Analysis of biological networks: Random Models*

Lecturer: Roded Sharan Scribe: Ori Folger and Keren Yizhak

Lecture 2, March 11, 2009

In this lecture we discuss fundamental concepts in network analysis and present random network models.

1 Concepts

We begin with introducing important concepts in network analysis.

1.1 Random (ER) graphs

The *Erdös-Rényi (ER) random graphs* model, also called simply *random graphs*, was presented by Erdös and Rényi [4] in the 1950s and 1960s. Erdös and Rényi characterized random graphs and showed that many of the properties of such networks can be calculated analytically.

Construction of an ER random graph with parameter $0 \leq p \leq 1$ and N nodes is by connecting every pair of nodes with probability p. Note that the resulting graph is a simple graph.

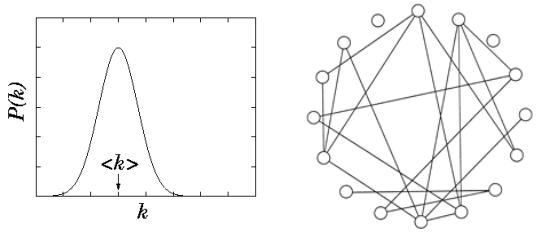

Figure 1: Source [4]. Degree distribution of a random graph, and an example of such a graph.

1.2 Scale-free networks

Many complex networks found in the real world feature an important property - most nodes have a few links to other nodes, but a small number of nodes are highly connected and have a huge number of links to other nodes. This leads to the observation that these networks do not have nodes with a typical number of neighbors, and in this sense these networks are *scale-free*. The modern investigation of scale-free networks began with Barabási and Albert[2].

In order to analyze these networks, we require several definitions. The *degree sequence* for a graph is defined as the vector $(d(1), d(2), \ldots, d(n))$ holding the degree information $d(v)$ of each node v in the graph.

*Based on a scribe by: Elena Kyanovsky, David Hadas, Lior Gavish and Ory Samorodnitzky.

The *degree distribution* for a graph, denoted by $P(k)$, is defined to be the fraction of nodes in the graph with a degree k. The degree distribution can be calculated thus:

$$P(k) = \frac{|\{v|d(v) = k\}|}{N} \tag{1}$$

where $d(v)$ is the degree of node v and N is the number of nodes in the graph. A network can be characterized in terms of its degree distribution. A directed graph has two separate degree distributions, namely the *in-degree distribution* and the *out-degree distribution*.

The *average degree* in a graph is denoted $d \equiv \sum_k kP(k)$. Note that the number of edges in the graph is given by $m = \frac{Nd}{2}$.

The degree distribution of several real networks is shown in Figure 2. All of these networks display a *power-law* degree distribution, which is defined by:

$$P(k) \propto k^{-c}, k \neq 0, c > 1 \tag{2}$$

The constraint $c > 1$ ensures the proper convergence of the total probability, i.e. the sum $\sum_{k=1}^{\infty} P(k)$. In typical networks, c takes values in the range $2 \leq c \leq 3$.

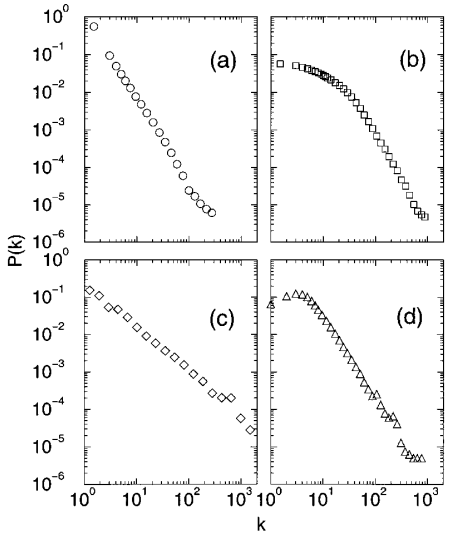

Figure 2: Source: [1]. The degree distributions of several real networks. The graphs are log-log graphs where both axes use a logarithmic scale. (a) Internet router connections; (b) movie actor collaborations; (c) co-authorship network of high-energy physicists; (d) co-authorship network of neuroscientists. Note how the graphs are near linear, indicating a power-law function.

And just as in the real networks, in a power-law degree distribution there are many nodes with low degrees and a small number of nodes with high degrees. The high degree nodes are highly important to the network connectivity and serve as *hubs*.

The term scale-free is also justified by the power-law degree distribution, since it can be scaled without altering the distribution. If we denote the distribution by $p(k)$, scaling the distribution by a factor a results in $p(ak) = g(a)p(k)$, which means that the distribution looks the same in every range of k. Thus, the power-law distribution has no natural scale and is scale-invariant (in fact, this is the only distribution with this property).

1.3 Clustering coefficient

An important measure of network cohesiveness is the *clustering coefficient*. As shown in Watts and Strogatz [10], in many complex networks we find *clusters* which are subsets of the network that display a high level of inner connectivity. The clustering coefficient measures the degree of clustering of a typical node's neighborhood. It is defined as the likelihood that any two nodes with a common neighbor are themselves connected. For example, in a friends network, it is the likelihood that two persons who have a common acquaintance also know each other. The clustering coefficient is a *local property* which describes the network structure of nodes which are close to each other.

To calculate the clustering coefficient, we first define the *node clustering coefficient*, $C(v)$, of a node v, as the proportion of pairs of connected neighbors of v out of the total number of pairs of neighbors of v. Every such pair of neighbors, for example u and w, forms a *triangle* with v, a cycle of length 3 (see figure 3). Let $t(v)$ be the number of triangles containing v. Then $C(v)$ can be calculated by

$$C(v) = \frac{t(v)}{\frac{1}{2}d(v)(d(v)-1)} \tag{3}$$

Where we define for $d(v) = 0$ and for $d(v) = 1$ that $C(v) = 0$.

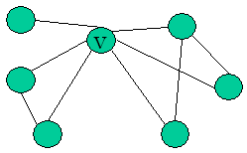

Figure 3: Calculating the clustering coefficient of a node. The node v has degree $d = 6$, and participates in $t(v) = 3$ triangles. Therefore, $C(v) = \frac{3}{\frac{1}{2}*6*5} = \frac{1}{5}$.

The *network clustering coefficient*, C, is defined as the average of $C(v)$ over all nodes in the network:

$$C = \frac{1}{N}\sum_{v} C(v) \tag{4}$$

Note that both $C(v)$ and C are between 0 and 1. When $C = 1$, all possible links exist and the graph is a *clique*. On the other extreme, when $C = 0$ there are no triangles in the graph (See figure 4).

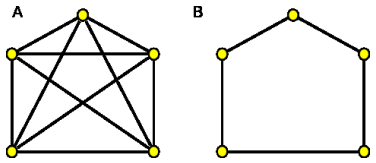

Figure 4: A. This 5 nodes clique has a clustering coefficient $C = 1$; B. This 5 nodes cycle has a clustering coefficient $C = 0$, as there are no triangles.

