

Oct 10, 2018

Working

Validating that a specific subnetwork has significantly more interactions than expected by randomness

PLOS One

Pascal Belleau¹, Astrid Deschênes¹, Martial Boutchueng-Djidjou², Robert L. Faure³¹Cold Spring Harbor Laboratory, ²H. Lee Moffit Cancer Center and Research Institut, ³Université Laval[dx.doi.org/10.17504/protocols.io.sdqea5w](https://doi.org/10.17504/protocols.io.sdqea5w)

Pascal Belleau

Cold Spring Harbor Laboratory



ABSTRACT

Network science is a cross-disciplinary domain that focuses on the representation, analysis and modeling of complex systems as networks or graphs. In computational biology, Cytoscape is a very popular open source software platform for network modeling and visualization. As an example, Cytoscape can be used to identify disease modules which are defined as connected subnetworks showing mechanistic evidence for a specific phenotype.

In the context of a new subnetwork identification, it is of interest to have access to measures of connectivity that help to assess the probability that this subnetwork could arise by chance.

The [R subnetsim package](#) uses bootstrapping to test the hypothesis that a specific subnetwork is more connected than randomly generated subnetworks with the same number of nodes.

Three measures of connectivity are validated by the R **subnetsim** package:

- the number of nodes in the subnetwork including all the first-degree neighbor nodes
- the number of links in the subnetwork including all the links of the first-degree neighbor nodes
- the number of links in the subnetwork

The [R subnetsim package](#) also offers the possibility to easily graph the results of the simulations.

This protocol provides the detailed method of a simulation analysis done on a specific subnetwork. To do so, the global subnetwork, from which the subnetwork is extracted, is needed in SIF format file.

TAGS

network-analysis

simulation

Show tags

EXTERNAL LINK

<https://doi.org/10.1371/journal.pone.0205180>

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Boutchueng-Djidjou M, Belleau P, Bilodeau N, Fortier S, Bourassa S, Droit A, Elowe S, Faure RL (2018) A type 2 diabetes disease module with a high collective influence for Cdk2 and PTPLAD1 is localized in endosomes. PLoS ONE 13(10): e0205180. doi: [10.1371/journal.pone.0205180](https://doi.org/10.1371/journal.pone.0205180)

 subnetsim_v0.2.1.pdf

PROTOCOL STATUS

Working

GUIDELINES

SIF Format Input

The Simple interaction file (SIF) format is one of multiple formats that Cytoscape can read. For the moment, it is the only format used by the R subnetsim package. SIF file can manually be generated as long as the basic rules as follows. Cytoscape provides

information about [SIF Format](#).

R software environment

[R software environment](#) must be installed on the computer that will be used to run the simulation analysis.

Install R subnetsim package version 0.2.1

1

The analysis must be done in a R. It can be done through a console using command lines or through an integrated development environment such as [RStudio](#).

The [R subnetsim package](#) version 0.2.1 can be downloaded from the website specified in the software section.

Alternatively, the R devtools package [R devtools package](#) can be used to download and install it (see command).

COMMAND

```
## Only run when devtools package is not already installed
install.packages("devtools")
devtools::install_github("belleau/subnetsim", ref = "0.2.1")

Install subnetsim package version 0.2.1 through devtools package.
```

SOFTWARE

subnetsim 0.2.1

Mac, Windows, Linux

[source](#) by Pascal Belleau, Astrid Deschênes, Martial Boutchueng-Djidjou and Robert L.

Load R subnetsim package

2

As with any R package, the [R subnetsim package](#) should be loaded with the following command.

COMMAND

```
## Loading subnetsim
library(subnetsim)

Loading the subnetsim package.
```

Load the global network and the tested subnetwork

3

The global network must be in a SIF format file. The `network()` function of the subnetsim package must be used to load the network. The tested subnetwork, when available in SIF format file, can be passed to the `network()` function to be loaded as a subsection of the global network object. The global network must be in a SIF format file. The tested subnetwork, when available in SIF format file, can be passed to the `network()` function to be loaded as a subsection of the global network.

NOTE

For the moment, only the simple interaction format (SIF format) files are accepted as input for the global network and the subnetwork.

COMMAND

```
network <- network(netFileName = networkFileInFormatSIF, subNetFileName = NULL)
```

Load only the global network.

COMMAND

```
network_with_subnetwork <- network(netFileName = networkFileInFormatSIF, subNetFileName = subnetworkFileInFormatSIF)
```

Load the global network and the tested subnetwork.

Generate the simulated subnetworks

4

The number of simulated subnetworks is specified by the user. The simulated subnetworks are generated through the *simuleSubNet()* function. Bootstrapping is used to generate subnetworks that have the exact same number of nodes as the tested subnetwork. When the subnetwork information is present in the network object, some of the parameters have to be left to *NULL*. When the network object only contains the information about the global network, all parameters must be filled.

NOTE

Beware that the function can take some time to process when the number of iterations is large.

COMMAND

```
## Run simulations when subnetwork is contained in the network object
simResults <- subnetwork(network = network_with_subnetwork, nbIter = 1000, seedV = 33333)
```

Run simulated subnetworks when the network object contains the information about the tested subnetwork.

COMMAND

```
## Run simulations when subnetwork is not part of the network object
simResults <- subnetwork(network = network, nbIter = 1000, nbNodes = 250, nbNodesOneLink=434, nbLinkOneLink=444, seedV = 33333)
```

Run simulated subnetworks when the network object does not contain the information about the tested subnetwork. All fields must be filled to pass the information about the subnetwork.

