Making sense of transcriptomics data v.008

Raffaele A. Calogero

February 12 2015

1 Introduction

The ultimate goal of any transcriptomic experiment is to discover functional patterns of biological response to conditions of interest (treatments, environmental influences, mutations etc). The biological interpretation of large gene lists (ranging in size from hundreds to thousands of genes) is still a challenging and daunting task. Over the last few decades, bioinformatics methods, using the biological knowledge accumulated in public databases, e.g. Gene Ontology Ashburner M (2000), made possible to systematically dissect large gene lists in an attempt to assemble a summary of the most enriched and pertinent biology Huang DW (2009). This practical session will focus on the use of data mining approaches to understand the biology underlying transcriptomic experiments.

2 ES to NPC differentiation experiment

The pluripotency and high proliferative capacity of embryonic stem (ES) cells make them an attractive source of different cell types for biomedical research and cell replacement therapies. ES cells have the potential to differentiate into all types of cell lineages including neural precursor cells (NPCs). NPCs can be expanded in large numbers for significant periods of time to provide a reliable source of cells for transplantation in neurodegenerative disorders and injury of the central nervous system. The experiment, we will use in this practical session, is based on mouse cell line 46C. Mouse ES cells were differentiated to neural progenitor cells (NPCs). Experiments were performed in duplicate, and ES cells NPCs were used for sequencing as well as proteomics. Each experimental point was done in duplicate. Total RNA was extracted and the polyA+ enriched fraction was sequenced by 50+50 nts PE stranded protocol (Table 1).

Table 1: **Summary of the differentiation experiment sequences**

Sample	**Type**	**fragments (milions)**
2IM1EB2	ES	25.77
2iM1EGRNA1	ES	119.52
GFB1EB2	NPC	61.01
GFP1EGRNA1	NPC	91.64

3 Sequence data preprocessing

The first step in a transcriptome sequence experiment is the definition of the level at which the analysis has to be done:

1. Gene-level

2. Transcript-level

3. Exon-level

Functional annotations are prevalently associated to gene-level. Thus, we will mainly focus, in this practical session, on gene-level information. However, to understand the effect of alternative splicing events in a biological framework there are ways of extending to transcript-level the gene-level functional annotations. For example, once functional paths are detected as enriched at gene-level, detected alternative splicing events are checked if they overlay to those functional paths. An other important decision that has to be made, at the beginning of a transcriptome sequence analysis, is the selection of the annotation to be used. The most used annotations at the present time are:

1. REFSEQ

2. UCSC

3. ENSEMBL

The main difference between the three annotation is the level of information richness and robustness. REFSEQ is a robust and manually curated transcript centric annotation. UCSC Bioinformatics Group itself does no generate sequences. UCSC creates the majority of the annotation tracks in-house, the annotations are based on publicly available data contributed by many labs and research groups throughout the world. The annotation is transcript centric. ENSEMBL provides a gene/transcript centric annotation. For selected species (eg. human, mouse, pig, rat), gene annotation may also include manual curation, i.e. reviewed determination of transcripts on a case-by-case basis. Furthermore, Ensembl imports annotation from FlyBase, WormBase and SGD. Ensembl transcripts displayed on website are products of the Ensembl automatic gene annotation

system (a collection of gene annotation pipelines), termed the Ensembl genebuild. All Ensembl transcripts are based on experimental evidence and thus the automated pipeline relies on the mRNAs and protein sequences deposited into public databases from the scientific community.
In this specific practical session we will use UCSC annotation. Having decided for gene-level and UCSC annotation now it is necessary to identify the bioinformatic tool that allow the conversion of reads in a number of counts for a specific gene. The number of tools that allow such conversion are many and a full description of all of them is out of the pourpose of this section. Steijger T (2013) paper provides a detailed evaluation of computational methods for transcript reconstruction and quantification from RNA-seq data. In their paper were evaluated 25 protocol variants of 14 independent computational methods for exon identification, transcript reconstruction and expression-level quantification from RNA-seq data. In this practical session we will use for counts generation rsem software Li B (2011). We selected such tool because it allow to collapse UCSC transcript information in gene-level information and it provides three types of counts format:

1. expected_count. This count is generally a non-integer value and is the expectation of the number of alignable and unfiltered fragments that are derived from a isoform or gene given the maximum likelihood estimated abundances.

2. TPM. This measure can be used directly as a value between zero and one or can be multiplied by a milion to obtain a measure in terms of transcripts per million.

3. FPKM or Fragments Per Kilobase of exon per Million reads. Fragment means fragment of DNA, so the two reads that comprise a paired-end read count as one. Per kilobase of exon means the counts of fragments are then normalized by dividing by the total length of all exons in the gene.

Differentially expression tools like edgeR and DESeq2 require the use of raw counts as input. However, the rsem expected_counts after rounding, on the basis of rsem developers, can be used as input for edgeR or DESeq2.
The problem with using raw read counts is that the origin of some reads cannot always be uniquely determined. If two or more distinct transcripts in a particular sample share some common sequence (for example, if they are alternatively spliced mRNAs or mRNAs derived from paralogous genes), then sequence alignment may not be sufficient to discriminate the true origin of reads mapping to these transcripts. One approach to address this issue involves discarding these multiple-mapped reads (multireads for short) entirely. Another involves partitioning and distributing portions of a multireads expression value between all of the transcripts to which it maps. So-called rescue methods implement this second approach in a naive fashion. RSEM improves upon this approach, utilizing an Expectation-Maximization (EM) algorithm to estimate maximum likelihood expression levels Daniel Standage Blog. These expected counts can then be provided as a matrix (rows = mRNAs, columns = samples) to programs such as:

1. EBseq
2. DESeq
3. edgeR

to identify differentially expressed genes.

