NetBox 1.0 Software: README

Contents

1 Introduction

NetBox is a Java-based software tool for performing network analysis on human interaction networks. It is pre-loaded with a Human Interaction Network (HIN) derived from four literature curated data sources, including the Human Protein Reference Database (HPRD) [1], Reactome [2,3], NCI-Nature Pathway Interaction (PID) Database [4], and the MSKCC Cancer Cell Map.

Currently, NetBox provides the analyzeNet.py method that is fully described in the manuscript: **Integrated Network Analysis Identifies Candidate Driver Genes in Glioblastoma** (Currently in review). This provides a simple command line interface for connecting genes into a network, identifying statistically significant "linker" genes, partitioning the network into modules, and executing two random background models. Results are then made available to the end user as an HTML web page and a series of network and attribute files, which can be loaded into Cytoscape [5] for visualization and further analysis.

2 System Requirements

To install NetBox, you must have:

- Java 1.5 or later.

- Python 2.5 or later.

3 Installation

3.1 On Mac OS X / Linux

To install NetBox on Mac OS X or Linux, you probably already have Python and Java installed, and installation should be very simple. Just follow these steps:

- Unzip/untar netbox in a directory of your choosing, e.g. /Users/xxx/netbox.
- Add NETBOX_HOME as an environment variable, and set it to your netbox directory.
- (Optional) Add NETBOX_HOME/bin to your global path.

For example, on Mac OS X, I have added the following to my .bash_profile:

```
export NETBOX_HOME="/Users/cerami/dev/sander/netbox/"
export PATH=$PATH:$NETBOX_HOME/bin
```

3.2 On Windows

To install NetBox on Windows, you may need to install Python first. Download it from: http://www.python.org/download/releases, and follow the installation instructions.

After that, you will need to:

- Unzip/untar netbox in a directory of your choosing, e.g. c:\netbox.
- Add NETBOX_HOME as an environment variable, and set it to your netbox directory. In Windows, this is usually done via the Control Panel. Complete instructions are available at: http://www.cs.usask.ca/~wew036/latex/env.html.
- (Optional) Add NETBOX_HOME/bin to your global path. In Windows, this is also done via the Control Panel. See the URL above, if you need detailed instruction.

4 Running netAnalyze.py

4.1 Configuration Options

To run netAnalyze.py, you must first create a configuration file. A sample configuration file is available in gbm_data/netbox1.props, and is shown below:

```
gene_file=ALTERED_GENE_LIST_HYPERMUTATORS_EXCLUDED.txt
title=title=GBM, Hypermutators Excluded; altered in 2% or more of all cases
shortest_path_threshold=2
p_value_threshold=0.05
```

```
num_global_trials=0
num_local_trials=0
```

Each of the configuration parameters is described in the table below.

Parameter	**Description**
gene_file	Name of the file containing your genes of interest. This file contains one gene per line, and genes can be specified with official HUGO gene symbols or Entrez Gene IDs. You can also comment out specific genes by pre-prepending them with a # mark.
title	[Optional] Title used in the final HTML generated report.
shortest_path_threshold	[Optional, default = 1] Sets the shortest path threshold used to connect genes in your input list. Can be set to 1 or 2. When set to 1, only genes which are directly connected via a shortest path of 1 will be included in the final network. When set to 2, NetBox will try to connect additional genes by identifying statistically significant "linker" genes which connect genes within your input list.
p_value_threshold	[Optional, default = 0.05] Sets the p-value threshold used to prune "linker" genes from the network. To identify statistically significant linker genes, NetBox uses the global degree of each linker gene within the Human Interaction Network (HIN) and the hypergeometric distribution to assess the probability that the linker gene would connect to the observed number of altered genes by chance alone. After FDR correction via Benjamini Hochberg [6], linker genes above the specified p_value_threshold are pruned from the network. If you are not sure which p_value_threshold to use for your specific network, we recommend you try the interactive option described below.