Przulj and colleagues [8] presented a comparison between the clustering coefficients of real networks and the analytical results from calculating the respective clustering coefficients of random (ER) networks with the same average degree. See Figure 5. As can be seen, real world clustering coefficients are considerably higher than those expected using random networks. In real world networks clustering is significant. For

Network	n	z	C measured	C for random graph
Internet [153]	6,374	3.8	0.24	0.00060
World Wide Web (sites) [2]	153,127	35.2	0.11	0.00023
power grid [192]	4,941	2.7	0.080	0.00054
biology collaborations [140]	1,520,251	15.5	0.081	0.000010
mathematics collaborations [141]	253,339	3.9	0.15	0.000015
film actor collaborations [149]	449,913	113.4	0.20	0.00025
company directors [149]	7,673	14.4	0.59	0.0019
word co-occurrence [90]	460,902	70.1	0.44	0.00015
neural network [192]	282	14.0	0.28	0.049
metabolic network [69]	315	28.3	0.59	0.090
food web [138]	134	8.7	0.22	0.065

Figure 5: Source: [8]. Number of nodes n, average degree z and clustering coefficient C for a number of different networks.

example, in the Internet network, the chance of two neighbors of a node to also be connected to each other is about 24% compared to the analytical result based on a random network which yields 0.06%.

It is evident from the presented table that the measured clustering coefficient is significantly higher than the clustering coefficient of comparable random networks.

1.4 Distance related measures

While a node's degree and its clustering coefficient are local properties which depend only on the structure of the node's neighborhood, there are also *global properties* which consider the entire network's structure.

The *distance* between two nodes is the length of the shortest path between them. This leads to the definition of the important network property of *average distance*, l, which is the average distance over all *connected* node pairs.

Another commonly used measure is the *betweeness* of a node, $b(v)$, which is the number of shortest paths betweens pair of nodes which pass through v. This is an indication of a node's centrality in the network.

Lastly, the *giant component* size, the size of the largest connected component in the network, is sometimes used as well.

1.5 Small world

We can find artificial examples for networks with various average distance values. For example, a clique has an average distance of 1, a tree's average distance is $\propto logN$ and a ring or a path has an average distance $\propto N$.

One of the phenomena of real networks is that the average distance through the network from one node to another is small compared to the network size. In social networks this effect is known as the small-world effect. We will analyze this effect on a random graph example (see 2.2.4).

2 Random models

We want to construct a random model which is similar to a real network in terms of:

1. Power-law degree distribution

	network	type	n	m	z	ℓ
social	film actors	undirected	449 913	25 516 482	113.43	3.48
	company directors	undirected	7 673	55 392	14.44	4.60
	math coauthorship	undirected	253 339	496 489	3.92	7.57
	physics coauthorship	undirected	52 909	245 300	9.27	6.19
	biology coauthorship	undirected	1 520 251	11 803 064	15.53	4.92
	telephone call graph	undirected	47 000 000	80 000 000	3.16	
	email messages	directed	59 912	86 300	1.44	4.95
	email address books	directed	16 881	57 029	3.38	5.22
	student relationships	undirected	573	477	1.66	16.01
	sexual contacts	undirected	2 810			
information	WWW `nd.edu`	directed	269 504	1 497 135	5.55	11.27
	WWW Altavista	directed	203 549 046	2 130 000 000	10.46	16.18
	citation network	directed	783 339	6 716 198	8.57	
	Roget's Thesaurus	directed	1 022	5 103	4.99	4.87
	word co-occurrence	undirected	460 902	17 000 000	70.13	

Figure 6: Source: [7]. Real networks measurements. n is the number of nodes; m is the number of links; z is the average node degree; l is the average distance. Note how the average distance l is very small even when the number of nodes, n, is very large.

2. Clustering coefficient
3. Average distance.

2.1 Motivation: Identifying network motifs

Network motifs, first studied by Uri Alon [6], are defined to be small sets of nodes that are connected in a specific wiring diagram and appear in the network in a significantly higher frequency than we would expect by random. These basic local elements/structures of the network usually have a functional role in the network that promotes their ubiquitous nature.

One such common functional element that can be seen in many biological (and other) networks is the *feed-forward-loop* (See figure 7). In transcription networks this 3-node sub-graph is quite common, and has a role in providing robustness and resistance to noise in gene regulation: Suppose that we want gene Z to be expressed only when transcription factor X is expressed. We would not like a momentary fluctuation of X's values to affect Z's expression, we only want Z to be expressed if X's levels are up for a significant period of time. For that purpose we add transcription factor Y. In order for Z to be expressed, both transcription factors, X and Y, have to be expressed. When X is first expressed it causes expression of Y. Y's levels gradually go up, and only when there are sufficient levels of both X and Y, Z is being expressed. On the other hand, when X's signal shuts down we want the system to immediately respond by shutting down Z's expression. This goal is achieved by the direct connection between X and Z: as soon as X's levels decline Z's expression is stopped (see Figure 8). This observation is explained by the following equations (5 and 6). Y is the protein concentrations and F is a threshold function. T is the activation thresholds (T_x = 0.1, T_y = 0.5, T_z = 0.8). We therefore get that $F(X,T)$ is the threshold on X activation. $F(X,T) = 0$ if $X < T$ and 1 if $X \geq T$. The same goes for Z concentration, which depends on X and Y concentrations.

$$\frac{dY}{dt} = F(X, T_y) - Y \tag{5}$$

$$\frac{dZ}{dt} = F(X, T_y)F(Y, T_z) - Z \tag{6}$$

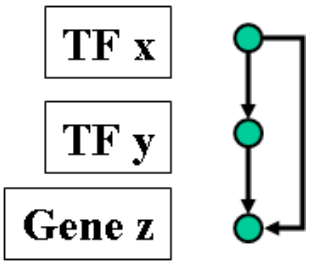

Figure 7: A feed forward loop motif in gene regulation networks.

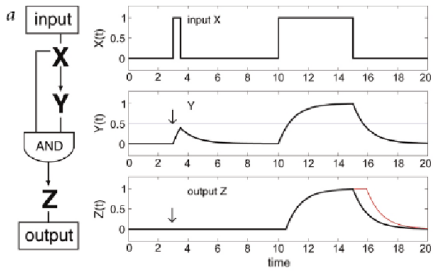

Figure 8: Concentration change over time, for the two TFs X and Y and for gene Z. The red line in the bottom graph indicates a moderate decline in Z's expression, when there isn't a direct edge between X and Z.