3.1 How counts were generated

```
#RSEM
#http://deweylab.biostat.wisc.edu/rsem/
#mapping and counting with RSEM
nohup  /someWhereInYourDisk/rsem-calculate-expression --paired-end
C4VKHACXX_2iM1EGRNA1_14s004164-1-1_Haase_lane514s004164_1_sequence.txt
C4VKHACXX_2iM1EGRNA1_14s004164-1-1_Haase_lane514s004164_2_sequence.txt
/someWhereInYourDisk/rsem-1.2.15/mm9/mm9 -p 64 2iM1EGRNA1 &

#STAR+htseq-count
#https://code.google.com/p/rna-star/
#mapping with STAR
nohup /someWhereInYourDisk/STAR_2.3.1n/STARstatic
--genomeDir /someWhereInYourDisk/genomes/mm9.star
--readFilesIn ./C4VKHACXX_2iM1EGRNA1_14s004164-1-1_Haase_lane514s004164_1_sequence.t
./C4VKHACXX_2iM1EGRNA1_14s004164-1-1_Haase_lane514s004164_2_sequence.txt
--runThreadN 64 2iM1EGRNA1 &
#htseq-count
#http://www-huber.embl.de/HTSeq/doc/overview.html
#convert sam in bam
nohup samtools view -Sb Aligned.out.sam -o Aligned.out.bam  &
#sort bam by read names
nohup samtools sort -n Aligned.out.bam Aligned.outS.bam &
htseq-count /someWhereInYourDisk/genomes/mm9/mm9.gtf
nohup htseq-count --stranded=STRANDED --order=name -q  Aligned.outS.bam
/someWhereInYourDisk/genomes/mm9/mm9.gtf  > Aligned.outS.counts &
```

3.2 Creating a gene-level count matrix

The four files, with the suffix *.genes.results* are present in the folder:

```
> paste(find.package(package="thu12feb05"),"/examples/", sep="")
```

```
[1] "/Library/Frameworks/R.framework/Versions/3.1/Resources/library/thu12feb05/example
```

The structure of each file is the following:

```
> dir <- dir(paste(find.package(package="thu12feb05"),"/examples/", sep=""))
> dir <- dir[grep(".genes.results",dir)]
> tmp <- read.table(paste(find.package(package="thu12feb05"),
+ "/examples/",dir[1], sep=""), sep="\t", header=T)
> head(tmp)

  gene_id      transcript_id.s. length effective_length expected_count   TPM
1       1 uc007aet.1,uc007aeu.1 3621.0          3596.90              0  0.00
2      10            uc011whv.1   26.0            14.61              0  0.00
3     100 uc007amd.1,uc007ame.1 4355.0          4330.90            536 41.04
4    1000            uc007dac.1 1403.0          1378.90            108 26.14
5   10000 uc008ajp.1,uc012ajs.1 1415.5          1391.40              0  0.00
6   10001            uc008ajq.1 2046.0          2021.90              3  0.49
   FPKM
1  0.00
2  0.00
3 19.10
4 12.16
5  0.00
6  0.23
```

where:

1. *gene_id* is the number defining the UCSC gene cluster in which multiple transcripts are collapsed. This is NOT Entrez gene-id

2. *transcript_id.s.* shows the transcripts collapsed in a gene

3.3 Differential gene expression

In Bioconductor are available different packages for differential expression analysis, see the *biocViews DifferentialExpression*. As highligted above we will use EBseq to detect the set of genes changing in expression in the ES to NPC differentiation. As detailed in Leng N (2013), EBSeq is an empirical Bayesian approach that models a number of features observed in RNA-seq data.

3.3.1 Exercise 1

In this package there is a function called *annotatingGenes*. Read its man page

```
Start R typing in a terminal window R
#load the package thu12feb05
```

```
library(thu12feb05)
#read the help for the function annotatingGenes
?annotatingGenes
```

and create a count matrix encompassing the four experiments using the symbol as gene identifier.

Using the information provided by EBSeq define the set of differentially expressed genes between ES and NPC. Please detect the subset of genes characterized by FDR <= 0.05 and absolute log2FC >= 1.

EBSeq relies on parametric assumptions that should be checked following each analysis. The QQP and DenNHist help to assess prior assumptions.

Please answer to the following questions:

1. How many genes are characterized by FDR <= 0.05 and absolute log2FC >= 1?
2. How many iteration are needed to get convergence for Alpha and Beta parameters?
3. On the basis of the QQP plot the parametric assumption that Beta prior is appropriate is satisfied?
4. On the basis of the DenNHist plot estimated distribution fits well to the data?

The script is available in section *Exercises solutions: exercise 1.*

3.3.2 Exercise 1.1

Using the Count table, *CountTable.txt* generated by Jonathon Blake run the same analysis in exercise 1. Please answer to the following question:

1. Is the counting approach affecting the number of differentially expressed genes?
2. If you compare the results obtained on the above mentioned count table with DESeq2, which is the overlap with the analysis performed using EBseq?