num_global_trials	[Optional, default = 0] Sets the number of random trials to execute in the global null model. The global null model assesses the level of global connectivity seen in your observed network by comparing the size (number of nodes and edges) of the largest component in the network to the largest components generated by randomly selected sets of genes known to be present in the Human Interaction Network (HIN). For example, if your input list contains 500 genes, and 300 of these have interaction information in the HIN, at each random trial, NetBox randomly selects 300 genes from the HIN and connects them via your specified shortest path threshold and p-value cut-off parameters. NetBox then determines the size of the largest component within this randomly generated network, and determines an empirical p-value by keeping track of the number of times this largest component equals or exceeds your observed largest component.
num_local_trials	[Optional, default = 0] Sets the number of random trials to execute in the local null model. The local null model assesses the statistical significance of the observed network modularity in relation to a null model of random networks of the same size and same degree distribution. In each random trial, NetBox uses a rewiring algorithm, such that the network remains the same size, and all genes maintain the same degree, but the choice of interaction partners is random [7]. NetBox then calculates the network modularity, and calculates the average and standard deviation for the entire set of random networks. The observed modularity score is then converted into a z-score, or scaled modularity score to measure the deviation of the observed network from its random null model [8].

4.2 Running netAnalyze.py

To run netAnalyze on Mac OS X or Linux, type `netAnalyze.py` and specify your configuration file. For example:

```
$ cd gbm_data
$ netAnalyze.py netbox1.props
```

On Windows, you would type something like:

```
C:\>cd gbm_data
C:\>c:\Python31\python.exe c:\netbox\bin\netAnalyze.py netbox1.props
```

Either way, you should then see output like this:

```
Welcome to NetBox.  Initializing Database.  Please wait a few moments...
```

```
Total number of genes read in:  517.
...
Input gene list consists of 517 genes.
Of these genes, 274 are in the reference network.
At shortest path threshold of:  2 and p-value cut-off of:  5E-2, I can connect 66
genes with 6 linker genes.
...
Final HTML Report is available at:  /Users/ceramie/dev/sander/netbox/gbm_data/report.html
```

4.3 Running netAnalyze.py in Interactive Mode

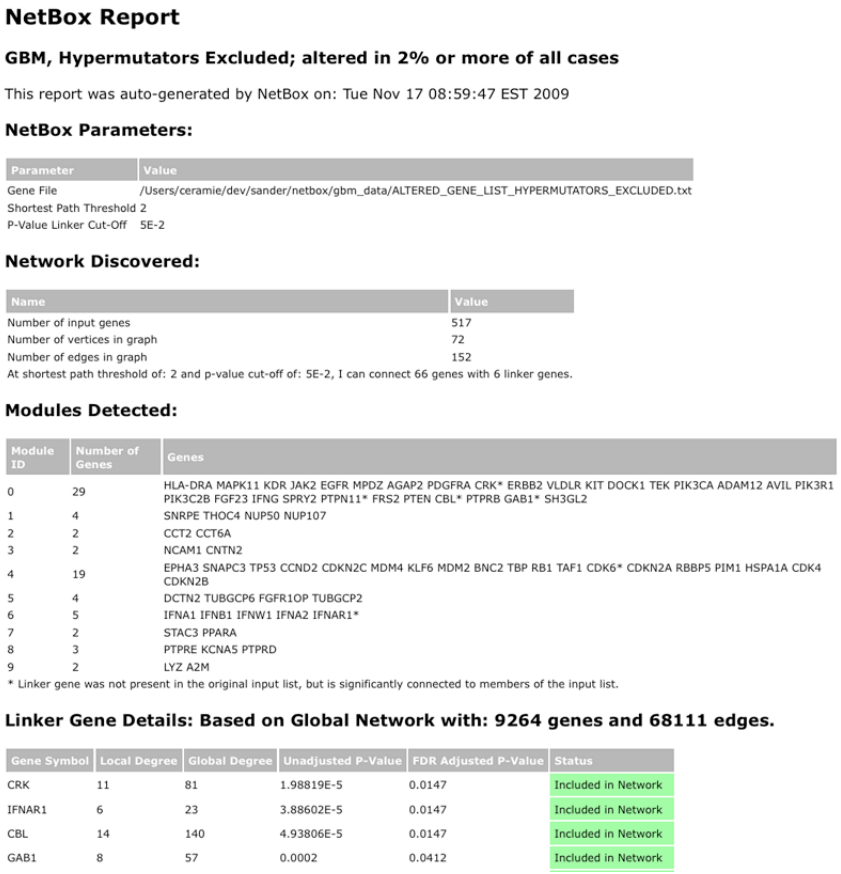
To run netAnalyze in interactive mode, specify the -i parameter. This enables you to interactively enter different p-value cut off values, and see how this affects the network discovered. For example:

```
$ netAnalyze.py netbox1.props -i
Input gene list consists of 517 genes.
Of these genes, 274 are in the reference network.
At shortest path threshold of:  2 and p-value cut-off of:  5E-2, I can connect 66
genes with 6 linker genes.
Enter [1] to accept network; [2] to enter new p-value threshold: 2
Enter new p-value threshold: 0.07
At shortest path threshold of:  2 and p-value cut-off of:  7E-2, I can connect 72
genes with 15 linker genes.
```