2.2 Generalized Random Graphs

As discussed earlier, the degree of many real networks follows a power-law distribution. However, the random graph model we have discussed so far, does not reproduce such a distribution. In 1978, Rodney and Canfield [3] suggested to improve the approximation of real networks by controlling the degree distribution of the network. The suggested *generalized random graph* model, creates a graph based on a given degree sequence. That is, it uniformly picks a graph in which node i has degree k_i, for a given $\{k_i\}$ set. This allows us to consider graphs with a specific degree distribution, and in particular graphs with power-law distributions. This model is helpful in analyzing the behavior of real networks, as will be seen below.

2.2.1 Construction using a matching algorithm

Let's consider how a generalized random network can be constructed. Rodney and Canfield [3] suggested a matching algorithm that includes random assignments of vertices to edges such that a vertex v is assigned $d(v)$ times, where $d(v)$ is the degree of vertex v. The algorithm is as follows:

1. Prepare $d(v)$ copies of each vertex v.
2. Randomly assign all the copies to edges. This done by creating a two column table where each column contains all the copies in a random order and creating a link for every row.
3. Repeat steps 1-2 if the resulting graph is not simple (i.e. contains cycles or duplicate edges).

We consider the different copies of each vertex to be the same and therefore may use different choices during the construction algorithm and still produce the same graph topology. In fact, each vertex that has

$d(v)$ copies, contributes $d(v)!$ different construction choices. Since copies are indistinguishable, if we have m edges, each graph topology can be generated in $2^m(\prod_v d(v)!)$ different ways with an equal probability and therefore the construction algorithm is uniform.

As noted above, the matching algorithm may introduce graphs with unity loops and double edges, and we reject such graphs and repeat the matching algorithm until we come up with a simple graph. However, this works only for graphs with low vertex degree. If we choose a graph with higher vertex degrees, we may fail to come up with a simple generalized random graph within a reasonable amount of trials.

2.2.2 Construction using a switching algorithm

In 2004, Milo et al. [5] suggested another method to generate a generalized random graph. This method picks any graph which conforms to the required degree distribution (i.e. node i has degree k_i, for a given $\{k_i\}$ set) and change it by performing a long series of random edge crosses, until it becomes a generalized random graph (we may even start with our original graph). Using this algorithm, we randomly select two pairs of connected nodes and cross their edges : for example $u-v, s-t \Longrightarrow u-s, v-t$ or $u-t, v-s$ (See Figure 9). The switching is legal only if the resulting graph remains simple. Therefore, if the resulting graph includes unit loops or double edges, we will skip this iteration. It was found empirically that each vertex should be switched on average $100|E|$ times to approximate a generalized random graph.

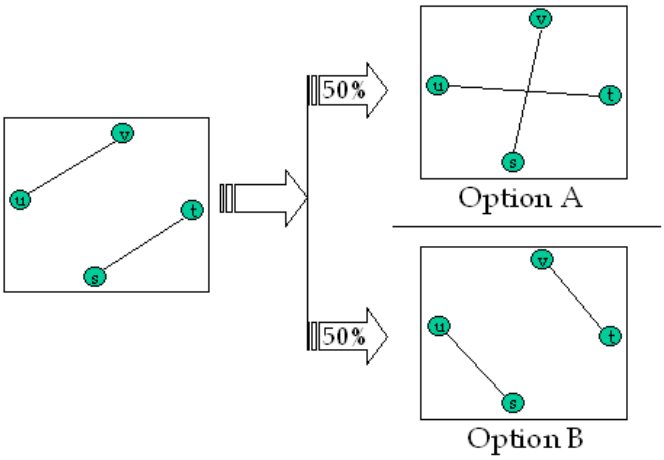

Figure 9: Source [5]. The Switching Algorithm

The procedure :

1. Randomly select two links $(u, v), (s, t)$.
2. Choose the crossing direction by a 50% chance: either edges $(u, s), (v, t)$ or edges $(u, t), (s, v)$.
3. Check if the graph that will result if the crossing take place is a simple graph, if so, perform the crossing.
4. Repeat steps 1-3 for $100|E|$ successful crossings.

2.2.3 Is this a good model?

We want to know if a graph constructed using the switching algorithm is "good" in terms of power-law degree distribution, average distance and clustering coefficient. Its clear that this graph has the same degree distribution as the original graph. Let's check for the other two parameters.

2.2.4 Average distance

Figure 10 shows that average distance estimations for generalized graphs are pretty close to the corresponding measures in real networks.

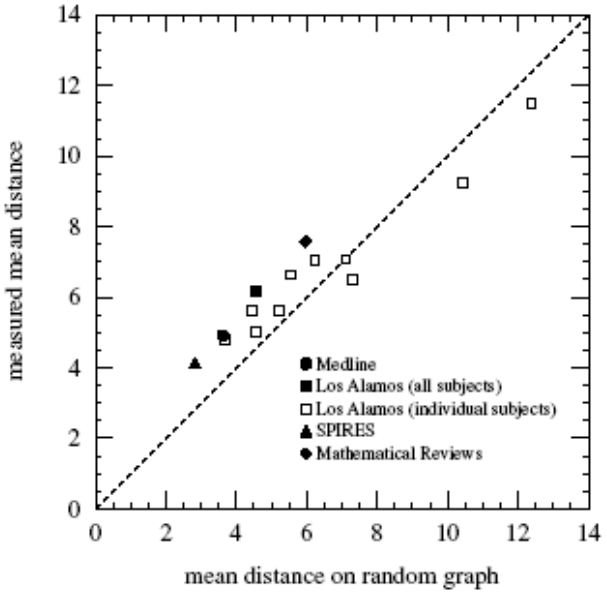

Figure 10: Source [7]. Estimated vs. real average distance for several real networks.

2.2.5 Clustering Coefficient

We now want to evaluate the Clustering Coefficient of this model. We will fix a vertex v and look at its i-th neighbor. This neighbor has $k(i)$ 'out-going' edges distributed according to $q_k(i) \sim [k(i)+1]P(k(i)+1)$. The probability that two of its neighbors (i and j) are connected is: $\frac{k(i)k(j)}{Nd}$. Taking an average for all i, j we obtain :

$$C = \frac{1}{Nd} \sum_{k(i),k(j)=1}^{\infty} q_k(i) q_k(j) k(i) k(j) = \frac{1}{Nd} \left(\sum_k q_k k \right)^2 = \frac{d}{N} \left(\frac{d_2 - d}{d^2} \right)^2$$

where $\frac{d}{N}$ is a cluster coefficient for ER graph and $d_2 = \sum k^2 P(k)$ is the second moment of the degree distribution. Let us analyze the $(\frac{d_2-d}{d^2})$ factor for power-law network: This coefficient is dependent on the second moment d_2. d_2 can be very large and it is finite, if and only if $c > 2$. Knowing that c is typically at range [2,3] for power-law networks, we can estimate d_2 for $c < 3$.

$$d_2 \approx k_{max}^{3-c} \approx N^{\frac{3-c}{c-1}} \tag{7}$$

From cluster coefficient formula we obtain:

$$C \approx N^{-\beta}, \beta = \frac{3c-7}{c-1} \tag{8}$$

That leads to the following conclusions:

- When $c = 7/3$ there is a phase transition.
- When $c > 7/3$ - C tends to 0 with increasing of N
- When $c < 7/3$ - C increases with graph size

Thus, for a good random graph network model we expect cluster coefficient to be $2 < c < 3$.

