Network Comparison

Gesine Reinert

Department of Statistics
University of Oxford
reinert@stats.ox.ac.uk

Theory of Big Data
University College London, 8 January 2015

Outline

1. What are networks?
2. Example: Protein-protein interaction networks
3. How to compare networks: alignment?
4. Network comparison based on subgraph counts
   - The main ideas
   - Netdis
   - Examples
5. Subsampling
6. Summary

Joint work with Waqar Ali, Robert Gaunt and Charlotte M. Deane

What are networks?

Networks are just graphs; they are described by a set of nodes (vertices) and a set of links (edges) between nodes.

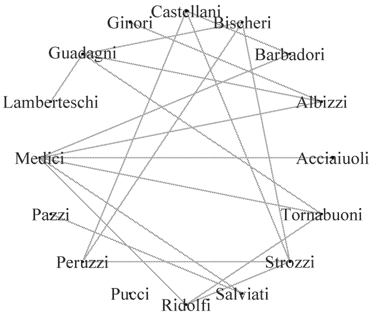
They are often described using summary statistics such as the average degree. The degree of a node is the number of other nodes to which it is connected by an edge.

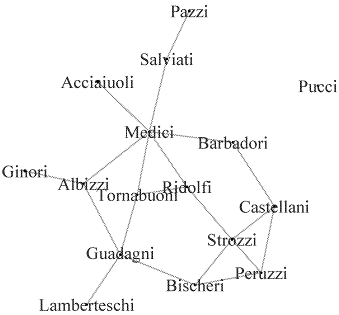
The density of a network is the number of edges divided by the possible number of edges.

Here we mostly look at examples where edges are single, undirected and unweighted, and there are no self-loops.

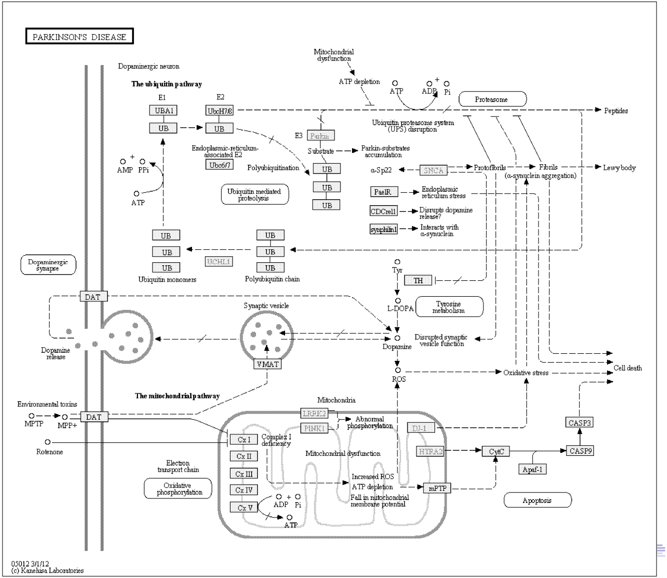
Here are some examples of networks (graphs).

Marriage relations between Florentine families





KEGG pathway for Parkinson’s disease



05012 3/1/12
(c) Kanehisa Laboratories

Yeast protein-protein interaction network: a hairball picture

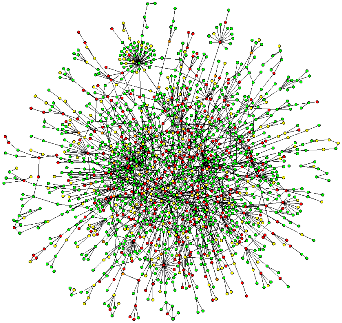


Figure : From *Yeong et al. (2001)*. Proteins are nodes, and two nodes are connected if a physical interactions between the protein has been observed. The colour of a node indicates the phenotypic effect of removing the corresponding protein (red = lethal, green = non-lethal, orange = slow growth, yellow = unknown).

Networks arise in a multitude of contexts, such as

- metabolic networks;
- protein-protein interaction networks;
- spread of epidemics;
- neural network of *C. elegans*;
- social networks;
- collaboration networks (Erdös numbers ...);
- Membership of management boards;
- World Wide Web;
- traffic networks;
- power grids.

