Analyst Training Program (ATP) Coding Assessment

This assignment is meant to evaluate your coding style, analytical approach, and writing ability. Solve the problems to the best of your ability and compose a document describing your results. The result document should contain a written description of your approach and the results you obtained. In a separate document or collection of documents, provide all source code used to solve the exercises. Please provide your result document as a pdf and your source code as plain text files.

1 Analysis of gene expression data

1. Write a Matlab function called *gctparse.m* that reads a GCT file. Details of the GCT file format can be found at:

   `http://www.broadinstitute.org/cancer/software/genepattern/gp_guides/file-formats/sections/gct`

   The function should accept a .gct file as input and return a Matlab structure with the following fields:

   - *ge*: a matrix of expression values [genes × samples]
   - *gn*: a cell array of row names
   - *gd*: a cell array of row descriptors
   - *sid*: a cell array of column names

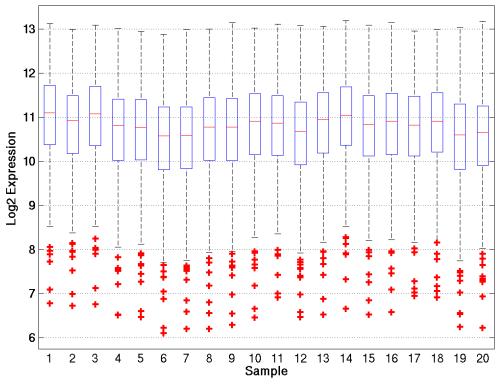
2. The file *gene_expr_500x204.gct* contains gene expression values for 500 genes (analytes) and 204 samples. Read this dataset and generate the following plots :

   (a) *distribution.png*: Distribution boxplots of log2 expression values of the first 20 samples. See Figure 1(a).

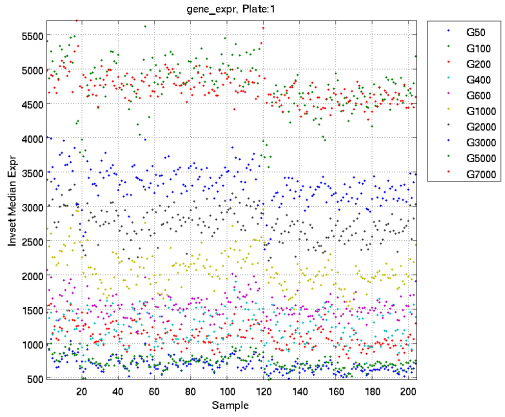
   (b) *profile.png*: A plot of the raw expression of the first 10 analytes vs the sample index. See Figure 1(b).

   (c) *calibplot.png*:A calibration plot of the raw expression of the first 10 analytes vs the analyte index. See Figure 1(c).

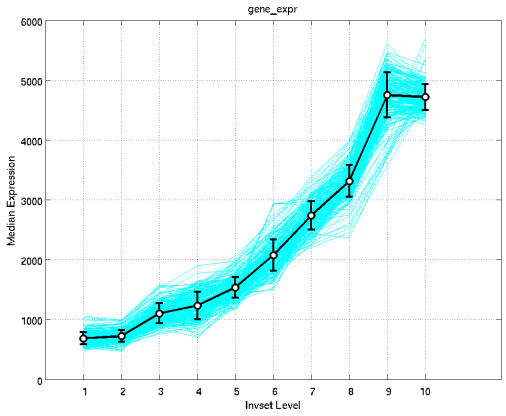
   (d) Provide Matlab source code for generating all plots.



