Manual for TimePath

Siddhartha Jain sj1@cs.cmu.edu

Requirements and General instructions

- The software is available at `www.sb.cs.cmu.edu/timepath`
- The github is located at `https://github.com/tmfs10/timepath`
- You must have Java 7.0 or above and Python 2.7+ installed on your machine.
- You must have a GML file visualizer like Cytoscape to view the graph visualization

Data needed

1. A protein protein interaction network in the format

   ```
   <protein> <interaction type> <protein> <probability>
   ⋮
   ```

   The columns have to be separated by the **tab** character. The `<interaction type>` can be one of `pp` (protein-protein), `ptm` (post-translational modification), and `pd` (protein-dna). The latter two are *directed* whereas `pp` is *undirected*. A sample protein-protein interaction network with gene names under the HGNC symbol naming scheme is provided in the file `ppi_ptm_pd_hgnc.txt`.

2. A list of transcription factors (TFs). A sample list containing 348 TFs is provided.

3. A list of TF-gene interactions in the format

   ```
   TF Gene Prior
   <TF> <Gene> <Prior>
   ```

   Again, the columns have to be tab separated. A sample TF-gene interaction file is provided in the file `human_encode_100.txt`.

4. A list of target genes for a particular time point (`TPtargetsfile`) is in the format

   ```
   <target gene 1 for phase 1> <target gene 1 for phase 2>...
   <target gene 2 for phase 1> <target gene 2 for phase 2>...
   ```

   and so on. The targets for each time point should be separated by a tab character. The targets should be ranked by how likely they are to be a target for that time point with the most likely target first. One can do this using p-values generated by a software like Deseq (for RNA-seq data). Another popular method is to rank the genes by most differentially expressed to least.

   Note: The number of targets for each time point must be the *same*. If they are not, please add in the dummy target XXXX as needed to make them the same.

5. The time series file should be in the format
`<gene name> <log-fold change for timepoint 1> <log-fold change for timepoint 2> ....`
`<gene name> <log-fold change for timepoint 1> <log-fold change for timepoint 2> ....`
It should also have a ***header*** row at the top. The log-fold change can be with respect to a 0 time point or with respect to a control condition depending how the exact needs.

Config file

Mandatory parameters

1. `numTPGenes` - The number of *targets* to select for each time point
2. `TPtargetsFile` - The path to the file containing targets for each time point as in point 4 in the “Data needed” section.
3. `timeseriesFilepath` - The path to the file containing the log fold change time series gene expression data. If you are merging time points together (we HIGHLY recommend merging time points to 4-5 phases for easier visualization. We merged the HIV expression time points down to 3 in our paper), then please have the log fold change for each merged time phase. So if you are merging time points 2 and 3 together, you could compute the log fold change of the average of 2 and 3 relative to time point 1 (or relative to the average of time points 2 and 3 of a control condition)

   NOTE: the number of time points in this file MUST be **equal** to the number of columns in the `TPtargetsFile`
4. `ppiNetworkFilepath` - The path to the file containing the protein-protein interaction network
5. `sourcesFilepath` - The path to the file containing the source proteins
6. `tfGeneFile` - The path to the file containing the TF-gene interactions
7. `pathFile` - Path to the file to which the pathways should be written to

Optional parametrs

1. `maxNumPaths` - Maximum number of candidate pathways to extract per source and target gene pair. Default value is 100,000.
2. `maxPathLength` - Maximum number of proteins per pathway. Default value is 10.
3. `rnaHitsFilepath` - Path to the file containing RNA hits if you have one
4. `numProteinsPerPhaseToSelect` - Number of proteins to rank per phase
5. `minRankingFoldChange` - Only proteins that show a big change in rank from one time point to another are selected (as we want to select the proteins that are uniquely relevant to a new phase)
6. `numGenesToDisplay` - This is the number of intermediate/TF genes to display per time point in the graph visualization. Default is 15.
7. `graphNodeWidth` - Width in pixels of the protein nodes in the graph in pixels. Default is 60.
8. `graphNodeHeight` - Height in pixels of the protein nodes in the graph in pixels. Default is 40.
9. `l1` - This is the penalty for including too many genes in the network. Increase to reduce number of genes selected. Default is 1

10. `l2` - this is the penalty for including too few targets in the network. Increase to increase number of targets explained. Default is 0.3

11. `L` - This is the size of the tabu list. Default is 200

12. `N` - This is the total number of iterations. Default is 1000

Running the algorithm

The command to run the algorithm is as follows :-

```
java -jar -Xmx8g timepath.jar <config file>
```

Here `<config file>` is the is the configuration file which is explained in the previous section

Compute protein rankings

1. The command to compute the overall protein rankings is

   ```
   python computeOverallProteinRanking.py <path file>
   ```

2. The command to compute *time point* specific protein ranking

   ```
   python computeTopProteinsForTP.py <config file>
   ```

   This will create a file `<path file>.phasegenes.txt` where `<path file>` is as entered in the config file

Visualization

You *must* compute the time point specific protein rankings (using `computeTopProteinsForTP`) before running the graph visualization. After doing that, run the following command

```
python createGraph.py <config file>
```

This will create a file named `<path file>.graph.gml` where `<path file>` is as entered in the config file. You can then load the gml file into a visualization software like Cytoscape to view the resulting network.