Data integration

Urminder Singh

May 3, 2018

Data integration using PCA

First I did PCA on my data sets. My, dataset looked separable in first two PC.

```
library(RColorBrewer)
library(scales)
library(readr)
library("igraph", lib.loc = "~/R/win-library/3.4")
library("ggfortify", lib.loc = "~/R/win-library/3.4")


## do PCA and cluster
protein_data <- read.table("Protein.txt", sep = ",", header = T)
rownames(protein_data) <- protein_data[, 1]
protein_data$Gene <- NULL
colnames(protein_data) <- paste("prot", as.character(1:length(protein_data)),
    sep = "")

gene_data <- read.table("Urminder.txt", sep = ",", header = F)
rownames(gene_data) <- gene_data[, 1]
gene_data$V1 <- NULL
colnames(gene_data) <- paste("gene", as.character(1:dim(gene_data)[2]), sep = "")

modified_protein_data <- subset(protein_data, (rownames(protein_data) %in% rownames(gene_data)))
modified_protein_data <- modified_protein_data[order(match(rownames(modified_protein_data),
    rownames(gene_data))), ]

gene_data_princomp <- prcomp(gene_data)
gene_data_princomp$sdev
```

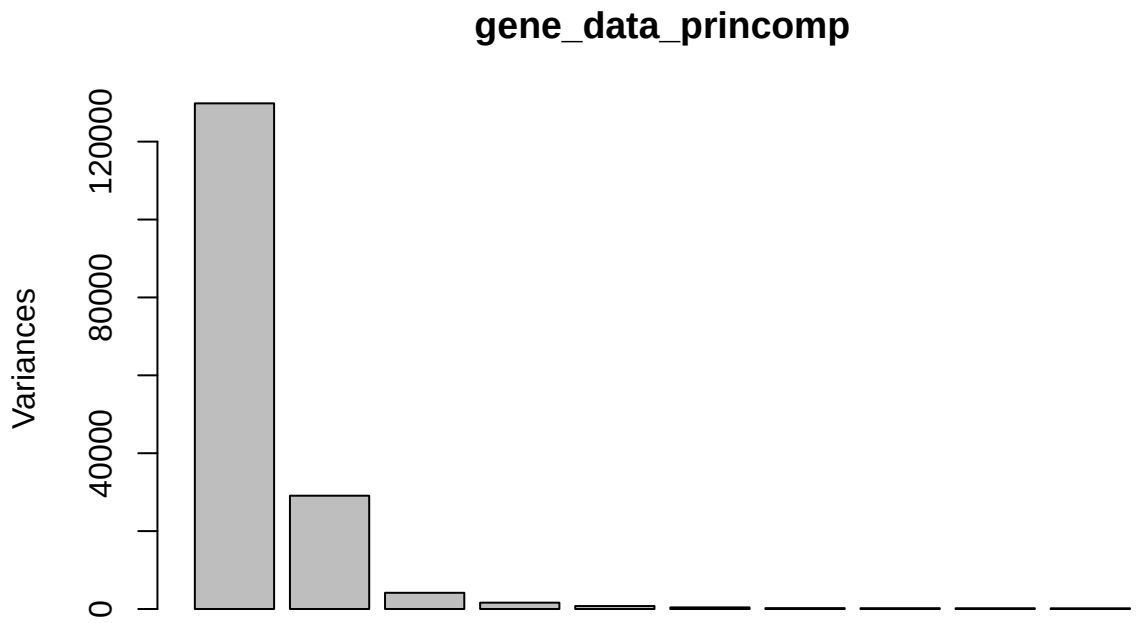
```
##  [1] 360.316076 170.570432  64.495121  40.382371  27.532421  20.328884
##  [7]  15.041356  14.444208  13.759547  13.316302  11.564073   9.955908
## [13]   9.101125   8.458803   8.383510   7.835331   7.054475   6.428347
```

```
summary(gene_data_princomp)
```