2.2.6 Summary

We have given an introduction to the use of random graphs as models of real-world networks and evaluated different properties for such a model. Most scale-free networks have cluster coefficient between $2 < c < 3$ and a giant component emerges when $c > 3.4788$ (see [7]). Though there are certain drawbacks that differ random graphs from real networks, this model is very popular and is the best studied model.

2.3 A biologically motivated scale-free random model

The models described so far focus on the topology of the network and not on its dynamics. Some of these models predict scale-free features and some of them do not; the random graph model do not reproduce a power-law degree distribution. However, we can construct generalized random graphs that have a power-law degree distribution, but not proper clustering coefficients.

All of these models leave open an important question: what is the mechanism responsible for the *emergence* of scale-free networks? The scale-free model, suggested by Barabási and Albert [2], tries to answer this question and is based on two observations:

- Real networks are hardly fixed, and they evolve and grow through time.
- In real networks, connections are not uniform and some vertices tend to have more connections as new vertices prefer to attach themselves to them. E.g., in a citation network (where each vertex is an article and there is a directed edge from an article to another article iff it refers to it) a popular article tends to have more references from new articles as compared to non-popular ones.

Barabási and Albert introduced a model that is based upon these two definitions:

1. **Growth** - The network starts from m_0 vertices and it grows as new nodes are added. Each added vertex has degree m ($m_0 \geq m$). The fixed degree is a reasonable assumption for some networks, such as the scientific citation network. In that network, most articles have a fixed number of references which does not vary much. In case the degree for the new vertex is fixed (m), the average degree in the graph after growth is $2m$, as each of the new m edges is counted for two vertices (after growth, the initial degrees of the first m_0 vertices can be ignored, as $m_0 \ll N$).
2. **Preferential attachment** - The probability that a new vertex will connect to an existing vertex, is not uniform. Rather, it is proportional to its degree. I.e, if $d(v)$ is the degree of vertex v, then:

$$P(u\ connects\ to\ v) = \frac{d(v)}{\sum d(v)} \tag{9}$$

Preferential attachment can be explained in biological networks by gene duplication. For example, we take the protein-protein interaction (PPI) network, where each protein is a vertex, and two proteins are connected iff they can interact with each other. During evolution, a gene may be duplicated, leading to a duplication of the protein it encodes. Assuming duplication preserves interactions, the new duplicated protein will have interactions with all the neighbors of the original protein. This leads to preferential attachment, as more connected proteins will gain more new neighbors.

2.3.1 Empirical evidences

Empirical evidences for this model should be focused on the preferential attachment assumption, as the growth assumption is obvious in real networks. This can be obtained when analyzing networks in which joining times for new vertices are known. Such networks are the citation networks or the actors' collaboration network (in which each vertex is an actor, and two actors are connected if they acted in the same film). Thus, we can try to estimate the probability that new edges will be assigned to a vertex, as a function of its connectivity and draw a cumulative distribution for that probability. In preferential attachment we presume linear attachment; hence we predict a linear graph for that function. Figure 11 shows such graphs for several real networks. It can be seen that indeed the attachment is linear, and differs from the prediction of models with no preferential attachment mechanism.

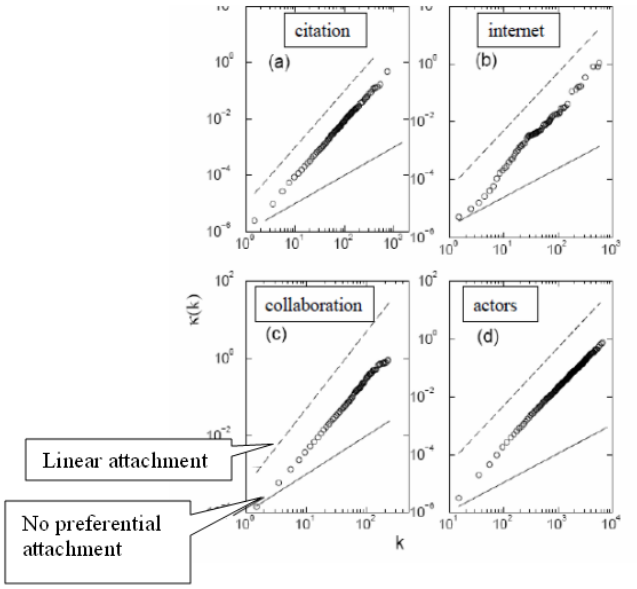

Figure 11: Source [1]. Cumulative preferential attachment for (a) the citation network; (b) the Internet; (c) the neuroscience scientific collaboration network; (d) the actor collaboration network. In all panels the dashed line corresponds to linear preferential attachment, and the solid line to no preferential attachment. The power-law can be seen clearly for preferential attachment models.

2.3.2 Model analysis - Power-law degree distribution

In order to analyze the degree distribution in the network, we should see how the dynamics of the network affect it. Each new vertex has m edges connected to the existing vertices. In order to find the probability to connect to any given k-degree vertex, we need to multiply Eq. 9 by $P(k)|V|$, where $P(k)$ is the probability to a k-degree vertex.

$$P(k)|V|\frac{k}{2|E|} = \frac{kP(k)}{2\frac{|E|}{|V|}} \tag{10}$$

The expression in the denominator is the definition of the mean degree and is equal to $2m$ (as explained

above). We therefore get:

$$\frac{kP(k)}{2m} \tag{11}$$

As each new vertex has m edges, in order to estimate the number of vertices with k-degree that will gain new edges from the new vertex, we should multiply the expression in (11) by m. Thus, we get:

$$\frac{mkP(k)}{2m} \quad = \quad \frac{kP(k)}{2} \tag{12}$$

which is independent of m. From the above expression we can learn about the dynamics of the degree distribution in the network. We shall now investigate the degree distribution in the transition between a network with N vertices to a network of $N+1$ vertices. Denote $P(k,N)$ as the probability for a k-degree vertex in a network with N vertices. When a new vertex is added, $kp(k)/2$ vertices with k degree will gain another edge and will become $(k+1)$-degree vertices, and the new network will contain $N+1$ vertices. If we take $k=m$ (the minimal degree), we deduce that in the transition from N to $N+1$ vertices, $mP(m,N)/2$ vertices become with degree $m+1$, and one m-degree vertex is added - the new vertex. Therefore, for $k=m$, the net change in the above transition is:

$$(N+1)P(m,N+1) - NP(m,N) \quad = \quad 1 - \frac{mP(m,N)}{2} \tag{13}$$

For $k>m$ two effects should be considered: the number of $k-1$ degree vertices that gained a new edge to become k-degree vertices and the number of k-degree vertices that gained a new edge and became of degree $k+1$. Therefore, for $k>m$ the net change in the transition is:

$$(N+1)P(k,N+1) - NP(k,N) \quad = \quad \frac{(k-1)P(k-1,N) - kP(k,N)}{2} \tag{14}$$