4.4 Output Generated by netAnalyze.py

By default, netAnalyze.py will generate four files for you:

- report.html: An HTML Report which summarizes the network analysis. This report summarizes your configuration parameters, the network discovered, the modules identified, details on all linker genes, and results of any null background tests. An example is shown in Figure 1.
- network.sif: The network discovered from your input gene list. The network is formatted in the Cytoscape Simple Interaction Format (SIF). An example is shown in Figure 2.
- modules.txt: Cytoscape attribute file containing gene to network module assignments, suitable for loading into Cytoscape. An example is shown in Figure 3.
- node_type.txt: Cytoscape attribute file containing node ALTERED / LINKER attributes, suitable for loading into Cytoscape. Using this file, you can visually identify your genes, identified as "ALTERED" v. statistically significant linker genes, identified as "LINKER".



NetBox Report

GBM, Hypermutators Excluded; altered in 2% or more of all cases

This report was auto-generated by NetBox on: Tue Nov 17 08:59:47 EST 2009

NetBox Parameters:

Parameter	Value
Gene File	/Users/ceramie/dev/sander/netbox/gbm_data/ALTERED_GENE_LIST_HYPERMUTATORS_EXCLUDED.txt
Shortest Path Threshold	2
P-Value Linker Cut-Off	5E-2

Network Discovered:

Name	Value
Number of input genes	517
Number of vertices in graph	72
Number of edges in graph	152

At shortest path threshold of: 2 and p-value cut-off of: 5E-2, I can connect 66 genes with 6 linker genes.

Modules Detected:

Module ID	Number of Genes	Genes
0	29	HLA-DRA MAPK11 KDR JAK2 EGFR MPDZ AGAP2 PDGFRA CRK* ERBB2 VLDLR KIT DOCK1 TEK PIK3CA ADAM12 AVIL PIK3R1 PIK3C2B FGF23 IFNG SPRY2 PTPN11* FRS2 PTEN CBL* PTPRB GAB1* SH3GL2
1	4	SNRPE THOC4 NUP50 NUP107
2	2	CCT2 CCT6A
3	2	NCAM1 CNTN2
4	19	EPHA3 SNAPC3 TP53 CCND2 CDKN2C MDM4 KLF6 MDM2 BNC2 TBP RB1 TAF1 CDK6* CDKN2A RBBP5 PIM1 HSPA1A CDK4 CDKN2B
5	4	DCTN2 TUBGCP6 FGFR1OP TUBGCP2
6	5	IFNA1 IFNB1 IFNW1 IFNA2 IFNAR1*
7	2	STAC3 PPARA
8	3	PTPRE KCNA5 PTPRD
9	2	LYZ A2M

* Linker gene was not present in the original input list, but is significantly connected to members of the input list.

Linker Gene Details: Based on Global Network with: 9264 genes and 68111 edges.

Gene Symbol	Local Degree	Global Degree	Unadjusted P-Value	FDR Adjusted P-Value	Status
CRK	11	81	1.98819E-5	0.0147	Included in Network
IFNAR1	6	23	3.88602E-5	0.0147	Included in Network
CBL	14	140	4.93806E-5	0.0147	Included in Network
GAB1	8	57	0.0002	0.0412	Included in Network

Figure 1. Example HTML report generated by NetBox netAnalyze.py.

4.5 Visualizing Network Modules and Linker Genes in Cytoscape

To visualize the network and network modules identified by netAnalyze.py, we recommend that you use Cytoscape, freely available for download at: `http://cytoscape.org/`.

The Cytoscape web site contains a complete User Guide. For now, we provide minimal steps on how to visualize the networks generated by NetBox.

