— Gene Set Enrichment —

Ulrich Mansmann, Manuela Hummel

Practical DNA Microarray Analysis, March 2008, Heidelberg
http://compdiag.molgen.mpg.de/ngfn/pma2008mar.shtml

Abstract. *This is the tutorial for one of the exercises on day 4 of the course on Practical DNA Microarray Analysis. The main topic is the gene set enrichment method introduced by Lamb et al. (2003).*

Gene set enrichment

Load data and packages. The workspace provided under the given link contains the expression set `eSet`, a data frame `pathways.U133A` with pathway information and a function `gene.set.enrichment.permut.rfc` for the gene set enrichment analysis. The expression data and clinical information (e.g. sex, age, tumour grade etc.) of 36 colorectal cancer patients is given. Of special interest are the differences in expression between cancer stages UICC II and UICC III.

```
> load(url("http://compdiag.molgen.mpg.de/ngfn/data/2008/mar/grouptesting.RData"))
> ls()
> library(affy)
> library(multtest)
> str(eSet)
> dims(eSet)
> pData(eSet)
```

We are interested in whether special pathways (noted in the data frame `pathways.U133A`) are enriched with differentially expressed genes. Which pathways are considered and how many probesets belong to each pathway?

```
> pw <- pathways.U133A$PATHWAY
> table(pw)
```

In order to perform the gene set enrichment procedure, it is necessary to define two gene groups. Group A consists of all probe sets that are related to cancer specific pathways, group B consists of all other probe sets. Some probe sets belong to multiple pathways. How many belong to 2 or 3 different pathways? How many different probe sets are related to cancer relevant pathways?

```
> ps <- pathways.U133A$AFFYPROBESET
> table(table(ps))
> sum(table(table(ps)))
> length(unique(ps))
```

There are `nrow(exprs(eSet))` probesets in total. The number of probesets in group A is given by `length(unique(ps))`. A group index vector is defined by

```
> group.ind <- rep("B", nrow(exprs(eSet)))
> group.ind <- ifelse(rownames(exprs(eSet)) %in% ps, "A",
+     group.ind)
```

First, the differential gene expression for each single gene will be quantified by using the t-statistic (Welch test when not supposing a common variance). Differential expression between UICC II and UICC III patients is of interest. In the data frame `pData(eSet)` this information is given in column "UICC". For the function `mt.teststat` from the `multtest` library the group index has to be 0/1 coded. This coding is achieved by

```
> UICC.gr <- pData(eSet)$UICC - 2
> table(UICC.gr)
```

The primary interest is to solve the question if the probe sets in the cancer related pathways show more differential expression than the remaining probesets. Therefore the direction of the differential expression can be ignored by taking the absolute value of the t statistics. Up- or down regulation of a gene between UICC II and UICC III patients is not taken into account and emphasis is put on regulation per se. The gene-wise quantification of the differential gene expression is performed by

```
> require(multtest)
> UICC.diff <- abs(mt.teststat(exprs(eSet), UICC.gr))
```

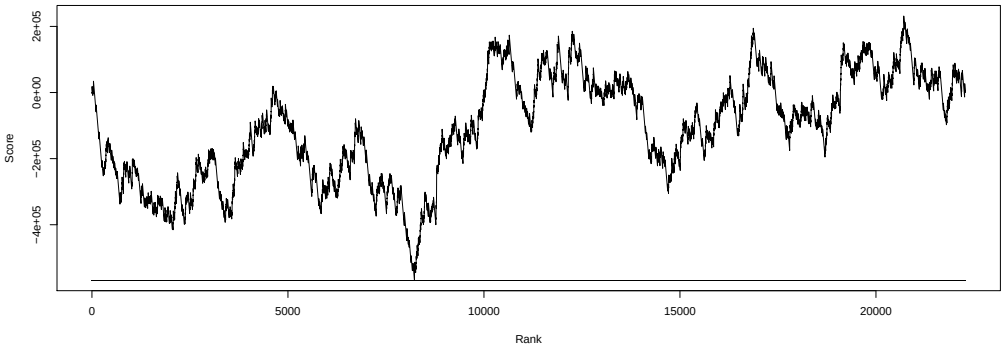
Now the statistics `Myy` for the gene set enrichment procedure can be calculated. The basic idea is to sort the `UICC.diff` values and to analyze at which positions members of group A and group B can be found. In a second step a score vector is created by assigning the score `nB` to each group A member and the score `-nA` to each group B member. In a third step the cumulative sum is calculated and the statistic `Myy` can be read off. In this case where we consider absolute values of t statistics as measure for differential expression, only large maxima of the cumulative sum will indicate particularity of gene group A.

```
> order <- order(UICC.diff, decreasing = TRUE)
> table(group.ind)
> UICC.score <- ifelse(group.ind == "A", table(group.ind)["B"],
+     -table(group.ind)["A"])
> UICC.score <- UICC.score[order]
> yy <- cumsum(UICC.score)
> M.yy <- max(yy)
> M.yy
```