When looking for a stationary degree distribution, which is independent of N, we have

$$P(k,N+1) = P(k,N) = P(k)$$

Solving this recursive formulae yields to the following expression:

$$P(k) \quad = \quad \begin{cases} \frac{(k-1)P(k-1)-kP(k)}{2} & k>m \\ \\ 1-\frac{mP(m)}{2} & k=m \end{cases} \tag{15}$$

From Eq. 15 we get

$$2P(k) + kP(k) = (k-1)P(k-1) \tag{16}$$

$$P(k) = \frac{(k-1)P(k-1)}{(k+2)} \tag{17}$$

Eq. 17 is the recursion formula and if we expand the recursion up till $k = m$ we will come up with:

$$P(k) = \frac{k-1}{k+2}P(k-1) = \frac{k-1}{k+2}\cdot\frac{k-2}{k+1}P(k-2) =$$
$$\frac{k-1}{k+2}\cdot\frac{k-2}{k+1}\frac{k-3}{k}P(k-3) = \frac{k-1}{k+2}\cdot\frac{k-2}{k+1}\frac{k-3}{k}\frac{k-4}{k-1}P(k-4) =$$
$$\frac{k-1}{k+2}\cdot\frac{k-2}{k+1}\frac{k-3}{k}\frac{k-4}{k-1}\cdots\frac{m}{m+2}\cdot\frac{2}{m+2} =$$
$$\frac{2m(m+1)}{k(k+1)(k+2)}$$

We can see that expressions eliminate each other and in the limit of large k this gives a power-law degree distribution, $P(k) \sim k^{-3}$.

2.3.3 Model analysis - Clustering coefficient

Empirical analysis of the Barabási-Albert model shows that the clustering coefficient C decreases with the network size, following approximately a power-law $C \sim N^{-\frac{3}{4}}$. Figure 12 shows that this is a more moderate slope to that observed in random graphs (where $C \sim N^{-1}$). However, it is still converges to 0 when we increase N.

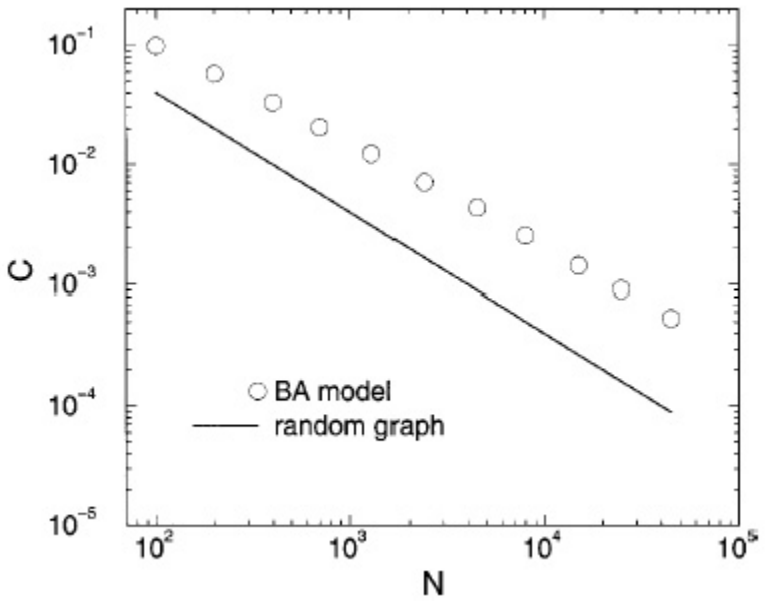

Figure 12: Source [1]. Clustering coefficient versus size of the Barabási- Albert model with $k = 4$, compared with the clustering coefficient of a random graph.

2.4 Geometric Random Graphs

2.4.1 Definition

We define a *geometric graph*, $G(V, r)$ above a metric space S and a distance measure δ, to be the graph with node set V of points in S and edge set:

$$E = \{(u, v)|(u, v \in V) \wedge (0 < \delta(u, v) \leq r)\} \tag{18}$$

That is, points in a metric space correspond to nodes and two nodes are adjacent if the distance between them is at most r. r is called the radius of the graph.
A *random geometric graph*, $G(n, r)$ above a metric space S and a distance measure δ, is a geometric graph with radius r on n nodes which were chosen randomly and independently from a uniform distribution over the points in S. For example, we can choose the two dimensional unit square to be our metric space and the Euclidean distance (2-norm) as the distance measure.

2.4.2 Global graph properties

As usual, we evaluate how good of a real-world network model a geometric random graph is by taking into consideration three global graph properties: degree distribution, average distance and clustering coefficient. Two out of these three standard network parameters show an improved fit to a GRN (Geometric Random Network) than to the popular scale-free model.

1. **Clustering coefficients:** Real-world networks typically have high clustering coefficients, but all the random network models we have seen so far lack that characteristic. Geometric random graphs capture spatial relationships as edges represent closeness in space. Intuitively it is clear that, especially in metric spaces of higher dimension, the distance between two neighbors of some node is likely to be small too. Thus, the likelihood of cluster formation in GRN should be high. Indeed, it turns out that geometric random graphs have better clustering coefficients, in terms of similarity to real biological networks, such as PPI-networks, than any of the models we have encountered so far. As can be seen in Figure 13, clustering coefficients of geometric random graphs have a very good fit with the clustering coefficients we see in PPI-networks, while there is not much similarity between clustering coefficients of PPI-networks to those of the generalized scale-free random models.

2. **Average distance:** Average distance of GRNs was also shown to be close to the small world characteristic of real-world networks.

3. **Degree distribution:** The major disadvantage of GRNs as models is their degree distribution. GRNs have Poisson degree distribution, while real-world networks show power-law degree distribution, closest to the distribution generated by the scale-free model.