- Start Cytoscape
- Select File ⇒ Import ⇒ Network, and choose the auto-generated network.sif file.
- Layout the network to your choosing, e.g. Layout ⇒ yFiles ⇒ Organic.

You can then choose to visualize the network modules and linker genes. To do so, you must first import the two auto-generated attribute files:

- File ⇒ Import ⇒ Node Attributes, and select the auto-generated modules.txt file.
- File ⇒ Import ⇒ Node Attributes, and select the auto-generated node_type.txt file.

```
BUB1B INTERACTS CENPP
HSPA2 INTERACTS MEOX2
HDAC2 INTERACTS SNW1
CPS1 INTERACTS SUCLA2
BIRC5 INTERACTS PPP1CC
HDAC3 INTERACTS NCOR2
TP53 INTERACTS TFDP1
GNA12 INTERACTS HSP90AA1
POLR2J INTERACTS SFRS6
HSP90AA1 INTERACTS RHOA
```

Figure 2. Example SIF network generated by NetBox netAnalyze.py.

```
MODULE
HLA-DRA = 0
SNRPE = 1
CCT2 = 2
CCT6A = 2
NCAM1 = 3
CNTN2 = 3
EPHA3 = 4
SNAPC3 = 4
TP53 = 4
```

Figure 3. Example module attribute file generated by NetBox netAnalyze.py.

You can then use the Cytoscape VizMapper to map the node attributes to any visual style you like. For example, you can map the network modules to discrete colors:

- In the Left Cytoscape Panel, Click the VizMapper Tab.
- Select Node Color.
- From the Pull-down Menu, select "MODULE".
- From the Mapping Type Menu, select "Discrete Mapping."
- Right Click on Node Color, and select Generate Discrete Values ⇒ Randomize.

The network modules will now be color-coded (see Figure 4).

You can also highlight linker nodes by marking them with a distinct shape. For example:

- In the Left Cytoscape Panel, Click the VizMapper Tab.
- Select Node Shape.
- From the Pull-down Menu, select "NODE_TYPE".
- From the Mapping Type Menu, select "Discrete Mapping."
- Next to "LINKER" select "DIAMOND".

Linker genes will now be visualized as diamonds (see Figure 5).

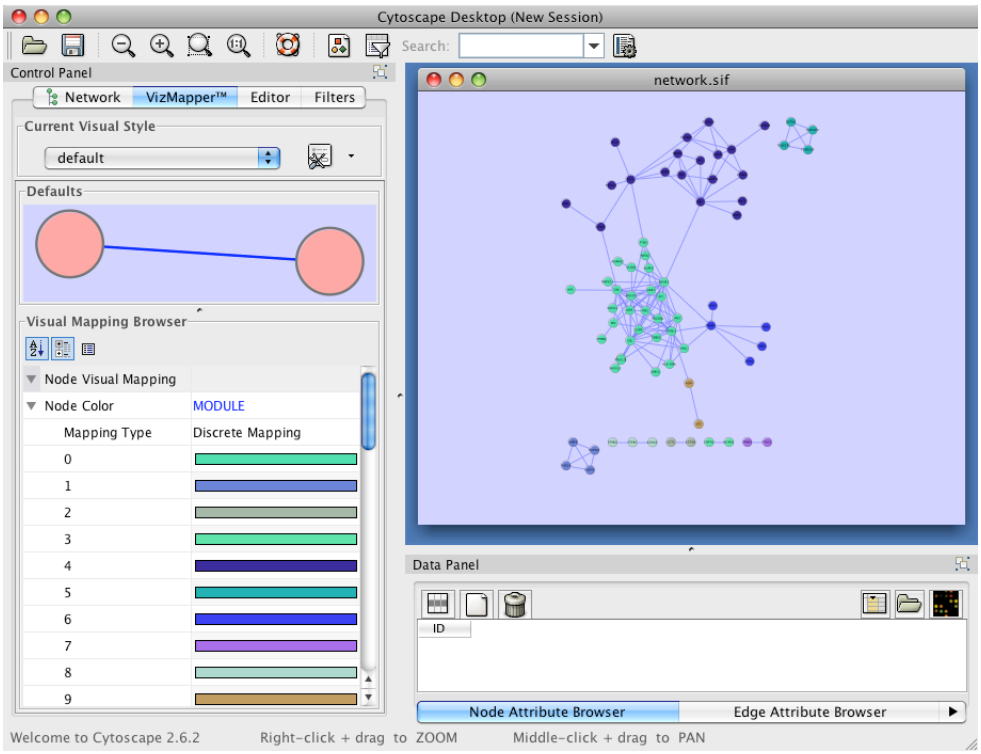


Figure 4. Visualizing Network Modules in Cytoscape.

