Statistical Inference for Networks

Systems Biology Doctoral Training Centre

Theoretical Systems Biology Module

MARCH 26-28, 2014

PROF. GESINE REINERT

www.stats.ox.ac.uk/people/academic_staff/gesine_reinert

WITH ANDREW ELLIOTT

Overview

Chapter 1: *Network summaries.* What are networks? Some examples from social science and from biology. The need to summarise networks. Clustering coefficient, degree distribution, shortest path length, motifs, between-ness, second-order summaries. Roles in networks, derived from these summary statistics, and modules in networks. Directed and weighted networks. The choice of summary should depend on the research question.

Chapter 2: *Protein interaction networks (PINs).* Protein interactions. Experimental interaction collection. Errors in networks. Predicting interactions. The STRING database. Global properties of PINs. Biological questions.

Chapter 3: *Models of random networks.* Classical Erdős-Renyi (Bernoulli) random graphs and their random mixtures, Watts-Strogatz small worlds and the modification by Newman, Barabasi-Albert scale-free networks, exponential random graph models. Gene duplication networks.

Chapter 4: *Sampling from networks.* Sampling designs: induced subgraph sampling, star sampling, snowball sampling. Within-graph versus between-graphs sampling.

Chapter 5: *Fitting a model.* Deriving the distribution of summary statistics. Parametric tests based on the theoretical distribution of the summary statistics (only available for some of the models). Quantile-quantile plots and other visual methods. Monte-Carlo tests based on shuffling edges with the number of edges fixed, or fixing the node degree distribution, or fixing some other summary. The particular issue of testing for power-law dependence. Tests carried out on the same network are not independent.

Chapter 6: *Statistical inference for networks: nodes and edges.* Inferring characteristics for a missing node from the existing network. Inferring missing edges and identifying false-positive edges.

Chapter 7: *Statistical inference for networks: motifs, modules, and communities.* Significance of motifs. Modules and communities.

Suggested reading

1. W. Ali, C. Deane and G. Reinert. Protein Interaction Networks and Their Statistical Analysis. *Handbook of Statistical Systems Biology* (eds M.P.H. Stumpf, D.J. Balding and M. Girolami), John Wiley & Sons Ltd, Chichester, UK, 2011.

2. U. Alon: *An Introduction to Systems Biology - Design Principles of Biological Circuits.* Chapman and Hall 2007.

3. R. Durrett: *Random Graph Dynamics*. Cambridge University Press 2007.

4. E.D. Kolaczyk. *Statistical Analysis of Network Data.* Springer, 2009.

5. M. Newman. *Networks: An Introduction.* Oxford University Press 2010.

6. S. Wasserman and K. Faust: *Social Network Analysis*. Cambridge University Press 1994.

7. D. Watts: *Small Worlds*. Princeton University Press 1999.

This part of the module will take place Wednesday, March 26 to Friday March 28, 2014, from 9:30 - 12:30 and 14:00-17:00 in the DTC. There will be a mixture of lectures and worked example in the mornings, and computer practicals in the afternoons. The assessment will take place through briefly presenting your results from the computer practicals in less than 5 minutes, between 16:00-17:00.

We shall use the `R` language, which is open source, and we will also have some advice available about how to implement the ideas in Python.

The notes may cover more material than the lectures. The notes may be updated throughout the course.

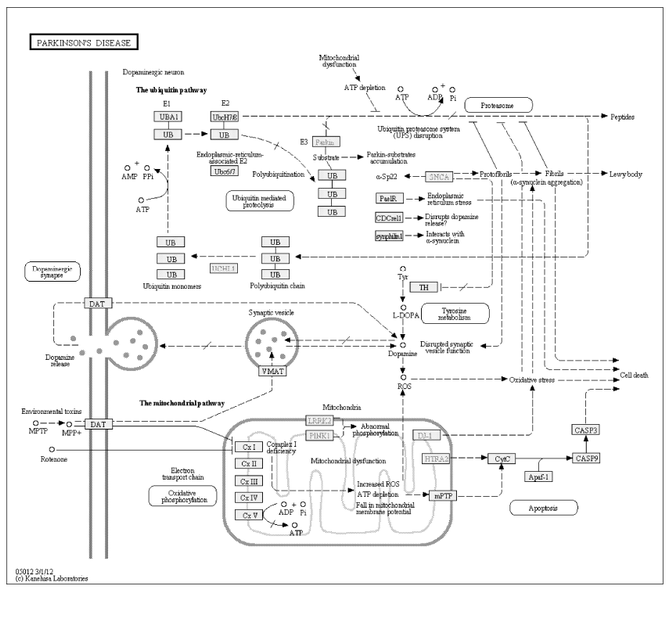
The statistical analysis of networks is a very complex topic, far beyond what could be covered in 3-day course. Hence the goal of the class is to give a brief overview of the basics, highlighting some of the issues to be addressed.

1 Network summaries

1.1 What are networks?

Networks are just graphs. Often one would think of a network as a connected graph, but not always. In this lecture course we shall use *network* and *graph* interchangeably.

Here are some examples of networks (graphs).



05012 3/6/12
(c) Kanehisa Laboratories

Figure 1: KEGG pathway: Parkinson's disease; `www.genome.jp/kegg/pathway/hsa/hsa05012.png`, 11/4/2012

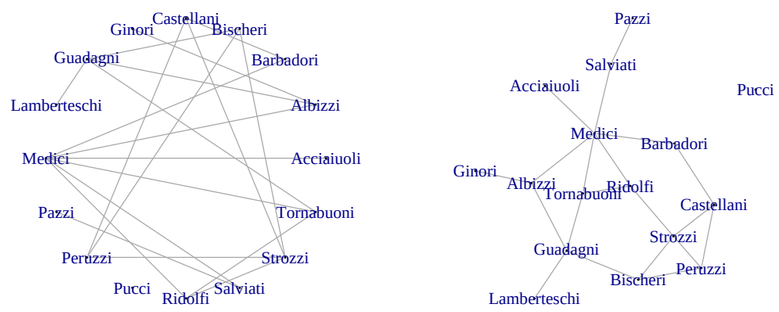


Figure 2: Marriage relations between Florentine families; two different graphical representations

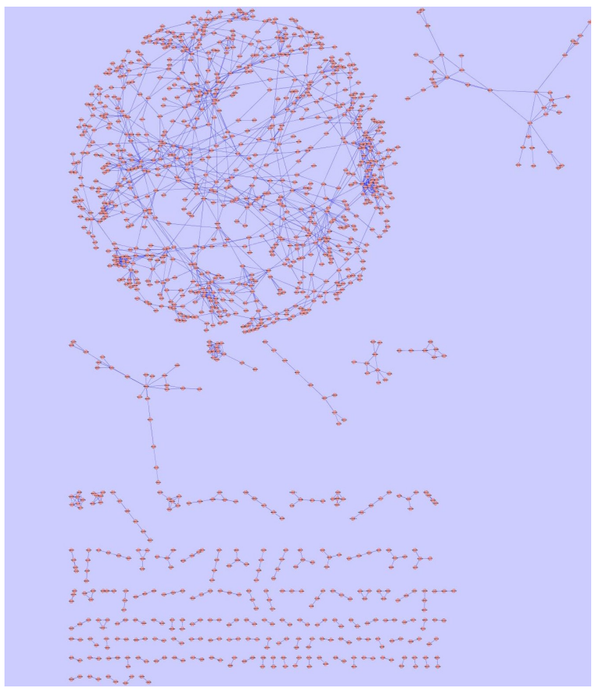


Figure 3: Yeast protein-protein interactions

Networks arise in a multitude of contexts, such as

- metabolic networks
- protein-protein interaction networks
- gene expression networks
- spread of epidemics
- neural network of *C. elegans*
- social networks
- collaboration networks (Erdős numbers ...)
- Membership of management boards
- World Wide Web
- power grid of the Western US

Challenges are

- Data collection - and conversion, for example methods to process raw data from microarrays
- Data bases
- Analysis of network topology, possibly in time and/or space

Unless the network is very small it appears like a hairball, and is difficult to analyse by just looking at it.

The study of networks has a long tradition in social science, where it is called *Social Network Analysis*. The networks under consideration are typically fairly small. In contrast, starting at around 1997, statistical physicists have turned their attention to large-scale properties of networks. Our lectures will try to get a glimpse on both approaches.

Typical networks in systems biology are

- *Protein-protein interaction networks*: There is typically no dynamics, and no spatial process; an issue are high false-positive and false-negative rates.
- *Gene regulatory networks*: The activity of genes is regulated by transcription factors, which are proteins that typically bind to DNA. Most transcription factors bind to multiple binding sites in a genome, resulting in gene regulatory networks.

- *Metabolic networks*: The chemical compounds of a living cell are connected by biochemical reactions which convert one compound into another. These reactions are catalyzed by enzymes, resulting in a biochemical network of reactions.
- *Cell signaling networks*: Cell signaling pathways interact with one another to form networks; these typically integrate protein-protein interaction networks, gene regulatory networks, and metabolic networks.

Research questions include

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?
- How did these biological networks evolve? Could mutation affect whole parts of the network at once?
- How similar are these networks? If we study some organisms very well, how much does that tell us about other organisms?
- How are these networks interlinked? Can we infer information from gene interaction networks that would be helpful for protein interaction networks?
- What are the building principles of these networks? How is resilience achieved, and how is flexibility achieved? Could we learn from biological networks to build man-made efficient networks?

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of nodes in the network?
- How to infer missing links, and how to check whether existing links are not false positives
- How to compare networks from related organisms?
- How to predict functions from networks?
- How to find relevant sub-structures of a network?

Statistical inference relies on the assumption that there is some randomness in the data. Before we turn our attention to modelling such randomness, let's look at how to describe networks, or graphs, in general.

The paradigm of Systems Biology:

Biological function arises from the integration of processes interacting across a range of spatio-temporal scale. Hence a multi-scale biomolecular modelling approach is required.

1.2 What are graphs?

A *graph* consists of *nodes* (sometimes also called *vertices*) and *edges* (sometimes also called *links*). We typically think of the nodes as actors, or proteins, or genes, or metabolites, and we think of an edge as an interaction between the two nodes at either end of the edge. Sometimes nodes may possess characteristics which are of interest (such as structure of a protein, or function of a protein). Edges may possess different weights, depending on the strength of the interaction. For now we just assume that all edges have the same weight, which we set as 1.

The *graph density* of a graph, with v vertices and e edges, is defined as the ratio between the number of edges e and the number of potential edges,

$$\frac{2e}{n(n-1)}.$$

Mathematically, we abbreviate a graph G as $G = (V, E)$, where V is the set of nodes and E is the set of edges. We use the notation $|S|$ to denote the number of elements in the set S. Then $|V|$ is the number of nodes, and $|E|$ is the number of edges in the graph G. If u and v are two nodes and there is an edge from u to v, then we write that $(u, v) \in E$, and we say that v is a *neighbour* of u.

If both endpoints of an edge are the same, then the edge is a *loop*. For now we exclude self-loops, as well as multiple edges between two nodes.

Edges may be *directed* or *undirected*. A *directed graph*, or *digraph*, is a graph where all edges are directed. The *underlying* graph of a digraph is the graph that results from turning all directed edges into undirected edges. Here we shall mainly deal with undirected graphs.

Two nodes are called *adjacent* if they are joined by an edge. A graph can be described by its $|V| \times |V|$ *adjacency matrix* $A = (a_{u,v})$;

$$a_{u,v} = 1 \text{ if and only if } (u, v) \in E.$$

As we assume that there are no self-loops, all elements on the diagonal of the adjacency matrix are 0. If the edges of the graph are undirected, then the adjacency matrix will be symmetric.

The adjacency matrix entries tell us for every node v which nodes are within distance 1 of v. If we take the matrix produce $A^2 = A \times A$, the entry for (u, v) with $u \neq v$ would be

$$a^{(2)}(u, v) = \sum_{w \in V} a_{u,w} a_{w,v}.$$

If $a^{(2)}(u, v) \neq 0$ then u can be reached from v within two steps; u is within distance 2 of v. Higher powers can be interpreted similarly.

Example: Adjacency matrix for marital relation between Florentine families (see Wasserman+Faust, p.744).

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

A *complete* graph is a graph such that every pair of nodes is joined by an edge. The adjacency matrix has entry 0 on the diagonal, and 1 everywhere else.

A *bipartite* graph is a graph where the node set V is decomposed into two disjoint subsets, U and W, say, such that there are no edges between any two nodes in U, and also there are no edges between any two nodes in W; all edges have one endpoint in U and the other endpoint in W. For example we create a metabolic network by taking U the set of enzymes, and V the set of metabolites. The adjacency matrix A can then be arranged such that it is of the form

$$\begin{bmatrix} 0 & A_1 \\ A_2 & 0 \end{bmatrix}.$$

1.3 Network summaries

The *degree* $deg(v)$ of a node v is the number of edges which involve v as an endpoint. The degree is easily calculated from the adjacency matrix A;

$$deg(v) = \sum_u a_{u,v}.$$

The *average degree* of a graph is then the average of its node degrees.

A *degree distribution* $(d_0, d_1, d_2, , \ldots)$ of a graph on n vertices is the vector of fraction of vertices with given degree;

$$d_i = \frac{1}{n} \times \text{ number of vertices of degree } i.$$

For directed graphs we would define the *in-degree* as the number of edges directed at the node, and the *out-degree* as the number of edges that go out from that node.

The *local clustering coefficient* of a vertex v is, intuitively, the proportion of its "friends" who are friends themselves. Mathematically, it is the proportion of neighbours of v which are neighbours themselves. In adjacency matrix notation,

$$C(v) = \frac{\sum_{u,w \in V} a_{u,v} a_{w,v} a_{u,w}}{\sum_{u,w \in V} a_{u,v} a_{w,v}}.$$

Here $0/0 := 0$.

The *average clustering coefficient* is defined as

$$\bar{C} = \frac{1}{|V|} \sum_{v \in V} C(v).$$

Note that

$$\frac{1}{2}\sum_{u\neq v\in V;w\neq u,v\in V} a_{u,v}a_{w,v}a_{u,w}$$

is the number of triangles involving v in the graph. Similarly,

$$\frac{1}{2}\sum_{u\neq w\in V} a_{u,v}a_{w,v}$$

is the number of *2-stars* centred around v in the graph. The local clustering coefficient describes how ”locally dense” a graph is. Sometimes the local clustering coefficient is also called the *transitivity*.

The *global clustering coefficient* or *transitivity* is defined as

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triples}}.$$

Note that $\bar{C} \neq C$ in general. Indeed $\bar{C}$ tends to be dominated by vertices with low degree, since they tend to have small denominators in the local clustering coefficient.

The global clustering coefficient in the Florentine family example is 0.1914894; the global clustering coefficient in the Yeast data is 0.1023149. The average clustering coefficient in the Florentine family example is 0.1395833.

For models of random networks often we consider the *expected clustering coefficient*

$$E(C) = \frac{3 \times \mathbf{E}(\text{number of triangles})}{\mathbf{E}(\text{number of connected triples})}.$$

Unfortunately all of the average clustering coefficient, the global clustering coefficient, and the expected clustering coefficient are often just called *the clustering coefficient* in the literature. Here we mean by *clustering coefficient* the global clustering coefficient, unless otherwise stated.

In a graph a *path* from node v_0 to node v_n is an alternating sequence of nodes and edges, $(v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_n)$ such that the endpoints of e_i are v_{i-1} and v_i, for $i = 1, \ldots, n$. A graph is called *connected* if there is a walk between any pair of nodes in the graph, otherwise it is called *disconnected*. The *distance* $\ell(u, v)$ between two nodes u and v is the length of the shortest path joining them. This path does not have to be unique.

We can calculate the distance $\ell(u, v)$ from the adjacency matrix A as the smallest power p of A such that the (u, v)-element of A^p is not zero.

In a connected graph, the *average shortest path length* is defined as

$$\ell = \frac{1}{|V|(|V| - 1)} \sum_{u \neq v \in V} \ell(u, v).$$

The average shortest path length describes how ”globally connected” a graph is.

Node degree, clustering coefficient, and shortest path length are the most common summaries of networks. Other popular summaries, to name but a few, are: the *between-ness of an edge* counts the proportion of shortest paths between any two nodes which pass through this edge. Similarly, the *between-ness of a node* is the proportion of shortest paths between any two nodes which pass through this node. The *connectivity* of a connected graph is the smallest number of edges whose removal results in a disconnected graph.

In addition to considering these general summary statistics, it has proven fruitful to describe networks in terms of *motifs*; these are building- block patterns of networks such as a feed-forward loop, see the book by *Alon*. Here we think of a motif as a subgraph with a fixed number of nodes and with a given topology. In biological networks, it turns out that motifs seem to be conserved across species. They seem to reflect functional units which combine to regulate the cellular behaviour as a whole.

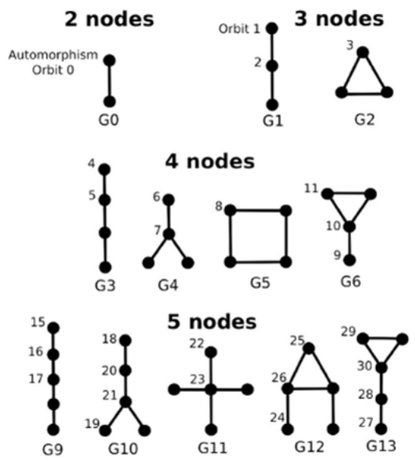


Figure 4: Small subgraphs; Przulj 2006

The decomposition of *communities* in networks, small subgraphs which are highly connected but not so highly connected to the remaining graph, can reveal some structure of the network.

Core and periphery structure detection can also shed light on the network.

Identifying *roles* in networks singles out specific nodes with special properties, such as hub nodes, which are nodes with high degree.

Looking at the ”spectral decomposition”, i.e. at eigenvectors and eigenvalues, of the adjacency matrix, provides another set of summaries, such as *centrality*.

The above network summaries provide an initial go at networks. Specific networks may require specific concepts. In protein interaction networks, for example, there is a difference whether a protein can interact with two other proteins simultaneously (party hub) or sequentially (date hub). In addition, the research question may suggest other summaries. For example, in fungal networks, there are hardly any triangles, so the clustering coefficient does not make much sense for these networks.

Excursion: Milgram and the *small world effect*.

In 1967 the American sociologist Milgram reported a series of experiments of the following type. A number of people from a remote US state (Nebraska, say) are asked to have a letter (or package) delivered to a certain person in Boston, Massachusetts (such as the wife of a divinity student). The catch is that the letter can only be sent to someone whom the current holder knew on a first-name basis. Milgram kept track of how many intermediaries were required until the letters arrived; he reported a median of six; see for example `www.uaf.edu/northern/big_world.html`. This made him coin the notion of *six degrees of separation*, often interpreted as everyone being six handshakes away from the President. While the experiments were somewhat flawed (in the first experiment only 3 letters arrived), the concept of *six degrees of separation* has stuck.

Exercise Set 1:

- Draw an undirected complete graph on 6 nodes, and write down its adjacency matrix. Determine the degrees of the 6 nodes. What is the clustering coeffient?
- Draw two different undirected graphs on 6 nodes where each node has degree 2, and write down their adjacency matrices. What are their clustering coefficients?
- Find the number of triangles in a graph (with no self-loops) using the adjacency matrix.
- We call a graph *simple* if it has no loops or multiple edges. Here we consider only undirected graphs. Let $G = (V, E)$ be a simple graph with $|V| = n$ nodes. Show that the sum of degrees in a simple graph is always even. (This is sometimes called the *Handshake Lemma*.)

2 Protein interaction networks

Protein-protein interactions (PPI) are the corner-stone of most biological processes taking place in cells. Proteins are the most versatile macromolecules in living systems and serve crucial functions in most biological processes. For example, they function as catalysts, transport and store other molecules such as oxygen, provide mechanical support and immune protection and control growth differentiation. Proteins are linear polymers built of monomer units called amino acids.

Most proteins function through interaction with other molecules, and often these are other proteins. These interactions are physical contacts.

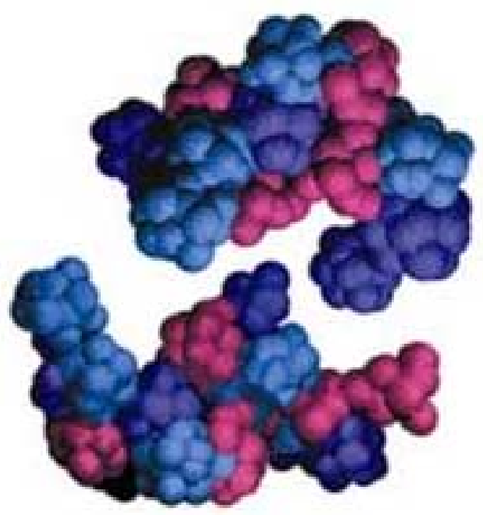


Figure 5: A sketch of two interacting proteins

There is an important distinction between transient and obligate protein interactions. Many proteins exist as parts of permanent obligate complexes such as multi-subunit enzymes, which may often fold and bind simultaneously. We can separate out the types of physical interactions we expect to observe into three specific classes:

- Multi-domain protein
- Stable complex
- Transient interactions

2.1 Interaction detection and datasets

Experimental techniques for interaction detection include

- Crystalised complex: low through-put
- Co-immunoprecipitation: low through-put
- Yeast 2 Hybrid Assay: high through-put
- Purification of complex followed by Mass Spectrometry; TAP-MS: high through-put

The high through-put methods have a high false-positive and a high false-negative rate. Error rate estimates range from 20 to 70 % in Saeed and Deane, 2006. Newer datasets have potentially less error.

Main protein interaction datasets, with partial overlap:

- BIND Biomolecular Interaction Network Database (Bader et al. 2001)
- DIP Database of Interacting Proteins: about 23,000 Yeast interactions (Xenarios et al. 2002)
- MINT Molecular INteractions Database (Zanzoni et al. 2002)
- BioGRID General Repository for Interaction Datasets (Breitkreutz et al. 2003)
- IREFINDEX is a combined database which includes BIND, BioGRID, CORUM, DIP, HPRD, InnateDB, IntAct, MatrixDB, MINT, MPact, MPIDB, MPPI and OPHID (Razick et al., 2008)
- HPRD Human Protein Reference Database: about 35,000 Human interactions (Keshava Prasad et al. 2009)
- HINT (High-quality INTeractomes) (Das and Yu 2012)

Most of these databases contain protein-protein interaction data only, though MINT and BIND also feature interactions involving non-protein entities such as promoter regions and mRNA transcripts. Most include yeast data, which is becoming a reference set, together with data from other species and some curated data coming from small-scale experiments. DIP is probably the most highly curated database of protein interactions. Curation in DIP is done manually by experts and also automatically using computational approaches.

The choice of database depends on the research question. See Mathivanan et al., 2006, for a comparison of human protein-protein interaction data bases.

2.2 Predicting interactions

Interactions are also predicted or inferred, based on other biological data. Methods include

Gene Neighbour

If two genes are bound to be neighbours in several different genomes, then a functional linkage is inferred.

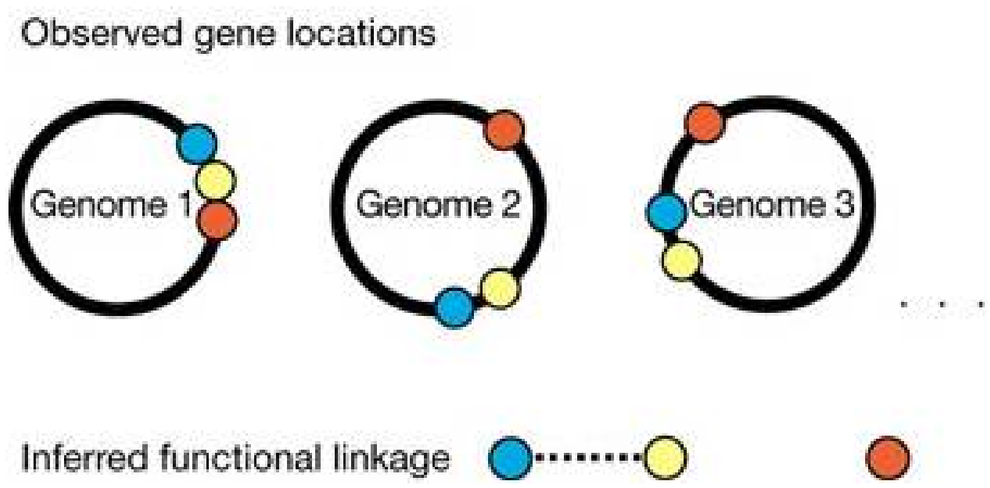


Figure 6: A sketch of the gene neighbour method

Rosetta Stone/ Gene Fusion

Some pairs of interacting proteins have homologs in other organisms which are fused to a single protein chain. This fused protein is called the fused domain or Rosetta Stone sequence. The two matches are functionally linked.

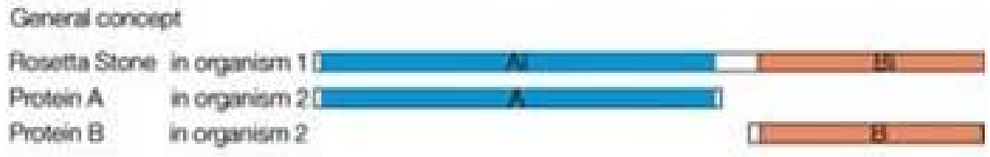


Figure 7: A sketch of the Rosetta Stone method

Phylogenetic profile

The phylogenetic profile has as number of entries the number of genomes which have been sequenced, and in each genome it is recorded whether or not the proteins in question are present. The proteins are then clustered based on the similarity of their phylogenetic profiles.

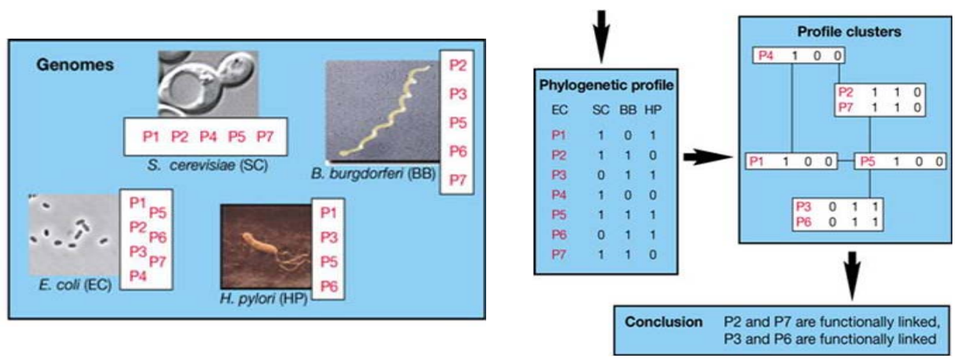


Figure 8: A sketch of the Phylogenetic profile method

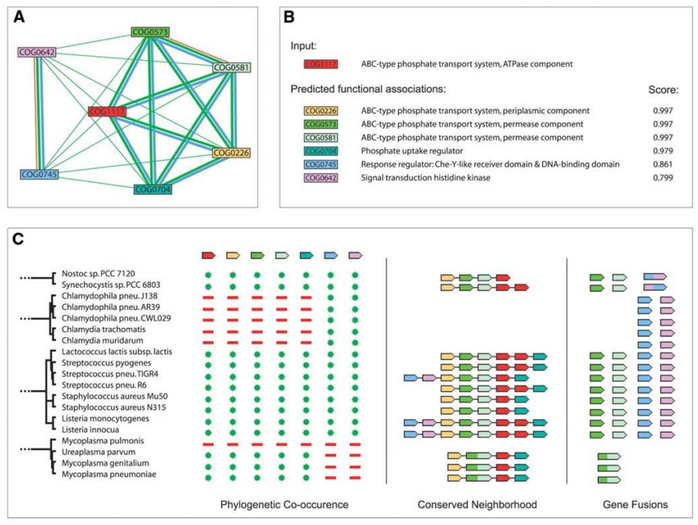


Figure 9: STRING: database of predicted interactions

When using observed as well as predicted interactions, the method of predicting interactions may be a confounding factor for what we would like to assess. From a statistical viewpoint it is often easier to use only "observed" interactions.

Sets of protein-protein interactions can be viewed as protein-protein interaction networks (PINs). In PINs, proteins are nodes, interactions are edges, edges are undirected and may or may not have weights. Here are some global summaries of PINs:

Statistic	Yeast DIP	Human HPRD
Nodes	4823	12937
Edges	17471	43496
Avg. degree	6.10	6.72
Avg. Clustering Coeff.	0.1283	0.1419
Avg. Shortest Path	4.14	4.40

How can we draw conclusions from these summaries?

3 Models of random networks

In order to judge whether a network summary is ”unusual” or whether a motif is ”frequent”, there is an underlying assumption of randomness in the network.

The randomness can stem from the construction of the network itself, such as in networks of disease transmissions, or from errors in the data - or from both.

Errors which can create randomness include

- There may be missing edges in the network. Perhaps a node was absent (social network) or has not been studied yet (protein interaction network).
- Some edges may be reported to be present, but that recording is a mistake. Depending on the method of determining protein interactions, the number of such *false positive* interactions can be substantial, of around 1/3 of all interactions.
- There may be transcription errors in the data.
- There may be bias in the data, some part of the network may have received higher attention than another part of the network.

Often network data are snapshots in time, while the network might undergo dynamical changes.

To understand the randomness mathematical models have been suggested.

3.1 Bernoulli (Erdős-Renyi) random graphs

The most standard random graph model is that of Erdős and Renyi (1959). The (finite) node set V is given, say $|V| = n$, and an edge between two nodes is present with probability p, independently of all other edges. As there are

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

potential edges, the expected number of edges is then

$$\binom{n}{2} p.$$

Each node has $n-1$ potential neighbours, and each of these $n-1$ edges is present with probability p, and so the expected degree of a node is $(n-1)p$. As the expected degree of a node is the same for all nodes, the expected degree of a randomly picked node is $(n-1)p$.

Similarly, the expected number of triangles in the graph is

$$\binom{n}{3}p^3 = \frac{n(n-1)(n-2)}{6}p^3,$$

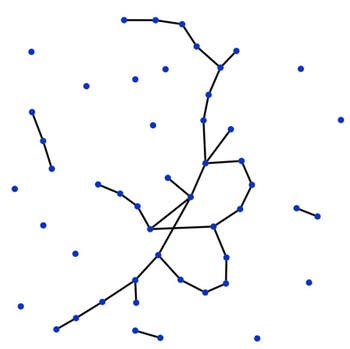
and the expected number of 2-stars is

$$\binom{n}{3}p^2.$$

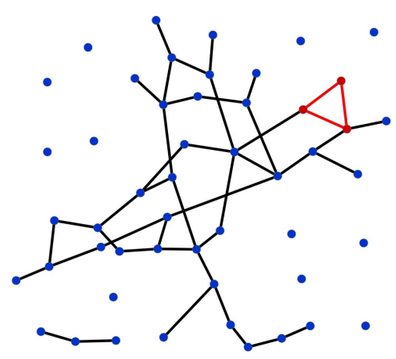
Thus, the expected clustering coefficient is

$$\frac{\binom{n}{3}p^3}{\binom{n}{3}p^2} = p.$$

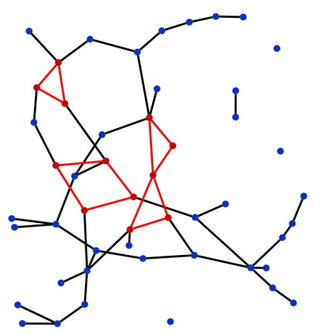
Bernoulli random graphs display a threshold-type behaviour for the appearance of small subgraphs. For example for the number of triangles when p/n is much smaller than 1, then we do not expect to see triangles; when it is much larger than 1 we would expect to see many triangles. Here is a simulation by Tiago Rito:



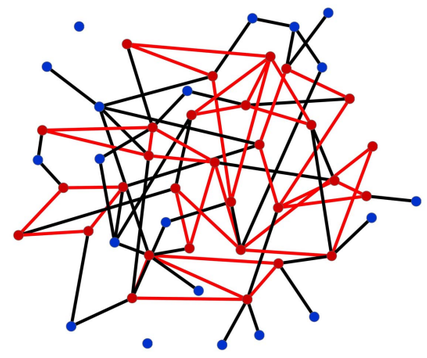
35 edges
0 triangles
0 squares



45 edges
1 triangle
0 squares



55 edges
3 triangles
1 square



80 edges
8 triangles
5 squares

A popular variant of the Bernoulli random graphs are Bernoulli random graphs with fixed degree sequence. Given a (real) network every vertex keeps its degree but the edges are then distributed at random. This model is also called the *configuration model.*

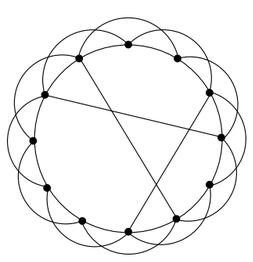
In a Bernoulli random graphs, your friends are no more likely to be friends themselves than would be a two complete strangers. This model is clearly not a good one for social networks. Below is an example from scientific collaboration networks (*N. Boccara, Modeling Complex Systems, Springer 2004, p.283*. We can estimate p as the fraction of average node degree and $n-1$; this estimate would also be an estimate of the clustering coefficient in a Bernoulli random graph.

Network	n	ave degree	$\bar{C}$	$C_{Bernoulli}$
Los Alamos archive	52,909	9.7	0.43	0.00018
MEDLINE	1,520,251	18.1	0.066	0.000011
NCSTRL	11,994	3.59	0.496	0.0003

Also in real-world graphs often the shortest path length is much shorter than expected from a Bernoulli random graph with the same average node degree. The phenomenon of short paths, often coupled with high clustering coefficient, is called the *small world phenomenon*. Remember the Milgram experiments!

3.2 The Watts-Strogatz model

Watts and Strogatz (1998) published a ground-breaking paper with a new model for small worlds; the version currently most used is as follows. Arrange the n nodes of V on a lattice. Then hard-wire each node to its k nearest neighbours on each side on the lattice, where k is small. Thus there are nk edges in this hard-wired lattice. Now introduce random shortcuts between nodes which are not hard-wired; the shortcuts are chosen independently, all with the same probability.



If there are no shortcuts, then the average distance between two randomly chosen nodes is of the order n, the number of nodes. But as soon as there are just a few shortcuts, then the average distance between two randomly chosen nodes has an expectation of order $\log n$. Thinking of an epidemic on a graph - just a few shortcuts dramatically increase the speed at which the disease is spread.

It is possible to approximate the node degree distribution, the clustering coefficient, and the shortest path length reasonably well mathematically; we will come back to these approximations later.

While the Watts-Strogatz model is able to replicate a wide range of clustering coefficient and shortest path length simultaneously, it falls short of producing the observed types of node degree distributions. It is often observed that nodes tend to attach to ”popular” nodes; popularity is attractive.

3.3 ”The” Barabasi-Albert model

In 1999, Barabasi and Albert noticed that the actor collaboration graph and the World Wide Web had degree distributions that were of the type

$$Prob(degree = k) \sim Ck^{-\gamma}$$

for $k \to \infty$. Here $C > 0$ is a constant. Such behaviour is called *power-law behaviour*; the constant γ is called the *power-law exponent*. Subsequently a number of networks have been identified which show this type of behaviour. They are also called *scale-free random graphs*.

To explain this behaviour, Barabasi and Albert introduced the *preferential attachment* model for network growth. Suppose that the process starts at time 1 with 2 nodes linked by m (parallel) edges. At every time $t \geq 2$ we add a new node with m edges that link the new node to nodes already present in the network. We assume that the probability π_i that the new node will be connected to a node i depends on the degree $deg(i)$ of i so that

$$\pi_i = \frac{deg(i)}{\sum_j deg(j)}.$$

To be precise, when we add a new node we will add edges one at a time, with the second and subsequent edges doing preferential attachment using the updated degrees.

This model has indeed the property that the degree distribution is approximately power law with exponent $\gamma = 3$. Other exponents can be achieved by varying the probability for choosing a given node.

Unfortunately the above construction will not result in any triangles at all. It is possible to modify the construction, adding more than one edge at a time, so that *any* distribution of triangles can be achieved.

3.4 Erdős-Renyi Mixture Graphs

An intermediate model is given by the Erdős-Renyi mixture model, also known as *latent block models* or *stochastic block models* (*Nowicky and Snijders (2001)*). Here nodes are of different types, say, there are L different types. Let α_q be the prior probability for a vertex to be of class q. Then edges are constructed independently, such that the probability for an edge varies only depending on the type of the nodes at the endpoints of the edge; $\pi_{q\ell} = \pi_{\ell,q}$ is the probability for a vertex from class q to be connected with a vertex from class ℓ.

Robin et al. (2008) have shown that this model is very flexible and is able to fit many real-world networks reasonably well, although it does not produce a power-law degree distribution. In particular they use it to model a metabolic network of *E.coli*; in their network, vertices are chemical reactions, and two reactions are connected if a compound produced by the first one is a part of the second one (or vice versa). Their network has 605 vertices and 1782 edges. They find that the best-fitting model is a model with $L = 21$ classes, and many of these classes gather reactions involving a same compound, such as chorismate, pyruvate, L-aspartate, and ATP.

3.5 Exponential random graph (p^*) models

Networks have been analysed for ”ages” in the social science literature, see for example the book by *Wasserman and Faust*. Here usually digraphs are studied, and the research questions are different from biological networks. Typical research questions could be

- Is there a tendency in friendship towards transitivity; are friends of friends my friends?
- What is the role of explanatory variables such as income on the position in the network?
- What is the role of friendship in creating behaviour (such as smoking)?
- Is there a hierarchy in the network?
- Is the network influenced by other networks for which the membership overlaps?

Exponential random graph (p^*) models model the whole adjacency matrix of a graph simultaneously, making it easy to incorporate dependence. Suppose that $\mathbf{X}$ is our random adjacency matrix. The general form of the model is

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\theta' \mathbf{z}(\mathbf{x})\},$$

where θ is a vector of model parameters and $\mathbf{z}(\mathbf{x})$ is a vector of network statistics; κ is a normalising quantity so that the probabilities sum to 1.

The simplest such model is that the probability of any edge is constant across all possible edges, i.e. the Bernoulli graph, for which

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\theta L(\mathbf{x})\},$$

where $L(\mathbf{x})$ is the number of edges in the network $\mathbf{x}$ and θ is a one-dimensional parameter.

3.6 Gene duplication networks

A more biologically plausible mechanism is *gene duplication and divergence* (*Bebek et al., 2006*); known to give a heavy-tailed degree distribution:

At each iteration t a node is chosen uniformly at random (parent) and duplicated by adding a new node to the graph (child) which retains all edges incident to the parent node.

The divergence is then introduced at each iteration by
i) deleting each edge in the child node independently with probability q;
ii) each node in the graph is independently connected to the child node with probability r/t.

A final step consists in assuring that, if a singleton is produced, this node will be connected to at least one uniformly chosen random node.

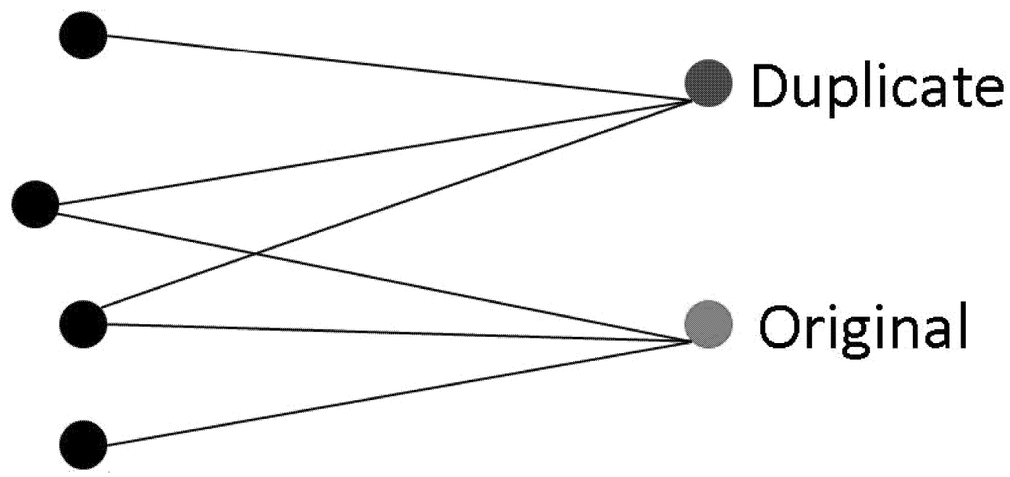


Figure 10: Duplication divergence model. The duplicate node loses some of the original links and creates new links.

While the basic duplication divergence model remains by far the most widely accepted one in protein interaction network literature, some studies have proposed enhancements such as mixture models combining DD and preferential attachment (*Ratmann et al. 2007*). Alternatives to the DD model have also been investigated, including a crystal growth model that captures the age-dependency of interaction density in the yeast interaction network along with hierarchical modularity (*Kim et al. 2008*).

Navlakha and Kingsford (2011) have developed algorithms to reconstruct the network history based on such models.

3.7 Specific models for specific networks

Depending on the research question, it may make sense to build a specific network model.

For metabolic pathways, a number of Markov models have been introduced.

When thinking of flows through networks, it may be a good idea to use weighted networks; the weights could themselves be random.

Exercise Set 2:

Consider an Erdős-Renyi mixture model with two types of nodes. Type 1, of which there are n_1 nodes, has edge probability p_1; whereas Type 2, of which there are n_2 nodes, has edge probability p_2; the edge probability for an edge between a Type-1 node and a Type-2 node is $p_{1,2}$. We are interested in the average node degree and the clustering coefficient.

- What should average node degree and the clustering coefficient be if $p_{1,2} = 0$? What if $p_{1,2} \neq 0$ but $p_1 = p_2 = 0$?
- What would the average node degree be in general?
- For the clustering coefficient, which types of triangles would you need to consider? What would you expect for the case $p_{1,2} = 0$? What if $p_{1,2} \neq 0$ but $p_1 = p_2 = 0$?

Further references

J.-J. Daudin, F. Picard, S. Robin (2006). A mixture model for random graphs. Preprint.

O. Frank and D. Strauss (1986). Markov graphs. *Journal of the American Statistical Association* **81**, 832-842.

S. Navlakha and C. Kingsford (2011). Network Archaeology: Uncovering ancient networks from present-day interactions. *PLoS Computational Biology* **7**, e1001119.

K. Nowicky and T. Snijders (2001). Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association* **455**, 1077-1087.

S. Milgram (1967). The small world problem. *Psychology Today* **2**, 60–67.

4 Sampling from networks

Often the whole network is very large and we only observe a portion of the network. If we want to draw inference from the whole network based on the sample then we need to take the specifics of our sample into account.

Sometimes we would like to focus on a small part of the network, to have a closer look. In this case we may like to draw samples from the network.

The main sampling schemes for networks which we shall consider are

1. induced subgraph sampling - we take a random sample of vertices and observe all edges between the vertices in the sample.

2. snowball sampling - in a 1-hop snowball sample we start with one vertex, sample all of its edges and neighbours and, depending on the scheme, the edges between its neighbours too. In a 2-hop snowball sample we start with one vertex and sample all vertices within distance 2 of the vertex (i.e. at most two edges away), and depending on the scheme, also sample the edges between those neighbours.

Here we assume that observations are made without measurement error, and that the only source of randomness is the sampling design. This will help to understand the issues which arise from sampling alone, not taking any probabilistic model for the network into account.

The general set-up is as follows.

Suppose that we have a population $U = \{1, \ldots, N_u\}$ of units and with each unit $i \in U$ is associated a value y_i of interest. Let

$$\tau = \sum_{i \in U} y_i$$

be the population total, and let

$$\mu = \frac{\tau}{N_u}$$

be the population average. We also need the population variance

$$\sigma^2 = \frac{1}{N_u} \sum_{i \in U} (y_i - \mu)^2.$$

Let $S = \{i_1, \ldots, i_n\}$ be a sample of n units from U and observe that we observe y_i for each element in the sample. The goal is to estimate μ and τ. In many situations S is far from being a random sample with replacement; in particular often some units are more likely than other units to be included in the sample.

Suppose that each unit $i \in U$ has probability π_i to be included in S. Then the *Horvitz-Thompson estimate* of τ is

$$\hat{\tau}_\pi = \sum_{i \in S} \frac{y_i}{\pi_i}.$$

The corresponding estimate of the mean is

$$\hat{\mu}_\pi = \frac{1}{N_u} \hat{\tau}_\pi.$$

Assume that $\pi_i > 0$ for all i. It can be shown that

$$\mathbf{E}\hat{\tau}_\pi = \tau,$$

so $\hat{\mu}_\pi$ is an unbiased estimator for τ.

Let

$$\pi_{i,j} = \mathbb{P}(i \in S, j \in S)$$

with $\pi_{i,i} = \pi_i$. Then similarly it can be shown that

$$\text{Var}\left(\hat{\tau}_\pi\right) = \sum_{i \in U} \sum_{j \in U} y_i y_j \left(\frac{\pi_{i,j}}{\pi_i \pi_j} - 1\right).$$

Assuming that $\pi_{i,j} > 0$ for all pairs, an unbiased estimator for $\text{Var}\left(\hat{\tau}_\pi\right)$ is given by

$$\hat{\text{Var}}\left(\hat{\tau}_\pi\right) = \sum_{i \in S} \sum_{j \in S} y_i y_j \left(\frac{1}{\pi_i \pi_j} - \frac{1}{\pi_{i,j}}\right).$$

For induced subgraph sampling, a simple random sample of n vertices is selected from the vertex set V, without replacement, and edges are observed for all vertex pairs in the sample. Then, for any $i \in V$,

$$\pi_i = \frac{n}{|V|}$$

and, for $i \neq j$,

$$\pi_{i,j} = \frac{n(n-1)}{|V|(|V|-1)} = \frac{\binom{n}{2}}{\binom{|V|}{2}}.$$

Note that these probabilities assume that we know $|V|$.

Star sampling is a special case of snowball sampling. A simple random sample S of n vertices is selected from the vertex set V, without replacement. For each vertex in the sample all edges that are incident to it are observed. In this case, for any $i \in V$,

$$\pi_i = \frac{n}{|V|}$$

and, for $i \neq j$,

$$\mathbb{P}(i \notin S, j \notin S) = \frac{\binom{|V|-2}{n}}{\binom{|V|}{n}}.$$

If we include all adjacent vertices of all vertices in S then $\pi_{i,j}$ stays the same, but the node inclusion probabilities change. For i,

$$\pi_i = \sum_{L \subset N(i)} (-1)^{|L|+1} \mathbb{P}(L),$$

where $N(i)$ is the set of all vertices which are adjacent to i, and i itself, and $\mathbb{P}(L)$ is the probability of selecting the set L when obtaining the sample S.

Inclusion probabilities for snowball sampling become increasing intractable to calculate beyond star sampling.

Variants

Instead of attaching a variable y_i to a vertex we could attach a variable $y_{i,j}$ to an edge. The Horvitz-Thompson estimator for the total is then

$$\hat{\tau}_\pi = \sum_{i,j \in S} \frac{y_{i,j}}{\pi_{i,j}}.$$

For the variance we need the probability $\pi_{i,j,k,l}$ that vertices i, j, k and l are included in the sample. Using $y_{i,j}$ as the indicator that an edge is present, this approach can be used to estimate the total number of edges.

Similarly we could attach a variable $y_{i,j,k}$ to triples of vertices. Using $y_{i,j,k}$ as the indicator that a triangle is present, this approach can be used to estimate the total number of triangles in the graph.

Dependent sampling

Usually we assume that we have i.i.d. observations and can estimate the parameters from these observations. When analysing networks we often observe only 1 network. If the network has n vertices then we observe $\binom{n}{2}$ edge indicator variables, but depending on the network model these indicator variables may not be independent.

Also sometimes our observations relate directly to summary statistics, and these may introduce dependence even when the edges in the model are independent. Here is an example: we calculate all shortest paths between vertices in a Watts-Strogatz small world, with ϕ the probability of rewiring.

Our data are usually just one graph, and we calculate all shortest paths. But there is much overlap between shortest paths possible, creating dependence:

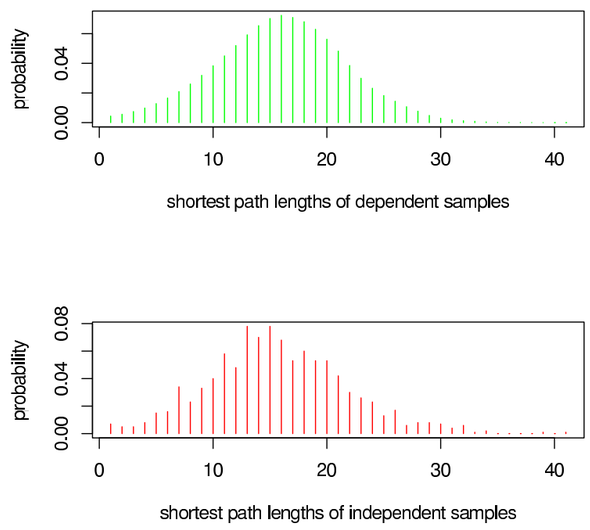


Figure 11: Simulation: $n = 500, k = 1, \phi = 0.01$

A final issue for scale-free networks: It has been shown (*Stumpf et al. (2005)* that when the underlying real network is scale-free, then a subsample on fewer nodes from the network will not be scale-free. Thus if our subsample looks scale-free, the underlying real network will not be scale-free.

In biological network analysis, is is debated how useful the concept of ”scale-free” behaviour is, as many biological networks contain relatively few nodes.

5 Fitting a model

5.1 Parametric methods

Parametric just means that we have a finite set of parameters which fully specify the model. For example:

Bernoulli (Erdős-Renyi) random graphs

In the random graph model of Erdős and Renyi (1959), the (finite) node set V is given, say $|V| = n$. We denote the set of all potential edges by E; thus $|E| = \binom{n}{2}$. An edge between two nodes is present with probability p, independently of all other edges. Here p is an unknown parameter.

5.1.1 Parameter estimation

In classical (frequentist) statistics we often estimate unknown parameters via the method of maximum likelihood.

The *likelihood* of the parameter given the data is just the probability of seeing the data we see, given the parameter.

Example: Bernoulli random graphs.

Our data is the network we see. We describe the data using the adjacency matrix, denote it by $\mathbf{x}$ here because it is the realisation of a random adjacency matrix $\mathbf{X}$. Recall that the adjacency matrix is the square $|V| \times |V|$ matrix where each entry is either 0 or 1;

$$x_{u,v} = 1 \text{ if and only if there is an edge between } u \text{ and } v.$$

The likelihood of p being the true value of the edge probability if we see $\mathbf{x}$ is

$$\mathcal{L}(p; \mathbf{x}) = \prod_{(i,j) \in E} \left\{ p^{x_{i,j}} (1-p)^{1-x_{i,j}} \right\}.$$

For example,

$$\begin{aligned}\mathcal{L}(0.5;\mathbf{x}) &= \prod_{(i,j)\in E}\left\{(0.5)^{x_{i,j}}(1-0.5)^{1-x_{i,j}}\right\} \\ &= \prod_{(i,j)\in E} 0.5 = 0.5^{|E|}.\end{aligned}$$

In general we can simplify

$$\begin{aligned}\mathcal{L}(p;\mathbf{x}) &= (1-p)^{|E|}\prod_{(i,j)\in E}\left(\frac{p}{1-p}\right)^{x_{i,j}} \\ &= (1-p)^{|E|}\left(\frac{p}{1-p}\right)^{\sum_{(i,j)\in E} x_{i,j}}.\end{aligned}$$

Note that $t = \sum_{(i,j)\in E} x_{i,j}$ is twice the total number of edges in the random graph.

To maximise the likelihood, we often take logs, and then differentiate. Here this would give

$$\begin{aligned} \ell(p;\mathbf{x}) &= \log \mathcal{L}(p;\mathbf{x}) \\ &= |E|\log(1-p) + t\log p - t\log(1-p); \end{aligned}$$

and

$$\frac{\partial \ell(p;\mathbf{x})}{\partial p} = -\frac{1}{1-p}(|E|-t) + \frac{t}{p}.$$

To find a maximum we equate this to zero and solve for p,

$$\begin{aligned} \frac{t}{p} &= \frac{1}{1-p}(|E|-t) \iff t(1-p) = p(|E|-t) \\ \iff t &= p|E| \iff p = \frac{t}{|E|}. \end{aligned}$$

We can check that the second derivative of ℓ is less than zero, so the fraction of edges that are present in the network,

$$\hat{p} = \frac{t}{|E|},$$

is our maximum-likelihood estimator.

Maximum-likelihood estimators have attractive properties; under some regularity conditions they would not only converge to the true parameter as the sample size tends to infinity, but it would also be approximately normally distributed if suitably standardized, and we can approximate the asymptotic variance.

Maximum-likelihood estimation also works well in Erdős-Renyi Mixture graphs when the number of types is known, and it works well in Watts-Strogatz small world networks when the number k of nearest neighbours we connect to is known. When the number of types, or the number of nearest neighbours, is unknown, then things become messy.

In Barabasi-Albert models, the parameter would be the power exponent for the node degree, as occurring in the probability for an incoming node to connect to some node i already in the network.

In exponential random graphs, unless the network is very small, maximum-likelihood estimation quickly becomes numerically unfeasible. Even in a simple model like

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\lambda_1 L(\mathbf{x}) + \lambda_2 S_2(\mathbf{x}) + \lambda_3 S_3(\mathbf{x}) + \lambda_4 T(\mathbf{x})\}$$

the calculation of the normalising constant κ becomes numerically impossible very quickly.

5.1.2 Markov Chain Monte Carlo estimation

A Markov chain is a stochastic process where the state at time n only depends on the state at time $n-1$, plus some independent randomness; a random walk is an example.

A Markov chain is *irreducible* if any set of states can be reached from any other state in a finite number of moves.

The Markov chain is *reversible* if you cannot tell whether it is running forwards in time or backwards in time.

A distribution is *stationary* for the Markov chain if, when you start in the stationary distribution, one step after you cannot tell whether you made any step or not; the distribution of the chain looks just the same.

There are mathematical definitions for these concepts, but we only need the main result here:

If a Markov chain is irreducible and reversible, then it will have a unique stationary distribution, and no matter in which state you start the chain, it will eventually converge to this stationary distribution.

We make use of this fact by looking at our target distribution, such as the distribution for $\mathbf{X}$ in an exponential random graph model, as the stationary distribution of a Markov chain.

This Markov chain lives on graphs, and moves are adding or deleting edges, as well as adding types or reducing types. Finding suitable Markov chains is an active area of research.

The `ergm` package has MCMC implemented for parameter estimation. We need to be aware that there is no guarantee that the Markov chain has reached its stationary distribution. Also, if the stationary distribution is not unique, then the results can be misleading. Unfortunately in exponential random graph models it is known that in some small parameter regions the stationary distribution is not unique.

5.1.3 Assessing the model fit

Suppose that we have estimated our parameters in our model of interest. We can now use this model to see whether it does actually fit the data.

To that purpose we study the (asymptotic) distributions of our summary statistics *node degree, clustering coefficient*, and *shortest path length*. Then we see whether our observed values are plausible under the estimated model.

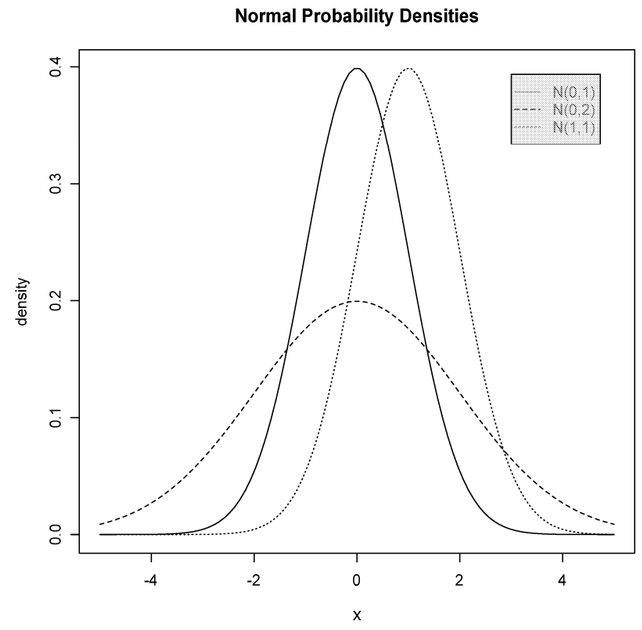
Often, secretly we would like to find that they are not plausible! Because then we can reject, say, the simple random graph model, and conclude that something more complicated is going on.

5.1.4 A quick review of distributions

Just a quick reminder of some classical distributions which often appear as limiting distributions.

The normal distribution $\mathcal{N}(\mu, \sigma^2)$

This distribution has mean μ and variance σ^2. Its shape is given by the Bell curve. Its density is awkward to manipulate, but probabilities can be calculated numerically.


Normal Probability Densities
density
x
N(0,1)
N(0,2)
N(1,1)
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4

For a normally distributed random variable, around 2/3 of the time it will be within σ (the standard deviation, square root of the variance) of the mean.

Around 95% of the time it will be within 2σ of the mean.

Around 99% of the time it will be within 3σ of the mean.

Thus if an observed value is further than 3σ away from μ, we would find that rather unusual; we would reject the null hypothesis that the data is normally $\mathcal{N}(\mu, \sigma^2)$ distributed at the level 1%.

The *Central Limit Theorem* tells us that, in a sequence of independent identically distributed observations with finite variance, the sample mean will converge to a normal distribution, and the standardised sample mean will be approximately standard normal.

Fact: If the observations are dependent, but only ”weakly” dependent, then the Central Limit Theorem still holds. (Another area of research.)

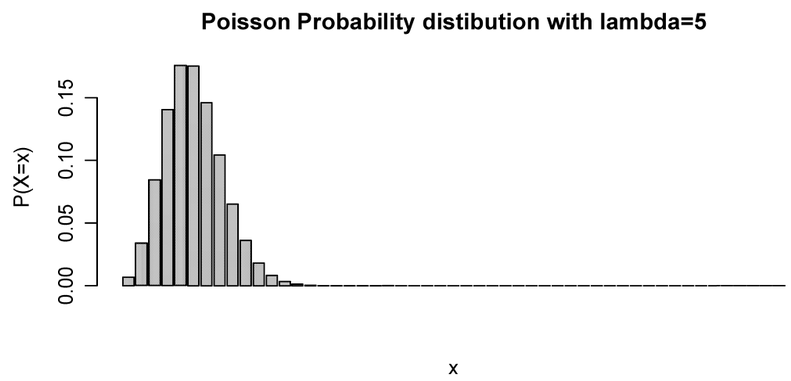
The Poisson distribution

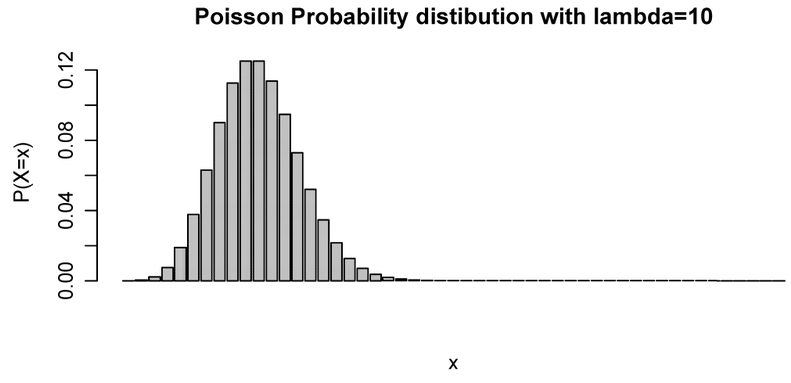
When considering the occurrence of ”rare” events, an approximation with a Poisson distribution is often more appropriate than a normal approximation.

The *Poisson distribution* lives on the non-negative integers; it has a parameter λ and X has Poisson distribution with parameter λ if

$$P(X = k) = e^{-\lambda}\frac{\lambda^k}{k!}, \quad k = 0, 1, 2, \ldots$$

It is relatively easy to calculate that mean and variance both equal to λ.


Poisson Probability distibution with lambda=5
P(X=x)
0.00
0.05
0.10
0.15
x


Poisson Probability distibution with lambda=10
P(X=x)
0.00
0.04
0.08
0.12
x

A famous result result states that if the expected number np of successes in n independent Bernoulli trials with probability of success p each is such that $np \to \lambda$ as $n \to \infty$, then the number of successes in these trials is approximately Poisson distributed with parameter λ.

Note: $np \to \lambda$ as $n \to \infty$ means that in each trial the probability of success is very small. A Poisson approximation is used for *rare events*.

The Poisson approximation is also good if the trials are only ”weakly” dependent.

5.1.5 The distribution of summary statistics in Bernoulli random graphs

In a Bernoulli random graph on n nodes, with edge probability p, the network summaries are pretty well understood.

The degree of a random node Pick a node v, and denote its degree by $D(v)$, say. The degree is calculated as the number of neighbours of this node. Each of the other $(n-1)$ nodes is connected to our node v with probability p, independently of all other nodes. Thus the distribution of $D(v)$ is Binomial with parameters $n-1$ and p, for each node v.

Typically we look at relatively sparse graphs, and so a Poisson approximation applies. If $\mathbf{X}$ denotes the random adjacency matrix, then, in distribution,

$$D(v) = \sum_{u:u \neq v} X_{u,v} \approx Poisson((n-1)p).$$

Note that the node degrees in a graph are not independent. We have seen last time that there is **no** graph on 6 nodes which has 5 nodes of degree 5 and 1 node of degree 1. So $D(v)$ does **not** stand for the average node degree.

How about the average degree of a node? Denote it by $\bar{D}$. Note that the average does not only take integer values, so would certainly not be Poisson distributed. But

$$\bar{D} \quad = \quad \frac{1}{n}\sum_{v=1}^{n} D(v) = \frac{2}{n}\sum_{v=1}^{n}\sum_{u<v} X_{u,v},$$

noting that each edge gets counted twice. As the $X_{u,v}$ are independent, we can use a Poisson approximation again, giving that

$$\sum_{v=1}^{n}\sum_{u<v} X_{u,v} \approx Poisson\left(\frac{n(n-1)}{2}p\right)$$

and so, in distribution,

$$\bar{D} \approx \frac{2}{n}Z,$$

where $Z \sim Poisson\left(\frac{n(n-1)}{2}p\right)$.

The local clustering coefficient of a random node Here it gets a little tricky already. Recall that the *local clustering coefficient* of a node v is,

$$\bar{C}(v) = \frac{\sum_{u,w \in V} X_{u,v} X_{w,v} X_{u,w}}{\sum_{u,w \in V} X_{u,v} X_{w,v}}.$$

The ratio of two random sums is not easy to evaluate. If we just look at

$$\sum_{u,w \in V} X_{u,v} X_{w,v} X_{u,w}$$

then we see that we have a sum of dependent random variables.

Most 3-tuples (u, w, v) and (r, s, t), though, will not share an index, and hence $X_{u,v}X_{w,v}X_{u,w}$ and $X_{r,s}X_{s,t}X_{r,t}$ will be independent. The dependence among the random variables overall is hence weak, so that a Poisson approximation applies. As

$$E \sum_{u,w,v \in V} X_{u,v}X_{w,v}X_{u,w} = \binom{n}{3} p^3,$$

we obtain that, in distribution,

$$\sum_{u,w.v \in V} X_{u,v}X_{w,v}X_{u,w} \approx Poisson\left(\binom{n}{3} p^3\right).$$

Similarly,

$$E \sum_{u,w \in V} X_{u,v}X_{w,v} = \binom{n}{2} p^2.$$

K. Lin (2007) showed that, for the average clustering coefficient

$$\bar{C} = \frac{1}{n} \sum_{v} C(v)$$

it is also true that, in distribution,

$$\bar{C} \approx \frac{1}{n\binom{n}{2}p^2} Z,$$

where $Z \sim Poisson\left(\binom{n}{3}p^3\right)$.

Example. In the Florentine family data, we observe 16 nodes, 20 edges, an average node degree of 2.5, and an average clustering coefficient of 0.1395833. Assess the null hypothesis that the data come from a Bernoulli random graph.

Let us assume that the null hypothesis is true. Then we estimate

$$\hat{p} = \frac{20}{\binom{16}{2}} = \frac{20 \times 2}{16 \times 15} = \frac{1}{6}.$$

As in a Bernoulli random graph $\bar{D} \approx \frac{2}{n} Z$, where $Z \sim Poisson\left(\frac{n(n-1)}{2} p\right)$, under the null hypothesis the average node degree would be $\bar{D} \approx \frac{1}{8} Z$, where $Z \sim Poisson(20)$. The probability under the null hypothesis that $\bar{D} \geq 2.5$ would then be

$$P(Z \geq 2.5 \times 8) = P(Z \geq 20) \approx 0.55,$$

so no reason to reject the null hypothesis.

Exercise Set 3:

Test the null hypothesis that the Florentine family data come from a Bernoulli random graph using a test based on the average clustering coefficient.

Shortest paths: Connectivity in Bernoulli random graph *Erdős and Renyi (1960)* showed the following "phase transition" for the connectedness of a Bernoulli random graph.

If $p = p(n) = \frac{\log n}{n} + \frac{c}{n} + O\left(\frac{1}{n}\right)$ then the probability that a Bernoulli graph, denoted by $\mathcal{G}(n, p)$ on n nodes with edge probability p is connected converges to $e^{-e^{-c}}$.

We use the O and o notation: $f(n) = O(g(n))$ as $n \to \infty$ if the fraction $\frac{f(n)}{g(n)}$ is bounded away from ∞. If $f(n) = o(g(n))$ as $n \to \infty$ then the fraction $\frac{f(n)}{g(n)}$ tends to zero as $n \to \infty$.

The *diameter* of a graph is the maximum diameter of its connected components; the diameter of a connected component is the longest shortest path length in that component.

Chung and Lu (2001) showed that, if $np \geq 1$ then, asympotically, the ratio between the diameter and $\frac{\log n}{\log(np)}$ is at least 1, and remains bounded above as $n \to \infty$.

If $np \to \infty$ then the diameter of the graph is $(1 + o(1))\frac{\log n}{\log(np)}$. If $\frac{np}{\log n} \to \infty$, then the diameter is concentrated on at most two values.

In the Physics literature, the value $\frac{\log n}{\log(np)}$ is used for the average shortest path length in a Bernoulli random graph. This has hence to be taken with a lot of grains of salt.

While we have some idea about how the diameter (and, relatedly, the shortest path length) behaves, it is an inconvenient statistics for Bernoulli random graphs, because the graph need not be connected.

5.1.6 The distribution of summary statistics in Watts-Strogatz small worlds

Recall that in this model we arrange the n nodes of V on a lattice. Then hard-wire each node to its k nearest neighbours on each side on the lattice, where k is small. Thus there are nk edges in this hard-wired lattice. Now introduce random shortcuts between nodes which are not hard-wired; the shortcuts are chosen independently, all with the same probability ϕ.

Thus the shortcuts behave like a Bernoulli random graph, but the graph will necessarily be connected. The degree $D(v)$ of a node v in the Watts-Strogatz small world is hence distributed as

$$D(v) = 2k + Binomial(n - 2k - 1, \phi),$$

taking the fixed lattice into account. Again we can derive a Poisson approximation when p is small; see K.Lin (2007) for the details.

For the clustering coefficient there is a problem - triangles in the graph may now appear in clusters. Each shortcut between nodes u and v which are a distance of $k + a \leq 2k$ apart on the circle creates $k - a - 1$ triangles automatically.

Thus a Poisson approximation will not be suitable; instead we use a *compound Poisson distribution*. A compound Poisson distribution arises as the distribution of a Poisson number of clusters, where the cluster sizes are independent and have some distribution themselves. In general there is no closed form for a compound Poisson distribution.

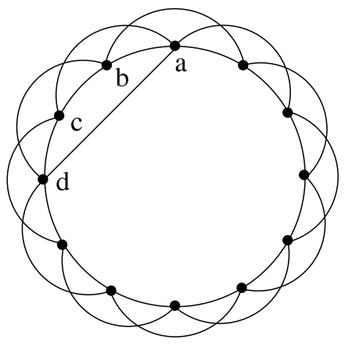


Figure 12: An example for triangles in a small-world network: one shortcut here creates 2 triangles

The compound Poisson distribution also has to be used when approximating the number of 4-cycles in the graph, or the number of other small subgraphs which have the clumping property.

It is also worth noting that when counting the joint distribution of the number of triangles and the number of 4-cycles, these counts are not independent, not even in the limit; a bivariate compound Poisson approximation with dependent components is required. See Lin (2007) for details.

The shortest path length Let $\mathcal{D}$ denote shortest distance between two randomly chosen points, and abbreviate $\rho = 2k\phi$. Then (Barbour + Reinert) show that uniformly in $|x| \leq \frac{1}{4}\log(n\rho)$,

$$Prob\left(\mathcal{D} > \frac{1}{\rho}\left(\frac{1}{2}\log(n\rho) + x\right)\right) \quad \approx \quad \int_0^\infty \frac{e^{-y}}{1 + e^{2x}y}\, dy$$

if the probability of shortcuts is small. If the probability of shortcuts is relatively large, then $\mathcal{D}$ will be concentrated on one or two points.

Note that $\mathcal{D}$ is the shortest distance between two randomly chosen points, **not** the average shortest path.

To illustrate the approximation we simulate 100 replicas, and calculate the average shortest path length in each network. We compare this distribution to the theoretical approximate distribution; we carry out 100 chi-square tests:

n	k	ϕ	$E.no$	mean p-value	max p-value
300	1	0.01	3	1.74 E-09	8.97 E-08
		0.167	50	0.1978	0.8913
	2	0.01	6	0	0
1000	1	0.003	3	1.65E-13	3.30 E-12
		0.05	50	0.0101	0.1124
	2	0.03	60	0.0146	0.2840

Thus the two statistics are close if the expected number $E.no$ of shortcuts is large (or very small); otherwise they are significantly different.

Recall: chi-square test of goodness-of-fit

To test whether a data set comes from a conjectured (*null*) distribution, we group our data into cells such that under the null distribution, the count in each cell is expected to be at least 5. Then we take the sum

$$X^2 = \sum_{\text{cells } i} \frac{(Observed_i - Expected_i)^2}{Expected_i}.$$

If the data come indeed from the null distribution, then X^2 will be approximately chi-squared distributed, with degrees of freedom "number(cells) - number(fitted parameters) - 1".

The p-value is the probability of seeing X^2 as least as large as the observed value if the null distribution is the correct distribution.

Aside: When comparing continuous distributions, the *Kolmogorov-Smirnov test* is another nonparametric alternative, as are *Wilcoxon tests*.

5.1.7 The distribution of summary statistics in Barabasi-Albert models

The node degree distribution is given by the model directly, as that is how it is designed.

The clustering coefficient depends highly on the chosen model. In the original Barabasi-Albert model, when only one new edge is created at any single time, there will be no triangles (beyond those from the initial graph). The model can be extended to match any clustering coefficient, but even if only two edges are attached at the same time, the distribution of the number of the clustering coefficient is unknown to date. The expected value, however, can be approximated:

Fronczak *et al.* (2003) studied the models where the network starts to grow from an initial cluster of m fully connected nodes. Each new node that is added to the network created m edges which connect it to previously added nodes. The probability of a new edge to be connected to a node v is proportional to the degree $d(v)$ of this node. If both the number of nodes, n, and m are large, then the expected average clustering coefficient is

$$EC = \frac{m-1}{8}\frac{(\log n)^2}{n}.$$

The average pathlengh ℓ increases approximately logarithmically with network size. If $\gamma = 0.5772$ denotes the Euler constant, then Fronczak *et al.* (2004) show for the mean average shortest path length that

$$E\ell \sim \frac{\log n - \log(m/2) - 1 - \gamma}{\log\log n + log(m/2)} + \frac{3}{2}.$$

The asymptotic distribution is not understood.

5.1.8 The distribution of summary statistics in Erdős-Renyi Mixture graphs

Given the label of a vertex, the conditional distribution of the degree of this vertex is Binomial.

The *connectivity* between class q and ℓ is the number of edges connecting a vertex from class q to a vertex from class ℓ. The expected connectivity between class q and ℓ is

$$\frac{1}{2}(n-1)\alpha_q\alpha_\ell\pi_{q,\ell}.$$

5.1.9 The distribution of summary statistics in exponential random graph models

The distribution of the node degree, clustering coefficient, and the shortest path length is poorly studied in these models. One reason is that these models are designed to predict missing edges, and to infer characteristics of nodes, but their topology itself has not often been of interest.

The summary statistics appearing in the model try to push the random networks towards certain behaviour with respect to these statistics, depending on the sign and the size of their factors θ.

When only the average node degree and the clustering coefficient are included in the model, then a strange phenomenon happens. For many combinations of parameter values the model produces networks that are either full (every edge exists) or empty (no edge exists) with probability close to 1. Even for parameters which do not produce this phenomenon, the distribution of networks produced by the model is often bimodal: one mode is sparsely connected and has a high number of triangles, while the other mode is densely connected but with a low number of triangles.

5.1.10 Gene duplication and divergence models

Estimation of the parameters is tricky. Recall our variant of the model:

The divergence is introduced at the t^{th} iteration by
i) deleting each edge in the child node independently with probability q;
ii) each node in the graph is independently connected to the child node with probability r/t.

The proportion q is often estimated by the number of interacting neighbours which are not shared by both gene duplicates versus the total number of neighbours the duplicates interact with. But this method fails to account for duplication that occur subsequent to the duplication being measured.

Gibson and Goldberg (2011) find that there is a large range of (q, r)-values which fit the number of edges and the degree distribution reasonably well, but the associated networks may have very different clustering coefficients.

They propose a variant of the model which includes the interaction sites which are involved in the interaction. They run their model backward in time, and are able to derive parameter estimates which capture the size, the clustering coefficients and the degree distribution of Yeast.

5.2 Nonparametric methods

What if we do not have a suitable test statistic for which we know the distribution? We need some handle on the distribution, so here we assume that we can simulate random samples from our null distribution. There are a number of methods available.

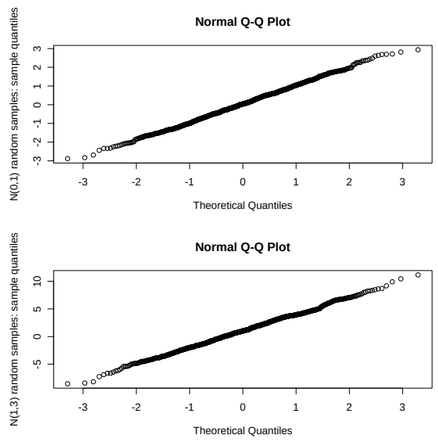
5.2.1 Quantile-quantile plots

It is often a good idea to use plots to visually assess the fit. A much used plot in Statistics a *quantile-quantile plot*.

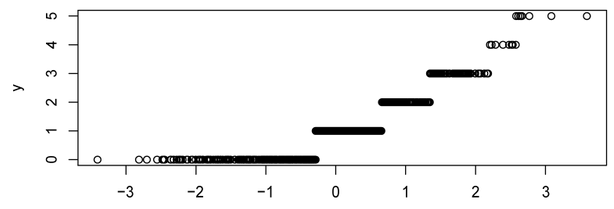
The *quantiles* of a distribution are its "percent points"; for example the 0.5 quantile is the 50 % point, i.e. the median. Mathematically, the *(sample) quantiles* q_α, are defined for $0 \leq \alpha \leq 1$ so that a proportion of at least α of the data are less or equal to q_α and a proportion of at least $1 - \alpha$ is greater or equal to q_α. There are many (at least 8) definitions of q_α if αn is not an integer.

We plot the quantiles of our observed (empirical) distribution against the quantiles of our hypothesised (null) distribution; if the two distributions agree, then the plot should result in a roughly diagonal line.

Example: Simulate 1,000 random variables from a normal distribution. Firstly: mean zero, variance 1; secondly: mean 1, variance 3. Both QQ-plots are satisfactory.



We can also use a quantile-quantile plot for two sets of simulated data, or for one set of simulated data and one set of observed data. The interpretation is always the same: if the data come from the same distribution, then we should see a diagonal line; otherwise not. Here we compare 1000 Normal (0,1) variables with 1000 Poisson (1) variables - clearly not a good fit.



5.2.2 Monte-Carlo tests

The Monte Carlo test, attributed to Dwass (1957) and Barnard (1963), is an exact procedure of virtually universal application and correspondingly widely used.

Suppose that we would like to base our test on the statistic T_0. We only need to be able to simulate a random sample $T_{01}, T_{02}, \ldots$ from the distribution, call it F_0, determined by the null hypothesis. We assume that F_0 is continuous, and, without loss of generality, that we reject the null hypothesis H_0 for large values of T_0. Then, provided that $\alpha = \frac{m}{n+1}$ is rational, we can proceed as follows.

1. Observe the actual value t^* for T_0, calculated from the data
2. Simulate a random sample of size n from F_0
3. Order the set $\{t^*, t_{01}, \ldots, t_{0n}\}$
4. Reject H_0 if the *rank* of t^* in this set (in decreasing order) is $\geqslant m$.

The basis of this test is that, under H_0, the random variable T^* has the same distribution as the remainder of the set and so, by symmetry,

$$\mathbf{P}(t^* \text{ is among the largest } m \text{ values }) = \frac{m}{n+1}.$$

The procedure is exact however small n might be. However, increasing n increases the power of the test. The question of how large n should be is discussed by Marriott (1979), see also Hall and Titterington (1989). A reasonable rule is to choose n such that $m \geqslant 5$. Note that we will need more simulations to test at smaller values of α.

An alternative view of the procedure is to count the number M of simulated values $> t^*$. Then $\hat{P} = \frac{M}{n}$ estimates the true significance level P achieved by the data, i.e.

$$P = \mathbf{P}(T_0 > t^* | H_0).$$

In discrete data, we will typically observe ties. We can break ties randomly, then the above procedure will still be valid.

Unfortunately this test does not lead directly to confidence intervals.

For random graphs, Monte Carlo tests often use shuffling edges with the number of edges fixed, or fixing the node degree distribution, or fixing some other summary.

Suppose we want to see whether our observed clustering coefficient is ”unusual” for the type of network we would like to consider. Then we may draw many networks uniformly at random from all networks having the same node degree sequence, say. We count how often a clustering coefficient at least as extreme as ours occurs, and we use that to test the hypothesis.

In practice these types of test are the most used tests in network analysis. They are called *conditional uniform graph tests*.

Some caveats:

In Bernoulli random graphs, the number of edges asymptotically determines the number of triangles when the number of edges is moderately large. Thus conditioning on the number of edges (or the node degrees, which determine the number of edges) gives degenerate results. More generally, we have seen that node degrees and clustering coefficient (and other subgraph counts) are not independent, nor are they independent of the shortest path length. By fixing one summary we may not know exactly what we are testing against.

”Drawing uniformly at random” from complex networks is not as easy as it sounds. Algorithms may not explore the whole data set. Even in Bernoulli random graphs, when the expected number of edges is moderate, so that a normal approximation would hold for the number of edges, then, asymptotically, the number of 2-stars and the number of triangles is already completely determined by the number of edges!

”Drawing uniformly at random”, conditional on some summaries being fixed, is related to sampling from exponential random graphs. We have seen already that in exponential random graphs there may be more than one stationary distribution for the Markov chain Monte Carlo algorithm; this algorithm is similar to the one used for drawing at random, and so we may have to expect similar phenomena.

Exercise:

We say that W has the Polya-Aeppli distribution with parameters a and λ if

$$Prob(W = w) = e^{-\lambda} a^w \sum_{c=1}^{w} \frac{1}{c!} \binom{w-1}{c-1} \left(\frac{\lambda(1-a)}{a} \right)^c \text{ if } w = 1, 2, \ldots,$$

and $Prob(W = 0) = e^{-\lambda}$. We observe a value of 24 for a count of a motif in a network where motif counts can be assumed to have a Polya-Aeppli distribution with known parameter $\lambda = 10$. Test the null hypothesis that $a = 0.5$ against the alternative that $a > 0.5$, at level $\alpha = 0.1$. To this purpose, you may use the following sorted averages of 49 i.i.d. samples from a Polya-Aeppli distribution with parameters $\lambda = 10, a = 0.5$.

$$2, 2, 4, 4, 5, 5, 5, 5, 5, 5, 5, 6, 7, 7, 7, 8, 8, 8, 8, 8, 9, 9, 9, 9, 9, 10, 10, 10, 11,$$

$$12, 12, 12, 13, 13, 13, 14, 16, 16, 16, 17, 17, 17, 17, 18, 19, 21, 21, 22, 32.$$

5.2.3 Scale-free networks

Barabasi and Albert introduced networks such that the distribution of node degrees the type

$$Prob(degree = k) \sim Ck^{-\gamma}$$

for $k \to \infty$. Such behaviour is called *power-law behaviour*; the constant γ is called the *power-law exponent*. The networks are also called *scale-free*, because of the following property.

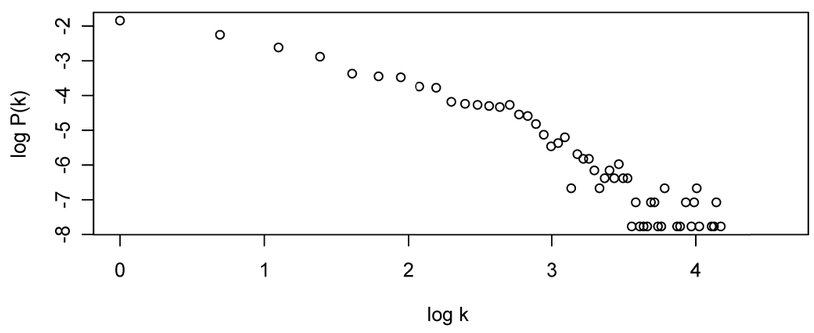
If $\alpha > 0$ is a constant, then

$$Prob(degree = \alpha k) \sim C(\alpha k)^{-\gamma} \sim C' k^{-\gamma},$$

where C' is just a new constant. That is, scaling the argument in the distribution changes the constant of proportionality as a function of the scale change, but preserves the shape of the distribution itself. If we take logarithms on both sides:

$$\begin{aligned} \log Prob(degree = k) &\sim \log C - \gamma \log k \\ \log Prob(degree = \alpha k) &\sim \log C - \gamma \log \alpha - \gamma \log k; \end{aligned}$$

scaling the argument results in a linear shift of the $\log$ probabilities only. This equation also leads to the suggestion to plot the $\log relfreq(degree = \alpha k)$ of the empirical relative degree frequencies against $\log k$. Such a plot is called a *log-log plot*. If the model is correct, then we should see a straight line; the slope would be our estimate of γ.



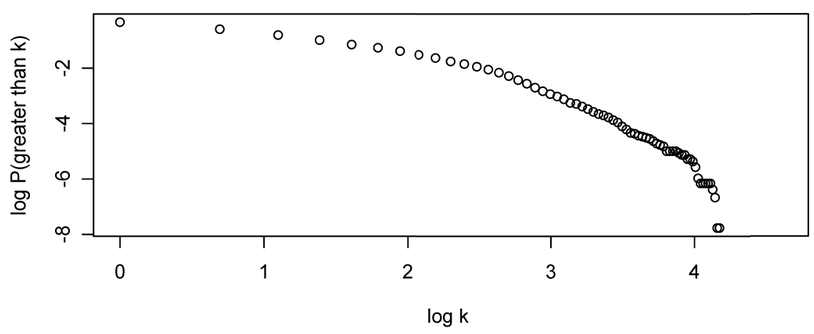


Figure 13: Yeast data log-log plots

These plots have a lot of noise in the tails. As an alternative, *Newman (2005)* suggests to plot the $\log$ of the empirical cumulative distribution function instead, or, equivalently, our estimate for

$$\log Prob(degree \geq k).$$

If the model is correct, then one can calculate that

$$\log Prob(degree \geq k) \sim C'' - (\gamma - 1) \log k.$$

Thus a log-log plot should again give a straight line, but with a shallower slope. The tails are somewhat less noisy in this plot.

In both cases, the slope can be estimated by least-squares regression: for our observations, $y(k)$ (which could be log probabilities or log cumulative probabilities, for example) we find the line $a + bk$ which minimises

$$\sum (y(k) - a - bk)^2.$$

As a measure of fit, the squared sample correlation R^2 is computed. For general observations $y(k)$ and $x(k)$, for $k = 0, 1, \ldots, n$, with averages $\bar{y}$ and $\bar{x}$, R is defined as

$$R = \frac{\sum_k (x(k) - \bar{x})(y(k) - \bar{y})}{\sqrt{(\sum_k (x(k) - \bar{x})^2)(\sum_k (y(k) - \bar{y})^2)}}.$$

It measures the strength of the linear relationship.

In linear regression, $R^2 > 0.9$ would be rather impressive. However, the rule of thumb for log-log plots is that

1. $R^2 > 0.99$
2. The observed data (degrees) should cover at least 3 orders of magnitude.

Examples include the World Wide Web at some stage, when it had around 10^9 nodes. The criteria are not often matched.

Further references

A.D. Barbour and G. Reinert (2001). Small Worlds. Random Structures and algorithms 19, 54 - 74.

A.D. Barbour and G. Reinert (2006). Discrete small world networks. Electronic Journal of Probability 11, 12341283.

G. Barnard (1963). Contribution to the discussion of Bartlett's paper. J. Roy. Statist. Soc. B, 294.

F. Chung and L. Lu (2001). The diameter of sparse random graphs. Advances in Applied Math. 26, 257–279.

M. Dwass (1957). Modified randomization tests for nonparametric hypotheses. Ann. Math. Stat. 28, 181–187.

A. Fronczak, P. Fronczak and J. A. Holyst (2003). Mean-field theory for clustering coefficients in Barabasi-Albert networks. Phys. Rev. E 68, 046126.

A. Fronczak, P. Fronczak, J. A. Holyst (2004). Average path length in random networks. Phys. Rev. E 70, 056110.

T.A. Gibson and D.A. Goldberg (2011). Improving evolutionary models of protein interaction networks. Bioinformatics 27, 376–382.

P. Hall and D.M. Titterington (1989). The effect of simulation order on level accuracy and power of Monte Carlo tests. J. Roy. Statist. Soc. B, 459.

K. Lin (2007). Motif counts, clustering coefficients, and vertex degrees in models of random networks. D.Phil. dissertation, Oxford.

F. Marriott (1979). Barnard's Monte Carlo tests: how many simulations? Appl. Statist. 28, 75–77.

M.E.J. Newman (2005). Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46, 323351.

M. P. H. Stumpf, C.Wiuf and R. M. May (2005). Subnets of scale-free networks are not scale-free: Sampling properties of networks. Proc Natl Acad Sci U S A. 2005 March 22; 102(12): 4221 - 4224.

6 Statistical inference for networks: nodes and edges.

In the statistical analysis of networks we are often interested in inferring ”local” properties, such as the existence of an edge or the characteristics of a node, from the position of the node, or the potential edge, in the network. Such inference has a long tradition in social network analysis; there logistic regression and loglinear regression models are used. Here we look at two examples which are inspired by social network analysis.

6.1 Example: Inferring characteristics of proteins from their position in a protein interaction network

The most often used method is *Majority Vote* (*Schwikowski et al., 2000*): observe the functional characteristics which the nearest neighbours of the target protein possess, and to select the function which occurs most frequently.

We can improve on this method with ideas from loglinear models to develop a *score* which we can use to classify proteins, say. Here is an elaborate example from *Chen et al. (2007)*.

As characteristics we consider structure (7 categories) and function (24 categories). From the protein-protein interaction network, we build an upcast set of category-category interactions. A category-category interaction is constructed by two characteristic categories from two interacting proteins.

Consider a protein x, within the set of all characteristic categories S, $S(x)$ includes the categories that protein x is classified into. If two proteins x and y interact, the category-category interaction is the edge between two characteristic categories, a and b ($a \in S(x)$, $b \in S(y)$), from each of two proteins (denoted by $a \sim b$). The upcast set of category-category interactions is a collection of all category-category interactions extracted from the protein-protein interaction network, which may be from one or multiple organisms.

Our scoring method is based on the heuristic assumption that the likelihood for a specific category to be observed in the query protein is roughly proportional to the product of the relative frequencies of observing this category in all pairs around the neighbours of a query protein.

The score for the query protein x with annotated neighbours $B(x)$, to be in a specific category a is proportional to the product $C(a, x)$. This is the product of the relative frequencies f of observing category a for all category-category interactions of x's neighbours in the prior data base;

$$C(a, x) = \prod_{\substack{b \in S(n) \\ n \in B(x)}} f(a \sim b),$$

where $f(\, a \sim b)$ is the relative frequency of category-category interaction $\{\, a \sim b\}$ among all category-category interactions.

We define our score $F(a, S(x))$ by

$$F(a, S(x)) := \frac{C(a, x)}{\sum_{k \in S} C(k, x)}.$$

The protein is then predicted to possess the characteristic category, or categories, with the highest score.

This score is derived as an analogy of the likelihood of observing category a in $S(x)$ if all edges in the category interaction network occurred independently. Heuristically this score serves as a measure for the chance of protein x having characteristic a.

The method can be extended to include two or more protein characteristics in the prediction of a specific protein characteristic. Then the category in a category-category interaction is now a vector containing all characteristics of the protein.

The *enhanced* method uses function information as well to predict structure, and also includes structure information when predicting function.

The protein structure is predicted the class with the highest probability.

A function prediction is counted as correct if one of the best three predicted categories is correct.

The accuracies of structure prediction using different methods:

Organism (DIP)	Predicted proteins	M.V.	F.	E. F.
D.Melanogaster	1262	0.35	0.17	**0.44**
C.Elegans	78	0.36	0.37	0.49
S.Cerevisiae	1608	0.39	0.31	**0.54**
E.Coli	150	0.57	0.70	**0.71**
M.Musculus	32	0.72	0.50	0.69
H.Sapiens	273	0.44	0.47	0.71

Underline: where the result outperforms M.V. with statistical significance

The accuracies of function prediction using different methods:

Organism (DIP)	Predicted proteins	M.V.	F.	E. F.
D.M	1275	0.53	0.67	**0.69**
C.E	85	0.38	0.55	0.71
S.C	1618	0.67	0.61	0.67
E.C	154	0.69	0.69	0.70
M.M	32	0.59	0.88	0.81
H.S	274	0.79	0.90	0.89

Underline: where the result outperforms M.V. with statistical significance

6.2 Example: Inferring protein interactions from protein characteristics using the protein interaction network

This time we use ideas from logistic regression to develop a score which we can use to predict and to validate protein interactions, based on the protein characteristics and the protein interaction network.

Here, network structure comes into play. We observe that the protein interaction network has a tendency to form triangles.

6.2.1 Tendency to form triangles

Let a, b, $c \in S$ be three characteristic vectors, and let N be the set of proteins in the protein interaction network. Assume that all of a, b and c are indeed observed in the proteins.

For each type of category-category pair $\{a, c\}$ with a fixed category b, the ratio of conditional probabilities $r_{abc} = \frac{P(a \sim c | a \sim b \sim c)}{P(a \sim c)}$ is then estimated by

$$\hat{r}_{abc} = \frac{\hat{P}(a \sim c | a \sim b \sim c)}{\hat{P}(a \sim c)} = \frac{\hat{P}(a \sim c,\ a \sim b \sim c)}{\hat{P}(a \sim b \sim c \sim a) + \hat{P}(a \sim b \sim c \not\sim a)},$$

where $\hat{P}(a \sim c)$ is the proportion of pairs of proteins x, y in N, with characteristics such that $a \in S(x), b \in S(y)$, which interact, relative to all pairs of proteins with such characteristics.

Note that, in contrast to the triangle rate score, we sum over proteins and keep characteristics fixed. Similarly, $\hat{P}(a \sim b \sim c \sim a)$ is the proportion of protein triplets, with given characteristics, which form a triangle, and $\hat{P}(a \sim b \sim c \not\sim a)$ is the proportion of protein triplets, with given characteristics, which form a line (but not a triangle).

For each organism (protein interaction network), $\bar{r}$ is the average of $\hat{r}_{abc}$ for all $a, b, c \in S$,

$$\bar{r} = \frac{\sum_{a,b,c \in S} \hat{r}_{abc}}{\frac{1}{2}|S|^2(|S|+1)}.$$

If $\bar{r} \ll 1$, the existence of the interacting partner tends to decrease the chance of interaction. If $\bar{r} \gg 1$, the interaction is more likely if two protein have an common interacting partner.

Table 1: Estimates of $\bar{r}$ from characteristic triplets

Type	Organisms	#obs. pairs†	#obs. triples	$\bar{r}$	S.E.	$\bar{r} > 1$
Structure	D.M.	23	94	5.6	2.76	$\star$
	S.C.	26	157	25.2	8.28	$\star$
	E.C.	20	105	9.6	4.14	$\star$
	H.S.	19	74	26.7	20.44	
Function	D.M.	110	534	48.3	67.6	
	S.C.	214	1850	55.9	91.36	
	E.C.	76	494	16.1	9.91	$\star$
	H.S.	60	350	76.8	125.25	

† number of different pairs $\{a, c\}$ forming triples $\{a \sim b \sim c\}$

$\star$ organism showing tendency of formation of triangles, 5% level of significance

The protein interaction network is used to build an upcast set of triplets of characteristic vectors. Here, A, B, C and D denote protein characteristics, whereas different shapes indicate different proteins. A protein may possess more than one characteristic. Our triplets are triangles and lines of three characteristic vectors according to their interacting patterns. A characteristic line is a specific pattern constructed by three vectors with two vector interactions among them. A characteristic triangle is formed by three vectors interacting with each other.

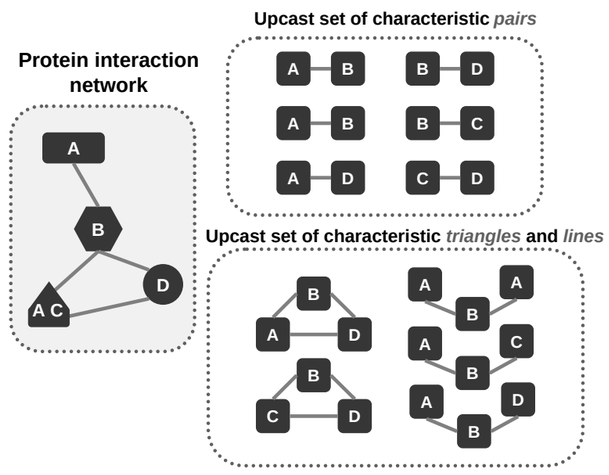


Figure 14: Example for the upcast set. Here we consider a single characteristic, the characteristic vector for a protein is a 1-vector. There are 3 single-category proteins and 1 two-category protein in the protein interaction network, which result in an upcast set of six characteristic pairs $\{A-B, A-B, A-D, B-D, B-C, C-D\}$.

Within the triplet interactions, we assess the odds to observe triangles versus lines around the query protein pair. More formally, let t_{xy} be the total frequency of all characteristic triangles around the query protein pair $\{x, y\}$; denoting by $z \in B(x, y)$ the set of all common neighbours of x and y in the protein interaction network,

$$t_{xy} = \sum_{z \in B(x,y)} \left[\sum_{v_a \in S(x), v_b \in S(y), v_c \in S(z)} f(v_a \sim v_c \sim v_b \sim v_a) \right],$$

where $f(v_a \sim v_c \sim v_b \sim v_a)$ is the frequency of triangle $\{v_a \sim v_c \sim v_b \sim v_a\}$ among all characteristic triangles in the prior data base.

Similarly, l_{xy} is the total frequency of all characteristic lines around the query protein pair $\{x, y\}$. We define the *triangle rate score*, $tri(x, y)$ for the protein pair $\{x, y\}$ as the odds of observing triangles versus lines among triangles and lines in its neighbourhood,

$$tri(x, y) = \frac{t_{xy}}{t_{xy} + l_{xy}}. \tag{1}$$

Heuristically, the higher the triangle rate score is, the higher the chance one would observe an interaction between the query protein pair.

6.2.2 The Receiver Operating Characteristic (ROC) curve

In order to put our scores to work we choose a threshold; all pairs with scores above that threshold would be classified as interacting, while all pairs below that threshold would be classified as non-interacting.

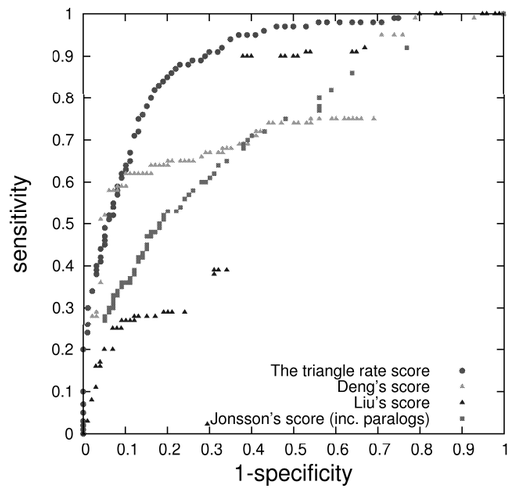
The choice of threshold depends on the desired sensitivity and specificity. The *sensitivity* is the fraction of correct predictions among all predicted positive pairs and the *specificity* is the fraction of correct predictions among all predicted negative pairs.

To assess our scores we use a Receiver Operating Characteristic (ROC) curve, which is a useful technique for examining the performance of a classifier; in our case the score “interacting” or “non-interacting” for a pair of proteins. The curve plots sensitivity against (1 minus specificity). Each point on a ROC curve is generated by selecting a score threshold for a method. We move the cutoff along the range of the score and record different sensitivities and specificities of a method. The closer the curve is to the upper left hand corner (i.e., the larger the area under curve), indicating that sensitivity and specificity are both high, the better the predictive score.

Validation procedure

While we are never completely certain that a prediction is correct, we assume that a positive prediction is correct if it is contained in our gold-standard positive (GSP) set, and that a negative prediction is correct if it is contained in our gold-standard negative (GSN) set. The GSP set is based on 8,250 hand-curated interactions in MIPS complexes catalog (MIPS-GSP). These positive interactions are identified if two proteins are within the same complex and if the interactions are confirmed by various experimental techniques. The set of gold-standard negatives (GSN) are random protein pairs which neither share protein localisation, nor expression nor homologous interaction data.

We have many more gold-standard negatives than positives. The unequal sizes of gold-standard sets may affect the ROC curve; when the cutoff is high, too many gold-standard negatives would cause a rapid increase in true negatives, which would result in artificially high specificity. To avoid this bias, we collect 300 samples of randomly selected pairs from the extensive GSN. Each sample is the same size as our GSP set. Predictions are verified against these 300 reference sets obtained by combining the GSP set and the sample from the GSN set. We test the difference between two ROC curves through a z-test for differences, at 5% significance level.



The ROC curves, 1 minus specificity vs. sensitivity, for predicting yeast protein interactions using domain interaction based approaches (Deng's score and Liu's score), a homology-based approach (Jonsson's score plus paralogs) and our network-based approach (the triangle rate score)

7 Statistical inference for networks: motifs, modules, and communities.

7.1 Motifs

Network *motifs* are small over-represented subgraphs with a fixed number of nodes and with a given topology. Motifs seem to be conserved across species, suggesting a link between protein evolution and topological features of the protein interaction network.

Here over-representation is judged in comparison to a statistical null model. Most commonly, significantly over-represented motifs are detected based on a conditional uniform graph test which preserve some characteristics of the network; for example keeping the degree distribution fixed, which results in a configuration model.

Picard et al. calculate the mean and the variance for motifs on 3 and 4 nodes in undirected graphs under the models

1. Bernoulli random graph (ER)

2. Random graphs with fixed degree sequence (FDD) (here they estimate the mean and standard deviation from simulations; and they also consider a version with fixed expected degrees)

3. Erdős-Renyi mixture models (ERMG).

Here are some examples of their results. For the mean:

motif	obs	ER	FDD	ERMG
H.Pylo				
2-stars	14,113	5704.08	14,113	13,602.97
triangles	75	10.85	66.91	52.82
3-stars	112,490	7676.83	112,490	93,741.08
E.coli				
2-stars	248,093	52,774.79	248,093	243,846.93
triangles	11,368	72.47	3579.49	10,221.17
3-stars	6,425.495	133,050.00	5,772,005.15	1,537,740.00

Note that in FDD the degree distribution is fixed to equal the degree sequence in the network.

For the standard deviation:

motif	ER	FDD	ERMG
H.Pylo			
2-stars	311.08	0	2659.18
triangles	3.40	7.80	20.41
3-stars	681.76	0	27,039.88
E.coli			
2-stars	1281.87	0	51,676.68
triangles	8.90	68.58	3041.98
3-stars	5089.62	0	1,672.086.51

The expectation and variance strongly depend on the model we choose for comparison.

We see that the variance is very different from the mean, so a Poisson approximation is not adequate. Also the counts are relatively low compared to the potential number of structures on the networks, thus a normal approximation (z-score) is not appropriate. Instead we use a *Polya-Aeppli distribution*, which is a special case of a compound Poisson distribution. It is obtained when the clump size has a geometric distribution. If the probability that a clump has size k is

$$Prob(k) = a^{k-1}(1-a),$$

and if the number of clumps is Poisson distributed with parameter λ, then we can write down the Polya-Aeppli distribution for the count W,

$$Prob(W = w) = e^{-\lambda}a^w \sum_{c=1}^{w} \frac{1}{c!}\binom{w-1}{c-1}\left(\frac{\lambda(1-a)}{a}\right)^c \text{ if } w = 1, 2, \ldots,$$

and

$$Prob(W = 0) = e^{-\lambda}.$$

The parameters of the Polya-Aeppli distribution can be calculated from the first two moments:

$$a = \frac{var - mean}{var + mean}$$

and

$$\lambda = (1 - a)mean.$$

The Polya-Aeppli distribution is a better fit to the count distribution, compared to the normal distribution; this result is fairly consistent across motifs and across networks. In particular the normal approximation can lead to false positive results: some motifs may be thought as being exceptional while they are not.