Extract the information from the simulated subnetworks

5

The *subnetwork()* function of the subnetsim package returns an object of class *subnetwork* than contains the information relative to the simulation.

The proportion of the simulated subnetworks that have higher measures of connectivity than the tested subnetwork are also accessible from the returned object. Those proportions are used to assess the probability that the tested subnetwork could arise by chance.

An example of the output:

```

> print(simulatedSubnetwork)
simulation subnetwork summary
Iterations:
[1] 500
Seed:
[1] 111111
--- Tested subnetwork ---
Number of nodes:
[1] 103
Number of nodes including first-degree nodes:
[1] 275
Number of links:
[1] 330
Number of links including all links from first-degree nodes:
[1] 830
--- Simulated subnetworks ---
Number of nodes in the simulated subnetworks including first-degree nodes:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  237.0  254.0  259.0  258.8  264.0  281.0
Number of links in the simulated subnetworks including all links from first-degree nodes:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  522.0  602.8  634.5  633.0  660.0  758.0
Number of links in the simulated subnetworks:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   73.0  108.0  123.0  124.2  139.0  201.0
Proportion of simulate subnetworks with more or equal number of link:
[1] 0
Proportion of simulate subnetworks with more or equal number of link
including all links from first-degree nodes:
[1] 0
Proportion of simulate subnetworks with more or equal number of nodes
including first-degree nodes:
[1] 0.004

```

COMMAND

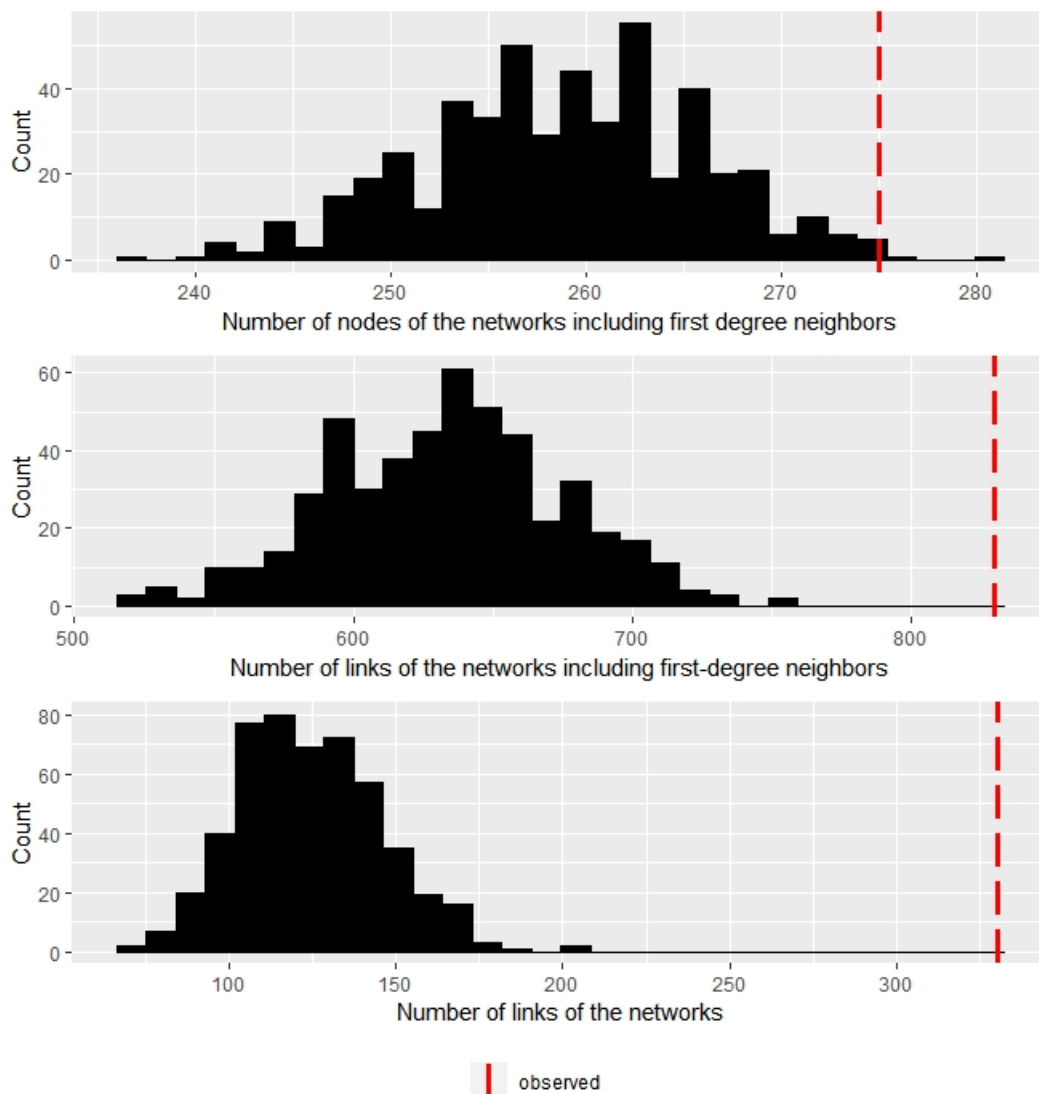
```
## Print a summary of the simulations
print(simResults)
```

See the summary information of the simulations.

Plot the results

6

A graph showing the position of the observed values compared to the simulated results can be easily generated using the *plot()* function of the subnetsim package. An example of the generated graph is shown below.



COMMAND

```
## Generate a graph of the simulation results
plot(simResults)

Plot the results from the simulations.
```



This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited