Examination Bioinformatics

Linköpings Universitet, IDA, Statistik

Course code and name:	732A51 Bioinformatics
Date:	2019/01/09, 8–12
Assisting teacher:	Krzysztof Bartoszek
Allowed aids:	The help material is included in the zip file **exam_help_material.zip**. In the zip **exam_material.zip** file you will also find data files and help scripts for your exam.
Grades:	A= $[18-20]$ points B= $[16-18)$ points C= $[14-16)$ points D= $[12-14)$ points E= $[10-12)$ points F= $[0-10)$ points
Instructions:	Provide a detailed report that includes plots, conclusions and interpretations. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in an appendix. If you are asked to do plots, then make sure that they are informative, have correctly labelled axes, informative axes limits and are correctly described. Points may be deducted for poorly done graphs. Name your digital part solution files as: **[your exam account]_[own file description].[format]** There are **THREE** assignments (with sub–questions) to solve. Include all code that was used to obtain your answers in your solution files. Make sure it is clear which code section corresponds to which question. Your code should be complete and readable, possible to run by copying directly into a script. Comment directly in the code whenever something needs to be explained or discussed.

Problem 1 (7p)

In the file `GSE7765.txt` (from `ftp://ftp.ncbi.nih.gov/pub/geo/DATA/SeriesMatrix/GSE7765/GSE7765-GPL96_series_matrix.txt.gz` and from a study on the effects of dioxin on breast cancer cell lines `https://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS2744`) you will find gene expression measurements for various genes from six samples. Use the code in `readGSE7765.R` to read the data into `R`. The rows are the genes and columns are the samples. Samples GSM188013, GSM188014, GSM188016 are controls (healthy) while samples GSM188018, GSM188020, GSM188022 are cases (cancerous).

a) Identify differentially expressed genes (choose the significance level yourself, so that it is useful). In the file `readGSE7765.R` you will find a function called *`ttest_for_GSE7765()`* that might be useful. For speed of calculations and easiness of implementation please consider using the `apply()` function, to compare all the genes between cases and controls. It might take a few seconds to run to compare all genes.

TIP: Think on what scale (e.g. normal, log) should the data be analyzed.

b) What would happen if in a) you had furthermore used the function *`p.adjust()`*? Is it sensible or not sensible to use it in this particular situation?

c) Assume that samples GSM188013 and GSM188018 are paired, i.e. both come from the same individual but the first was taken from healthy tissue while the second from cancerous tissue. Provide some plots that compare the gene expressions in a meaningful way (e.g. if it helps think that cases were dyed red while controls green). If possible and meaningful mark on your plots the genes found in a).

Problem 2 (7p)

You are given two DNA sequences ATGGA and ATCGTA. Choose some distance function (for which these two sequences are valid input) and calculate the distance between these two sequences under the chosen function.

Perform (manually) a global alignment between the two sequences. Choose all necessary parameters yourself (however they should be non–trivial, i.e. **NOT** 0, and furthermore meaningful). Do not forget to report the dynamic programming matrix and how one obtains the optimal solution from it.

Explain what an alignment is from a biological point of view.

Problem 3 (6p)

In the file `SFXN5.fasta` you will find part of the DNA deposed in GenBank entry AY044437.1. The provided DNA is part of the SFXN5 gene (found on chromosome 2) and includes the code for the human sideroflexin 5 protein. This protein is related to citrate (derivative of citric acid, found in citrus fruits) transport.

The goal of this exercise is to find the protein code of the sideroflexin 5 protein, i.e. correctly translate the nucleotide sequence into the protein sequence. Your final provided sequence (in a fasta file) should be only the sequence of the protein without any extras. Explain what you did, why and motivate why you think your translation is correct.

TIP: You might want to look at **ape**'s functions: *`ape::complement()`*, *`ape::read.FASTA()`*, *`ape::trans()`* and the options they have.

The correctly translated protein sequence should contain the substring GELEE. Take note that you may **NOT** use the presence of this substring to motivate correctness of your translation.