Manual of sunrise

SUbNetworks RIch in Significant Elements

Contents

1 introduction

sunrise is an application of Cytoscape 3.x which searches subnet significantly enriched in low p-values. An additive score is computed from p-values got by genome-wide association study or gene expression. These values are matched to the nodes of a protein protein interaction network. The the highest score subnet gives information about the touched pathways which gives indications on the involved functions.

2 Tutorial

2.1 Get Data

Data consists of a list of genes or proteins associated with p-values typifying the feature to analyze between two populations. These p-values must be corrected for all possible bias for example the linkage disequilibrium. The input is a table names of gene/protein and p-values (example: bionet1_p-value.csv).

2.2 Prepare Network

After extraction from database and adaptation to the problem studied, the undirected network must be under a format readable by Cytoscape as SIF, XGMML, JSON ...(example: bionet1_net.xgmml)

the names of the genes/proteins must be the same in the data and in the network. Notably, every node must be associate with a p-value. Obviously, some genes or proteins can be absent from the network. In this case, this analysis loses in precision.

2.3 Compute Additive Score

The p-values must be transformed in an additive score. In this example, the Beta Uniform Mixture Model (see 6) is used. A script R based on BioNet transforms the table names of gene/protein and p-values in table names of gene/protein and score. The critical parameter is the Upper Bound of False Discovery Rate.

Pay attention to separator (";" in this script) and the column titles to import in Cytoscape: Shared Name and chosen name for score (example: bionet1_scr.csv).

2.4 Import Network And Data

The network is imported by *File > Import > Network > File*.
It is better to remove duplicated edges and self-loops by *Edit > Remove Duplicated Edges* and *> Remove Self-Loops*.
The reference network must be entirely connected, use *Connected Nodes in Table* 3.5.1 to select the sub-network of interest, generally the largest. If not it can be divided.
The table is imported by *File > Import > Table > File*. Pay attention to options notably the key and to the type of column (must be Double, see 4).
Several scores are possible, they refer to different column names (see 3.1.1).

2.5 Compact Network

The function *Compact Network* 3.2.1 creates a compact network where the optimal sub-network is included (see algorithm 5). Tables of the compact network and the reference network contain information to link each to other (see glossary 4) and also due to the function *Select in Reference Selected in Compact* 3.3.1.

Obviously, the compact network depends not only on reference network but still on positive and negative score nodes. The compact network may give an idea of optimal due to the score of positive node group. The iterate searching 3.2.3 can complete this idea. If all nodes have the same sign, the searching has no interest.

2.6 Launch Several Shots

Launch genetic search by *Search By Genetic Algorithm* 3.2.2 on the compact network. The input fill proposes default parameters which may be changed. The option of several shots allows to launch several random search with random seed. In option one shot, the seed is used for start, same seed gives same result.

Several values of parameters can be tested. The parameters with significant effect are the number of generations and number of mutations, to a lesser extent the size of the population.

2.7 Select an Optimum

During the search, the score of sub-network is displayed in a result window and, at the end, the found optimal sub-network is displayed with the corresponding seed. The seed can be kept by *Keep Parameters in Table*, menu of result window 3.2.2.

By the option *One Research Starting from Seed*, the kept seed allows to create the optimal group, to list it and select it in compact network and reference network (menu of result window 3.2.2). The sub-network can be created by *File > New > Network > From selected nodes, all edges*.
The selection functions, notably *Score of Selected Nodes* 3.4.1, are useful for analyzing the result. The example session bionet1.cys shows the got result pointed out by (1). The table and style in Cytoscape allows to complete the analysis.

2.8 Complement

Coloring nodes according to the scores is possible by Cytoscape styles using a color gradient. Pay attention to position of zero to get a clearly style (example style_sunrise.xml). If several types of scores, several styles based on the score columns.

If there are nodes with null score (to keep links if p-value/score of nodes is missing in network), they are grouped in compact. The found network can be cleaned by deleting null score nodes with degree = 1 unnecessary for the optimum (select and delete in Cytoscape).