4 Datasets

From here you can decide to use the data generated with RSEM or by Jonathan. You can also go over the following workflow with one dataset and then the other.

5 Annotating differentially expressed genes

In Bioconductor are present specific organism-specific annotation packages:

```
> library(Mus.musculus)
> head(keytypes(Mus.musculus))
```

```
[1] "GOID"       "TERM"       "ONTOLOGY"   "DEFINITION" "ENTREZID"
[6] "PFAM"
```

```
> head(columns(Mus.musculus))
```

```
[1] "GOID"       "TERM"       "ONTOLOGY"   "DEFINITION" "ENTREZID"
[6] "PFAM"
```

keys and *columns* allow to extract as a dataframe any of the annotation using the function *select*:

```
> head(keys(Mus.musculus, keytype="ENTREZID", pattern="^2"), n=6)
```

```
[1] "20005" "20014" "20015" "20016" "20017" "20018"
```

```
> ann.df <- select(Mus.musculus, keys=keys(Mus.musculus, keytype="ENTREZID"),
+ columns=c("GENENAME", "SYMBOL"), keytype="ENTREZID")
> head(ann.df)
```

```
  ENTREZID SYMBOL                                             GENENAME
1    11287    Pzp                               pregnancy zone protein
2    11298  Aanat                    arylalkylamine N-acetyltransferase
3    11302   Aatk                   apoptosis-associated tyrosine kinase
4    11303  Abca1 ATP-binding cassette, sub-family A (ABC1), member 1
5    11304  Abca4 ATP-binding cassette, sub-family A (ABC1), member 4
6    11305  Abca2 ATP-binding cassette, sub-family A (ABC1), member 2
```

5.0.3 Exercise 2

Using the above annotation package and the *select* function annotate the genes detected as DE in exercise 1. Plot an histogram of log2FC of DE genes
Please answer to the following question:

1. There is any particular characteristic in the identified DEs, when looking at log2FC histogram?

The script is available in section *Exercises solutions: exercise 2.*

5.1 Orthologs annotation conversion

The Stockholm Bioinformatics Centre has compiled INPARANOID data packages containing orthologs and paralogs. These data packages can be downloaded from BioConductor and are named hom.Xx.inp.db where Xx is the central species, such as At, Ce, Dm, Dr, Hs, Mm, Rn and Sc. For the human package, hom.Hs.inp.db, this means that the ortholog mappings are between Humans and cattle, humans and mice and so on. Information how to handle INPARANOID databases is available in the vignette of *AnnotationFuncs* package.

5.1.1 Exercise 2.1

Starting from the *de.selected* extract the mouse symbols and convert them in human symbols. Attach the human gene symbols to de.selected dataframe, de.w.orth, and save the dataframe as tab delimited file.

```
browseVignettes("AnnotationFuncs")
```

The script is available in section *Exercises solutions: exercise 2.1*

5.1.2 Exercise 2.2

Starting from the *de.w.orth* add the swiss prot id using the *Mus.musculus*. The function that has to be used is *select*. Attach the swiss prot ids to de.w.orth dataframe and save the dataframe as tab delimited file.

Please answer to the following questions:

1. There is any duplicated symbol?
2. There is any duplicated Uniprot id?

The script is available in section *Exercises solutions: exercise 2.2*

6 Search for chromosome positional enrichments

Since we are investigating the effect of differentiation it would be interesting to see if differential expression is linked to chromatin reorganization. How can we address this question? We can have a look at the location of DE genes in the chromosomes. We generated a function, *plotChrs*, that allows the visualization of the overall number of DE with respect to chromosomes and it also allows the plotting of the position of the DE on chromosomes. Check the manual page for it.

```
?plotChrs
```

The search for feature enrichment is a widely used method to characterize a set of genes. While several tools have been designed for nominal features such as Gene Ontology annotations or KEGG Pathways, very little has been proposed to tackle numerical features such as the chromosomal positions of genes. De Preter K (2008) a positional gene enrichment analysis method PGE for the identification of chromosomal regions that are significantly enriched in a given set of genes. The strength of their method relies on an original query optimization approach that allows to virtually consider all the possible chromosomal regions for enrichment, and on the multiple testing correction which discriminates truly enriched regions versus those that can occur by chance.

6.0.3 Exercise 3

Please produce the plots available for the set of DE genes we have identified in exercise 1

The script is available in section *Exercises solutions: exercise 3*.

Use the Entrez gene-ids associated to the set of DE genes that we have identified in exercise 1 and run an analysis on `http://homes.esat.kuleuven.be/~bioiuser/pge/` Please answer to the following questions:

1. Plotting the number of DE on chr, are they uniformly distributed?

2. Plotting the number of DE on chr, are they distributed proportionally to the chromosome size?

3. Plotting the genes location on chromosome do you see any locus enriched of DEs?

4. PGE identifies gene-enriched loci?

7 Gene ontology enrichment

Gene Ontology enrichment is the most basic functional analysis that can be run on a set of differentially expressed genes. Specifically we will use the Bioconductor package *goseq*. The requirement of goseq package are quite limited, since it requires a simple named vector, which contains two pieces of information:

1. **Measured genes**: all genes for which RNA-seq data was gathered for your experiment. Each element of your vector should be named by a unique gene identifier.

2. **Differentially expressed genes**: each element of your vector should be either a 1 or a 0, where 1 indicates that the gene is differentially expressed and 0 that it is not.