When assessing the significance of two or more motifs, their dependence has to be taken into account, or else we resort to the Bonferroni correction, dividing the significance level of our tests by the number of tests carried out and using this as new significance level for each individual test.

But we need to find out whether the null model is suitable for subgraph counts before reaching statistical conclusions.

Here is a list of candidate small subgraphs again, see *Przulj 2006*, where they are called *graphlets*.

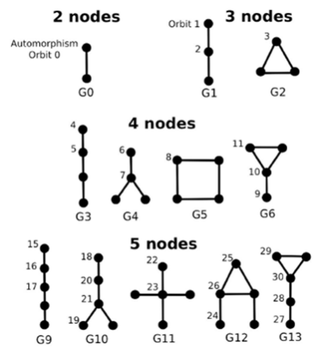
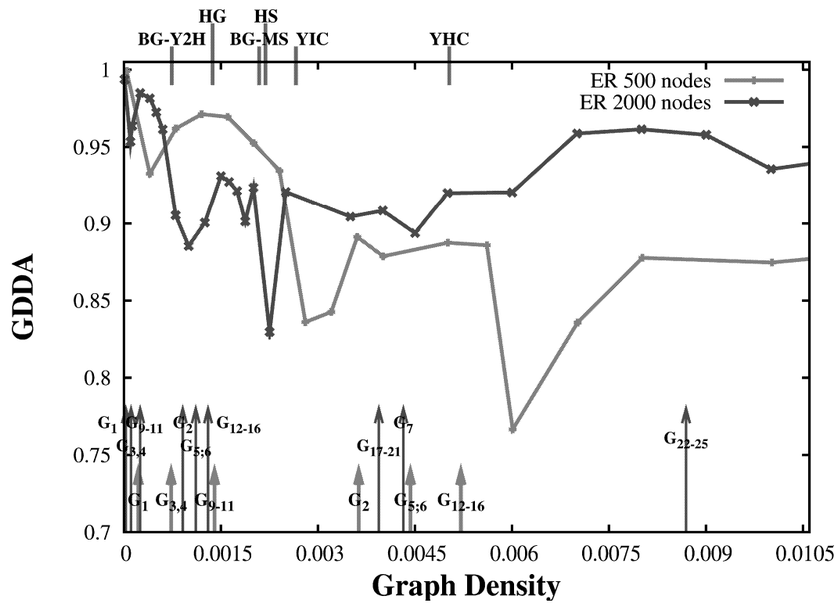


Figure 15: Small subgraphs

We can compare networks based on their combined graphlet counts using the Graphlet Distribution Degree Agreement (GDDA), a score based on graphlets and orbits (*Przulj et al. 2004*). But the threshold behaviour comes into play (*Rito et al., 2010*), as follows.


HG
HS
BG-Y2H
BG-MS
YIC
YHC
ER 500 nodes
ER 2000 nodes
GDDA
Graph Density
1
0.95
0.9
0.85
0.8
0.75
0.7
0
0.0015
0.003
0.0045
0.006
0.0075
0.009
0.0105
G1
G9-11
G2
G12-16
G3,4
G5;6
G7
G17-21
G22-25

Why would protein interaction networks operate near the threshold of the appearance of small graphlets? We conjecture:

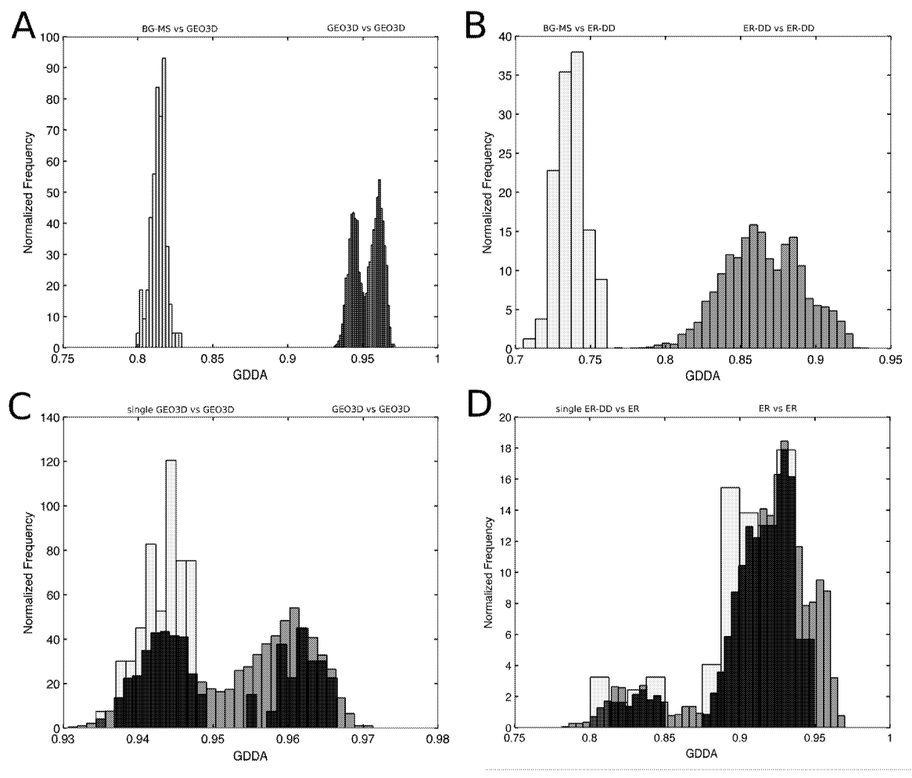
A small number of potential interactions, so that interactions are specific, makes the network efficient.

Some redundancy in the network makes the network robust against small errors.

However the comparison to a Bernoulli random graph may not be correct.

To address how to interpret the output from a graph comparison based on GDDA, for the ER model and the GEO3D model, graphs of 500, 1000 and 2000 nodes with increasing graph density were generated using. The graphs were subsequently used as query networks in the software and compared with 50 networks of the same model, to ascertain typical GDDA scores if the model is correct.

We compare this with using PINs as input networks. In the figure each value represents the average agreement of 50 networks. The graph densities of the PPI networks considered are indicated on the top x axis. The graph density values where the expected number of occurrences of a specific graphlet is approximately equal to 1, for an ER graph with 500 and 2000 nodes are respectively indicated by the short and long arrows along the x axis.


A
BG-MS vs GEO3D
GEO3D vs GEO3D
Normalized Frequency
GDDA
B
BG-MS vs ER-DD
ER-DD vs ER-DD
Normalized Frequency
GDDA
C
single GEO3D vs GEO3D
GEO3D vs GEO3D
Normalized Frequency
GDDA
D
single ER-DD vs ER
ER vs ER
Normalized Frequency
GDDA

None of the models under investigation here fits. We also tried preferential attachment and gene duplication networks. In *Rito et al. (2012)* we find that the Yeast PIN is highly heterogeneous, which makes model fitting very difficult.

But remember - the model depends on the research question; here we were interested in finding motifs.

7.2 Modules and communities

Finding modules, or communities, which make up the network, has been of interest not only for social networks. Statistically speaking, we would like to apply a *clustering method* to find out more about the structure of the network. There is an abundance of clustering methods available.

A much used algorithmic approach is the algorithm by Newman and Girvan, see for example *Newman and Girvan (2004)*. Recall that the betweenness of an edge is defined to be the number of shortest paths between node pairs that run along the edge in question, summed over all node pairs. The algorithm of Girvan and Newman then involves simply calculating the betweenness of all edges in the network and removing the one with highest betweenness, and repeating this process until no edges remain. If two or more edges tie for highest betweenness then one can either choose one at random to remove, or simultaneously remove all of them.

As a guidance to how many communities a network should be split into, they use the *modularity*. For a division with g groups, define a $g \times g$ matrix $\mathbf{e}$ whose component e_{ij} is the fraction of edges in the original network that connect nodes in group i to those in group j. Then the modularity is defined to be

$$Q = \sum_i e_{i,i} - \sum_{i,j,k} e_{i,j} e_{k,i},$$

the fraction of all edges that lie within communities minus the expected value of the same quantity in a graph where the nodes have the same degrees but edges are placed at random. A value of $Q = 0$ indicates that the community is no stronger than would be expected by random shuffling.

Unfortunately the Newman-Girvan algorithm does not provide a measure for statistical significance. The approach by *Handcock et al (2007)* in contrast can assess statistical significance of the clusters, for an Erdős-Renyi mixture model. They propose a new model, the latent position cluster model, under which the probability of a edge between two nodes depends on the distance between them in an unobserved Euclidean space, and the nodes' locations in the latent space arise from a mixture of distributions, each corresponding to a cluster. They propose two estimation methods: a two-stage maximum likelihood method and a fully Bayesian method that uses Markov chain Monte Carlo sampling. The former is quicker and simpler, but the latter performs better.

Lewis et al. (2010) show that there is no one scale of interest in the community structure of the yeast protein interaction network, and that almost all proteins lie in a functionally homogeneous community at some scale. They are able to identify a range of resolution parameters that yield the most functionally coherent communities. Also they trace the community membership of a protein through multiple scales.

8 Some topics we have not covered

Many networks, such as food webs, have a *hierarchical* structure.

Networks may also be *dynamic*; our data are snapshots in time. The yeast interactome appears to exhibit organized modularity where a small proportion of proteins the 'hubs' interact with many partners. These hubs fall into one of two categories: 'party' hubs, which interact with most of their partners simultaneously, and 'date' hubs, which bind different partners at different locations and times. The biological role of topological hubs may vary depending upon the timing and location of the interactions they mediate, see *Han et al. (2004)*.

Following up from network motifs, it is interesting to look at nodes which are ”special” in networks, leading to *roles* in networks. These could be nodes with high degree, so-called *hubs*, or these could be nodes which have low degree but high between-ness, indicating that they may link fairly separate part of the network, or modules. Depending on the scientific question one may like to identify other roles.

We can apply a whole range of network measures to analyse PIN data, but should bear in mind:

Nothing in Biology Makes Sense Except in the Light of Evolution.

(Theodosius Dobzhansky, 1900-1975)

Further references

C. J. Anderson, S. Wasserman, B.Crouch (1999). A p* primer: logit models for social networks. Social Networks 21, 37–66.

J-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G.F. Berriz, L. V. Zhang, D. Dupuy,A. J. M Walhout, M. E. Cusick, F. P. Roth, and M. Vidal (2004). Evidence for dynamically organized modularity in the yeast protein-protein interaction network. Nature, 430, 8893.

Handcock, M.S., Raftery, A.E., and Tantrum, J.M. 2007. Model-based clustering for social networks. Journal of the Royal Statistical Society. **170**:122.

P. W. Holland, S. Leinhardt (1981). An exponential family of probability distributions for directed graphs (with discussion). Journal of the American Statistical Association, 76, 33-65.

F. Kepes ed. (2007). Biological Networks. Complex Systems and

Interdisciplinary Science Vol. 3. World Scientific, Singapore.

A. Lewis, N. S. Jones, M. A. Porter, and C. M. Deane (2010). The function of communities in protein interaction networks at multiple scales. BMC Systems Biology 4, 100.

M.E.J. Newman and M. Girvan, (2004). Finding and evaluating community structure in networks. Physical Review E. 69 026113.

G. Palla, A.-L. Barabasi, T. Vicsek (2007). Quantifying social group evolution. Nature 446, 664–667.

F. Picard, J.-J. Daudin, M. Koskas, S. Schbath, S. Robin (2008). Assessing the exceptionality of network motifs. Journal of Computational Biology 15, 1–20.

T. Rito, Z. Wang, G. Reinert and C.M. Deane (2010). How threshold behaviour affects the use of subgraphs for network comparison. Bioinformatics, 26 (18), vi611-7.

T. Rito, C.M. Deane. and G. Reinert (2012). The importance of age and high degree, in protein-protein interaction networks. Journal of Computational Biology, to appear.

G.L. Robins, T.A.B. Snijders, P. Wang, M. Handcock, P. Pattison (2007). Recent developments in exponential random graph (p*) models for social networks. Social Networks 29, 192-215 (2007). http://www.stats.ox.ac.uk/ snijders/siena/RobinsSnijdersWangHandcockPattison2007.pdf

G.L. Robins, P. Pattison, Y. Kalisha, D. Lushera (2007). An introduction to exponential random graph (p*) models for social networks. Social Networks 29, 173-191.

Addendum

Summaries based on spectral properties of the adjacency matrix.

If λ_i are the eigenvalues of the adjacency matrix A, then the spectral density of the graph is defined as

$$\rho(\lambda) = \frac{1}{n} \sum_i \delta(\lambda - \lambda_i),$$

where $\delta(x)$ is the delta function. For Bernoulli random graphs, if p is constant as $n \to \infty$, then $\rho(\lambda)$ converges to a semicircle.

The eigenvalues can be used to compute the kth moments,

$$M_k = \frac{1}{n}\sum_i (\lambda_i)^k = \frac{1}{n}\sum_{i_1,i_2,\ldots,i_k} a_{i_1,i_2}a_{i_2,i_3}\cdots a_{i_{k-1},i_k}.$$

The quantity nM_k is the number of paths returning to the same node in the graph, passing through k edges, where these paths may contain nodes that were already visited.

Because in a tree-like graph a return path is only possible going back through already visited nodes, the presence of odd moments is an indicator for the presence of cycles in the graph.

The *subgraph centrality*

$$Sc_i = \sum_{k=0}^{\infty} \frac{(A^k)_{i,i}}{k!}$$

measures the ”centrality” of a node based on the number of subgraphs in which the node takes part. It can be computed as

$$Sc_i = \sum_{j=1}^{n} v_j(i)^2 e^{\lambda_i},$$

where $v_j(i)$ is the ith element of the jth eigenvector.

Entropy-type summaries

The structure of a network is related to its reliability and speed of information propagation. If a random walk starts on node i going to node j, the probability that it goes through a given shortest path $\pi(i,j)$ between these nodes is

$$\mathcal{P}(\pi(i,j)) = \frac{1}{d(i)} \sum_{b \in \mathcal{N}(\pi(i,j))} \frac{1}{d(b)-1},$$

where $d(i)$ is the degree of node i, and $\mathcal{N}(\pi(i,j))$ is the set of nodes in the path $\pi(i,j)$ excluding i and j.

The *search information* is the total information needed to identify one of all the shortest paths between i and j and is given by

$$S(i,j) = -\log_2 \sum_{\pi(i,j)} \mathcal{P}(\pi(i,j)).$$

Similarly, an entropy can be defined based on the predictability of a message flow.

Further reading:

L. da F. Costa, F.A. Rodrigues, P.R. Villas Boas, G. Travieso (2007). Characterization of complex networks: a survey of measurements. Advances in Physics 56, Issue 1 January 2007, 167 - 242.