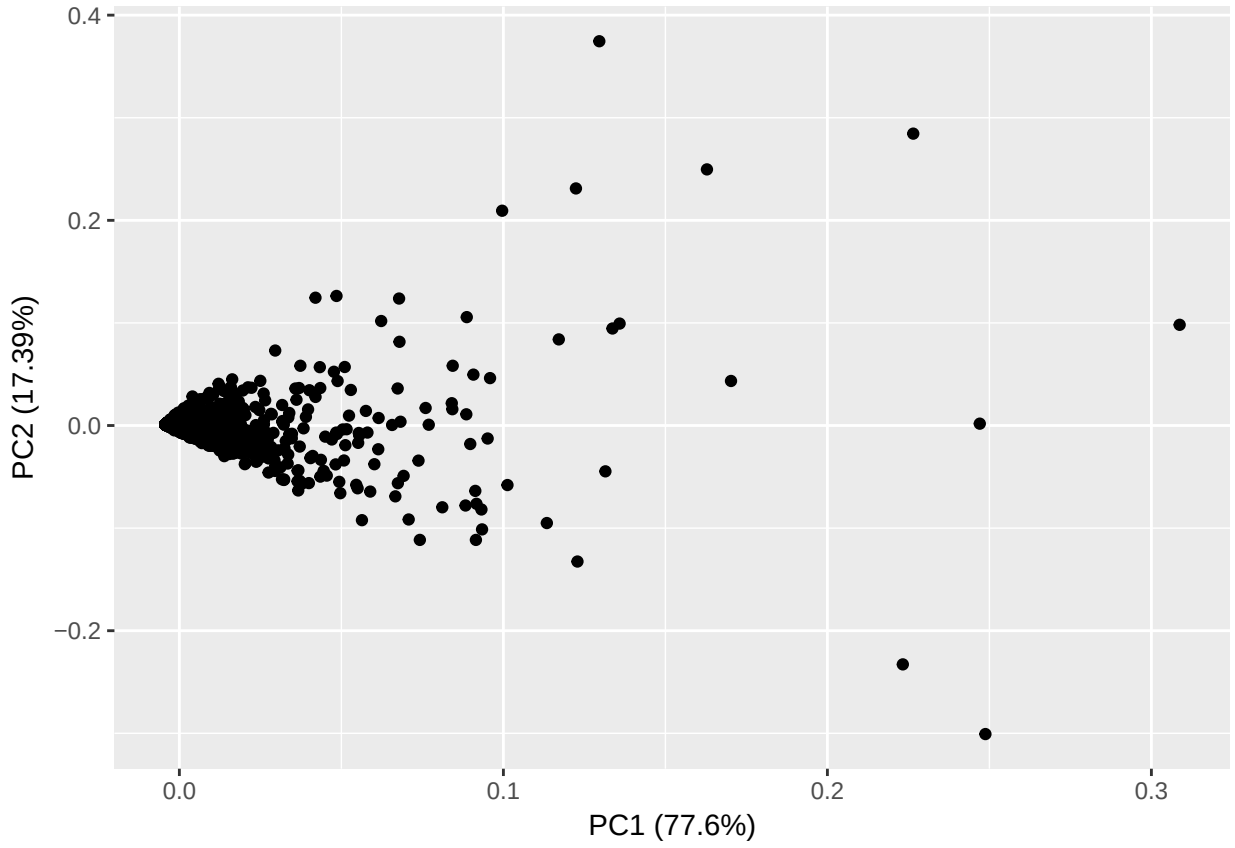
```
## Importance of components:
##                             PC1      PC2      PC3      PC4      PC5
## Standard deviation     360.316 170.5704 64.49512 40.38237 27.53242
## Proportion of Variance   0.776   0.1739  0.02486  0.00975  0.00453
## Cumulative Proportion    0.776   0.9499  0.97481  0.98456  0.98909
##                             PC6      PC7      PC8      PC9     PC10
## Standard deviation     20.32888 15.04136 14.44421 13.75955 13.31630
## Proportion of Variance  0.00247  0.00135  0.00125  0.00113  0.00106
## Cumulative Proportion   0.99156  0.99292  0.99416  0.99529  0.99635
##                            PC11    PC12   PC13    PC14    PC15    PC16
## Standard deviation     11.5641 9.95591 9.1011 8.45880 8.38351 7.83533
## Proportion of Variance  0.0008 0.00059 0.0005 0.00043 0.00042 0.00037
```

```
## Cumulative Proportion   0.9971 0.99775 0.9982 0.99867 0.99909 0.99946
##                           PC17    PC18
## Standard deviation     7.0545 6.42835
## Proportion of Variance 0.0003 0.00025
## Cumulative Proportion  0.9998 1.00000
```

```
plot(gene_data_princomp)
```




```
autoplot(gene_data_princomp)
```




```
protein_data_princomp <- prcomp(modified_protein_data)
protein_data_princomp$sdev
```

```
##  [1] 6898.342084 1703.695397 1002.853134  663.486355  452.060609
##  [6]  336.112518  271.203533  235.482514  166.871332  149.554317
## [11]  123.271845  105.560789   63.730017   59.238677   54.708824
## [16]   50.502845   47.278183   43.964010   39.799707   35.531645
## [21]   35.010075   32.339500   31.484667   29.432591   28.932963
## [26]   26.404680   26.077002   25.254523   24.668188   23.125259
## [31]   22.285150   22.109131   21.233764   20.496756   20.376578
## [36]   19.248365   18.450566   18.011888   17.479869   17.149242
## [41]   16.776924   16.455807   15.714343   15.371303   15.158995
## [46]   14.688860   14.464465   14.097894   13.651137   13.081357
## [51]   12.944349   12.685668   12.306176   12.043728   11.691841
## [56]   11.534715   11.112884   10.942634   10.760879   10.548111
## [61]   10.434852   10.289533   10.136478    9.665030    9.424115
## [66]    9.326926    8.957756    8.824237    8.454055    8.294230
## [71]    8.047271    7.882738    7.816768    7.571425    7.523266
## [76]    7.419548    7.221595    7.088708    6.944116    6.676578
## [81]    6.311712    6.273805    6.127517    5.945582    5.658020
## [86]    5.623398    5.423458    5.056695
```

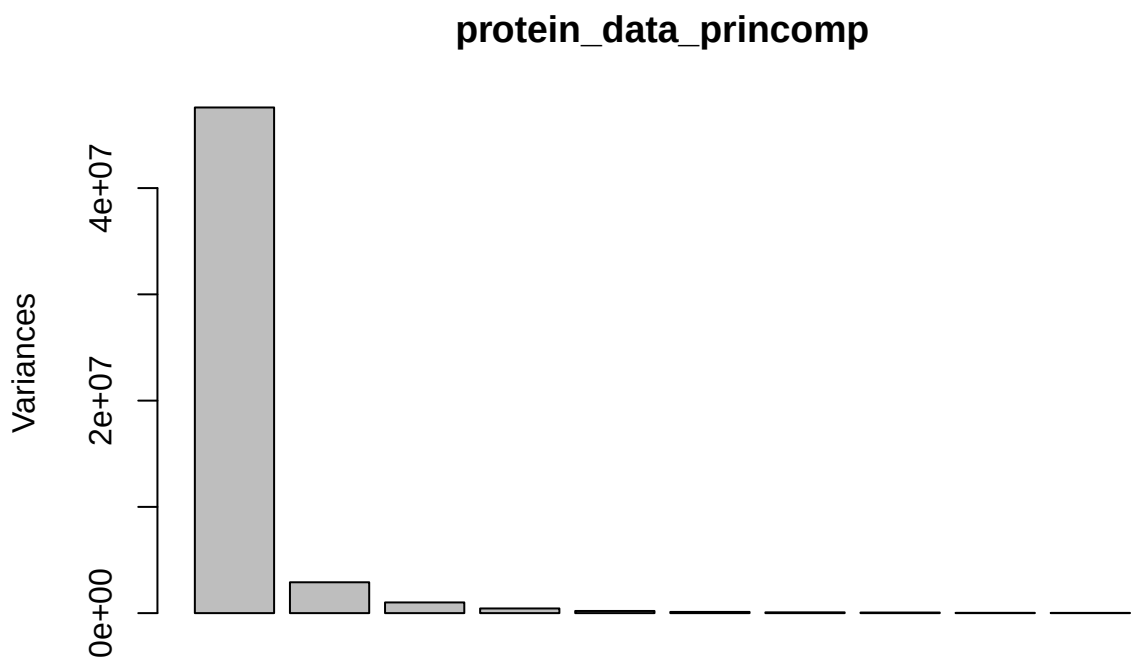
```
summary(protein_data_princomp)
```

```
## Importance of components:
##                             PC1       PC2       PC3       PC4       PC5
## Standard deviation     6898.3421 1.704e+03 1.003e+03 663.48635 452.06061
```

