Bioinformatics III

Prof. Dr. Volkhard Helms
Trang Do
Summer Semester 2021

Saarland University
Chair for Computational Biology

Exercise Sheet 5

Due: May 27, 2021 12:00

Submit your solutions to trangdht.bioinfo@gmail.com with two attachments: (1) A ZIP file containing all your source code files, potential result files, figures and whatever else is needed to generate your solution, (2) a PDF file containing your answers. Subject of the email should be in the following format: BI3A5_LastName1_LastName2.

Differential Gene Expression Analysis, Gene Annotation Enrichment Analysis, Motif Sequence Finding

Exercise 5.1: Differential Gene Expression Analysis (50 points)

In this exercise, you are going to perform differential gene expression (DGE) analysis to find significantly deregulated genes between two biological conditions. Specifically, you are going to investigate the transcriptomes of human bronchial epithelial cells exposed to either cigarette smoke (labeled S) or air (labeled R) assessed at 1h, 2h, 4h and 24h post-exposure.
The file provided in the supplement contains a table of expression values from single-channel microarrays. The columns correspond to exposed conditions (smoke or air) and assessed time points (2h, 4h and 24h) in 3 replicates (A, B and C). The rows correspond to the probes on the microarray platform. The first two columns contain the identifier for each probe, as well as the associated gene symbol. Note that gene symbols can occur more than once, since microarrays often contain probes for multiple transcript variants of a gene.
So far, the dataset has been downloaded from Gene expression omnibus (GEO) and a *quantile normalization* has been applied. The probes were annotated with gene symbols and those without an associated gene name, as well as reference probes used for normalization, were removed.

(a) Write a Python program that reads the table into a data structure. Create methods that calculate the following values for every probe $i \in \{1, \ldots, n\}$ in the dataset:

(1) Let S_i and R_i be the lists of expression values for probe i with condition S and R assessed at a time point t. Implement a method to calculate the *arithmetic means* μ_{S_i}, μ_{R_i} and the *variances* $\sigma^2_{S_i}, \sigma^2_{R_i}$. Show these values computed at $t = 2h$, $4h$ and $24h$ in your report.

(2) Implement method **get_LFC()** to calculate the *log fold change* ($logFC$) between the two conditions:

$$logFC_i = log_2\left(\frac{\mu_{S_i}}{\mu_{R_i}}\right)$$

at time point t. What is the benefit of taking the logarithm of the fold change?

(3) Since $logFC_i$ alone is not a statistical test, we need some way of calculating a confidence level for the fold changes. For this reason, we are going to perform a gene-specific t-test to calculate a p-value p_i from S_i and R_i in the method **get_pval()**.
(Note: You can use the stats.ttest_ind function from the scipy package (with parameter equal_var = True).
In your report, shortly explain the meaning of the *p-value* returned by this function. Which problems could this specific function for calculating p-values face on this dataset? What would be a possible solution?
(Hint: Take a look at the previously calculated variances)

(4) The next step is multiple testing correction. For simplicity, we are going to implement the *Bonferroni*-correction in the method **adjust_pval()**. The adjusted p-value p_i^{adj} is calculated from p-value p_i as:

$$p_i^{adj} = min(1.0, n * p_i)$$

where n is the number of tests (i.e. the number of p-values). Why is p-value adjustment necessary? Which benefits and disadvantages would the *Benjamini-Hochberg*-method bring compared to *Bonferroni*?

(b) Finally, filter your table for 10 probes with the lowest *adjusted p-value* at $t = 2h$, $4h$ and $24h$. For each of these probes, please report the following information:

- Probe identifier
- Gene symbol
- Log fold change $logFC_i$
- Adjusted and unadjusted p-value (p_i^{adj}, p_i)

Exercise 5.2: Gene Functional Annotation Enrichment Analysis (30 points)

In this exercise you are going to add protein function annotations from the Gene Ontology (GO) to the differentially expressed genes from Exercise 1. The human GO annotation file containing the accession numbers for the protein database UniProtKB and GO terms specific for each protein is provided as supplementary data.

(a) The GO annotation file is tab–separated, apart from the initial header. The relevant columns are:

- Column 0: Name of the protein or gene database. Skip all entries that are not from UniProtKB.
- Column 1: Accession number of the gene or protein in the database.
- Column 2: Exactly one alternative name for the gene or protein.
- Column 4: GO identifier of the annotation.
- Column 8: Indicator whether the annotation belongs to the cellular component (C), molecular function (M) or biological process (P) ontology. Skip all entries that do not belong to the biological process ontology.