It is important to note that GO enrichment in RNAseq experiments is different from microarray experiments. Indeed, it is necessary to take in account the length bias present in the dataset under consideration Young MD (2010). Furthermore, to account for the length bias inherent to RNAseq data when performing a GO analysis one cannot simply use the hypergeometric distribution as the null distribution for category membership, which is instead appropriate for data without DE length bias, such as microarray data. GO analysis of RNA-seq data requires the use of random sampling in order to generate a suitable null distribution for GO category membership and calculate each categories significance for over representation amongst differentially expressed genes. However random sampling is computational expensive and it can be approximated using Wallenius distribution, which is a generalization of the hypergeometric distribution where items are sampled with bias.

7.0.4 Exercise 4

Following the instuction on the vignette of *goseq*

```
library(Biobase)
library(goseq)
openVignette()
```

create a vector called de.universe where the set of DE identified in exercise 1 are indicated by 1 and all the other genes are indicated by 0. The names of the vector elements are the gene symbols. Follow the vignette instructions and run the enrichment procedure for BP and MF GO classes. Using the function *plotGO* have a look at the GO layer at which the GO enriched term is located. Using the function *deInGO* identify the subset of DE associated to the enriched GO terms.

The script is available in section *Exercises solutions: exercise 4*.
Please answer to the following questions:

1. Which class of GO, BP, CC or MF produces a set of enriched set of GO terms?

2. Which is the GO layer at which the enriched GO belongs?

8 Topology-based Pathway Analysis of RNASeq data

The Bioconductor package ToPASeq implements several different methods for topology-based pathway analysis of gene expression data from microarray and RNA-Seq technologies.

8.1 graphite

In order to gather curated information about human pathways, developers of graphite collected data from the four public databases that were emerged as reference points for the systems biology community.

1. KEGG, Ogata H (1999)

2. Biocarta

3. Reactome, Matthews L (2009)

4. NCI/Nature Pathway Interaction Database, Schaefer CF (2009)

5. SPIKE, Paz A (2011)

6. HumanCyc, Caspi CF (2010)

7. Panther, Mi H (2013)

The KEGG database has been in development by Kanehisa Laboratories since 1995, and is now a prominent reference knowledge base for integration and interpretation of large-scale molecular data sets generated by genome sequencing and other high-throughput experimental technologies. KEGG is the only pathway database not in biopax format, they use the KGML format. Reactome, backed by the EBI, is one of the most complete repository; it is frequently updated and provides a semantically rich description of each pathway. KEGG Pathways (KGML format) provides maps for both signaling and metabolic pathways. BioCarta is a developer, supplier and distributor company of reagents and assays for biopharmaceutical and academic research. Through an "open source" approach, this community-fed forum constantly integrates emerging proteomic information from the scientific community. It also catalogs and summarizes important resources providing information for over 120,000 genes. BioCarta pathway data in biopax format are available through NCI website. NCI (NCI/Nature Pathway Interaction Database) is a highly-structured, curated collection of information about known biomolecular interactions and key cellular processes assembled into signaling pathways. This was a collaborative project between the NCI and Nature Publishing Group (NPG). Panther data are a comprehensive, curated database of pathways, protein families, trees, subfamilies and functions geenrated by the University of Southern California. HumanCyc is part of the BioCyc database collection of pathways. Finally, SPIKE is a database for human curated signaling pathways backed by Tel Aviv University.

8.2 TopologyGSA

TopologyGSA represents a multivariable method in which the expression of genes is modelled with Gausian Graphical Models with covariance matrix reflecting the pathway topology. It uses the the Iterative Proportional Scaling algorithm to estimate the covariance matrices. The testing procedure is a two-step process. First the equality of covariance matrices is testes via a likelihood ratio test. Then, when the null hypothesis of equality of covariance matrices is not rejected, the differential expression is testes via multivariate analysis of variance. On the other hand, when the convariance matrices are not equal, then Behrens-Fisher method for testing the equality of means in a two sample problem with unequal covariance matrices is employed.

8.2.1 Exercise 5

Have a look at the pathways available in the graphite databases e.g.:

```
library(ToPASeq)
names(kegg)
```

Use counts table generated in exercise 1. Make analysis with the above mentioned tools, using only genes characterized by sum greater than 10 expected_counts and using panther database.

Please answer to the following questions:

1. How many paths are specific only to one graphite database?
2. How many paths are in common between graphite databases?
3. Which method provide some results?
4. Does any of the results fits to the biology of the ES to NPC differentiation?
5. Plot the pathway that show link with the biology of ES to NPC differentiation.

The script is available in section *Exercises solutions: exercise 5.*

8.2.1.1 Exercise 5.1

Using the information provided by the ToPASeq vignette check if some path are detected as enriched, using:

1. panther db
2. deseq2 as normalization
3. symbol as id

Permutations should be kept greater than 100 but is incompatible with the time available for this practical session, please use 10.

8.3 DEGraph

This multivariable method assumes the same direction in the differential expression of genes belonging to a pathway. It performs the regular Hotelling's T2 test in the graph-Fourier space restricted to its first k components, which is more powerful than test in the full graph-Fourier space or in the original space.

8.3.0.2 Exercise 5.2

Repeat the above analysis using DEGraph. If there are paths with an Overall.p smaller than 0.05 plot them using the function *plotPath*. In case some path is found enriched, what happen if I change database? Are the detected enriched path also present in other databases? Whats happen if the search is performed in a different database encompassing the same path found enriched?

