Genomics

Freie Universität Berlin, Institut für Informatik
Peter N. Robinson, Rosario M. Piro
Wintersemester 2015/2016

6. Übungsblatt vom 26. Januar 2016
Diskussion am 4. Februar 2016

---

1. ChIP-seq

   You have performed a ChIP-seq for cells *D. melanogaster* (fruit fly; genome size: 165 Mbp) and mapped a total of 3 million reads of length 36 bp. The average length of the DNA fragment from which the single-end reads were obtained was 152 bp. You roughly estimate that at a read length of 36 bp about 95% of the genome will mappable.

   (a) Estimate the overall coverage level and briefly describe the concepts on which the estimation is based.

   (b) With this estimation, what is the probability of observing a peak height of 3 or more reads by chance?

   (c) Why is the use of an overall coverage level, as used in (a) and (b), problematic and what can be done as an improvement?

2. FDR

   Explain the concept of false discovery rate and describe how it can be used to filter the peaks obtained in a ChIP-seq experiment (based on the simplified concepts used in the question 1).

3. ChIP-seq technology

   Write down the order of steps in the ChIP-seq procedure. Some of the followiing items belong to the same steps. Identify this in your answer. Some of the items are not related to ChIP-seq. Cross them out

   - Formaldehyde
   - Immunoprecipitation
   - Antibody
   - Cross link DNA and protein
   - 5′ end next generation sequencing
   - de novo motif inference
   - read mapping
   - homozygosity mapping
   - fragmentation
   - peak calling
   - variant calling

4. Principle components analysis

   In this exercise, we will perform PCA on a dataset that can downloaded here. http://www-genome.stanford.edu
   Download the file `Select_Elutriation.txt`. This is yeast RNA expression data that should show some nice clusters. Put the file in the directory from which you start R from (or adjust the path). The following code shows how to load the data and turn it into a matrix we can use for PCA. The authors

of the original paper performed a log transform on the inverted data, which we will not question here, but it will help us to get nice results.

Please note this exercise is exploratory and for you to gain intuition. You do not need to hand in your results, but please come prepared to discuss this!

```
> path<-"Select_Elutriation.txt"  ## adjust as necessary!
> dat <- read.table(path,header=FALSE)
> X <- as.matrix(dat)
> ngenes <- dim(X)[1]
> narrays <- dim(X)[2]
## 2) Log transform data
> Xt <- log(1/X)
```

We will first show how to use a built-in function in R, and then you will repeat some of the analysis using matrix operations.

To perform PCA on this data is as simple as the following line (use the names command to see what data is returned)

```
>  pca <- prcomp(Xt)
> names(pca)
[1] "sdev"     "rotation" "center"   "scale"    "x"
```

First of all you should plot a screeplot

```
 # Eigenvalues
eig <- (pca$sdev)^2
 # Variances in percentage
variance <- eig*100/sum(eig)
# Cumulative variances
cumvar <- cumsum(variance)
eig.active <- data.frame(eig = eig, variance = variance,
                     cumvariance = cumvar)
```

What do you think about the results of PCA analysis? Is this an effective dimensionality reduction? How many PCs do we need to explain "most" of the variability? The following code is a nice way of plotting this kind of data. Alternatively, you may try `head(eig.active)` or `> summary(pca)`.

```
 barplot(eig.active[, 2], names.arg=1:nrow(eig.active),
       main = "Variances",
       xlab = "Principal Components",
       ylab = "Percentage of variances",
       col ="steelblue")
# Add connected line segments to the plot
lines(x = 1:nrow(eig.active),
      eig.active[, 2],
      type="b", pch=19, col = "red")
```

We can now plot the array variables according to PC 1 and 2

We will use a helper function to extract loadings

```
# Correlation between variables and principal components
var_cor_func <- function(var.loadings, comp.sdev){
  var.loadings*comp.sdev
```

```
  }

# Variable correlation/coordinates
loadings <- pca$rotation
sdev <- pca$sdev

var.coord <- var.cor <- t(apply(loadings, 1, var_cor_func, sdev))
head(var.coord[, 1:4])
# Plot the correlation circle
a <- seq(0, 2*pi, length = 100)
plot( cos(a), sin(a), type = 'l', col="gray",
      xlab = "PC1",  ylab = "PC2")

abline(h = 0, v = 0, lty = 2)

# Add active variables
arrows(0, 0, var.coord[, 1], var.coord[, 2],
      length = 0.1, angle = 15, code = 2)

# Add labels
text(var.coord, labels=rownames(var.coord), cex = 1, adj=1)
```

Do it yourself To finish this exercise, examine and run the following code. Try to explain what it is doing in terms of the mathematics of PCA. There are three segments.

```
 covm <- cov(Xt)
e <- eigen(covm)

eval <- e$values


#  Plot results

x <- seq(1:narrays)  ## The numbers of the eigengenes
par(las=1, mar=c(4, 4, 4, 4))
plot.new()
plot.window(range(x), range(eval))
lines(x, eval)
points(x, eval, pch=16, cex=2)
par(col="grey50", fg="grey10", col.axis="grey10")
axis(1, at=seq(0, narrays, 1))
axis(2, at=seq(0, max(eval), 0.2))
box(bty="u")
mtext("Principle Component", side=1, line=2, cex=1.2)
mtext("Variance", side=2, line=2, las=0, cex=1.2)
```

```
covm <- cov(Xt)
e <- eigen(covm)


pc1 <- Xt %*% e$vectors[,1]
pc2 <- Xt %*% e$vectors[,2]

## Calculate max and min to make axes (using integer values)
pc1max <- ceiling(max(pc1))
```

```
pc1min <- floor(min(pc1))
pc2max <- ceiling(max(pc2))
pc2min <- floor(min(pc2))

pc1range <- seq(pc1min,pc1max,1)

#  Plot results


plot(pc1,pc2,pch='.',xlab='Principle Component 1',
        ylab='Principle Component 2',bty='o',
        xaxp=c(pc1min,pc1max,pc1max-pc1min),
        yaxp=c(pc2min,pc2max,pc2max-pc2min))
```

and finally

```
covm <- cov(Xt)
e <- eigen(covm)


## 4) Plot the first three eigenvectors

par(mfrow=c(3,1))

plot(e$vectors[,1],xlab='Time Point',ylab='Coeefficients',type='l',
        ylim=c(-0.15,0.65))
points(e$vectors[,1], pch=16, cex=2)
plot(e$vectors[,2],xlab='Time Point',ylab='Coeefficients',type='l',
        ylim=c(-0.8,0.4))
points(e$vectors[,2], pch=16, cex=2)
plot(e$vectors[,3],xlab='Time Point',ylab='Coeefficients',type='l',
        ylim=c(-0.6,0.6))
points(e$vectors[,3], pch=16, cex=2)
```