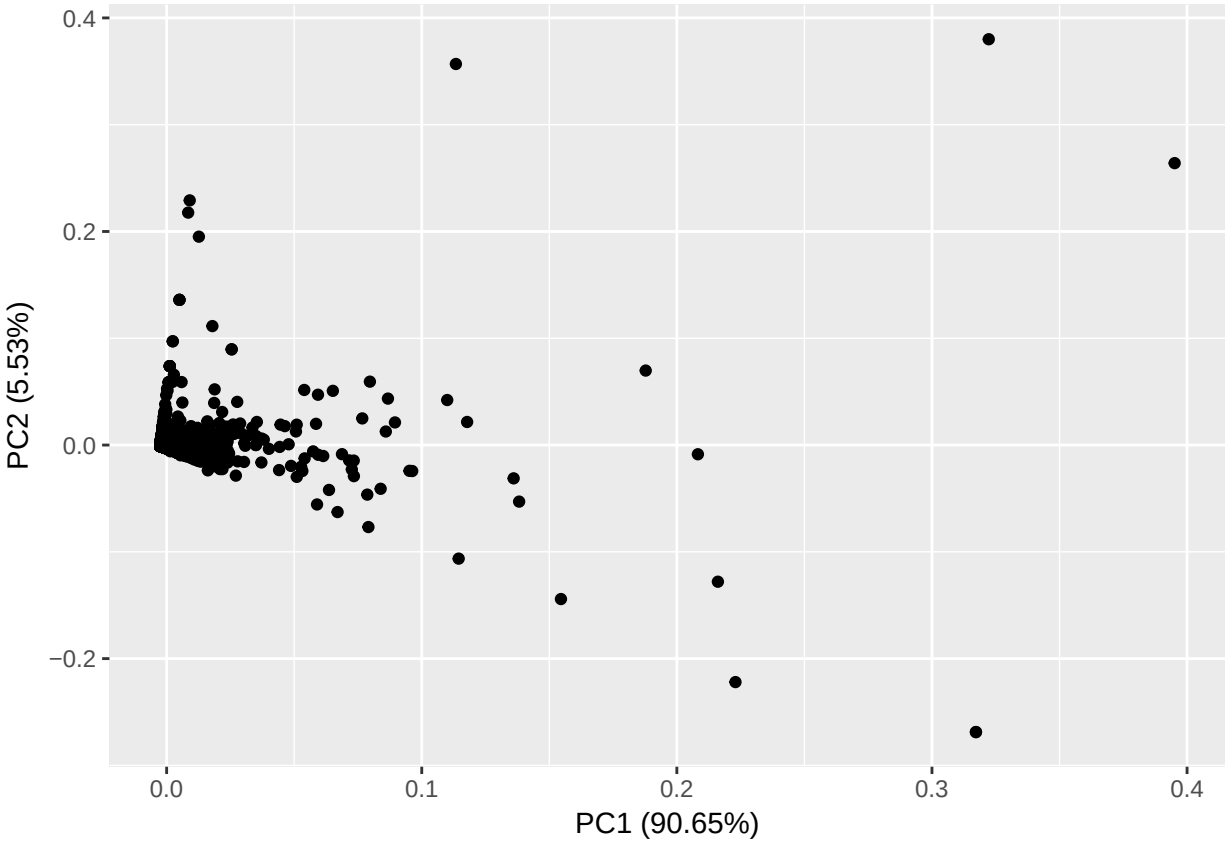
```
## Proportion of Variance    0.9065 5.529e-02 1.916e-02   0.00839   0.00389
## Cumulative Proportion     0.9065 9.618e-01 9.809e-01   0.98933   0.99322
##                             PC6      PC7       PC8       PC9      PC10
## Standard deviation     336.11252 271.2035 235.48251 166.87133 149.55432
## Proportion of Variance   0.00215   0.0014   0.00106   0.00053   0.00043
## Cumulative Proportion    0.99537   0.9968   0.99783   0.99836   0.99878
##                            PC11      PC12     PC13     PC14     PC15
## Standard deviation     123.27184 105.56079 63.73002 59.23868 54.70882
## Proportion of Variance   0.00029   0.00021  0.00008  0.00007  0.00006
## Cumulative Proportion    0.99907   0.99929  0.99936  0.99943  0.99949
##                            PC16     PC17     PC18     PC19     PC20
## Standard deviation     50.50285 47.27818 43.96401 39.79971 35.53165
## Proportion of Variance  0.00005  0.00004  0.00004  0.00003  0.00002
## Cumulative Proportion   0.99954  0.99958  0.99962  0.99965  0.99967
##                            PC21     PC22     PC23     PC24     PC25
## Standard deviation     35.01008 32.33950 31.48467 29.43259 28.93296
## Proportion of Variance  0.00002  0.00002  0.00002  0.00002  0.00002
## Cumulative Proportion   0.99969  0.99971  0.99973  0.99975  0.99976
##                            PC26     PC27     PC28     PC29     PC30
## Standard deviation     26.40468 26.07700 25.25452 24.66819 23.12526
## Proportion of Variance  0.00001  0.00001  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99978  0.99979  0.99980  0.99981  0.99982
##                            PC31     PC32     PC33     PC34     PC35
## Standard deviation     22.28515 22.10913 21.23376 20.49676 20.37658
## Proportion of Variance  0.00001  0.00001  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99983  0.99984  0.99985  0.99986  0.99987
##                            PC36     PC37     PC38     PC39     PC40
## Standard deviation     19.24836 18.45057 18.01189 17.47987 17.14924
## Proportion of Variance  0.00001  0.00001  0.00001  0.00001  0.00001
## Cumulative Proportion   0.99987  0.99988  0.99989  0.99989  0.99990
##                            PC41     PC42    PC43    PC44    PC45    PC46
## Standard deviation     16.77692 16.45581 15.7143 15.3713 15.1590 14.6889
## Proportion of Variance  0.00001  0.00001  0.0000  0.0000  0.0000  0.0000
## Cumulative Proportion   0.99990  0.99991  0.9999  0.9999  0.9999  0.9999
##                            PC47    PC48    PC49    PC50    PC51  PC52  PC53
## Standard deviation     14.4645 14.0979 13.6511 13.0814 12.9443 12.69 12.31
## Proportion of Variance  0.0000  0.0000  0.0000  0.0000  0.0000  0.00  0.00
## Cumulative Proportion   0.9999  0.9999  0.9999  0.9999  0.9999  1.00  1.00
##                           PC54  PC55  PC56  PC57  PC58  PC59  PC60  PC61
## Standard deviation     12.04 11.69 11.53 11.11 10.94 10.76 10.55 10.43
## Proportion of Variance  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## Cumulative Proportion   1.00  1.00  1.00  1.00  1.00  1.00  1.00  1.00
##                           PC62  PC63  PC64  PC65  PC66  PC67  PC68  PC69
## Standard deviation     10.29 10.14 9.665 9.424 9.327 8.958 8.824 8.454
## Proportion of Variance  0.00  0.00 0.000 0.000 0.000 0.000 0.000 0.000
## Cumulative Proportion   1.00  1.00 1.000 1.000 1.000 1.000 1.000 1.000
##                           PC70  PC71  PC72  PC73  PC74  PC75 PC76  PC77
## Standard deviation     8.294 8.047 7.883 7.817 7.571 7.523 7.42 7.222
## Proportion of Variance 0.000 0.000 0.000 0.000 0.000 0.000 0.00 0.000
## Cumulative Proportion  1.000 1.000 1.000 1.000 1.000 1.000 1.00 1.000
##                           PC78  PC79  PC80  PC81  PC82  PC83  PC84  PC85
## Standard deviation     7.089 6.944 6.677 6.312 6.274 6.128 5.946 5.658
## Proportion of Variance 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## Cumulative Proportion  1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
```

```
##                           PC86  PC87  PC88
## Standard deviation     5.623 5.423 5.057
## Proportion of Variance 0.000 0.000 0.000
## Cumulative Proportion  1.000 1.000 1.000
```

```
plot(protein_data_princomp)
```