8.4 clipper

This multivariable method is similar to the topologyGSA as it uses the same two-step approach. However, the Iterative Proportional Scaling algorithm was subsituted with a shrinkage procedure of James-Stein-type which additionally allows proper estimates also in the situation when number of samples is smaller than the number of genes in a pathway. The tests on a pathway-level are followed with a search for the most affected path in the graph.

8.4.0.3 Exercise 5.3

Repeat the above analysis using clipper. If there are paths check alphaMean in res(cli). If no paths are present repeat the analysis using *var* in test parameter. If enriched paths are present plot them using the function *plotPath*.

8.5 SPIA

In SPIA two evidences of differential expression of a pathway are combined. The first evidence is a regular so called overrepresentation analysis in which the statistical significance of the number of differentially expressed genes belonging to a pathway is assessed. The second evidence reflects the pathway topology and it is called the pertubation factor. SPIA assumes that a differentially expressed gene at the begining of a pathway topology (e.g. a receptor in a signaling pathway) has a stronger effect on the functionality of a pathway than a differentially expressed gene at the end of a pathway (e.g. a transcription factor in a signaling pathway). The pertubation factors of all genes are calculated from a system of linear equations and then combined within a pathway. The two evidences in a form of p-values are finally combined into a global p-value, which is used to rank the pathways.

8.5.0.4 Exercise 5.4

Repeat the above analysis using SPIA. Check if there is any pFdr < 0.05 using *res* function. What happen if you change test statistics? If there are paths with an pFdr smaller than 0.05 plot them using the function *plotPath*.

8.6 TAPPA

TAPPA was among the first topology-based pathway analysis methods. It was inspired in chemointformatics and their models for predicting the structure of molecules. In TAPPA, the gene expression values are standardized and sigma- transformed within a samples. Then, a pathway is seen a molecule, individual genes as atoms and the energy of a molecule is a score defined for one sample. This score is called Pathway Connectivity

Index. The difference of expression is assessed via a common univariable two sample test - Mann-Whitney.

8.6.0.5 Exercise 5.5

Repeat the above analysis using TAPPA. If enriched paths are present plot them using the function *plotPath*.

8.7 TBS

TBS is another method that works with gene-level statistics and a list of differentially expresed genes. The pathway topology is incorporated as the number of downstream differentially expressed genes. The gene-level log fold-changes are weigted by this numeber and sumed up into a pathway-level score. A statistical significance is assessed by a permutations of genes.

8.7.0.6 Exercise 5.6

Repeat the above analysis using TBS. If enriched paths are present plot them using the function *plotPath*.

8.8 PWEA

PathWay Enrichment Analysis (PWEA). This is actually a weigthed form of common Gene Set Enrichment Analysis (GSEA). The weights are called Topological Influence Factor (TIF) and are defined as a geometic mean of ratios of Pearson's correlation coefficient and the distance of two genes in a pathway. The weights of genes outside a pathway are assigned randomly from normal distribution with parameters estimated from the weights of genes in all pathways. A statistical significance of a pathway is assessed via Kolmogorov-Simirnov-like test statistic comparing two cumulative distribution functions with class label permutations.

8.8.0.7 Exercise 5.7

Repeat the above analysis using PWEA. If enriched paths are present plot them using the function *plotPath*.

8.9 FGNet

FGNet allows to perform a Functional Enrichment Analysis (FEA) on a list of genes or expression set, and transform the results into networks. The resulting functional networks provide an overview of the biological functions of the genes/terms, and allows to easily see links between genes, overlap between clusters, finding key genes, etc. FGNet takes as input a query list of genes selected by the user, and builds and displays networks of genes based in the existence of common functional terms that are enriched in certain subsets of genes of the list. By doing this, the tool allows to disclose groups/clusters of genes that have similar annotations and so they may have similar biological function in the cell. FGNet builds the functional networks, based on data from a previous functional enrichment analysis (FEA). The package provides the functions to perform the FEA through four specific tools:

1. DAVID with Functional Annotation Clustering (DAVID-FAC), which measures relationships among annotation terms based on their co-association with subsets of genes within the query gene list

2. GeneTerm Linker, a post-enrichment tool, which focuses on clearing and sorting the results from a previous modular enrichment analysis. This is achieved by filtering general terms with low information content (i.e. cellular process or protein binding) and redundant annotations (i.e. metabolic process and primary metabolic process)

3. TopGO allows an enrichment analysis tool based on Gene Ontology (GO) that tests GO terms while accounting for the topology of the GO graph to eliminate local similarities and dependencies between GO terms.

4. GAGE (Luo et al), a gene set enrichment analysis (GSEA) tool. It searches for functional enrichment in gene sets (i.e. KEGG pathways, Reactome, GO) and allows including a signal value -like expression changes- to rank the genes and then to identify the enrichment in functional terms that are altered (i.e. changed in genes UP and DOWN) or altered consistently in one direction (UP or DOWN).

8.9.0.8 Exercise 6

Write in two files the subsets of up and dw regulated genes.

1. Following the FGNet vignette information, run an enrichement analysis using topGO, using only the subset of **up-regulated** genes and BP GO terms. Please remember to use Symbols and Mouse annotation database. Do not give a name to the job a folder is made by the software.

