Institut für Informatik
Praktische Informatik und Bioinformatik
Prof. Dr. Ralf Zimmer
Dr. Gergely Csaba
Dr. Florian Erhard

Wintersemester 2015/2016

GoBi: Excercise 4

Differential Expression

Deadline: Thursday, 21.01.2016, 14:00

Save your solution to /home/proj/biocluster/genprakt/${stud_account}/Solution4. Also provide an executable jar file (containing the sources) in this directory that allows to reproduce your results. The jar should print a usage info if invoked without parameters.
The base path for all relative paths in the tasks is:

`/home/proj/biosoft/praktikum/genprakt/assignments/a4`

Task 1 (Differential expression detection evaluation):

This task can be solved using R alone, i.e. you do not need to prepare an executable jar file here!

We applied our read simulator from the previous assignment to simulate reads of differentially regulated genes and used our feature counting program to compute gene-wise read counts.

There are two conditions with the corresponding read count matrices
`data/diff_simulation/(f1,f2)/simul.readcounts`

and the corresponding simulated reads in three replicates (with 200bp fragment length 10% standard deviation) in
`data/diff_simulation/(f1,f2)/(r1,r2,r3)/read.mappinginfo`.

Your task is to evaluate DEseq, edgeR and limma based on the read count matrices and the input to the reads simulator as a reference. These three methods are integrated into EnrichmentBrowser so that you can easily apply them after preparing the count data in the EnrichmentBrowser data format (see the documentation of `read.eset` in `doc/EnrichmentBrowser.pdf` on page 25).

You find example input files in `data/eb_example`.

Prepare the simulated data in the corresponding format and derive the DE data with all three methods (DEseq, limma, edgeR) using the script:

`scripts/de_rseq.R`

Create four plots to compare the three methods with respect to their Type I error control, their DE to non-DE discrimination performance, their FDR control and their total DE detection performance:

(a) Type I error control: Plot the cumulative distributions of p-values for non-DE genes for each of the three methods and compare against the uniform distribution.

(b) Discrimination performance: Create ROC curves for each of the three methods (in a single plot) with the respective p-value as ranking criterion.

(c) FDR control: Plot the computed FDR (based on the Benjamini-Hochberg correction of the respective p-values) against the true FDR (based on the reference)

(d) DE detection performance: Plot the computed FDR against the total number of detected genes.

In addition, compute one statistic for each of the four plots and write those in a tab separated file (columns: statistics, rows: method):

(a) Type I error control: What fraction of p-values of non-DE genes is below 5%

(b) Discrimination performance: Compute the area under the ROC curve (AUROC)

(c) FDR control: What is the true FDR for the the computed FDR of 10%

(d) DE detection performance: How many genes are detected with an FDR of 10%

Task 2 Analyze real data:

Extract gene-unique transcriptome NRP-s per gene from the ENCODE data set (processed in the previous assignment) for all mapper methods and invoke EnrichmentBrowser with all three DE-methods.

The report should contain:

- Overall consistency: cumulative plot of number of genes found differentially expressed in at least X mapper x DE-method combinations
- consistency plots grouped by DE-method (3 plots) and mapper (4 plots)
- vulcano plots (scatter plot log_2(foldchange) vs $-log_{10}$(p-value) for all mapper x DE-method combination.

Task 3 (EM algorithms - Christmas edition):

This task can be solved using R alone, i.e. you do not need to prepare an executable jar file here!

