**Parts of the MDC algorithm**

1) Input files

The software requires two files as input. The first one is the data file which contains a table with the features and all the samples of the different time intervals and can be either a txt or a csv file.

The second one, is an UTF-8 or ANSI format txt file, in which the correlation between the sample class labels, for which the clustering will take place, along with their corresponding samples in the different time intervals have to be specified.

For Data Matrix Files from the Gene Expression Omnibus (GEO), the following steps are required in preparation of those two files.

For the data file, the user needs to:

* Download the Series Matrix File txt file of the desired GSE Series from the Gene Expression Omnibus.
* Delete everything from within the txt file up until (and) the line ”!series\_matrix\_table\_begin”.
* Delete the very last line “!series\_matrix\_table\_end”.
* Save the file and exit

For the correlation file, the user needs to create a txt file in UTF-8 or ANSI format, like the example below.

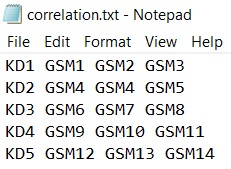


Fig 1. Correlation file example.

In this particular example, KD1 is a sample class label and GSM1, GSM2 and GSM3 correspond to that label’s samples at (e.g.) three time intervals. The naming of the sample class labels is arbitrary.

The following have to be true for the correlation file:

**1)** One space per word, **2)** No space at the end of each line, **3)** No space at the end of the file, **4)** No number as the first label letter.

2) Data Preprocessing

To avoid errors, MDC replaces NaN values in the Data file if any, with zeroes. The user is advised to use the preferred imputation technique on the Data file before it is feeded into MDC.

MDC creates the 3D data matrix based on the input files. The dimensions are , where is the number of sample class labels (labels in the correlation file), is the number of genes (features in the data file) and the timesteps (number of same-sample class label samples in the correlation file).

Scale normalization to the range 0 - 1 is used by default on the 3D data matrix for easier training and debugging.

Principal Component Analysis (PCA) is used to reduce the very high dimensionality of the Data file before is it feeded into the MDC training. MDC provides different options for the number of principal components as discussed in the “ MDC PARAMETERS” session.

3) Neuron Initialization

Grid-like neuron initializations, as in conventional Self-Organizing Map implementations, are not possible due to time-related data. The user has to input the number of neurons and the initialization technique. The different options are discussed in the “ MDC PARAMETERS” session.

4) Neural Network Training

MDC utilizes competitive learning theory to adjust neuron positions in time-dependent, high dimensionality feature space and assign them as clustering centers. The Euclidean distance is first computed between the input sample and all the weights of the neurons in the neural network. The neuron that has the smallest distance is declared as the best matching unit and its weight along with its nearest neighbor neurons are re-adjusted to closer mimic the input sample. MDC is able to train the neural network given the input samples and the neurons weights as matrices and not just as vectors, containing both the spatial and the temporal information.

The update function for the weights is defined as:

|  |  |
| --- | --- |
|  | (1) |

where:

: is the weights matrix of neuron with index at time , with being the current iteration,

is a random data point of the dataset,

: the BMU,

: the learning rate, which decreases exponentially:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (2) | |  |  |  |

with being a constant, and

**:** the neighborhood function, which dictates the cooperation between neurons. It decreases exponentially and includes a reducing Gaussian distance function [5]:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (3) | |  |

where isthe Euclidean distance between a neighbor neuron and the BMU which is calculated by the Frobenius Norm of the neuron matrices difference:

|  |  |
| --- | --- |
|  | (4) |

is the standard deviation of the initial Euclidean distances of the randomly initiated neurons and is a constant.

More information about the training parameters at the “ MDC PARAMETERS” session.

5) Clustering

The clustering at the end of the neural network training is based on the Euclidean distances between the labels (3D data points) and the neurons (3D data points).