Associate the GO annotation ID of the entry with the differentially expressed genes (with $p_{adj} < 0.05$) found at different assessed time points ($t = 2h, 4h$ and $24h$)

(b) Implement a function that returns the n most common and n least common GO identifiers in a set of differentially expressed genes ($p_{adj} < 0.05$). If there are several GO identifiers that are associated with the same number of genes, choose the ones with the lower lexicographical order first. In your report, list the 20 most common annotations including how often they occur in the differential gene set at $t = 2h, 4h$ and $24h$.

(c) The enrichment of GO annotation for a gene set against a background set can be performed with a hypergeometric test with the probability P:

$$P = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}$$

which is the probability of observing k genes from a cluster of n genes by chance in a biological process category containing m genes from a total genome size of N genes. Implement method

hypergeometric_test() to compute the hypergeometric p-value for each GO annotation associated to the differentially expressed genes:

$$p = \sum_{i=k}^{min(n,m)} \frac{\binom{m}{i}\binom{N-m}{n-i}}{\binom{N}{n}}$$

(d) Apply Bonferroni correction (**adjust_pval()** from Exercise 5.1b) to the newly calculated p-value and report 20 biological processes enriched with the lowest adjusted p-value at $t = 2h$, $4h$ and $24h$. Are there any differences between the most common terms and significant terms? Comment on the changes in differential transcriptomes between two exposure conditions (cigarette smoke versus air) along the given time points.

Exercise 5.3: Position-specific scoring matrix (PSSM) (20 points)

PSSMs are useful for representing binding site or motifs in biological sequences. In this exercise, you will implement a a Python class **PSSM** (Position-Specific Scoring Matrix) to compute and visualize PSSMs for a finding motif from a given set of aligned sequences.

(a) The Python class **PSSM** should read in a list of l sequences and contain:

(1) **get_FrequencyMatrix()** - a method returning the frequency matrix containing the position-specific occurrences of each nucleotide from l sequences $f_{i,j}$:

$$f_{i,j} = \frac{n_{i,j}}{\sum_{i=1}^{A} n_{i,j}}$$

where $n_{i,j}$ are the occurrences of residue i at position j and A is the size of the alphabet.

(2) **get_CorrectedFrequencyMatrix()** - a method returning the corrected frequency matrix $f'_{i,j}$:

$$f'_{i,j} = \frac{n_{i,j} + p_i k}{\sum_{i=1}^{A} n_{i,j} + k}$$

where p_i is the prior residue probability for residue i and k is a random pseudo-weight.

What is the advantage of using the corrected frequency matrix over the frequency matrix?

(3) **get_ScoringMatrix()** - a method returning the score matrix $S_{i,j}$:

$$S_{i,j} = \frac{f'_{i,j}}{p_i}$$

where $f'_{i,j}$ is the computed corrected frequency matrix.

(4) **plot_SequenceLogo()** - method plotting the sequence logo representing a frequency matrix or scoring matrix. In this graphical representation, letters (nucleotide bases) are stacked at each sequence position in decreasing order of frequency/score. The letters' relative sizes indicate their frequency/score in the sequences. An example of the expected graphic is found in Figure 1.
Hint: You can use matplotlib for this task and the same coloring scheme for the letters as shown in the example.

(b) Use the implemented methods to compute the frequency matrix, corrected frequency matrix and scoring matrix to find the motif from the following 8 sequences:

TCACACGTGGGA
GGCCACGTGCAG
TGACACGTGGGT

CAGCACGTGGGG
TTCCACGTGCGA
ACGCACGTTGGT
CAGCACGTTTTC
TACCACGTTTTC

with $p_A = p_T = 0.325$, $p_G = p_C = 0.175$ and $k = 1$. In your report, include the computed matrices with their corresponding sequence logo and add your interpretation.

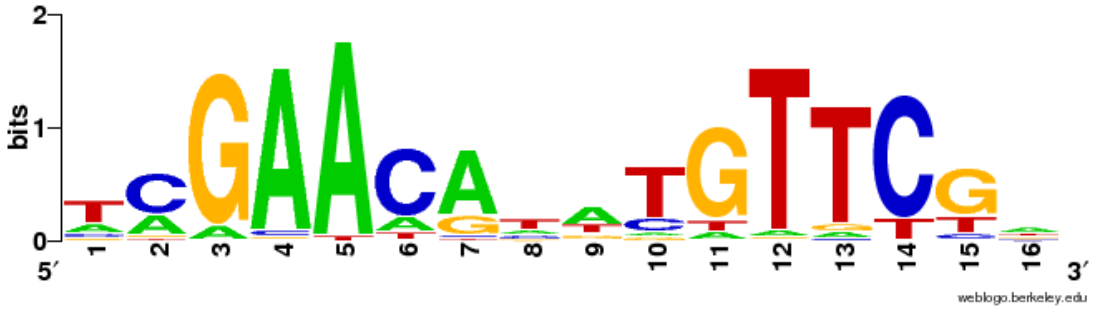

Figure 1: An example sequence logo