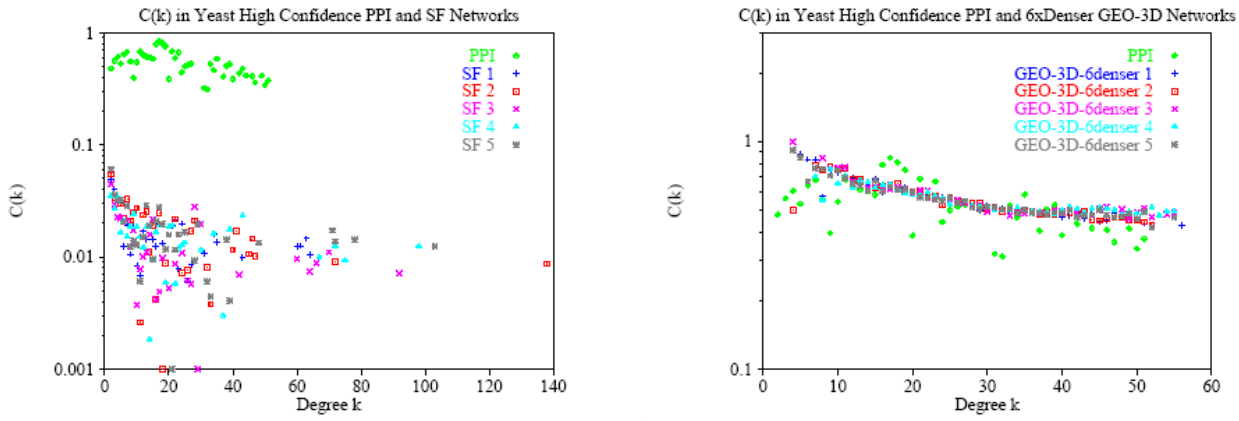

Figure 13: Source [9]. Clustering coefficient vs. degree in different random models and in a real PPI-network. The geometric graphs in this example were obtained from 3D points in a cube. r was selected such that the mean degree is equal to that of the PPI-network.

2.4.3 Local graph properties

Local graph properties, such as small, over represented patterns, may be important and characteristic to the network's (biological) function. Thus, it is important that our model will be able to imitate local properties of real-life networks too.

Network motifs, As mentioned in 2.1, are defined to be small sets of nodes that are connected in a specific wiring diagram and appear in the network in a significantly higher frequency than we would expect in random. Prŵulj *et al.* enumerated all 29 possibilities for connected graphs on 3-5 nodes (called *graphlets*), as can be seen in Figure 14. They identified the graphlets in the yeast's PPI-network and counted their frequencies of appearance. Then, they compared each graphlet's frequency with its frequency in different kinds of random networks. It was shown, that PPI-networks are closest to geometric random graphs with respect to the graphlets' distribution. That is to say, PPI-networks and GRNs have a similar motif profile (see Figure 15). This is a remarkable and surprising property of GRNs, and we wouldn't necessarily predict it a-priori.

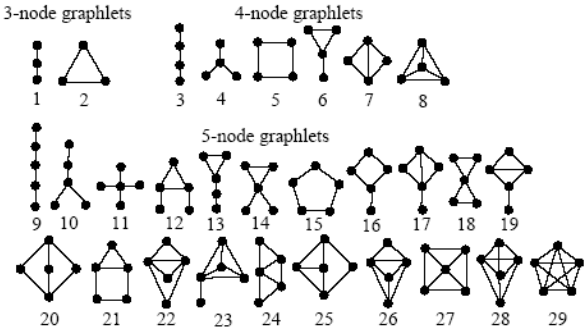

Figure 14: Source [9]. All possible 3-5-node graphlets.

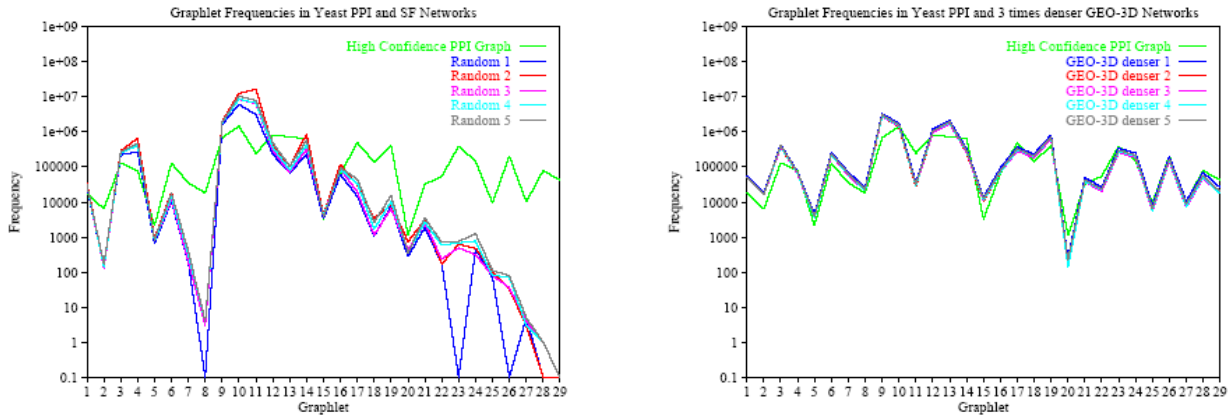

Figure 15: Source [9]. Motif frequency in different random models and in a real PPI-network.

To conclude, not only does GRN serves as good network model with respect to global graph properties, but it also has the advantage of portraying the local structure of important biological networks.

2.5 Exponential Random Graphs

2.5.1 Definition

By now we have seen a variety of random network models, yet each and every one of them lacked one or more of the properties we would like our network model to have. Is it possible to find a random network model to fulfill all our demands? We will now present a general method for developing such random network model.

According to our observations of real-world networks, we can come up with a set of constrains we would like our network model to satisfy (number of edges/nodes, clustering coefficients, degree distribution, local graph properties, etc.). All the random network models we have seen so far were generative ones: they defined a randomized procedure creating a graph to (hopefully) fulfill these constrains. Exponential random graphs present a different approach: instead of creating a graph, define a distribution over some ensemble of graphs, and sample from this distribution. To make a good model we would like to ensure that the expected value of each property will be equal to its observed value.

We denote our set of observations over m graph properties, $\{X_1, X_2, \ldots, X_m\}$ as $X = \{x_1, x_2, \ldots, x_m\}$. Our model contains a large finite set of graphs, $G = \{G_1, G_2, \ldots, G_n\}$, and a probability function over the set G, $\sum_i P(G_i) = 1$, such that $\sum_j P(G_j)X_i(G_j) = x_i$, where x_i is the observed value of property X_i and $X_i(G_j)$ is the value of property X_i in the graph G_j. Sampling graphs from G according to P ensures that in expectation we will get a graph which posses all qualities $\{x_1, \ldots, x_m\}$.

2.5.2 Choosing the distribution

We have m observations (or constraints) and n graphs. Since usually n is much larger than m, there are many degrees of freedom in determining a probability function over G that will fulfill our m constraints. That is, there are many distribution functions to choose from. Which one of them should we take?

Out of all choices we pick the probability function that maximizes the entropy. The *entropy*, S, of a distribution is defined to be:

$$S(P) \quad = \quad -\sum_G P(G) \ln P(G) \tag{19}$$

This approach is based on the assumption, borrowed from physics, that natural systems tend to maximize their entropy. By choosing the probability function with the maximal entropy, we choose the probability function that assumes as little excessive information as possible.