The number of nodes varies from less than 100 (social networks) to 4.34 billion pages (the Internet, Monday, 05 January, 2015) and beyond.

A big question

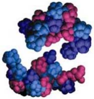
How do we compare networks which are of different size and which may contain different nodes, yet which are hypothesized to be related?

Example: Protein-protein interaction networks

Proteins are

- large and complex organic molecules built of units called amino acids
- very versatile
- serve crucial functions in most biological processes

Most proteins function through interactions (which are physical contacts) with other molecules, and often these are other proteins.



Interaction detection

There is a large amount of experimental results for interaction data available.

But error rate estimates range from 20 to 70%, see *Saeed and Deane 2006*.

Big question: how do proteins evolve? Is evolution linked to interactions?

Network representation: A protein-protein interaction network has proteins as nodes, and two nodes are connected by an edge if the two proteins physically interact.

These networks typically have of the order of 10,000 nodes, and 40,000 edges; they are sparse.

Network alignment

To compare two networks, networks can be "aligned" (Singh et al. 2008, Phan and Sternberg 2012, Flannick et al. 2008, Liao et al. 2009, Alkan and Erten 2014, Hu et al. 2014, Patro and Kingsford 2012).

These methods aim to identify matching nodes between networks and use these matching nodes to identify exact or close subnetwork matches.

But exact graph matching is NP-hard.

Network alignment issues

Approximate network alignment methods are usually computationally intensive and for protein-protein interaction networks tend yield an alignment which contains only a relatively small proportion of the network.

In protein-protein interaction networks, finding matching nodes is hard: specific interaction matches seem not to be the rule, but rather the exception (Shou et al. 2011, Lewis et al. 2012).

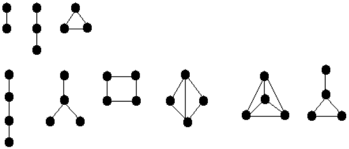
Network alignment based methods typically return results in the form of node or edge mappings and not a distance between networks. The closest to giving a distance is MI-GRAAL (Kuchaiev and Przulj 2011) which can be used to generate phylogenetic trees on small networks.

Network comparison based on functional modules

In protein-protein interaction networks, proteins are seen to form functional modules: cellular functions are carried out by modules consisting of a few interacting proteins. Such motifs seem to be conserved across species.

Often we are interested in how a group of nodes jointly achieve a certain function (and then possibly how to interrupt that process).

Hence rather than matching nodes or edges, we use small graphs as units for network comparison. Here are small graphs on 2, 3 and 4 nodes:



Trees as a comparison summary

If we have a distance (or score) between pairs of networks we can build a tree which reflects these distances.

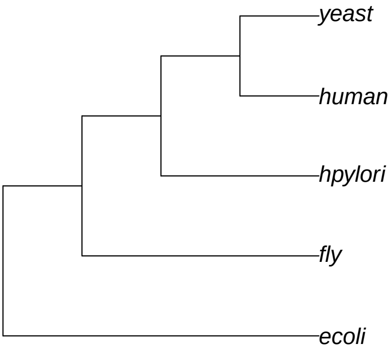
There are such tree building methos available; standard methods are called UPGMA and Neighbour-Joining.

Here we mostly use UPGMA trees: At each step, the nearest two clusters are combined into a higher-level cluster.

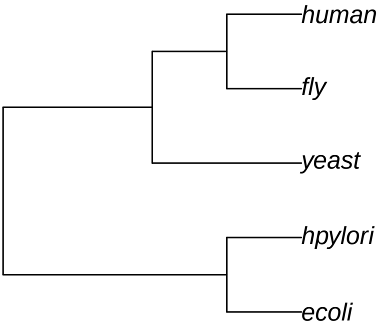
The distance between any two clusters A and B is taken to be the average of all distances between pairs of objects, one from A and the other one from B.

Using subgraph counts: not a good idea?

One could directly use Euclidean distance between the subgraph counts of, say, all 4 node subgraphs, creating a vector of length 6 for each network. Unfortunately this method is too crude for protein interaction networks:



(a) Tree based on subgraph counts



(b) Accepted (NCBI) tree

The main ideas

Instead of the networks themselves, compare the ensembles of all node neighbourhoods (ego-networks) for the networks of interest.