5 Software Implementation Notes

The NetBox software is written in the Java and Python programming languages. It uses Hibernate and the Java HyperSQL embedded database to store the Human Interaction Network (HIN) and Entrez Gene information, and the Java JUNG library for all graph operations.

6 Software Updates

To check for updates to NetBox, please go to: `http://cbio.mskcc.org/netbox`.

7 Questions?

If you have questions regarding NetBox, please email Ethan Cerami: cerami AT cbio.mskcc.org.

References

1. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, et al. (2009) Human protein reference database–2009 update. Nucleic Acids Res 37: D767-72.

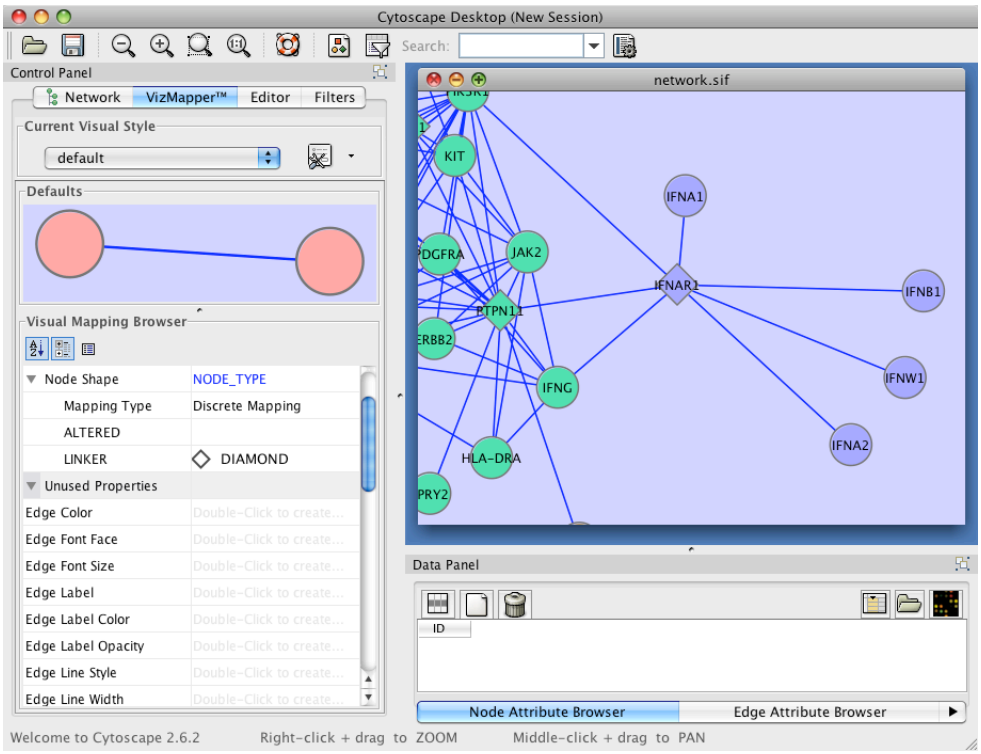


Figure 5. Visualizing Linker Genes in Cytoscape.

2. Joshi-Tope G, Gillespie M, Vastrik I, D'Eustachio P, Schmidt E, et al. (2005) Reactome: a knowledgebase of biological pathways. Nucleic Acids Res 33: D428-32.

3. Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, et al. (2009) Reactome knowledgebase of human biological pathways and processes. Nucleic Acids Res 37: D619-22.

4. Schaefer CF, Anthony K, Krupa S, Buchoff J, Day M, et al. (2009) PID: the Pathway Interaction Database. Nucleic Acids Res 37: D674-9.

5. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, et al. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13: 2498–2504.

6. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B (Methodological) 57: 289–300.

7. Maslov S, Sneppen K (2002) Specificity and stability in topology of protein networks. Science 296: 910–913.

8. Wang Z, Zhang J (2007) In search of the biological significance of modular structures in protein networks. PLoS Comput Biol 3: e107.