Visualization and normalization

Missing data

In this lab, you will work with one of the first microarray datasets, from 1998. Spellman *et al.* wrote a paper on transcriptomics during the cell cycle of yeast, tracking genome-wide expression after using a number of different methods to synchronize the cell cycle. They used a "home-made" type of two-color cDNA microarray. First, have a look at the abstract of the paper (PMCID PMC25624).

a. Start by downloading the data, from http://genome-www.stanford.edu/cellcycle/data/rawdata/combined.txt. You can do that by hand, or use read.table(url("http://genome-www.stanford.edu/cellcycle/data/rawdata/combined.txt"), ...) directly. Inspect the data using the commands you learned about last week. Also create an image of the data matrix. What do you notice?

b. What is the range of expression in the data? Hint: use na.rm=TRUE if you run into trouble.

c. Find the *fraction* of missing values. The *number* of missing values can be found using something like:

```
> length(which(is.na(as.matrix(data[,-1]))))
```

d. Clearly, we should deal with this missing data, but doing this yourself is out of the scope of this course. You can find a cleaned and imputed version of the Spellman dataset, imputed.txt, on Blackboard. Load and inspect it; how many samples and genes have been removed? Check whether any missing values are left.

e. Generate a heatmap (using heatmap(), or pheatmap() in the corresponding package). What do you observe?

Normalization

a. Visualize the expression distributions of all microarrays left in ndata using boxplots. What do you notice?

b. From here on, we will focus on the samples taken after alpha factor arrest. Create a new data frame "afdata" that contains just these microarrays and regenerate the boxplots. What do you see?

c. Mean-normalize the expression distributions using scale() and visualize the result.

d. Do the same for mean-variance normalization.

e. What type of normalization do you think is needed here? Create a new matrix "nrmdata" that contains the data normalized using the approach you selected.

Optional

Quantile normalization is an extreme form of normalization. All measurements are replaced by the average of all measurements with the same rank, over all samples. As a result, the distributions of the measurements will be **exactly** the same after quantile normalization.

f. In R, quantile normalization is included in the preprocessCore library in BioConductor:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("preprocessCore")
```

Use its function normalize.quantiles() to perform quantile normalization and visualize the results.

Visualization

a. For visualization, it helps to work with a subset of the data, to avoid too much computation. Select n genes (say, $n = 1000$) as follows:

```
> sds <- apply(nrmdata,1,sd)
> ind <- order(sds,decreasing=TRUE)
> subdata <- nrmdata[ind[1:n],]
```

What types of genes does this select?

b. Visualize "subdata" using heatmap() and pairs(). Do you notice anything peculiar? You may have to zoom in...

c. Corroborate your previous answer by visualizing the correlation matrix.

d. Use qplot() to visualize the first few genes (5 or so) as a function of time, in a scatterplot. Add a non-linear trend line. The time values, for the x axis, can be generated using seq(0,119,7).

e. To get more insight in the data, we can perform Principal Component Analysis. Use prcomp() to generate a PCA and use plot() and summary() to inspect the results. How many PCs are needed to explain at least 90% of the variation?

f. Generate a biplot. Can you say anything about the distribution of the genes (the points)? Do you notice anything about the samples (arrows)? Use scaling before performing the PCA to see if you can get a clearer view.

g. Inspect the first 9 PCs, by plotting the first columns of the matrix p$rotation. What do you notice? Hint: try to use par(mfrow=c(3,3)) to place all plots in a single figure.

Optional

a. Investigate how PCA results,in particular the fraction of retained variance, changes as a function of n, the number of genes included in subdata (see a. above).

b. Investigate how the PCA results (without scaling) differ between the normalized and original data.

c. Install and use FactoMineR to generate a PCA plot of subdata and t(subdata). Can you explain what you see?