2. Create a html report and have a look at it. Extract the genes involved in networks manipulating the iGraph.RData present in the folder that was created.

3. Repeat the analysis for the **down-regulated** genes

4. Combine the two list of genes that generate networks and repeat the above analysis on those. Have a look at the html report.

The script is available in section *Exercises solutions: exercise 6.*

9 Exercises solutions

9.1 Exercise 1

```
samples.dir <- paste(find.package(package="thu12feb05"),"/examples/", sep="")
samples <- dir(samples.dir)[grep(".genes.results",dir(samples.dir))]
counts <- lapply(paste(samples.dir,samples, sep=""), function(x){
      tmp <- annotatingGenes(filename=x, org="mm9")
          counts.tmp <- tmp$expected_count
})
counts.df <- t(data.frame(matrix(unlist(counts), nrow=length(samples), byrow=T)))

tmp <- annotatingGenes(filename=paste(samples.dir,samples[1], sep=""), org="mm9")
dimnames(counts.df) <- list(as.character(tmp$Symbol),
gsub(".genes.results","",samples))
```

```
library(EBSeq)
#library size
sizes=MedianNorm(counts.df)
#experimental groups
conditions=as.factor(c("ES","ES","NPC","NPC"))
#
de <- EBTest(Data=as.matrix(counts.df), Conditions=conditions,
sizeFactors=sizes, maxround=15, QtrmCut=50)
#checking for convergence given 50 iterations
de$Alpha
de$Beta
#getting the posterior probability
pp=GetPPMat(de)
#detecting the subset of genes with a FDR <= 0.05
de.found=rownames(pp)[which(pp[,"PPDE"] >= 0.95)]
#detecting log2FC
gene.fc <- PostFC(de)
#extracting the posterior log2FC
de.found.fc <- gene.fc$PostFC[which(names(gene.fc$PostFC)%in%de.found)]
#FDR <= 0.05 |log2FC| >= 1
de.found <- names(de.found.fc)[which(abs(log2(de.found.fc)) >= 1)]
#checking the parametric assumptions
par(mfrow=c(1,2))
QQP(de)
par(mfrow=c(1,2))
```

```
DenNHist(de)
```

Comments on exercise 1: because the number of replicates is very low we cannot expect an optimal parameter estimation as well as detection of differentially expressed genes. This not only apply to EBSeq but also to other methods as highlighted in Sonenson publication Soneson C (2013).

9.2 Exercise 2

```
library(Mus.musculus)
keytypes(Mus.musculus)
columns(Mus.musculus)
#looling at the keys characteristics
head(keys(Homo.sapiens, keytype="ENTREZID"), n=2)
#looling at the keys characteristcs sharing a specific pattern
head(keys(Mus.musculus, keytype="ENTREZID", pattern="^2"), n=6)
ann.df <- select(Mus.musculus, keys=keys(Mus.musculus, keytype="ENTREZID"),
columns=c("GENENAME", "SYMBOL"), keytype="ENTREZID")
head(ann.df)

#annotating the set of differentially expressed genes
head(de.found)
ann.de <- select(Mus.musculus, keys=de.found, columns=c("ENTREZID",
"GENENAME"), keytype="SYMBOL")
head(ann.de)
#combining with differential expression
#calculating log2 FC for the subset of genes selected ad DE
de.fc <- log2(de.found.fc[which(abs(log2(de.found.fc)) >= 1)])
identical(ann.de$SYMBOL, names(de.fc))
#if FALSE
#ann.de <- ann.de[order(ann.de$SYMBOL),]
#de.fc <- de.fc[order(names(de.fc))]
#identical(ann.de$SYMBOL, names(de.fc))
de.selected <- as.data.frame(cbind(de.fc, ann.de))
write.table(de.selected, "de.selected.txt", sep="\t", row.names=FALSE)
#looking at FC distribution in DE
hist(de.selected$de.fc, breaks=50)
```

9.2.1 Exercise 2.1

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("AnnotationFuncs","hom.Mm.inp.db"))
library(AnnotationFuncs)
library(hom.Mm.inp.db)
library(Mus.musculus)
library(Homo.sapiens)
hom.Mm.inp()
hom.Mm.inpHOMSA

symbols <- as.character(de.selected$SYMBOL)

symbols.hs <- getOrthologs(symbols, hom.Mm.inpHOMSA, 'HOMSA',
pre.from=org.Mm.egSYMBOL2EG, pre.to=org.Mm.egENSEMBLPROT,
post.from=org.Hs.egENSEMBLPROT2EG, post.to=org.Hs.egSYMBOL)

de.hs <- de.selected[which(symbols%in%names(symbols.hs)),]
de.hs <- de.hs[order(as.character(de.hs$SYMBOL)),]
symbols.hs <- symbols.hs[order(names(symbols.hs))]
identical(names(symbols.hs), as.character(de.hs$SYMBOL))
de.hs <- cbind(de.hs, as.character(symbols.hs))
dimnames(de.hs)[[2]][5] <- "Hs.SYMBOL"
de.mm <- de.selected[setdiff(seq(1, dim(de.selected)[1]),
which(symbols%in%names(symbols.hs))),]
de.mm <- cbind(de.mm, rep(NA, dim(de.mm)[1]))
dimnames(de.mm)[[2]][5] <- "Hs.SYMBOL"
de.w.orth <- rbind(de.hs, de.mm)
de.w.orth <- as.data.frame(de.w.orth)
```

9.2.2 Exercise 2.2