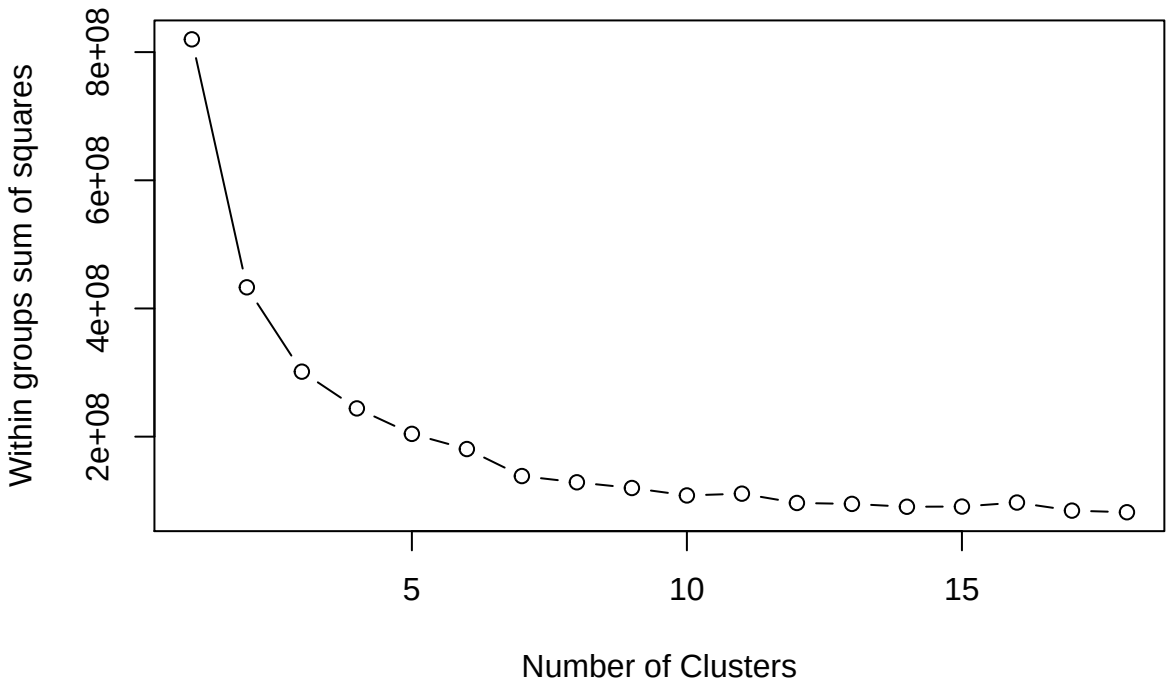



```
autoplot(protein_data_princomp)
```