Familiarize yourself with the *mclust* R package. Nikolaus (not Santa Claus!) has put some files into the data folder (`data/xmas/data1/2/3a/3b.tsv`) and gave us some hints what to do with them:

(a) data1 contains data from a bivariate normal distribution

(b) data2 contains data from a mixture of 8 bivariate normal distributions; all components have zero correlation and the same variance, and the variance is equal in both dimensions

(c) data3a/b contain both data from mixtures of 4 bivariate normal distributions; all components have the same covariance matrix

Your task is to estimate the parameters of the distributions and to write them to a tab separated file (with header line) with columns

(a) Dataset: data1/data2/data3a/data3b

(b) Component: Integer indicating the mixture component (i.e. 1-8 for data2, 1-4 for data3a and data3b)

(c) mean.x: Mean of the component in the first dimension

(d) mean.y: Mean of the component in the second dimension

(e) var.x: Variance of the component in the first dimension (i.e. $S_{1,1}$ when S is the covariance matrix)

(f) var.y: Variance of the component in the second dimension (i.e. $S_{2,2}$ when S is the covariance matrix)

(g) rho: Correlation of the two dimensions in the component (i.e. $S_{1,2} = S_{2,1}$ when S is the covariance matrix)

Call the R script `scripts/merryxmas.R <table> <out.png>` with your table to find out what kind of gift Nikolaus had in mind for us.

Task 4 (Fragment length estimation):

Extend your feature count program from the previous assignment with the option to report the fragment lengths for transcript-mapping read-pairs. For such read pairs you can derive the fragment length from the start and end position in the transcript (and subtracting introns).

Create two files:

- unique lengths: derived from read pairs with a single fragment length containing the number of reads supporting one given fragment length.
- ambigous lengths: derived from read pairs that could correspond to fragments of at least two different lengths. For all reads pairs of this type write a line containing the different fragment lengths.

If we assume that fragment lengths follow a normal distribution, its parameters can be estimated using the following likelihood function:

$$L(\mu, \sigma^2 | l, z) = \prod_{i=1}^{n} \sum_{j=1}^{m_i} \delta_{z_i,j} f(l_{i,j} | \mu, \sigma^2)$$

Here, $l_{i,j}$ are the unique and ambiguous fragment lenghts of read pair i (j is always 1 for *unique length*, and $i \in \{1..m_i\}$ when the length is ambiguous) and z_i is the index of the true fragment length. The Kronecker delta $\delta_{a,b}$ is 1 if $a = b$ and 0 otherwise. $f(x|\mu, \sigma^2)$ is the density function of the normal distribution with mean μ and variance σ^2.

As the z_i are not observed, μ and σ^2 cannot be estimated directly. Instead, an EM algorithm can be used for the following incomplete-data likelihood function:

$$L(\mu, \sigma^2 | l) = \prod_{i=1}^{n} \sum_{j=1}^{m_i} f(l_{i,j} | \mu, \sigma^2)$$

E step: Estimate weights

$$w_{i,j} = \frac{f^{\mu_k, \sigma_k}(l_{i,j})}{\sum_j f^{\mu_k, \sigma_k}(l_{i,j})}$$

M step: Updated estimates of μ and σ^2 can be computed as the w weighted sample mean and variance of all fragment lengths, respectively.

Your algorithm must output the current parameter estimates and the log likelihood for each step.

You find BAM-files corresponding to the simulations under

`data/diff_simulation/(f1,f2)/(r1,r2,r3)/(gsnap,star,contextmap,tophat2).bam`
Reads have been simulated with a fragment length mean of 200 and a standard deviation of 80.

For each run, create a line plot showing (from left to right) the current mean estimates and a horizontal line corresponding to the true mean. In addition, provide error bars indicating the

current standard deviation and dotted lines above and below the true mean line to indicate the true standard deviation.

Try to initialize your EM with either ML estimates from unique lengths or by setting all weights to $1/m_i$. Derive the likelihood function and stop your EM algorithm when the likelihood improvement is smaller than 1E-8. Discuss the plots for both choices.

Finally, compute for each ambiguous read pair the posterior probabilities (likelihood of specific length divided by sum of all likelihoods for possible lengths). Create plots showing the cumulative distributions of the length with maximal posterior probability for read pairs with exactly two possible lengths, with three to five possible lengths, with six to ten possible lengths and with more than ten possible lengths.

You find a short introduction to the EM algorithm in:

Do, CB, Batzoglou, S (2008). What is the expectation maximization algorithm? Nat. Biotechnol., 26, 8:897-9. `http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html`