

## README: IMM (Infinite Mixture Model-based algorithm)

Using IMM involves the following steps:

1. Convert the input file to the format required by IMM (Perl script written by Ka Yee)
2. Run IMM, which is available from <http://homepages.uc.edu/~medvedm/BioinformaticsSupplement.htm>. The output of IMM represents posterior probabilities for each pair of objects. IMM is written by Dr. Mario Medvedovic in C++. The current implementation of IMM is available as a windows executable<sup>1</sup>.
3. Convert the pairwise probabilities to a similarity matrix. (Perl script written by Ka Yee). Then, cluster the output posterior probabilities from IMM. The Java bytecode files is available. Alternatively, you may use Matlab or any of your favorite clustering implementation.

Perl scripts and Java bytecode files for steps 1 and 3 are available. We make absolutely no guarantee that these scripts and bytecode files will run properly on your system.

Java SDK 1.4 for both windows and linux can be downloaded from <http://java.sun.com/j2se/1.4/download.html>, while Perl for windows can be downloaded from <http://www.activeperl.com>. Java and Perl should come as part of mac OS 10.2.

### To uncompress the archive:

- On linux:  

```
tar -xzf pre_post_process.tar.gz
```
- On windows: WinZip can extract these files. There are probably other windows applications that can do it as well.

### Format of input files:

Tab-delimited text files:

- The first row is a header row.
- Column 1: gene identifiers (eg. Gene names or ORF)
- Column 2: documentation or other gene names
- Column 3 onwards: expression values for repeated measurements of the same experiments are in consecutive columns

### Step 1: conversion of input format

This step can be skipped. You can format your input file in Excel or any other spreadsheet program. For details of IMM input format, please refer to “Readme.txt” from the IMM downloaded archive.

The Perl script “txt2inp.perl” can be used to convert input files from the input format described above to the input format for IMM.

- On linux or unix:

---

<sup>1</sup> Unfortunately, we do not yet have a linux or mac executable for the second IMM step. So, you need to have access to a windows machine to run IMM for now.

```
./txt2inp.pl <input filename> <# experiments> <# replicates>
```

- On windows:

```
perl txt2inp_win.pl <input filename> <# experiments> <# replicates>
```

**Example:** to convert “test\_10gene\_5expt\_2rep.txt” (which consists of 10 genes, 5 experiments and 2 repeated measurements) to the input format for IMM:

- On linux or unix:

```
./txt2inp.pl test_10gene_5expt_2rep.txt 5 2
```

- On windows:

```
perl txt2inp_win.pl test_10gene_5expt_2rep.txt t 5 2
```

Output: file “test\_10gene\_5expt\_2rep.inp”

## **Step 2: run IMM**

Run IMM, which is available from

<http://homepages.uc.edu/~medvedm/BioinformaticsSupplement.htm>. Please read

“Readme.txt” from the IMM archive for detailed instructions.

**Example:** use the suggested “parameters.prm” and “posthoc\_parameters.prm” files, double-click on GIMM, and then double-click on “post\_hoc”.

Output: “test\_10gene\_5expt\_2rep.zm”

## **Step 3: clustering the posterior probabilities**

### **Step 3a: conversion to similarity matrix**

The Perl script “zm2simMatrix.perl” converts the output zm file from “post\_hoc” to a similarity matrix for clustering algorithms.

- On linux or unix:

```
./zm2simMatrix.pl <zm filename> <# genes> <# samples> <# burnin>
```

- On windows:

```
perl zm2simMatrix_win.pl <zm filename> <# genes> <# samples> <# burnin>
```

Note that <# samples> is equivalent to “n\_samples”, and <# burnin> is equivalent to “burn\_in” in the parameter file “posthoc\_parameters.prm” for IMM.

**Example:** to convert “test\_10gene\_5expt\_2rep.zm” to “sim\_matrix.txt”:

- On linux or unix:

```
./zm2simMatrix.pl test_10gene_5expt_2rep.zm 10 20000 10000
```

- On windows:

```
perl zm2simMatrix_win.pl test_10gene_5expt_2rep.zm 10 20000 10000
```

### **Step 3b: cluster the similarity matrix**

Please note that you can apply any clustering implementation that takes the similarity matrix as input for this step. The java bytecode files provided here were used for the results in the manuscript.

## **How to run the java executable?**

- These bytecode files (with the “.class” suffix) produce clustering results from **hierarchical agglomerative** clustering algorithms using the pairwise posterior probabilities generated from IMM.
- General format:  

```
java hieclustSim -r <# genes> -NoC_range <range of number of clusters> -
step <step size of number of clusters> -alg <algorithm> -doc <documentation
filename> input similarity matrix
```
- To see all these options:  

```
java hieclustSim -
```
- Interpretation of options:
  - -alg <algorithm>, where <algorithm> can be:
    - **-alg avg** : average linkage (default, if unspecified)
    - **-alg complete**: complete linkage
    - **-alg single**: single linkage
  - -doc <documentation filename>, where <documentation filename> is the file with gene names and documentation, e.g. “test\_10gene\_5expt\_2rep.txt”. For a detailed description of format, see “doc\_javacode.pdf”.
- Examples:
  - To cluster the 10 genes in the input file “test\_10gene\_5expt\_2rep.txt” to produce 3 clusters using **average-link**:  

```
java hieclustSim -r 10 -NoC_range 3 3 -step 1 -alg avg -doc
test_10gene_5expt_2rep.txt sim_matrix.txt
```

This command should produce clustering result “OutSim\_AvgLink\_3.txt”.

### Format of output files:

- A tab-delimited file “OutSim\_<algorithm>\_<number of clusters>.txt” with the following headings  
GeneID<tab>ORF<tab>GeneName<tab>Cluster#(number of clusters<newline>
- GeneID’s are unique identifiers for genes. They follow the order of the genes appearing in the input file. For example, the first gene in the file is 0, and the second gene is 1 etc.
- Genes with the same cluster number belong to the same cluster. For example, gene #0 (G1) and gene #1 (G2) were assigned to the same cluster (# 2) in “OutSim\_AvgLink\_3.txt”.