LGBIO2010 – Bioinformatics

Assignment 3

Gene Expression Analysis

1 Gene Expression Data

In this assignment you will analyze gene expression data and gradually select most informative markers for a diagnosis task. The data is made of 54613 probeset expression values from microarrays of 179 different patients, suffering from Glioblastoma (a form of brain cancer). We search in particular for expression markers for predicting whether a patient suffers from *Glioblastoma* or *No tumor*[1]. The data has already been preprocessed and is available as a compressed file, either **`Glioblastoma.zip`** or **`Glioblastoma.Rdata`**, on Moodle. You have two options to start working on this data.

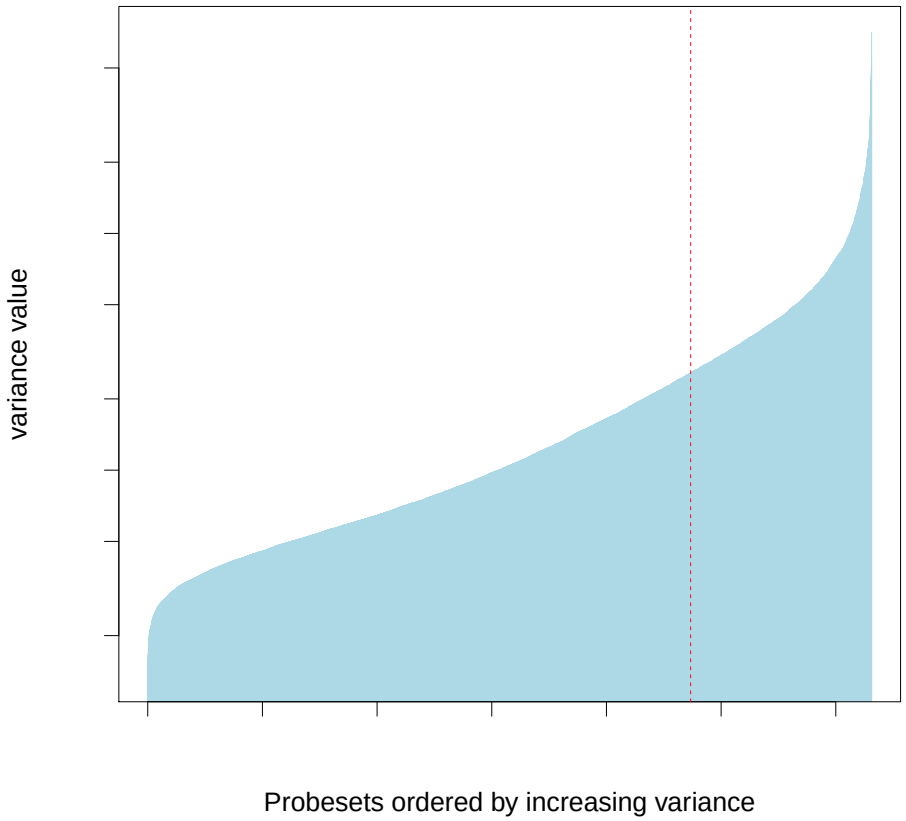
1. Download the **`Glioblastoma.zip`** archive file and uncompress it (= unzip it) to produce **`Glioblastoma.csv`**. Such a **`.csv`** file contains a large two dimensional table in which the respective entries are separated by a coma. Most data analysis softwares can take such a coma separated file as input. Each line contains 54615 entries corresponding to the columns of the table in the following order: A string specifying the patient ID, followed by 54613 probeset expression values, and a string which can either be *Glioblastoma* or *No tumor*. This status corresponds to the ***two conditions of interest*** in the subsequent analysis. The first line contains the names of the columns: an empty string, followed by the identifiers of the 54613 probesets, followed by the name of the last column: "labels". The subsequent 179 lines contain the actual data.

   You should read the **`Glioblastoma.csv`** file using **`R`** and store its content in an appropriate data structure to further process it. For example, the **`read.csv`** function will be useful to read a **`.csv`** file and store it in a **`R`** data frame.

2. Alternatively, you can download the **`Glioblastoma.Rdata`** from Moodle and then directly load it as a data frame called **`data`** in a R session through **`load("Glioblastoma.Rdata")`**. Both options should be equivalent but the second is faster.

[1] The original work was published in http://www.cell.com/cancer-cell/fulltext/S1535-6108(06)00084-5.

2 Non-specific filtering



Note: the above plot is just illustrative of your task. It does **not** correspond to the actual data you must process.