Compare the networks using a statistic which is based on the subgraph contents of the ensembles.

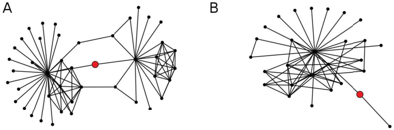
Heuristics: A given biological function is normally credited to a community of interacting proteins which act together to perform that function in the cell. Closely related species will have, on average, more of these communities in common.

Ali et al. Bioinformatics (2014)

2-step ego-networks through snowball sampling

A single node is picked in the network (the **ego**), then all nodes which are directly connected to it are picked, as well as the edges between these nodes.

In the next step all nodes which are connected to the nodes picked in the previous step are chosen, as well as all the edges between them. Here is an example from the yeast protein-protein interaction network.



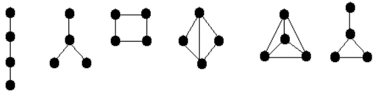
The egos for the ego-networks were A) YDR196C (a kinase) and B) YDR406W (related to transmembrane transport); they are coloured red.

The ensemble of ego-networks

A network with, say, 5000 nodes gives rise to 5000 (often overlapping) ego-networks.

We compare two networks through the small graph counts in their ensembles of ego-networks. Suppose we focus on all small graphs on 4 nodes.

For each of the ego-networks we count the number of occurrences of each of the 6 possible small graphs on 4 nodes.



An example

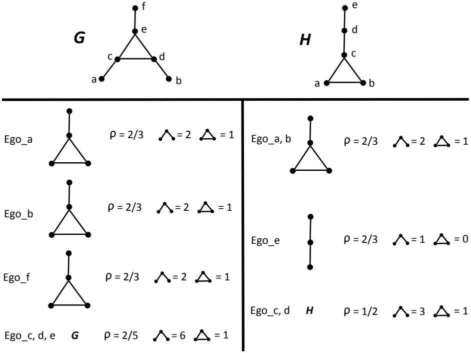


Figure : Overview of the Netdis method on a pair of networks. Each network is associated with vectors of subgraph counts, calculated from all its two-step ego-networks (this figure shows counts only for subgraphs of three nodes).

Netdis: comparing two networks

Given the expected counts for each small graph w on 4 nodes, we use

$$S_w(G) = \sum_{i \text{ node in } G} \Big(\text{No. of } w \text{ in ego-network of } i - \text{Expected no.} \Big)$$

For two networks G and H, we calculate

$$netD_2^S = \frac{1}{\sqrt{M}} \sum_w \left(\frac{S_w(G) S_w(H)}{\sqrt{S_w(G)^2 + S_w(H)^2}} \right),$$

where M is so that $netD_2^S \in [-1, 1]$. *Netdis* is defined as

$$netd_2^S = \frac{1}{2}(1 - netD_2^S) \in [0, 1].$$

Similarly for: small graphs on 3 nodes, on 5 nodes, ...

Centred counts: Expected numbers?

In theoretical models the expected small graph counts in an ego-network are typically not known, due to the additional dependence introduced by conditioning on the ego-network.

For protein-protein interaction networks, there is currently no suitable statistical model for subgraph counts available (*Rito et al. 2010, 2012*).

For each species we only have one realisation of the network available; standard statistical estimation methods which rely on a number of samples do not apply.

We estimate the expected small graph counts from the ensemble of ego-networks with similar density from a *gold-standard* network.

