

TimesVector (v1.0)

User Manual

1. Pre-requisites

Required python modules:

- 1) Scipy
- 2) Numpy

Required R libraries:

- 1) Skmeans

2. Installing TimesVector

Download TimesVector at http://biohealth.snu.ac.kr/TimesVector/download/TimesVector_v1.0.tar.gz

Uncompress file

```
> unzip TimesVector_v1.0.tar.gz
> export TIMESVECTOR=/home/inukj/my_softwares/TimesVector/bin
> export PATH=$PATH:$TIMESVECTOR
```

3. Running TimesVector

```
> bin/TimesVector
usage: bin/TimesVector [ h | gctdco ]
```

This script runs TimesVector.

Parameters(all mandatory):

- g The path to the gene expression file
- c Number of classes (INT)
- t Number of time points per class (INT)
- d Type of data ['m': Microarray, 'n': NGS]
- k K number of clusters (INT)
- o Output directory for results
- h Show this message

All parameters are mandatory.

The gene expression file is the only required input file. The format of the gene expression file is shown in Section 4.

-c is the number of sample conditions (or phenotypes) in the gene expression file (INTEGER)

-t is the number of time points in each sample condition (INTEGER)

-d is the type of the data. If gene expression data is from microarray data 'm'. If data is from high throughput sequencing data (i.e., RNA-seq) 'n' (CHARACTER).

-k is the number of clusters desired to detect (INTEGER). We recommend to choose a K close to the following equation.

$$K = -85.71 + 28.57x,$$

where x is the product of C (# of conditions) and T (# of time points).

-o is the output directory for the clustering results

The gene expression file of GSE11651 is included in the 'example' directory, "GSE11651_data.txt".

The command line for executing TimesVector using the example data will be as follows,

```
> TimesVector -g example/GSE11651_data.txt -c 5 -t 3 -d m -k 400 -o results
```

4. Gene Expression File Format

The gene expression file is a **TAB** delimited gene expression matrix.

Header

The first line of the file serves as a header.

The first column "GeneID" of the header is mandatory and must be used as is.

The following columns represent each sample conditions and their associated time points. The conditions need to be in order as well as the time points. Each column represents a single time point of a condition.

The name of a column follows the following syntax:

"Condition"_"Time Point"

The condition and time point are separated by an under line ("_"). The name of the condition and time point can be any string of characters.

For example, for a time-series data with three conditions (A, B and C) with three time points (20min, 40min and 60min), the header will look as follows:

```
GeneID A_20min A_40min A_60min B_20min B_40min B_60min C_20min C_40min C_60min
```

Gene expression values

Each row following the header represents a gene. The first column represents the gene ID. The remaining columns represent the gene expression value associated with the condition and time point of each column.

A toy example file with five genes will look as follows:

GeneID	A_20min	A_40min	A_60min	B_20min	B_40min	B_60min	C_20min	C_40min	C_60min
P53	5	9	10	6	8	9	8	4	2
bZIP	3	13	18	4	15	21	5	1	1
WRKY	25	27	28	24	25	26	25	20	15
ERF	8	11	12	9	10	11	9	8	5

5. Output files

There are a total of three sets of output files.

- 1) The first set of output files are – Kx.cluster, Kx.prototype. These files are the result files output from skmeans. The 'x' will be in integer, which represents the number of clusters to be detected (the input parameter "-k").

The Kx.cluster file shows the total list of genes and their assigned cluster ID.

The Kx.prototype file shows the centroid values of each cluster

- 2) The second set of output files are the result files of cluster classification – DEP, SEP and NEP cluster files.

DEP_clusters.dat: The list of clusters identified as DEP type clusters.

DEP_genes.dat: The list of genes in the DEP type clusters.

SEP_clusters.dat: The list of clusters identified as SEP type clusters.

SEP_genes.dat: The list of genes in the SEP type clusters.

NEP_clusters.dat: The list of clusters that failed to be classified as a DEP or SEP.

NEP_genes.dat: The list of genes in that failed to be assigned to a DEP or SEP cluster.

- 3) The third set of output files are the visualized plots of DEP and SEP clusters and their genes. These are located in the "plots" directory within the output directory.

The plots for the cluster representatives are located in the DEP_clusters and SEP_clusters directories.

e.g., results/plots/DEP_clusters/cluster_2_repr.pdf

The plots for the genes in each cluster are located in the DEP_genes and SEP_genes directories.

e.g., results/plots/DEP_genes/cluster_2_genes.pdf