2.9 More Complicated Example

The session is named largeNet.cys. The network is extract from BioGRID (thebiogrid.org). Scores are computed from GWAS p-value from biological measures by

$$score = log(threshold) - log(p - value)$$

with threshold of 5%, 1% and 0.1%.

Only genes with p-value are kept as nodes in the network. However, if an edges has only one extremity with p-value, it is kept to do not loose links, the score of this extremity will be set to zero. So, there a a lot of null score nodes.

Results are identified as

- Network marked by _Genetic is got by 3.2.2
- Network marked by _Genetic&iIeration got by 3.2.2 is completed by 3.2.3
- Network marked by _clean is got by deleting unnecessary null score node (see above), duplicated edges and self-loop present in largeNet are similarly deleted

3 Function of Application

3.1 Score Id and Compact Table

3.1.1 Input/Change Name of Score Column

To select the work score name or to change it if several score types, corresponding to a Double column title. The current score name is displayed. This parameter is kept during the session, included in name of compacted network and in its table (see glossary 4).

3.1.2 Create Void Compact Network Table

Create the columns in a compact network according to 4. They are void and must be filled by copy or by hands. Warning to the coherence of score id and reference network.

3.1.3 Copy Compact Network Table

Copy network table values from a compact network to another. Copy only the parameters of genetic algorithm present in the current compact network. Ensure that all titles of

column are created in the right target network. The simplest method is by $File > New > Network$ followed by the wanted option.

3.2 Search Max Score Subnetwork

3.2.1 Compact Network

Compact the reference network in a new network whose name is made of reference network name and score column name:

$Name\ of\ Network\_C\_Score\ Identifier$.

This operation cannot be redone without renaming the previous network.
The groups of positive score nodes are named:

$p\_Name\ of\ the\ Highest\ Score\ Node\ from\ Group$.

Names of negative score nodes which are not all kept are not changed even grouped with null score nodes (see algorithm 5). Nodes are places at the mean coordinates from reference or at one point if reference network has not view.

The table of the compact network contains information to go back to the reference network and the reference network contains the fate of its every node, which insures the link between these two networks (see glossary 4).

3.2.2 Search By Genetic Algorithm

The input form is filled by the parameters by default or kept by user during a previous step (see glossary 4). They can be changed before launching the genetic search. Choice between two options:

- **Several Researches**, their Number: enter the number of random shots, the displayed seed is not used, seeds are randomized;
- **One Research Starting from Seed**: the seed (long integer), generally filled in a previous step.

At the end result are displayed in a result window. Its menu contains two specific commands according to the option.

- **Several Researches**: the best result is displayed, even if the process is stopped by cancel; the menu is:
  - **Keep Parameters in Table**, update the compact network table with the parameters of the best result and this result ;
  - **Node List by One Research**, display the list of nodes with the best result as one shot without keeping parameters.
- **One Research Starting from Seed**: The result and the node list is displayed; the menu is:
  - **Select Nodes in Compact**, idem 3.4.3;
  - **Select Nodes in Reference**, idem 3.3.1.

3.2.3 Iterate Selection to Positive Node

Starting from a connected selection, iterate searching positive score nodes by the highest score path while the score of selection is increasing. Pay attention that, generally, union of sub-optimal subnetworks is not sub-optimal, so, by iteration, the got subnet can be far from the optimal. This greedy algorithm can give an idea of it or refine the solution of genetic algorithm.

3.2.4 Extend Selection to Positive Node

Extend a connected selection to positive score nodes by the highest score path. All possible paths with the new score are proposed. This function is useful to include a node which seems necessary in optimal. Same mechanism than above.

3.3 Only By Selection in Compact

3.3.1 Select in Reference Selected in Compact

This function uses the column fate (glossary 4) to select from compact net the corresponding nodes in reference. It work only on a compact network with coherent network table (if not error message).

3.3.2 Contents of Selected Compact Nodes

List the nodes from the reference network contained in the selected compact nodes. Several nodes can be selected. Only name of positive score nodes are changed in compact network as explained in 3.2.1. Same remark as above.