Example: simulated networks

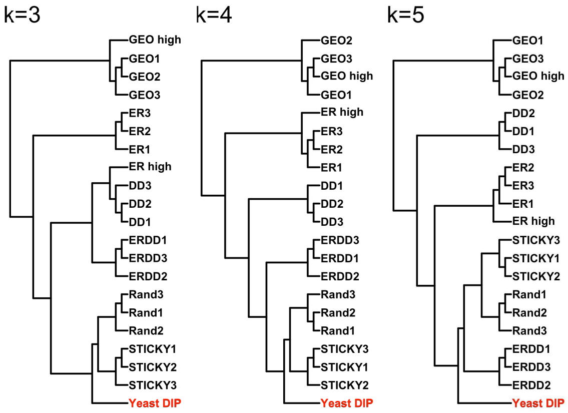
First we test our method on 3 simulated networks each of

- Bernoulli random graphs (ER)
- geometric 3-d random graphs (GEO) (see *Penrose 2003*)
- the stickiness index model (see *Przulj and Higham 2006*)
- the duplication and divergence model (DD) of *Middendorf 2005*
- the Bernoulli model with fixed degree distribution model
- the randomisation algorithm of *Maslov and Sneppen 2002*.

The parameters were chosen so that the number of nodes and edges closely match the yeast DIP protein-protein interaction network. We use the yeast DIP core dataset (*Salwinski et al. 2004*) as gold standard network.

To assess the effect of graph densities we also included the "*ER high*" and "*GEO high*" graphs with far higher graph densities.

UPGMA-based phylogenetic tree of simulated networks



Example: Protein-protein interaction network comparison

The network data of all species, apart from human, was taken from DIP; only those species with a coverage of at least 15% are included. For the human data set we used the more complete HPRD database.

Coverage is here a rough estimate of how many proteins have been probed for interactions given the expected proteome of the organism. We define it as a percentage by taking the number proteins (nodes of the network) divided by the estimated number of genes in the genome of the organism.

Species	# Genes	Nodes	Edges	Coverage	density ×1000
Hsap	21,224	9,223	36,631	43.9	0.8
Dmel	13,917	7,565	22,800	54.3	0.8
Scer	6,692	5,078	22,103	86.2	1.7
Ecoli	4,303	2,968	11,604	68.9	2.6
Hpyl	1,553	714	1,361	45.9	5.3

UPGMA-based phylogenetic trees of all species with a genome coverage of at least 15% in DIP and HPRD

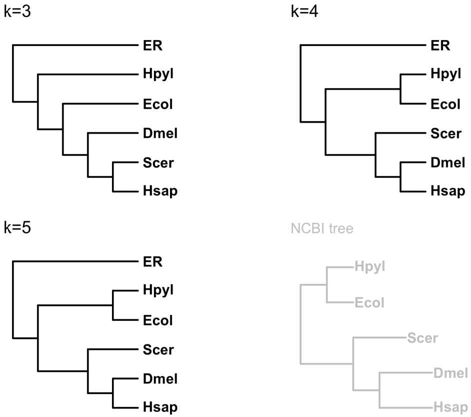
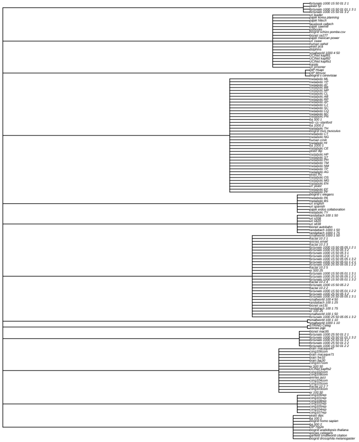


Figure : The currently accepted NCBI phylogeny between the species is shown shaded in the bottom right.

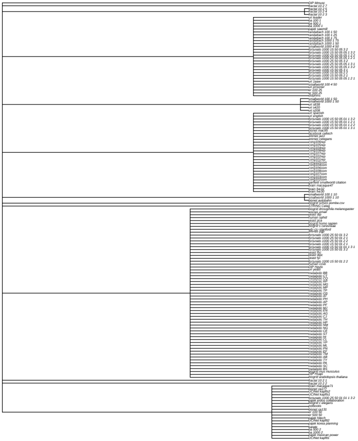
Example: 151 networks

Onnela et al. (2012) constructed a taxonomy of a large collection of networks obtained from a variety of sources. We used all unweighted and undirected networks from this set, resulting in a total of 151 networks. These come from across the biological and social domains as well as model simulations.

