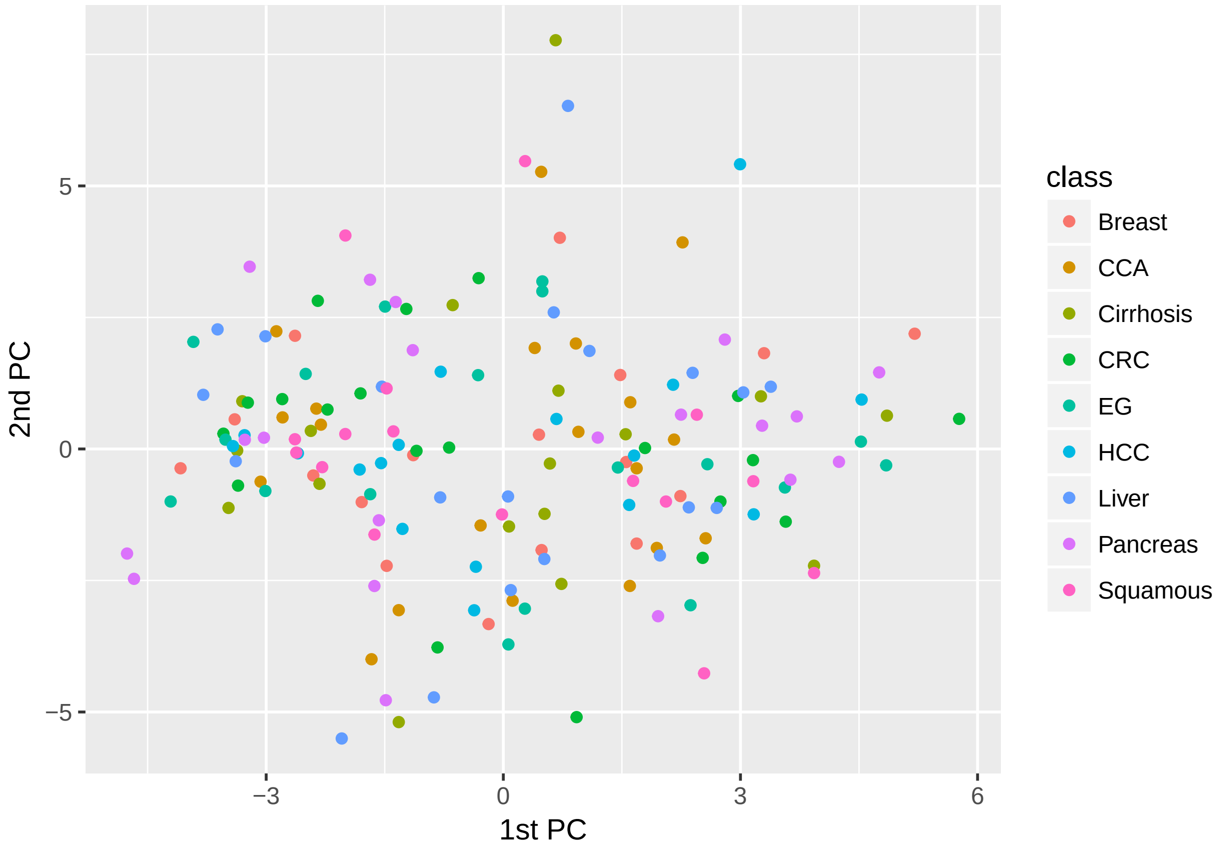


Figure 3: First and second PC

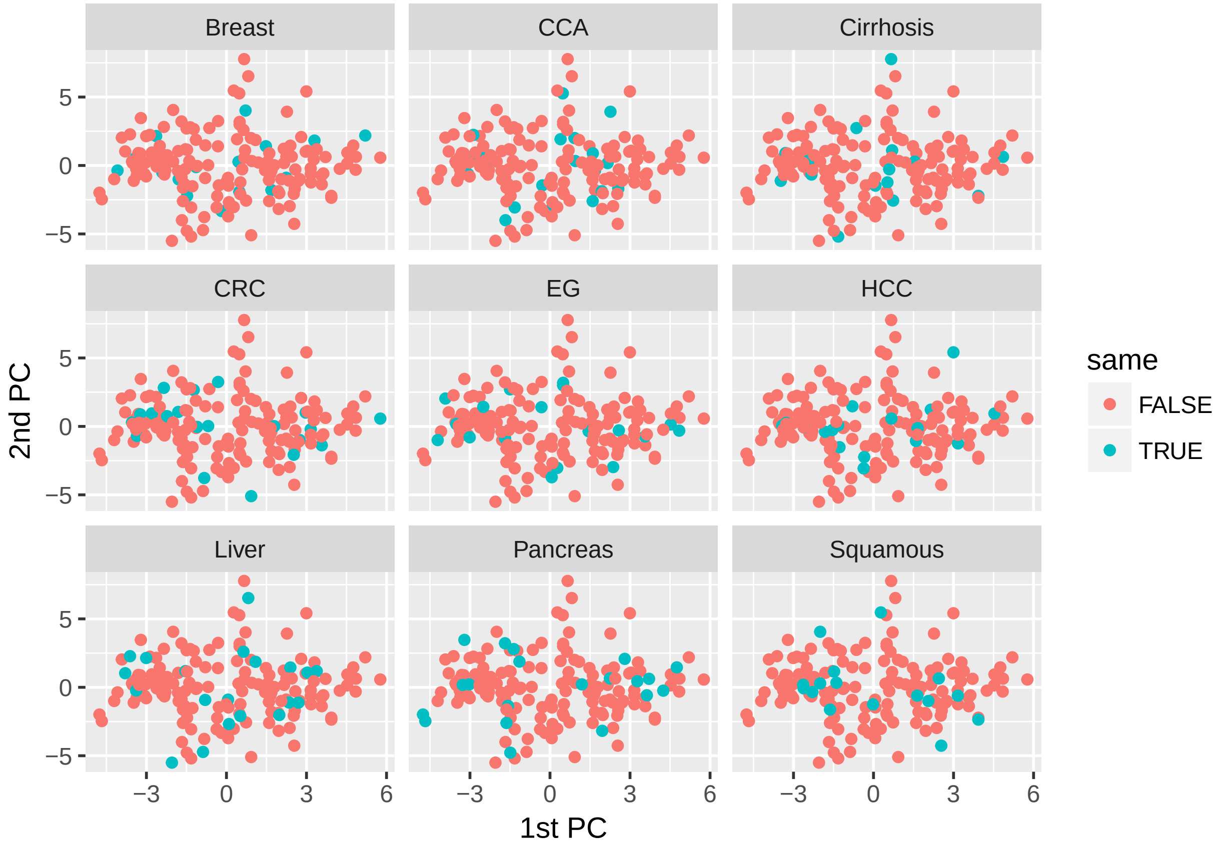


Figure 4: One vs rest plots for 1st and 2nd PC.

From this plot we see that the first two principal captures variance of the residual matrix in more or less perpendicular directions making nice round'ish cloud. From this cluster we can't quite see if some classes cluster together. To more clearly see if this is the case we make same plot for each class where we distinguish between points belonging to the class or not.

```
class.pc <- data.frame(class = rep(unique(classes),each=length(pc$class)),
           target =  rep(pc$class,length(unique(classes))),
           pc1 = rep(pc$V1,length(unique(classes))),
           pc2 = rep(pc$V2,length(unique(classes)))) %>%
  mutate(same = class == target)

ggplot(class.pc, aes(pc1,pc2,col=same)) +
  geom_point() +
  facet_wrap(~class)+
  ylab("2nd PC") +
  xlab("1st PC")
```

Here we don't see any significant clustering of certain classes and we can't use the 1st and 2nd PC to seperate points belonging to different classes.