3.4 By Selection in Any Net

3.4.1 Score of Selected Nodes

Display the scores of every selected node and their sum. The score identifier is displayed in the header of the column. The type of score must be chosen if it is not before and can be changed by 3.1.1.

3.4.2 List Selected Nodes

Copy names of selected nodes in the text box. So they can be copied in a spreadsheet or used to select these nodes in another network.

3.4.3 Select Nodes by a Name List

When viewing a network, open a text editor where list of node names can be pasted or typed, one name by line. Selection of nodes of this network by clicking select. Insure that is the good network in title of editor.

3.4.4 Select Edges by a Name List

Idem Select Nodes by a Name List.

3.5 Complementary Functions

3.5.1 Connected Nodes in Table

Every group of connected nodes is identified by $CN9$ in the node column "connected", $\_NC$ for isolated nodes. $Tools > NetworkAnalyser > Subnetwork\,Creation > Extract\,Connected\,Co$ can be also used. Useful for keeping only the main part of the network.

3.5.2 Nodes Common to Networks

Display the list of nodes, identified by name, common to several networks (intersection of networks).

3.5.3 List Nodes & Within Nodes

This function lists the nodes of the global network and nodes in nested networks (modules). The results in the text box can be simply copied in a spreadsheet through the clipboard.

3.5.4 Transfer Network Coordinates

Transfer node coordinates from a network to another network using names to match nodes.

4 Glossary

These fields are used by th application and generally not directly. All words and operations depend on the chosen score. Its identifier is named **Score_Id**, chosen or changed by 3.1.1 .

Fate_C_Score_Id (String column in reference net table) what became of the nodes in compact (see algorithm 5)

- the positive nodes are grouped in a node whose name is $p\_Max\ Score\ Node\ Name$, name of the highest node score from the group ;
- the negative nodes kept because being in the highest score paths ;
- the null score nodes grouped with positive groups or negative nodes according to their nearest neighbors ;
- the negative nodes not kept, their attribute is $\_NotInCompact$;

Reference_Net (String column in compact net table) origin of compact network

Score_Id (String column in compact net table) score column used for compact net

Population (Integer column in compact net table) Size Of Population *

Generation_Nb (Integer column in compact net table) Number of Generations *

Mutation_Nb (Integer column in compact net table) Number of Mutations *

Seed (Long column in compact net table) seed for one shot search *

Best_Score (Floating Point/Double column in compact net table) the score got with the kept parameters for memory.

* Fill by default in the input form in function of the number of nodes.

Pay attention that column are shared inside a collection and the value too according to the shared name.

5 Algorithm

This algorithm searches the highest score subnet. A score is associated to every node, edge weights are not used. Scores must be additive at large, a multiplication by a positive number should not change the result.

The network must be fully connected. Groups of nodes are simply described as an array of bits, what motivates to use a genetic algorithm.

The here algorithm only works on high connected network, what is the case of biological networks. It is better that the network is undirected without loop and duplicated edges. The two next steps correct these defects.

5.1 Group Positive Nodes

All connected positive nodes are gathered in groups. The score of every group is computed by adding. The groups are reconnected by the edges linking nodes inside groups, parallel edges ant auto-loops being eliminated. Indeed, if a positive node is in optimal subnet, all nodes connected to it can only increase the total score and, so, belong to the optimal subnet.

5.2 Keep the Shortest Paths

Connecting the positive groups needs to pass through the negative nodes. The best paths must be chosen to have the highest score. Scores being associated to nodes, edges are doubled and scores of ends are transferred to edges. After reversing edge signs, the problem is brought back to finding the shortest paths between nodes in a graph. Here, Dijkstra's algorithm between positive nodes is used.

The nodes in these paths belong to the optimal subnet. Indeed if a longer path containing lower score node was supposed to belong to it, a higher score path can be always found among the shortest paths as defined above. So, a compacted network containing the optimal subnet is built.