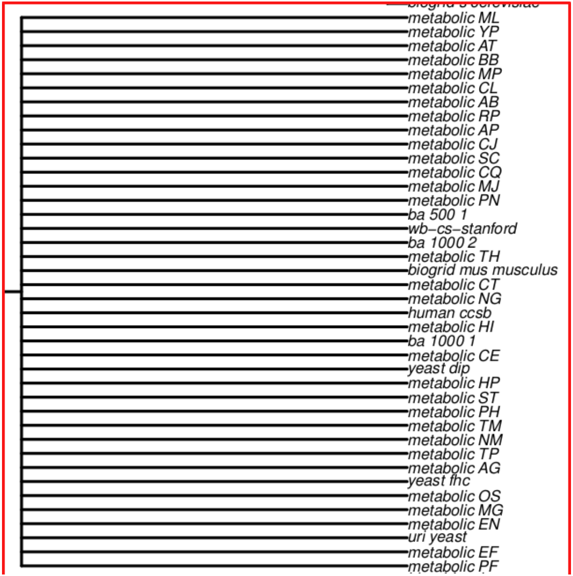
Netdis on 151 networks



151 networks clustered by the method of Onnela et al



A clade


biogrid s cerevisiae
metabolic ML
metabolic YP
metabolic AT
metabolic BB
metabolic MP
metabolic CL
metabolic AB
metabolic RP
metabolic AP
metabolic CJ
metabolic SC
metabolic CQ
metabolic MJ
metabolic PN
ba 500 1
wb–cs–stanford
ba 1000 2
metabolic TH
biogrid mus musculus
metabolic CT
metabolic NG
human ccsb
metabolic HI
ba 1000 1
metabolic CE
yeast dip
metabolic HP
metabolic ST
metabolic PH
metabolic TM
metabolic NM
metabolic TP
metabolic AG
yeast fhc
metabolic OS
metabolic MG
metabolic EN
uri yeast
metabolic EF
metabolic PF

151 networks: comparison

We first manually created a taxonomy by simply grouping the data based on type and assuming no further branching within groups. In total we identify 13 groups, such as protein interaction, congressional voting, metabolic networks, ER random graphs etc.

We then created taxonomic trees for these networks using Netdis and Onnela's method. The resulting trees were split to give 13 groups.

We compared the 13 groups from each of the methods to the manually created groups using the adjusted Rand index for cluster similarity. We also generated Monte-Carlo p-values for the similarity values by generating 50,000 samples from the null distribution, creating a random tree topology with the same number of leaves as the data set, and we then calculated the cluster similarity with the manual grouping.

151 network: comparison results

The best performing method is Netdis with similarity index 0.011 (p-value: 0). Even without correcting for the background expectation, a better grouping is achieved (Rand: 0.01, p-value: 2×10^{-5}) than Onnela *et al.*'s method, which has a similarity index of 0.006 (p-value: 0.001).

We also compared to the network alignment method MI-GRAAL; many of the networks were so large that MI-GRAAL failed to give an aligment.

We also tried ...

Using neighbour-joining as tree reconstruction method instead of UPGMA gives similar results.

We have also used Netdis for

- trade network time series (dense);
- metabolic networks (bivariate, directed).

The choice of gold-standard set is not as crucial as it may seem:

When comparing simulated data from different network models, not using any expectation already works.

For protein-protein interaction networks, not using any expectation does not work, but using other protein-protein interaction networks as gold-standard sets give similar results.

We currently use as general gold standard a Bernoulli random graph which has ego-networks with a large range of densities.

Subsampling

For large networks, comparing as many ego-networks as there are nodes in the sample is computationally intensive. Instead: only sample a fraction of the nodes.

The bootstrap:
Random sample of size n, say
Draw M observations out of the n, with replacement
Calculate the statistic of interest for this sample of size M
Repeat many times
Use the standard deviation in these samples to estimate standard deviation in the population.

The underlying idea is that of Russian dolls - the bootstrap samples should relate to the original sample just as the original sample relates to the unknown population (count the freckles on the faces of Russian dolls).

Empirical measures

Each observation can be represented by a point mass in space.
The average of these point masses is called empirical measure: a random quantity taking values in the set of measures.

This empirical measure will converge to a limit if the conditions are right; just like the law of large numbers.

Just like for real-valued random quantities, for independent identically distributed observations an approximation by a Gaussian measure holds.

