Manual and algorithm of SUbNetworks RIch in Significant Elements by Python (sunrise)

1 introduction

sunrise searches subnet significantly enriched in low p-values. It needs a network as a protein protein interaction network and an additive score (from p-values obtained by genome-wide association study or gene expression). The highest score subnet gives information about the pathways or functions influenced by the score.

The Python scripts are based on an approximation of the Steiner tree problem described in the algorithm section.

2 Format of files

The common part of files names is the name of the network (net_name). To differentiate the files, a suffix is added:

- input network: .sif
  $node < sep > interaction < sep > node$
- input nodes: _n.txt
  $node < sep > score$, with header as $shared\ name < sep > score$
- output reduced network: _r.sif
  $node\ group < sep > topos\ or\ toneg < sep > node\ group$
- output nodes: _r-n.txt
  $group\ name < sep > score < sep > node\ names\ in\ the\ group$,
  with header $shared\ name < sep > score < sep > name\_in\_group$
- output edges: _r-e.txt
  $edge\ identifier < sep > weight$ (see algorithm)
- output resulting groups: _o-r.txt, list of node groups
- output resulting nodes: _o-n.txt, list of nodes

$< sep >$ is the separator, generally a tab character (None for separator by default of Python or another of your choice). These formats are compatible with those used in Cytoscape.

3 Scripts

The script to input localization of file and options of input and output is sunrise_param.py

the scripts or reducing and searching solution are:

- sunrise_reduce.py creates a directed subnetwork which contains the optimal subnetwork with the maximal score. So it generates 3 files ready to be read by the following script.

- surrise_search.py searches an approximation of the optimal solution by the below algorithm. It displays or saves this solution according to the options.

The script sunrise_utils.py contains the input and output functions and some network functions.

4 Algorithm

The algorithm adapts an approximation of the Steiner tree problem: minimum spanning tree on the metric closure of the graph.

The network rid of parts not contributing to optimal and is prepared to use algorithm based on edge weights (Dijkstra and Kruskal):

1. positive nodes are grouped and kept connected, the got network is rid of duplicated edges and self loops, negative nodes of degree 1 are recursively removed ;

2. edges are duplicated as two opposite edges, a weight is attributed in function of in node: 0 if positive and opposite of score if negative, so negative scores are transferred to edge ;

3. the reduced network is obtained by keeping the shortest path (link of maximum scores by negative) between positive nodes computed as an undirected networks, in this directed network the direction does not change the sum of weight along the path.

The network is ready to searching the approximation of optimal:

1. the list of the shortest path between positive is sorted by weight (opposite of negative score of nodes between positive) ;

2. the list is browsed in this order, nodes in paths are cumuled in a set, browsing stops when all positive are in the set ;

3. at every item the score is computed, the set of nodes is kept when a maximum is reached and the subnetwork generated by the set is connected.

At the end, the subnetwork obtained by the maximal score is the approximation of optimum.

October 14, 2022 by daniel.rovera@gmail.com - Institut Curie