5.3 Start the Genetic Algorithm

Nodes are sorted by increasing score and the adjacency of the compacted network is as an array of bit set. A start node and a size are randomly drawn. A sub-tree is built from them by depth first search and gives a member of the population.

The positive nodes are connected by negative nodes which may be redundant. So, every member is cleaned by removing the redundant negative nodes. A list of nodes ordered by score is formed by negative nodes not linked to one degree positive node. Every node is successively deleted in the member and only nodes not changing the number of positive nodes are kept. This negative clean operation is used also in the search by genetic algorithm.

So the population of random members, arrayed in a container of bit set sorted by score, is the starting point of the next step.

5.4 Search by Genetic Algorithm

Cross-overs followed by mutations are iterated. Cross-over is an exchange of parts where one position and two subnets are randomly drawn. The several random positions are flipped according to a random number of mutations under a threshold. After these two operations, the two got groups of nodes are generally not connected.

To overcome this problem, a first step keeps only the connected group of the highest score by depth first search. A second step removes the redundant negative nodes as during the initialization (negative clean operation). Generation after generation, the members of population are always subnets. Only the highest score subnet are kept, the size of population is fixed. Whoever has the highest score is displayed when the number of generations is reached.

After several random shots starting from different populations, the optimal subnet is approached by playing with the different parameters, notably the number of generations and the maximum number of mutations.

5.5 Case of zero scores

Some scores can be set to zero (as nodes from network without data kept to do not loose links). During compacting reference network, these nodes are grouped with connected positive nodes and the rest of null score nodes are grouped with a connected negative node. There is no null score node int any compact network. So, null score nodes are neutral on the result.

6 Beta Uniform Mixture Model

The Beta Uniform Mixture Model applied to p-values is based on this probability density function:

$$f(x|a,\lambda) = \lambda + (1-\lambda)ax^{a-1} \qquad 0 < \lambda \; and \; a < 1$$

where p-value: x; shape parameters of Beta function: a, b=1; noise factor λ.

The parameters a and λ are computed by fitBumModel() from library R BioNet by maximum of log(likelihood).

See the graph of the first example from BioNet Tutorial in 1 where
Mixture parameter (lambda): 0.536
shape parameter (a): 0.276

The figure 2 shows an interpretation of this representation to fix a threshold of a false discovery rate (fdr). The false discovery rate in function of the p-value τ is the ratio of areas:

$$\frac{\int_C}{\int_A + \int_C}$$

it must be less than the threshold fdr (4.387218e-05 in the BioNet example). After computing, the got formula is:

$$\tau(fdr) = (\frac{\pi - \lambda fdr}{fdr(1-\lambda)})^{\frac{1}{a-1}} \; with \; \pi = \lambda + (1-\lambda)a$$

The score is an adjusted log likelihood ratio which depends on the accepted false positive rate and the p-values x as:

$$S_{fdr}(x) = (a-1)(ln(x) - ln(\tau(fdr))$$

References:

Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values
Stan Pounds and Stephan W. Morris

Identifying functional modules in protein protein interaction networks: an integrated exact approach
Marcus T. Dittrich, Gunnar W. Klau, Andreas Rosenwald, Thomas Dandekar and Tobias Mller