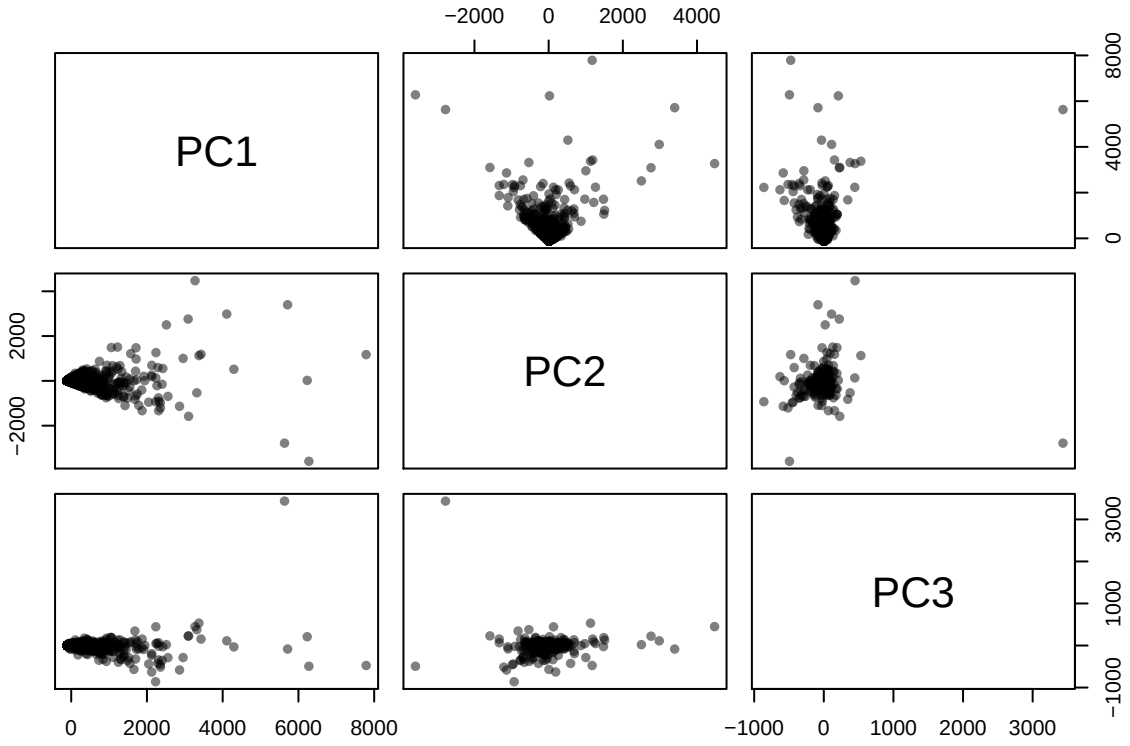


From the summary and plots we can see the most variation can be explaind using first 3 PC in both cases. Next I tried to cluster my data in this reduced space. The PCA space projects the data into a space which maximizes variance hence it is easy to cluster or separate our data. I used k-means clustering to cluster my data. To find k I plotted the within group sum-of-squares and picked the elbow point. There are other sophisticated methods like using the gap-statistic too. The code is shown below.

```
## find num clusters
wss <- (nrow(gene_data) - 1) * sum(apply(gene_data, 2, var))
for (i in 2:18) wss[i] <- sum(kmeans(gene_data, centers = i)$withinss)
plot(1:18, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")
```

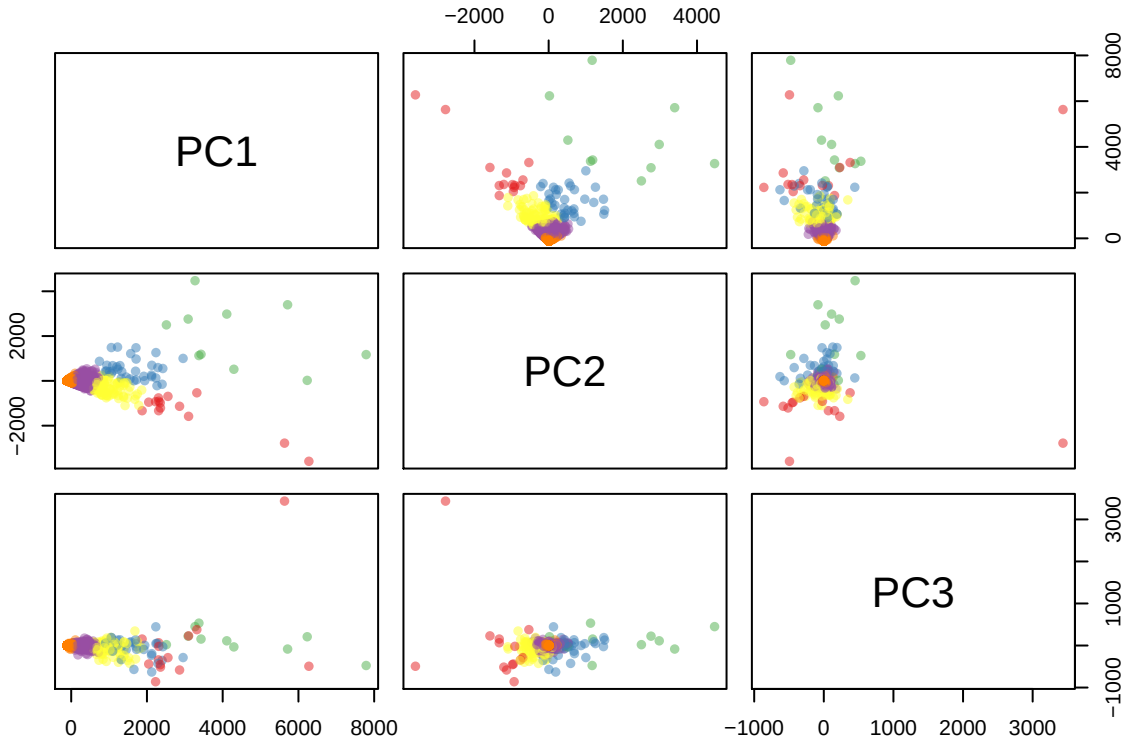



```
k = 6
# take first 3 PC as most variation is captured by first 3 PC
gcomp <- data.frame(gene_data_princomp$x[, 1:3])
plot(gcomp, pch = 16, col = rgb(0, 0, 0, 0.5))
```