(a) distribution.png



(b) profile.png



(c) calibplot.png

Figure 1: Analysis of gene expression data

2 Analysis of flow cytometry data

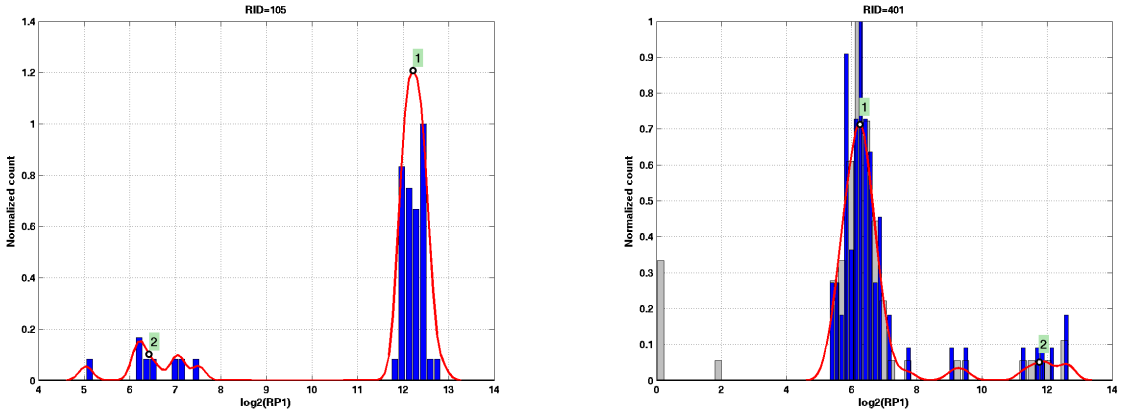
A multiplex flow cytometer can detect and measure the fluorescence intensity of up to 500 different classes of particles suspended in a fluid sample.

The file *P08.gct* contains the flow-cytometer output of a single sample. Each row in the file represents data from a single detected particle. For each particle the following information is recorded:

- *RID*: the particle class id [1-500]
- *RP1*: the fluorescent signal intensity of the particle
- *TIME*: the time elapsed since the start of the read (in 100th of a second)

Figure 2 shows the distribution of the log2 signal intensity of two classes of particles [RID=105, 401]. While one expects the intensity of a given type of particle to be stationary, distribution plots such as the ones shown in Figure 2 ignore the temporal aspects of the signal.

Analyze the sample using Matlab to determine if the intensity of the particles drifts over time. Describe your findings in not more that 250 words. Support your answer with figures. Provide source code for all analysis.



(a) RID=105 (b) RID=401

Figure 2: Distributions of flow-cytometry data

3 Regression modeling of gene expression data

It is commonly observed that gene expression data is highly correlated. You will leverage this knowledge to build inference models that predict the expression of genes based on the expression of other predictor genes. You are provided with the following data:

train.gct is the training set of gene expression profiles of 22268 genes across 2519 samples.

predictors.grp is a list of 979 predictor gene ids (a subset of the genes listed in *train.gct*).

test.gct is a test set of expression profiles of the 979 predictor genes for 96 samples.

1. Choose any multi-variate regression algorithm and build a prediction model using the training set *train.gct* (Note that you do not have to implement the regression algorithm and can employ in-built or external code). Use the 979 genes listed in *predictors.grp* as the predictor variables in your model. The remaining 21289 genes are the dependent variables.

2. Evaluate the training error of the model using a suitable metric(s) and plots.

3. Apply the model on *test.gct* to generate a predicted dataset of profiles *predicted.gct*. This is a gct format file of the model output (including the predictor genes) and should have the dimensions 22268 genes $\times$ 96 samples.

Provide commented source code in Matlab or other language for all steps in the analysis.

4 Shell scripting

Most installations of unix or GNU/Linux provide command-line utilities that can simplify common text processing tasks. Write a shell script using one or a combination of these utilities to complete the following tasks:

1. Extract the sample names from the file *gene_expr_500x204.gct* used in Q1. Save the list (one sample name per line) to a text file called *samples.grp*

2. Rename the sample names in *samples.grp* to a three-character fixed width format with zero-padded digits. For example the names 'A1','B9' and 'C10' are renamed as 'A01', 'B09' and C10 respectively. Sort the list alphabetically and save the list to a text file called *sorted_samples.grp*

Provide source code and the output files.