Another way of looking at it is to think of the following experiment: suppose we have n bins, a bin for each graph in G, and q infinitesimal identical particles of size $1/q$ each. We throw each of the particles independently, with equal probability to fall in each of the bins. The result of such an experiment defines a distribution over G (e.g., if r particles fell into bin i than $P(G_i) = r/q$). We repeat the experiment until the distribution we obtain satisfies $\{x_1, \ldots, x_m\}$. This is a simple experiment with multinomial distribution. If we calculate which of the valid results it is most likely to obtain in the experiment, we will see that it is the probability function that has the maximum entropy: The probability to obtain an outcome $\{k_1, k_2, \ldots, k_n\}$, where k_i denotes the number of particles that fell into bin i, is:

$$Pr(\{k_1, k_2, \ldots, k_n\}) = W \cdot (\frac{1}{q})^q \tag{20}$$

where

$$W = \frac{q!}{k_1!k_2!...k_n!} \tag{21}$$

The most probable result is the one which maximizes W. Rather than maximizing W directly we can equivalently maximize any monotonic increasing function of W. We choose to maximize

$$\ln W = \ln \frac{q!}{k_1!k_2!...k_m!} \tag{22}$$

$$= \ln q! - \sum_{i=1}^{n} \ln k_i! \tag{23}$$

We take the limit $q \to \infty$. Using Stirling's approximation, we get:

$$\lim_{q\to\infty} \ln W = q \ln q - \sum_{i=1}^{n} k_i \ln k_i$$

We devide it by q:

$$\ln q - \sum_{i=1}^{n} \frac{k_i}{q} \ln k_i$$

$$= \ln q - \sum_{i=1}^{n} P(G_i) \ln k_i$$

$$= \ln q - \sum_{i=1}^{n} P(G_i) \ln(qP(G_i))$$

$$= \ln q - \sum_{i=1}^{n} P(G_i) \ln q - \sum_{i=1}^{n} P(G_i) \ln P(G_i)$$

$$= \left(1 - \sum_{i=1}^{n} P(G_i)\right) \ln q - \sum_{i=1}^{n} P(G_i) \ln P(G_i)$$

$$= - \sum_{i=1}^{n} P(G_i) \ln P(G_i)$$

$$= - \sum_{G} P(G) \ln P(G)$$

$$= S(P)$$

Meaning, in order to maximize W, we should maximize the entropy.

2.5.3 The entropy-maximizing distribution

We chose the probability function for our model, $P(G)$, and now we wish to write it explicitly. We seek a distribution P that maximizes S (entropy) while ensuring $\sum_i P(G_i) = 1$ (i.e., P is a probability function over G) and for each $1 \leq i \leq m$, $\mathbb{E}(i) = x_i$. To do that we apply the Lagrange Multipliers technique. We get:

$$\frac{\partial}{\partial P(G)} \left[S + \alpha \left(1 - \sum_{G} P(G) \right) + \sum_{i} \theta_i \left(x_i - \sum_{G} P(G) X_i(G) \right) \right] = 0 \tag{24}$$

where α and $\{\theta_i\}$ are Lagrange multipliers. This gives

$$\ln P(G) + 1 + \alpha + \sum_i \theta_i X_i(G) = 0 \quad (25)$$

which yields

$$P(G) = \frac{1}{Z} \exp\left(-\sum_i \theta_i X_i(G)\right) \quad (26)$$

where Z is a normalization factor.

Clearly, $Z = e^{1+\alpha}$. Since $P(G)$ is a probability function we can also write Z as $-(\sum_G \sum_i \theta_i \cdot X_i(G))$. Calculating Z analytically is impossible since it requires summing over all the graphs in G (G's size is enormous). Since we cannot write Z explicitly we have a problem writing $P(G)$ explicitly. In complex models the θ_i's also cannot be calculated analytically, but they can be estimated using numerical methods.

2.5.4 Sampling

Seemingly we have come to a dead end: we know which distribution we want for our model, and we know how to write its equation, but we cannot express it explicitly. However, it turns out that though we cannot calculate the distribution explicitly, we can sample graphs from G according to this distribution. This is done using the Markov Chain Monte-Carlo (MCMC) technique. For practical purposes sampling according to the desired distribution is sufficient.

2.6 Monte Carlo Markov Chains (MCMC)

Markov chains are useful in a variety of calculation and optimization problems in bioinformatics which at first look seem unrelated to the technique. The methods used are described as *Markov Chain Monte-Carlo (MCMC)* methods. We apply a basic MCMC method called *Metropolis*, in order to sample according to a distribution without knowing explicitly its values. The idea is to construct a Markov chain on the set of graphs G and use the chain's statistics to estimate the distribution $P(G)$.

2.6.1 Markov chain

We introduce *discrete-time finite Markov chains* in abstract terms as follows. Consider some finite discrete set S of possible "states", labeled $\{E_1, E_2, \ldots, E_s\}$. At each of the unit time points $t = 0, 1, 2, 3, \ldots$, a Markov chain process occupies one of these states. In each time step t to $(t+1)$, the process either stays in the same state or moves to some other state in S according to some well-defined probabilistic rule described in more detail below. The process is called *Markovian*, and follows the requirements of a Markov chain if it has the following distinguishing characteristics:

1. **The memoryless property.** If at some time t the process is in state E_i, the probability that one time unit later it is in state E_j depends only on E_i, and not on the past history of the states it was in before time t. That is, the current state is all that matters in determining the probabilities for the states that the process will occupy in the future.

   $$Pr(S_i|S_{i-1} \ldots S_1) = Pr(S_i|S_{i-1}) \quad (27)$$

   where S_i is the state occupied at time i.

2. **The time homogeneity property.** Given that at time t the process is in state E_i, the probability that one time unit later it is in state E_j is independent of t.

For the Exponential Random Graphs (ERG) model, we define a set of valid operations on G, T, such that the following hold:

1. *Reachability*: Each graph $G_i \in G$ can be reached from any other graph $G_j \in G$ by applying some finite sequence of valid operations on G_j.

2. *Closeness*: The graph we obtain by applying a valid operation on some graph $G_i \in G$ is also in G.

For example, if G is the set of graphs with some fixed degree sequence, a valid operation may be crossing over edges.

The discrete-time finite Markov chain suitable for an ERG model is a Markovian process over the set of states $G = \{G_1, G_2, \dots G_n\}$, where the transition probability from state G_i to state G_j, denoted p_{ij}, is defined by the likelihood of transforming G_i to G_j by applying some valid operation from T on G_i.

We write the transition matrix

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}$$

It is easy to see that the transition probabilities for transforming one graph to another in t steps are encoded by $\mathbf{P^t}$. For example, the probability of transforming graph G_i to graph G_j in two time units is: $p^2_{ij} = \sum_l p_{il}p_{lj}$, which yields exactly $\mathbf{P^2}$.