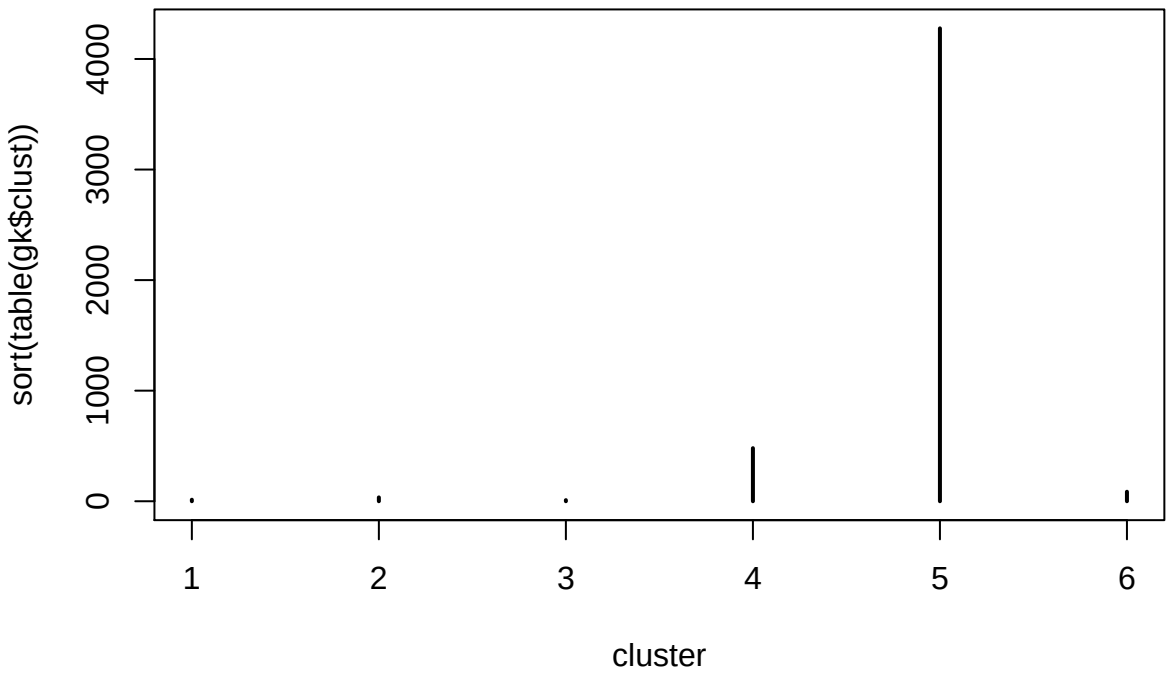



```r
gk <- kmeans(gcomp, 6, nstart = 25, iter.max = 1000)

palette(alpha(brewer.pal(9, "Set1"), 0.5))
plot(gcomp, col = gk$clust, pch = 16)
```




```
# get the genes in cluster
plot(sort(table(gk$clust)), xlab = "cluster")
```




```
clust <- names(sort(table(gk$clust)))
gc1 <- row.names(gene_data[gk$clust == clust[1], ])
gc2 <- row.names(gene_data[gk$clust == clust[2], ])
gc3 <- row.names(gene_data[gk$clust == clust[3], ])
gc4 <- row.names(gene_data[gk$clust == clust[4], ])
gc5 <- row.names(gene_data[gk$clust == clust[5], ])
gc6 <- row.names(gene_data[gk$clust == clust[6], ])

print(length(gc1))
```

```
## [1] 10
```

```
print(length(gc2))
```

```
## [1] 14
```

```
print(length(gc3))
```

```
## [1] 35
```

```
print(length(gc4))
```

```
## [1] 86
```

```
print(length(gc5))
```

```
## [1] 480
```

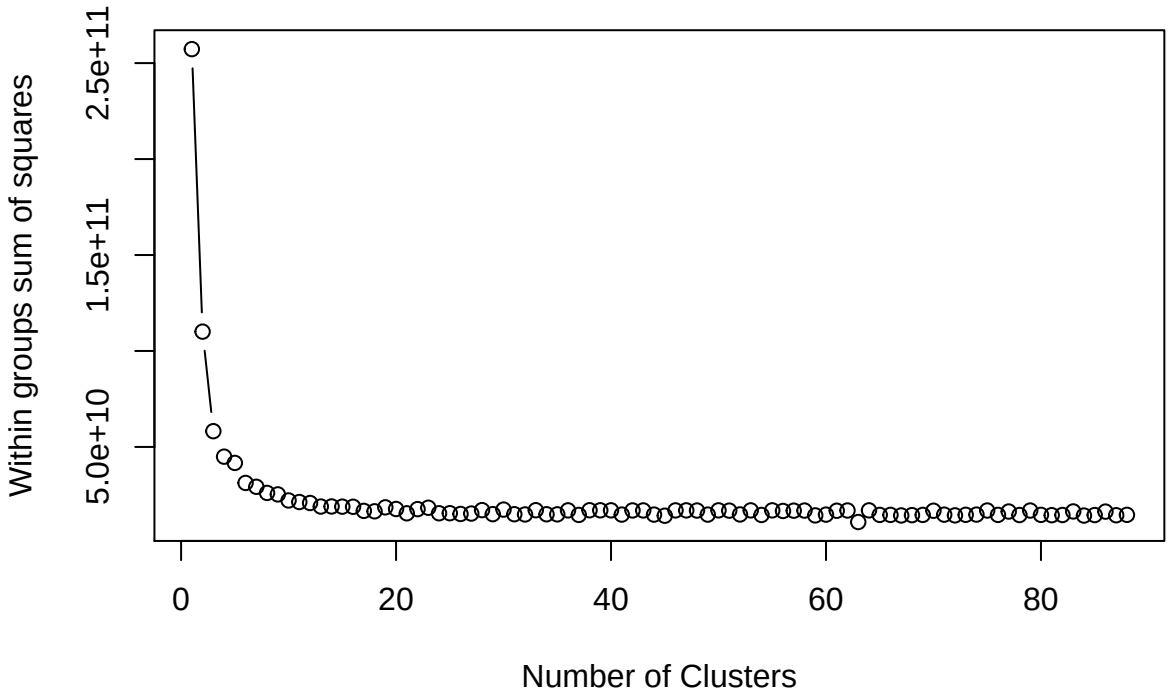
```
print(length(gc6))
```

```
## [1] 4277
```