```
prot.df <- select(Mus.musculus, keys=as.character(de.w.orth$SYMBOL),
columns=c("UNIPROT","SYMBOL"), keytype="SYMBOL")
#duplicated symbols
sum(duplicated(prot.df[,1]))
#duplicated proteins
sum(duplicated(prot.df[,2]))
```

9.3 Exercise 3

```
#exercise 3 DE localization
##plotting genes on chrs
#plotting DE in chrs in function of chr length
de.lst <- plotChrs(de.df=de.selected, org="Mus.musculus",
genome="mm10", plot="bar")
```

```
#plotting DE locations on chrs
de.lst <- plotChrs(de.df=de.selected, org="Mus.musculus",
genome="mm10", plot="chart")
```

9.4 Exercise 4

```
library(goseq)
#creating the gene universe
de.universe <- rep(0, length(as.character(tmp$Symbol)))
names(de.universe) <- as.character(tmp$Symbol)
#tagging differentially expressed genes
de.universe[which(names(de.universe)%in%de.found)] <- 1
#mm9 genome and symbols id are supported by goseq
supportedGenomes()[grep("mm9", supportedGenomes()$db),]
supportedGeneIDs()[grep("Symbol",supportedGeneIDs()$db),]

#null distribution
pwf <- nullp(de.universe,"mm9","geneSymbol")
#GO enrichment using Wallenius distribution
GO.wall <- goseq(pwf,"mm9","geneSymbol")
#only Molecular Function
GO.MF <- goseq(pwf,"mm9","geneSymbol",test.cats=c("GO:MF"))
#only Biological processes
GO.BP <- goseq(pwf,"mm9","geneSymbol",test.cats=c("GO:BP"))
#subsetting by significance
enriched.BP <- GO.BP[which(p.adjust(GO.BP$over_represented_pvalue,
method="BH")<.05),]
write.table(enriched.BP, "enriched.BP.txt")
enriched.MF <- GO.MF[which(p.adjust(GO.MF$over_represented_pvalue,
method="BH")<.05),]

#plotting GO Term
plotGO(go.term="GO:0030154", go.class="BP")

#extracting de genes from enriched GO term
de.celdiff <- deInGO(go.term="GO:0030154", de.universe=de.universe,
org="Mus.musculus")
write.table(de.celdiff, "de.celdiff.txt", sep="\t",row.names=FALSE)
```

9.5 Exercise 5

```
library(ToPASeq)

#have a look at the paths available
names(kegg)
names(biocarta)
names(nci)
names(spike)
names(humancyc)
names(panther)
path.all <- c(names(kegg), names(biocarta), names(nci),
names(spike), names(humancyc), names(panther))
#all paths
length(path.all)
[1] 998
paths <- unique(c(names(kegg), names(biocarta), names(nci),
names(spike), names(humancyc), names(panther)))
#unique paths
length(paths)
[1] 986
#common paths
common <- intersect(intersect(intersect(intersect(intersect(names(kegg),
names(biocarta)),names(nci)),names(spike)),names(humancyc)),names(panther))

#loading counts
samples.dir <- paste(find.package(package="thu12feb05"),"/examples/", sep="")
samples <- dir(samples.dir)[grep(".genes.results",dir(samples.dir))]
counts <- lapply(paste(samples.dir,samples, sep=""), function(x){
      tmp <- annotatingGenes(filename=x, org="mm9")
          counts.tmp <- tmp$expected_count
})

counts.df <- t(data.frame(matrix(trunc(unlist(counts)),
nrow=length(samples), byrow=T)))

tmp <- annotatingGenes(filename=paste(samples.dir,samples[1], sep=""), org="mm9")
dimnames(counts.df) <- list(as.character(tmp$Symbol),
gsub(".genes.results","",samples))
group <- c("ES","ES","NPC","NPC")
counts.df1 <- counts.df[rowSums(counts.df) > 10,]
counts.df1 <- as.matrix(counts.df1)
```

```
#converting in uppercase gene symbols
dimnames(counts.df1)[[1]] <- toupper(dimnames(counts.df1)[[1]])
top <- TopologyGSA(counts.df1, group, panther, type="RNASeq",
IDs = "symbol", nperm=10, norm.method="DESeq2")
res(top)
##comments: This method requires more samples than nodes in a pathway.
Therefore there is an empty output in the example above

#DEGraph
#DEGraph
deg <- DEGraph(counts.df1, group, panther, type="RNASeq",
IDs = "symbol", norm.method="DESeq2")
deg$res[1:3,]
names(which(deg$res[,1] <= 0.05))
#3 interesting paths
names(panther)[which(names(panther)%in%names(which(deg$res[,1] <= 0.05)))]
#Plotting interesting networks
plotPath(my.path="mRNA splicing", path.db="panther", type="symbol")

#clipper
cli <- Clipper(counts.df1, group, panther, type="RNASeq",
IDs = "symbol", test="mean")
res(cli)

#SPIA
spi <- SPIA(counts.df1, group, panther, type="RNASeq", test="DESeq2",
IDs = "symbol", logFC.th=1)
res.spi <- res(spi)
res.spi[which(res.spi$pFdr <= 0.05),]
#google or Pubmed for
# Alzheimer disease-presenilin pathway NPC
#or
#neural progenitor cell Alzheimer
#or
#neural progenitor cell presenilin
#or
#Integrin neural progenitor cell

