

Mini Assignment 1: Function Programming

This mini assignment has three parts, each asking you to write a function. Each part has its own due date.

You can use attached .mat file with MATLAB workspace for testing. This workspace contains the data stored as a table and as a cellarray. Test your function on both. The only thing that should be different is the function call.

Your code and function should be well commented, no magic numbers, following proper programming practices, avoiding unnecessary loops.

Part A:

Write a function 'SampleNormalizationRF' that takes as an argument a matrix, of type double. The function computes the sum of values for each column in the matrix (use function 'sum') and then divides each value in the column of the matrix by its sum.

The function returns resulting matrix.

The purpose of this function is to perform relative frequency (RF) normalization of genomic data.

Part B:

Write a function 'SampleCorrelation' that takes as arguments: 1) a matrix of type double, 2) A string, indicating the type of correlation to perform ('Spearman' or 'Pearson')

For each sample (column) in the input matrix, the function computes an average correlation to all other samples (columns). The type of correlation should be of the type specified by the second input argument.

The function returns a vector with the average correlations. The vector will have the same length as the number of columns in the matrix.

Things to consider:

- Check if the input is valid
- Initialize default return vector
- Check description for the function 'corr' in MATLAB

The purpose of this function is for each sample to compute an average correlation to all other samples.

Part 3:

Write a function 'MarkLowCounts' that takes as arguments:

- A matrix, of type double,
- A number Q, such that $0 \leq Q \leq 1$, indicating the quantile level to establish the filtering threshold

The function computes:

1) the threshold level for filtering the low expressed features

suggestion: use function 'quantile' and 'reshape' to compute overall level of expression at the quantile level Q:

```
[MatrixAsVector = reshape(datain, [1 size(datain,1)*size(datain,2)]);]
```

It is just a suggestion.

2) for each feature (row of the matrix) – check and mark if its expression is above the threshold in at least 1 sample (column)

The function returns a boolean vector (True/False) indicating which of the features are below the provided quantile. The values should be **True for below** the Q threshold and False otherwise.

The purpose of this function is to identify lowly expressed features.