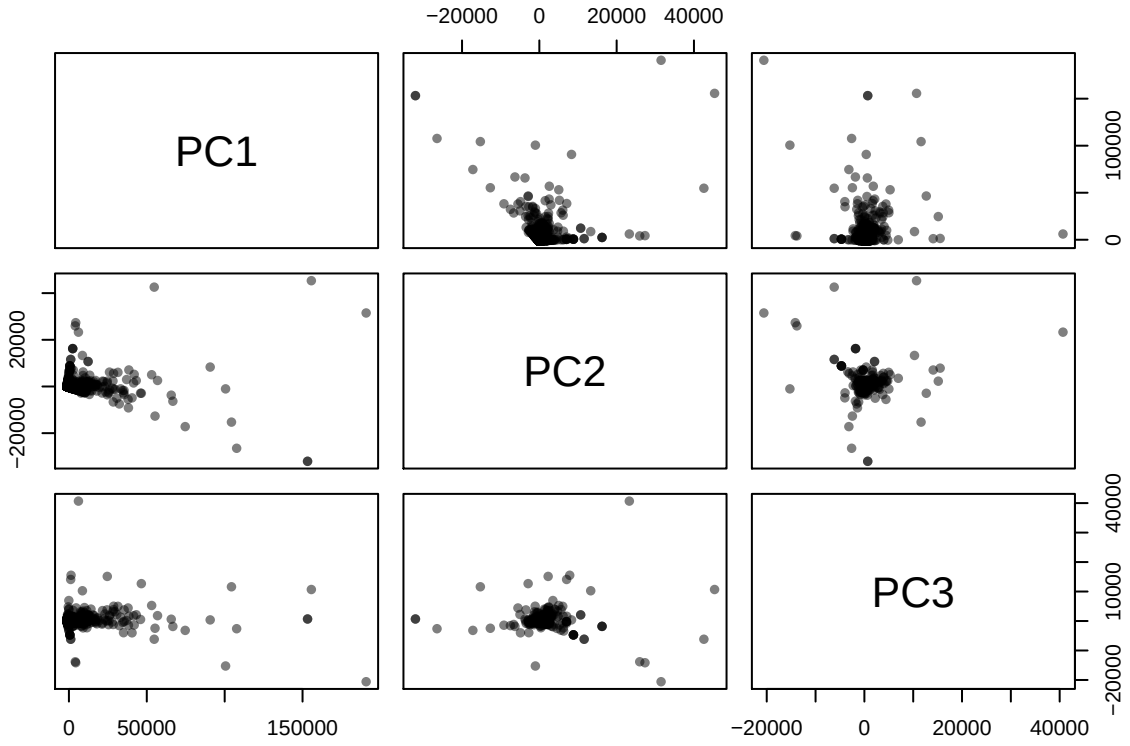
After doing k-means we can see we have 6 different clusters all of which look nicely separated from each other. I would beleive that these clusters represent genes which have somewhat similar expression patterns and hence they got clusters. my cluster gc6 is the biggest with 4277 genes.

I repeated the same k means clustering apprach on protein dataset.

```
# do for protein find num clusters
wss <- (nrow(modified_protein_data) - 1) * sum(apply(modified_protein_data,
    2, var))
for (i in 1:88) wss[i] <- sum(kmeans(modified_protein_data, centers = i)$withinss)
plot(1:88, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")
```

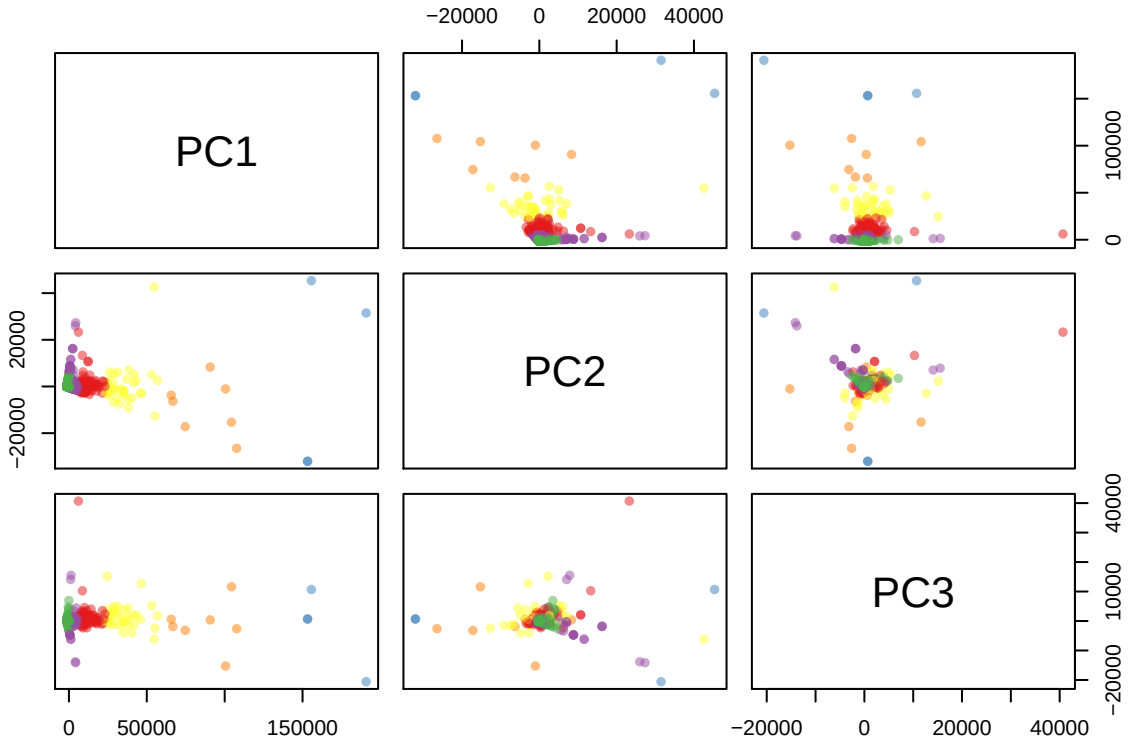



```
k = 6
# take first 3 PC as most variation is captured by first 3 PC
pcomp <- data.frame(protein_data_princomp$x[, 1:3])
plot(pcomp, pch = 16, col = rgb(0, 0, 0, 0.5))
```