#TAPPA
tap <- TAPPA(counts.df1, group, panther, type="RNASeq",
IDs = "symbol", gene.stat="stats")
```

```
#TBS
tbs <- TBS(counts.df1, group, panther, type="RNASeq",
IDs = "symbol", logFC.th=1, nperm=10, test="DESeq2", gene.stat="test")

#PWEA
pwe <- PWEA(counts.df1, group, panther, type="RNASeq",
IDs = "symbol", nperm=10, test="DESeq2")
which(res(pwe)$p.adj < 0.05)
```

9.6 Exercise 6

```
de.selected <- read.table("de.selected.txt", sep="\t", header=T)
de.up <- as.character(de.selected$SYMBOL)[which(de.selected$de.fc > 0)]
writeLines(de.up, "de_up.txt")
de.dw <- as.character(de.selected$SYMBOL)[which(de.selected$de.fc < 0)]
writeLines(de.dw, "de_dw.txt")
dir.create("UP")
dir.create("DW")
library(FGNet)
FGNet_GUI()
#Follow the vignette to answer to exercise questions
browseVignettes("FGNet")
#Ones the results are generated load the iGraph.RData
and extract the symbols of the genes involced in networks.
library(igraph)
load("folder-created-by-FGNet/iGraph.RData")
genes.up <- names(iGraph$commonGtSets[[]])
writeLines(genes.up, "genes_topGO.up.txt")

#repeat the same for down-regulated genes.
load("folder-created-by-FGNet/iGraph.RData")
genes.dw <- names(iGraph$commonGtSets[[]])
writeLines(genes.dw, "genes_topGO.dw.txt")
gene.dw <- readLines(con <- file("genes_topGO.dw.txt"))
gene.up <- readLines(con <- file("genes_topGO.up.txt"))
#combine up and dw in a list and save as file
gene.updw <- c(gene.dw, gene.up)
writeLines(gene.updw, "gene_updw.txt")
#analyse only the genes that were creating networks
#have a look at the results
```

10 R-Bioconductor information

```
> sessionInfo()

R version 3.1.1 (2014-07-10)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] C/UTF-8/C/C/C/C

attached base packages:
[1] parallel  stats4    stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] Mus.musculus_1.1.2
 [2] TxDb.Mmusculus.UCSC.mm10.knownGene_3.0.0
 [3] org.Mm.eg.db_3.0.0
 [4] GO.db_3.0.0
 [5] RSQLite_1.0.0
 [6] DBI_0.3.1
 [7] OrganismDbi_1.8.0
 [8] GenomicFeatures_1.18.3
 [9] GenomicRanges_1.18.4
[10] AnnotationDbi_1.28.1
[11] GenomeInfoDb_1.2.4
[12] IRanges_2.0.1
[13] S4Vectors_0.4.0
[14] Biobase_2.26.0
[15] BiocGenerics_0.12.1

loaded via a namespace (and not attached):
 [1] BBmisc_1.9              BatchJobs_1.5           BiocParallel_1.0.3
 [4] Biostrings_2.34.1       GenomicAlignments_1.2.1 RBGL_1.42.0
 [7] RCurl_1.95-4.5          Rsamtools_1.18.2        XML_3.98-1.1
[10] XVector_0.6.0           base64enc_0.1-2         biomaRt_2.22.0
[13] bitops_1.0-6            brew_1.0-6              checkmate_1.5.1
[16] codetools_0.2-10        digest_0.6.8            fail_1.2
[19] foreach_1.4.2           graph_1.44.1            iterators_1.0.7
[22] rtracklayer_1.26.2      sendmailR_1.2-1         stringr_0.6.2
[25] tools_3.1.1             zlibbioc_1.12.0
```

References

et al Ashburner M. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics*, 20:25–29, 2000.

et al Caspi CF. The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Res.*, 38:D473–9, 2010.

et al De Preter K. Positional gene enrichment analysis of gene sets for high-resolution identification of overrepresented chromosomal regions. *Nucleic Acids Res.*, 36:D1–6, 2008.

Lempicki RA Huang DW, Sherman BT. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nucleic Acid Research*, 37:1–13, 2009.

et al Leng N. Ebseq: an empirical bayes hierarchical model for inference in rna-seq experiments. *Bioinformatics*, 29:1035–1043, 2013.

Dewey CN Li B. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC Bioinformatics*, 12:323–339, 2011.

et al Matthews L. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res.*, 37:D619–22, 2009.

et al Mi H. Panther in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Res.*, 38:D377–86, 2013.

et al Ogata H. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 27:29–34, 1999.

et al Paz A. Spike: a database of highly curated human signaling pathways. *Nucleic Acids Res.*, 39:D793–9, 2011.

et al Schaefer CF. Pid: the pathway interaction database. *Nucleic Acids Res.*, 37: D674–9, 2009.

Delorenzi M Soneson C. A comparison of methods for differential expression analysis of rna-seq data. *BMC Bioinformatics*, 14:91–109, 2013.

et al Steijger T. Assessment of transcript reconstruction methods for rna-seq. *Nature Methods*, 10:1177–1184, 2013.

et al Young MD. Gene ontology analysis for rna-seq: accounting for selection bias. *Genome Biology*, 11:R14, 2010.