The result can be presented as a figure that shows the course of the cumulative sum. Note that for the `type` parameter must be given the letter 'l' (for 'line') and not the number '1'. Additionally the position of elements of group A is spotted by points.

```
> plot(0:length(yy), c(0, yy), type = "l", xlab = "Rank",
+     ylab = "Score")
> loc.A <- group.ind[order] == "A"
> xxx <- which(loc.A)
> yyy <- rep(min(yy), length(xxx))
> points(xxx, yyy, pch = ".")
```

Unfortunately, the figure does not look very convincing. There is an up and down of the line, the spots of group A can be found everywhere. Of course the figure could be improved by adding some information about the density of the A points.



Permutation of Genes

In the next step the permutation test will be performed in order to obtain a useful statistical assessment for answering the question of interest. The permutation test is performed by the function `gene.set.enrichment.permut.rfc` that can be found in the workspace.

```
> gene.set.enrichment.permut.rfc(UICC.diff, group.ind, 1000)
```

Interpret the result of the test. Is it possible to state, that there is no difference between the differential expression in probe sets which belong to cancer related pathways and the remaining probe sets on the array? The question can be formulated more specific by looking at a single pathway within the group of cancer related pathways. The names of the nine pathways under consideration are written into the object pathways.

```
> pathways <- unique(pathways.U133A$PATHWAY)
```

The following analysis will be restricted to the 1407 probesets which where extracted. The relevant probesets are listed in `PSS.pathway` and the relevant information on differential gene expression is given in `UICC.pathway`.

```
> UICC.pathway <- UICC.diff[group.ind == "A"]
> PSS.pathway <- rownames(exprs(eSet)) %in% ps
> PSS.pathway <- rownames(exprs(eSet))[PSS.pathway]
```

For analyzing the androgen receptor signaling pathway one needs to define a group variable which indicates the probe sets belonging to this specific pathway. Then the analysis can be performed.

```
> pss.special <- unique(ps[pw == pathways[1]])
> group.special <- ifelse(PSS.pathway %in% pss.special,
+     "A", "B")
> gene.set.enrichment.permut.rfc(UICC.pathway, group.special,
+     10000)
```

Simply by changing the index number for a pathway helps to check all candidates. The following loop generates a table with the entire relevant information. The number of permutations is set to 10000. This calculation can take some time. Now you have the chance to go for a coffee.

```
> pathway.results <- NULL
> for (i in 1:9) {
```

```
+     pss.special <- unique(ps[pw == pathways[i]])
+     group.special <- ifelse(PSS.pathway %in% pss.special,
+         "A", "B")
+     res <- gene.set.enrichment.permut.rfc(UICC.pathway,
+         group.special, 10000)
+     pathway.results <- rbind(pathway.results, unlist(res))
+ }
> rownames(pathway.results) <- pathways
```

Kolmogorov-Smirnov Test

As an alternative to the presented permutation approach one could use the Kolmogorov-Smirnov test in order to test whether the distribution of gene ranks is the same for both gene groups.

```
> order <- order(UICC.pathway, decreasing = T)
> pathway.results.KS <- numeric(length(pathways))
> for (i in 1:9) {
+     pss.special <- unique(ps[pw == pathways[i]])
+     group.special <- ifelse(PSS.pathway %in% pss.special,
+         "A", "B")
+     ranksA <- which(group.special[order] == "A")
+     ranksB <- setdiff(1:length(group.special), ranksA)
+     res <- ks.test(ranksA, ranksB, alternative = "greater")
+     pathway.results.KS[i] <- res$p.value
+ }
> names(pathway.results.KS) <- pathways
```

Permutation of Samples

A disadvantage of the presented strategy is that it does not account for dependencies between genes. Therefore an alternative is to permute phenotype labels instead of genes in order to preserve the correlation structure between genes. We would have to change our permutation function so that for each permutation of samples t-statistics are computed for each gene. In every permutation step this results in a new ordering of genes, for which again the cumulative sum and the maximum statistic can be derived and compared to the observed one, as it was done before. This approach is equivalent to the gene set enrichment analysis suggested by Subramanian et al. (2005).

References

Lamb J, Ramaswamy S, Ford HL, Contreras B, Martinez RV, Kittrell FS, Zahnow CA, Patterson N, Golub TR, Ewen ME. A mechanism of cyclin D1 action encoded in the patterns of gene expression in human cancer. Cell 2003; 114(3): 323-34.

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. PNAS 2005; 102(43): 15545-15550.