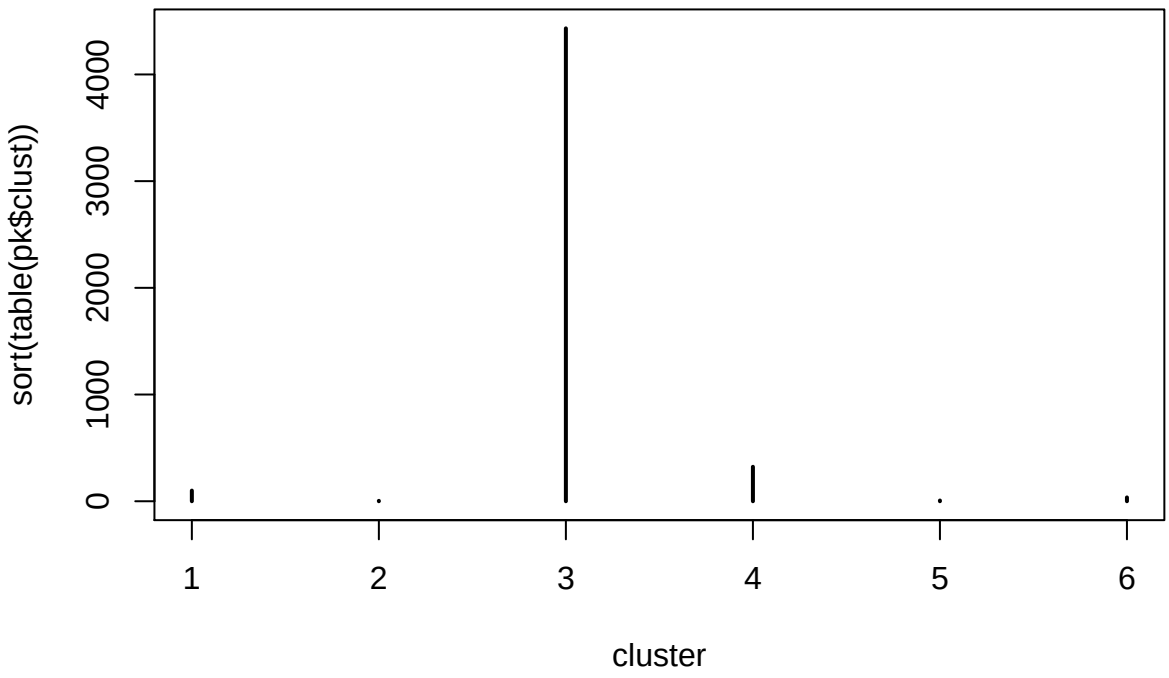



```
pk <- kmeans(pcomp, k, nstart = 25, iter.max = 1000)

palette(alpha(brewer.pal(9, "Set1"), 0.5))
plot(pcomp, col = pk$clust, pch = 16)
```




```
# get the genes in cluster
plot(sort(table(pk$clust)), xlab = "cluster")
```




```
pclust <- names(sort(table(pk$clust)))
pc1 <- row.names(gene_data[pk$clust == pclust[1], ])
pc2 <- row.names(gene_data[pk$clust == pclust[2], ])
pc3 <- row.names(gene_data[pk$clust == pclust[3], ])
pc4 <- row.names(gene_data[pk$clust == pclust[4], ])
pc5 <- row.names(gene_data[pk$clust == pclust[5], ])
pc6 <- row.names(gene_data[pk$clust == pclust[6], ])

print(length(pc1))
```

```
## [1] 4
```

```
print(length(pc2))
```

```
## [1] 7
```

```
print(length(pc3))
```

```
## [1] 36
```

```
print(length(pc4))
```

```
## [1] 100
```

```
print(length(pc5))
```

```
## [1] 323
```

```
print(length(pc6))
```

```
## [1] 4432
```

We see that the cluster sizes are similar for both protein and gene. I further took the intersection of the largest clusters and found 4045 genes overlap. I then took only these 4045 genes to build my network. The reason behind this is little intuitive. I am trying to get rid of genes and proteins which vary significantly in their expression profiles. Finally, I integrate two datasets by keeping common genes/proteins in the largest cluster. It is safe to assume that this set of genes behave in a coherent way assuming that there is a relationship between gene expression and the protein expression datasets. Further network analysis on this set of genes can reveal if my hypothesis is correct or not.

```
finalset <- intersect(pc6, gc6)

head(finalset)
```

```
## [1] "Mrpl15"  "Lypla1"  "Tcea1"   "Rgs20"   "Atp6v1h" "Rb1cc1"
```

Reference

Some part of my code was taken from this tutorial http://www.everydayanalytics.ca/2014/06/pca-and-k-means-clustering-of-delta-aircraft.html.