Source R and comments: BioNet/src/R/Statistic

April 25, 2019 by daniel.rovera@gmail.com - Institut Curie

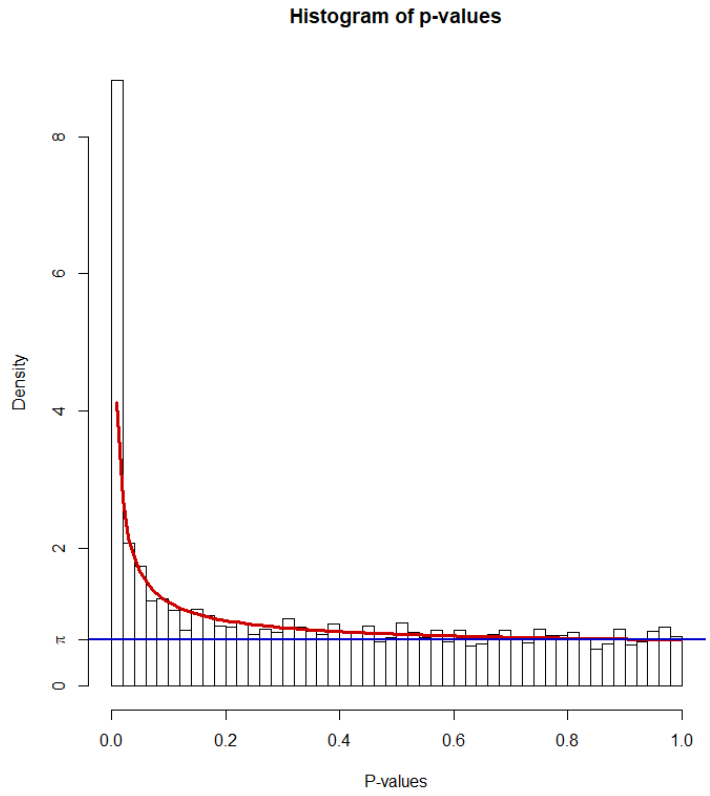

Figure 1: Density of p-value, the noise limit: $\pi = \lambda + (1 - \lambda)a$

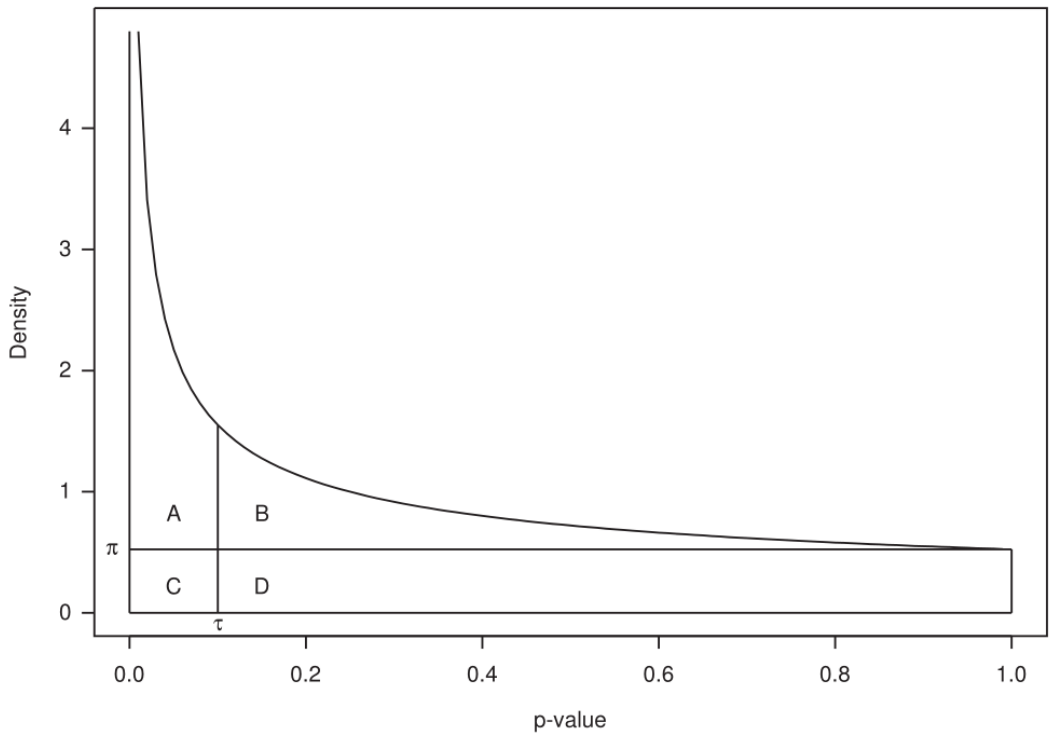

Figure 2: τ: p-value threshold. A area: true positives. B area: false negatives. C area: false positives. D area: true negatives.