We say that the bootstrap works when the bootstrap empirical measure can be approximated by a Gaussian measure centred around the true measure.

The theoretical arguments proving that the bootstrap works rely on large independent samples.

But in dependent observations the standard deviation would be estimated wrongly.

In time series: blockwise bootstrap: Kuensch (1989), Carlstein et al. (1998): sample a whole block of observations in the time series, use the block to approximate the standard deviation.

Bootstrapping in dependency graphs

Holmes and R. (2004): For random variables we can construct a graph with the random variables as the nodes.

Two nodes are linked by an edge if and only if the corresponding random variables are dependent.

The set S_i of all neighbours of a node i is then the set of all random variables which are dependent on the node random variable.

We sample nodes uniformly without replacement. To capture the dependence structure, we include not only this node but its 1-step ego-network in the empirical measure.

We can show that if the dependency graph is a regular graph (all nodes have the same degree) and if the dependency neighbourhoods are small, then the bootstrap works (with numerical bound, using Stein's method).

Details: general setting

Assume that $\mathbf{w} = (w_1, \ldots, w_n)$ is an exchangeable vector (of weights) such that all entries are positive and they sum to 1. Put

$$c^2 = c_n^2 = \mathbb{E}(w_1 - 1)^2$$

and

$$\lambda_n = \left\{ \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^{n} (w_n - 1)^2 - c^2 \right) \right\}^{\frac{1}{2}}$$

and consider the empirical measure

$$\xi_n = \frac{1}{\sqrt{n}} \sum (w_j - 1) \delta_{x_j}.$$

Let ζ_n be a Gaussian random measure with the same covariance structure as ξ_n.

Holmes and R. (2004)

For smooth test functions H,

$$
\begin{aligned}
&|\mathbb{E}H(\xi_n) - \mathbb{E}H(\zeta_n)| \\
&\quad \leq \quad \frac{1}{\sqrt{n}} \left\{ \mathbb{E}(w_1^3 + 3w_1^2) + 2(\mathbb{E}((w_1 - 1)^4)^{\frac{1}{2}}) \right\} + 3\lambda_n.
\end{aligned}
$$

In the case of i.i.d. observations the bound is $\frac{20}{\sqrt{n}}$.

Details: dependency graphs

Assume that the neighbourhood size is constant, say, γ. Put $\kappa = \frac{M}{\gamma}$; assume that this is an integer. Also assume that $M \leq n$.

Choose indices according to

$$(k_1, \ldots, k_n) \sim Mult\left(\kappa; \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right).$$

Put

$$Q_n = \frac{1}{\kappa} \sum_{i=1}^{\kappa} k_i \frac{1}{\gamma} \sum_{j \in S_i} \delta_{x_j}.$$

Centre Q_n, call the centred measure ξ_n, and compare to a centred Gaussian random measure ζ_n with the same covariance structure:

For smooth test functions H,

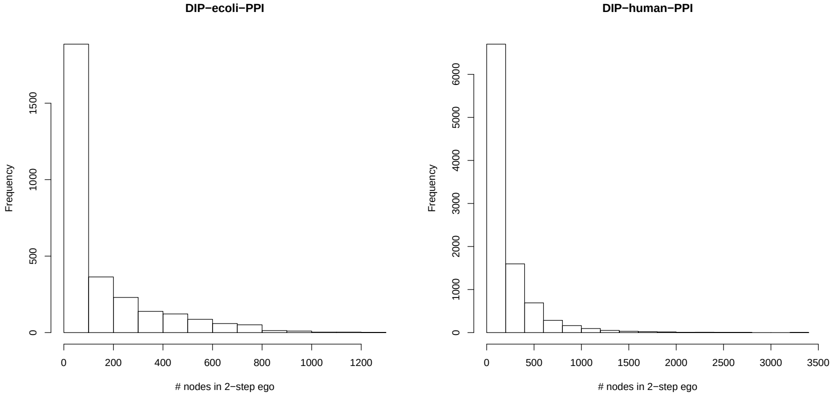
$$\begin{aligned} & |\mathbb{E}H(\xi_n) - \mathbb{E}H(\zeta_n)| \\ \leq \quad & \frac{n}{M^{\frac{3}{2}}} \left\{ \frac{7\gamma^2\sqrt{n}}{\sqrt{M}} + \frac{1}{2} + 11\sqrt{\gamma} + 4\gamma \right\}. \end{aligned}$$

If M and n are of the same order then the bound improves with n.