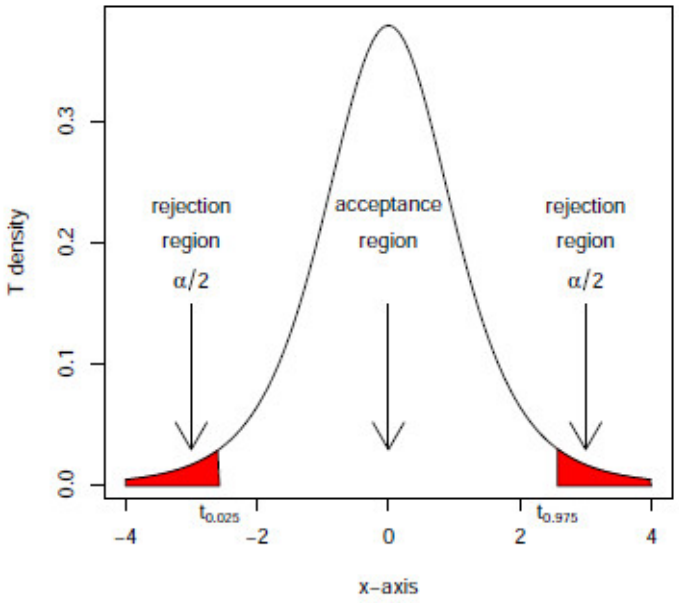
2.1. Rank the various probesets by increasing variance on all samples. Keep only 25% of probesets with the larger variances

2.2. Report how many probesets are kept after this non-specific filtering and the minimal variance considered for a probeset to be kept.

2.3. Report a table with the names and variances of the top 5 probesets with the largest variances.

For the rest of this assignment, you will only consider the dataset restricted to those features that have not been filtered out at this step.

3 Differentially expressed probesets



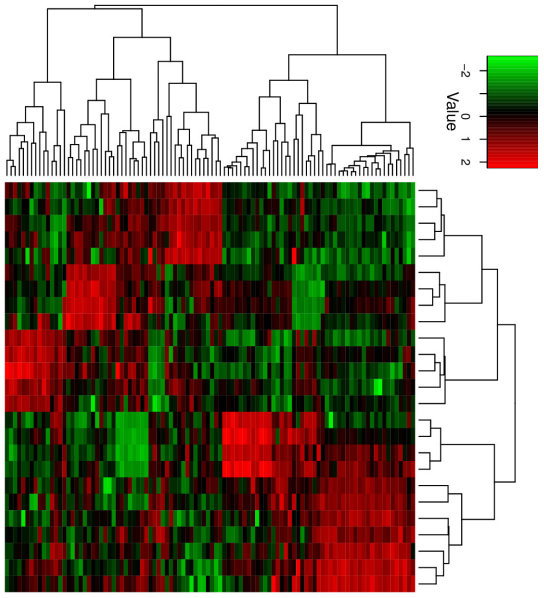
3.1. Rank the various probesets using a t-test by increasing p-values to distinguish between the two conditions under study.

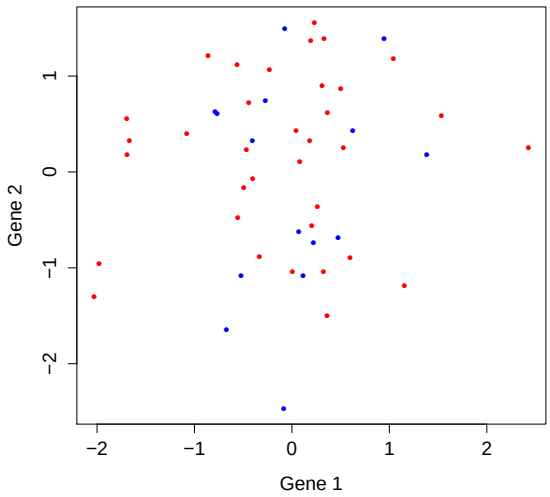
3.2. How many features (= probesets) are deemed statistically significant according to such a t-test? Consider a 5% significance level and no specific correction at this point.

3.3. How many features are considered significant after a Bonferroni correction of the original t-test? Report a table including the names of the top 10 most discriminating features and the associated corrected p-values in increasing order.

3.4. How many features are considered significant after a FDR correction of the original t-test? Report a table including the names of the top 10 most discriminating features and the associated FDR corrected p-values in increasing order.

4 Heatmaps and data visualization





Note: the above plots are just illustrative of your tasks. They do **not** correspond to the actual data you must process.

4.1. Report a heatmap[a] of all samples along the 50 most differentially expressed features.

4.2. A heatmap illustrates two hierarchical clusterings, respectively along the features (= probesets) and along the samples. Discuss to which extent the samples are clustered consistently with the conditions of interest[b]. Explain the reason(s) of possible difference(s) if any or why the clustering is fully consistent with the conditions of interest.