2.6.2 Stationary distribution

Let p_i denote the probability of reaching G_i at time t. The vector $v = (p_1, \dots, p_n)$ is a stationary distribution (over G) if it does not change at $t + 1$, i.e.: $p_i = \sum_k p_k p_{ki}$. In particular, $v = v\mathbf{P}$ (v is an eigenvector of $\mathbf{P}$ with eigenvalue 1). After achieving a stationary distribution, the probability of being in a state is independent of the starting position, and the occurrence frequency of a state is equal to its stationary probability.

It can be shown that finite, irreducible (that is, each state is reachable from each other state), aperiodic Markov chains admit a unique stationary distribution, and approach it as t increases.

Thus, if we have a desired distribution on G in mind, that is we have a given v, our goal is to find $\mathbf{P}$ such that its unique stationary distribution will be v.

2.6.3 Detailed balance

If v is the stationary distribution of a process, the transition probabilities r_{ij} of the reverse process can be obtained via Bayes' rule:

$$r_{ij} = \frac{p_j p_{ji}}{p_i} \tag{28}$$

In fact, the reverse process' stationary distribution is also v:

$$\sum_i p_i r_{ij} = \sum_i p_j p_{ji} = p_j \sum_i p_{ji} = p_j \tag{29}$$

Suppose that at some time point t, our Markov chain reaches a distribution $q = (q_1, \ldots, q_n)$, where q_i is the probability of occupying G_i at time t, so that for all pairs i, j: $q_i p_{ij} = q_j p_{ji}$. This property of $(q_1, \ldots, q_n)$ is called *detailed balance*. Summing over all i we get:

$$\sum_i q_i p_{ij} = \sum_i q_j p_{ji} = q_j \sum_i p_{ji} = q_j \tag{30}$$

That is, for every j, the probability of occupying G_j at time $t+1$ is equal to the probability of occupying G_j at time t. In other words, q is a stationary distribution of our Markov chain. If our chain is irreducible and aperiodic, we conclude that a distribution q that satisfies the detailed balance equations is necessarily the stationary distribution.

The last conclusion is important, since many times we cannot calculate the stationary distribution of a Markov chain directly, but we do know how to find it via calculating a distribution which satisfies the detailed balance equations.

A Markov chain is considered *reversible* if the probability to reach state S_i from state S_j is equal to the probability of reaching S_j from S_i. It is easy to see, that a Markov chain has a distribution which satisfies the detailed balance equations if and only if it is reversible. i.e., $p_{ij} = p_{ji}$.

$$r_{ij} = \frac{p_j p_{ji}}{p_i} = \frac{p_i p_{ij}}{p_i} = p_{ij} \tag{31}$$

Clearly, the reversibility of the Markov chain we construct for an ERG model is determined by the character of the valid operations set, T, and the probability assigned to each operation. Hence, choosing an adequate operation set ensures that the ERG's Markov chain has a distribution which satisfies the detailed balance equations.

We wish to sample from the distribution $v = (p_1, \ldots, p_n)$ which is not computable analytically. If we knew how to construct a chain whose transition probabilities satisfy the detailed balance equations when reaching distribution v, then, since v is the stationary distribution, sampling from this Markov chain throughout a long enough period of time is equivalent to sampling according to the distribution v. In the next section it will be shown that it is possible to construct a Markov chain so that its stationary distribution is exactly the one we are interested in sampling from, $P(G)$.

2.6.4 The Metropolis algorithm

The *Metropolis* algorithm is a sampling algorithm. It is one of the simplest MCMC methods. The aim of the Metropolis algorithm is to construct an aperiodic irreducible Markov chain having some prescribed stationary distribution. In our case the desired stationary distribution is $P(G)$.

The Metropolis algorithm requires two components:

1. A symmetric *proposal mechanism*: the probability to move from state E_i to state E_j is equal to the probability to move from state E_j to state E_i. This can be achieved by defining the set of valid operations, T, to be symmetrical. Formulated using the notations we use for ERG models the requirement is $p_{ij} = p_{ji}$.
2. Acceptance of a move with probability $min(1, Pr(E_j)/Pr(E_i))$. In our notation, $min(1, P(G_j)/P(G_i))$

The Metropolis algorithm defines the sampling distribution (the transition probability over the Markov chain), as follows: the probability of moving to state E_j from state E_i is the product of the transition

probability from E_i to E_j and the probability of accepting a move from E_i to E_j. Formulated in our terms we get that the metropolis transition probability is:

$$q(G_j|G_i) = p_{ij} \cdot min(1, P(G_j)/P(G_i)) \tag{32}$$

Metropolis has characteristics of a greedy algorithm: if $P(G_i)$ is bigger than $P(G_j)$ accept this move with probability 1. Otherwise accept the move with some probability related to the ratio between $P(G_j)$ and $P(G_i)$:

$$Pr(accept\ move\ from\ G_i\ to\ G_j) \quad = \quad \begin{cases} 1 & P(G_j) > P(G_i) \\ P(G_j)/P(G_i) & \text{otherwise} \end{cases} \tag{33}$$

Claim: Metropolis sampling satisfies detailed balance.
Proof:

$$P(G_i)q(G_j|G_i) = P(G_i) \cdot p_{ij} \cdot min(1, P(G_j)/P(G_i)) \tag{34}$$

$$= p_{ij} \cdot min(P(G_i), P(G_j)) = p_{ji} \cdot min(P(G_j), P(G_i)) = P(G_j)q(G_i|G_j)$$

Recall (2.6.3) that the fact that the probability function $P(G)$ satisfies detailed balance with the transition function, shows it to be the stationary distribution of our Markov chain. That means that if we use the Metropolis transition probability in our Markov chain, after some sufficient time, we will sample graphs from G according to the desired distribution $P(G)$.

Note, that in order to write the transition function, q, explicitly we only need to be able to calculate expressions of the form $P(G_i)/P(G_j)$. Since in these expressions the normalization factor, Z, cancels out, calculating them is easy. Luckily, Monte Carlo algorithms of this type are straightforward to implement and appear to converge quickly allowing us to study quite large graphs.

3 Conclusions

Considering all the models described above, we conclude that no model perfectly fits the networks we observe in practice. In other words, every model captures only some of the attributes of real networks. We have also seen that statistical methods, such as exponential random graphs, pose an alternative to the generative techniques. Nevertheless, generalized random graphs are the most popular in the field, and are commonly used in the (applied) literature.

References

[1] R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[2] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.

[3] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *J. Comb. Theory, Ser. A*, 24(3):296–307, 1978.

[4] P. Erdos and A. Renyi. On the evolution of random graphs. *MTA Mat. Kut. Int. K ozl.*, 5:17–61, 1960.

[5] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences. May 2004.

[6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[7] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[8] N. Przulj. Graph theory analysis of protein-protein interactions. 2005.

[9] N. Prẑulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale free or geometric? *arXiv:q-bio.MN/0404017*, 2004.

[10] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–42, 1998.