For other bootstrap sampling methods on networks see Bhattacharyya and Bickel (2014).

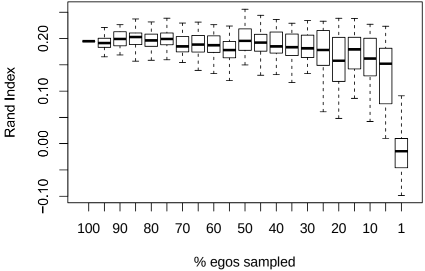
Subsampling ego-networks

Heuristics: 2-step ego-networks correspond to (strong) dependence.
Spread of ego-network sizes:



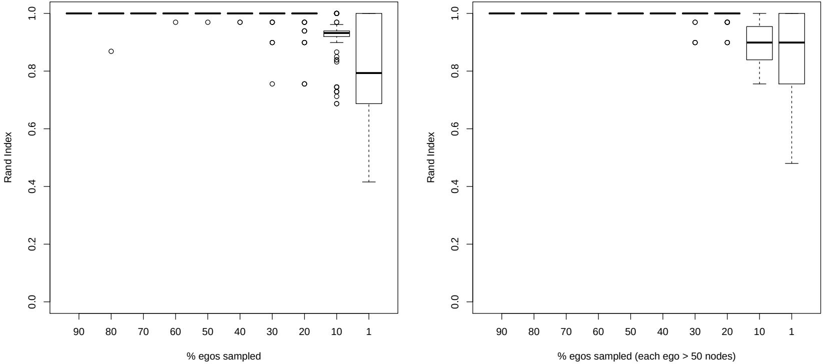
Robustness

Sampling only 20% of the ego-networks of the networks to be compared (same fraction for each network) still gives essentially the same tree. For the 151 networks:

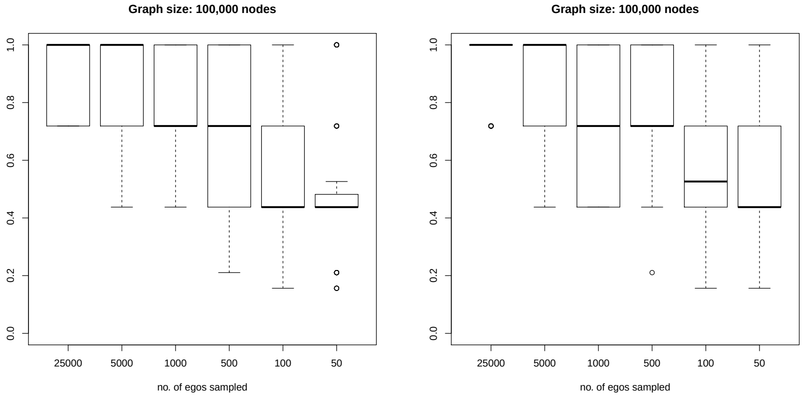


Egonetwork-size matters

The subsampling improves when a lower bound on the size of the ego-networks in introduced, for example that they contain at least 50 nodes. On the simulated data set on 5000 nodes:



100,000 nodes: any ego-networks vs at least 50 nodes



Features of the approach

Netdis does not require exact or similar one-to-one matches; instead it adopts a many-to-many approach which compares neighbourhoods in a given graph density region.

Our approach provides a way to compare networks which is not based on exact topological matches.

The success of Netdis points towards the hypothesis that interaction neighbourhoods may play a crucial role in the evolution of proteins.

While developed with PPI networks in mind, Netdis is applicable to network comparison for other types of networks.

Netdis can work even when sampling only a small fraction of the ego-networks, making it attractive for the analysis of larger complex network data sets.

Acknowledgements

Systems Biology
Doctoral Training Centre
SYS bio

EPSRC
Engineering and Physical Sciences
Research Council

OXFORD
MARTIN
SCHOOL