4.3. Report a 2-D plot of all samples along the 2 most significant features according to your differential expression analysis. Use some specific coding (*e.g.* a color code) to label each sample with its respective condition (*Glioblastoma or No tumor*). Report the names of the chosen features along the x- and y-axis of your plot. Do you conclude from such a plot that the 2 conditions can be perfectly or well distinguished? Specify for each of these two features whether the corresponding probeset tends to be under-expressed or over-expressed among *Glioblastoma or No tumor* patients. Explain how such an analysis is consistent or not with the previous heatmap.

4.4. Report a 2-D plot of all samples along the 2 least significant features among those kept after the non-specific filtering. Use some specific coding (*e.g.* the same color code as before) to label each sample with its respective condition (*Glioblastoma or No tumor*). Report the names of the chosen features along the x- and y-axis of your plot. Discuss this visualization and comment it in contrast to the previous 2-D plot.

[a] Hint: check the **`heatmap.2`** function of the **`gplots`** R package.
[b] Hint: check the **`RowSideColors`** or **`ColSideColors`** arguments of the **`heatmap.2`** function.

5 Predictive Models

5.1. Split the available samples into a 80 % training and 20 % test. Specifically use the first 143 samples as training and the last 36 samples as test.

5.2. Fit a linear SVM on the training using all probesets obtained after non-specific filtering (see section 2). We recommend the **`LiblineaR`** R package. Report a table with the 10 most important features according to the absolute weight values of such a linear model. The table should contain the identity of those 10 probesets and the absolute weight value associated to each of them in the linear SVM. This table should also contain the respective ranks of those features according to the ranking of p-values computed in section 3. Do you observe whether the most important features of a SVM model match the most differentially expressed probesets? Why?

5.3. Compute the labels (either *Glioblastoma* or *No tumor*) of the ***test*** samples predicted by the linear SVM estimated on the ***training set***. Report a confusion matrix between predicted and true labels on the ***test set***. What is the classification accuracy of the SVM on the test?

Hints:

- You are invited to consult the documentation of the **`LiblinearR`** package. Once a model has been built from a training sample (**`model <- LiblineaR(...)`**), the model parameters can easily be accessed as **`model$W[1,]`**

6 Feature selection

6.1. Compute the mutual information between each feature and the class label (*Glioblastoma* versus *No tumor*) on all samples. We recommend the **mRMRe** R package. Report a table with the top-10 features (here probesets) which exhibit the largest mutual information (MI) with the class label. This table should contain 3 columns: the ranking of each feature (from 1 to 10), the feature name and the MI value.

6.2. Select the 10 most relevant features according to the mRMR algorithm. Report a table containing the rank of each feature, here the rank denotes the iteration of the mRMR algorithm where such a feature is selected, the feature name and the associated mRMR scores[a].

6.3. Explain the similarities and/or differences between both tables computed here. Contrast these results with the feature rankings computed in section 3 and section 5. Do they differ? Why?

[a] At each iteration of mRMR, a specific feature maximizing a score is selected. This score evaluates the difference between relevance and redundancy (see the course slides).

Hints:

- You are invited to consult the documentation of the **mRMRe** package. The key methods you should use here are **mim(...)** and **mRMR.classic(...)** to perform a feature selection using the mRMR algorithm. Note that this package also implements a further extension for which an ensemble of such selection filters is built on repeated resamplings of the data and then aggregated. You can ignore this extension here.
- The **mRMRe** package computes mutual information between numerical variables and/or *ordered* factors. In **R**, a class label is typically represented by a **factor**, which is not necessarily ordered. Assuming **label** is indeed an unordered factor, you can transform it into an ordered factor simply as **label <- factor(label, ordered=TRUE)**.
- The R command **fs <- mRMR.classic(...)** assigns to the variable **fs** a specific **R** *object*, which is a specific data structure made of several *attributes*. The following instructions could be useful.
  - **class(fs)** to know the class of such an object,
  - **names(attributes(fs))** to get the names of the attributes of such an object,
  - **fs@***XXX* to access the specific attribute *XXX* of such an object.

Results

Upload on Moodle a **zip archive file**, for this assignment, containing

- a PDF report including your answers to all questions in a frame in the present document. Your report should mention any software resource you have been reusing (typically publicly available R packages). Please report benefits/problems you experienced while using existing public softwares.
- the R code you have been writing for this assignment.

Submitting the information requested above for grading your work implies that each member of your group is accepting the *anti-plagarism* policy summarized below.

I hereby certify that the results and code that I will submit for this project is coming from my own work and that of my teammate (if any in my group). The submitted works will not be (even partial) copy/paste from information found on the internet or from the work of other groups.

I also certify that I will not distribute any answer or code related to these projects, in person or on any repository (github, bitbucket, Facebook groups, ...) accessible to anybody beyond my teammate, even after the deadlines.

Any violation of the above statements will be considered as cheating and will be reported as such to the President of the Jury.