

# Chapter 1

## A Network Analysis Primer

MICHAEL P.H. STUMPF<sup>1</sup> AND CARSTEN WIUF<sup>2</sup>

<sup>1</sup>*Centre for Bioinformatics, Division of Molecular Biosciences, Imperial College London*

<sup>2</sup>*Bioinformatics Research Center, Aarhus University  
m.stumpf@imperial.ac.uk, wiuf@birc.au.dk*

Graph methods form a cornerstone of modern systems biology. In this chapter we review the fundamental apparatus of statistical descriptors and measures of graph properties. There is no single meaningful statistic that can describe all aspects of a network and we present a range of different measures that, when combined and critically evaluated, allow us to gain non-trivial insights into the architecture of complex networks in biology.

### 1.1. Introduction

Following the enormous advances in functional genomics and molecular biology, it is now possible to at least contemplate studying cellular processes at the level of a whole cell, rather than in isolation. Molecular networks, such as protein interaction,<sup>1–3</sup> metabolic<sup>4</sup> and gene regulation networks,<sup>5,6</sup> aim to capture such sets of biological processes in a single and coherent framework. In reality, of course, these different networks are intricately connected and interwoven inside a cell: protein products will interact with each other, regulate the expression of genes as well as digesting nutrients and catalysing basic biochemical reactions in a cell's metabolism. We are still a long way away from being able to consolidate these different networks into a realistic *in silico* organism.

The analysis and interpretation of present network data is, however, already challenging enough. Since the late 1990s, research has been aided considerably by the work of a host of physicists (see Refs. 7–10 for mainly physics-oriented reviews). While the models proposed have, despite their elegant simplicity, been able to explain certain aspects of complex biological networks, they increasingly reach the limit of their usefulness given the amount of data becoming available. New models, based on sound statistical principles and informed by bioinformatics, are now slowly taking their place. These networks, especially their union, form the scaffold for further systems biology investigations, and their understanding will

crucially underlie the success of the fledgling discipline of synthetic biology.

One of the central problems in the analysis of the detailed data we are confronted with now is to understand the intricate interplay between the functioning of these networks on the one hand, and their evolution on the other. While evolution clearly will not give rise to biological systems that fail spectacularly, recent research has shown that not everything found in nature has necessarily been honed by natural selection. There is indeed, as argued forcefully by Michael Lynch, a perfectly plausible explanation for any feature of biological networks in terms of a neutral evolutionary theory.

A generic problem of evolutionary analyses is, however, that evolutionary processes are highly stochastic and historically contingent. Therefore the variability inherent in evolutionary dynamics frequently masks the average behaviour and as a result, evolutionary biology has been intimately tied to statistical inference ever since it started to become a quantitative rather than a merely descriptive science. Hence the two-fold scope of this book, which puts roughly equal weight on evolutionary and statistical issues surrounding network evolution. Our aim is to present a selection of views related to how we can understand and analyse networks and their evolution<sup>11</sup> in a statistically sound manner.

## 1.2. Types of Biological Networks

At the molecular level we can distinguish very coarsely between three types of molecular networks.

**Metabolic networks** aim to describe the basic biochemistry inside a cell. Biologically important reactions have been described in terms of reaction pathways and metabolic networks are systematic collections of such biochemical data.

**Transcriptional networks** consist of genes where a directed edge is added between two genes if one regulates the transcription of the other gene.

**Protein interaction networks** in which an undirected edge is drawn between each pair of proteins where there is evidence of a physical or biochemical interaction.

Making these distinctions and simplifications must necessarily neglect details of the biological processes.<sup>12</sup> In reality these networks will be highly and intricately interconnected and factorising them into distinct networks will ultimately underestimate the biological complexity. These molecular networks are supplemented by physiological networks (such as the arterial and neuronal networks in higher organisms), which are not covered in this volume. Moreover, at the level of the population these networks are complemented by a higher level of networks which include food webs, ecological and epidemiological interaction and contact networks,<sup>13,14</sup> and ultimately for humans, social networks.<sup>15</sup> While we do not believe it is appropriate to push analogies which frequently do not hold up to closer scrutiny the mathematical

formalism and the statistical problems are frequently transferable. At a more ambitious level we may in fact need to include ecological interactions in order to understand the evolution and function of networks at the molecular level. This is, for example, likely to be the case when we compare different bacterial organisms, where levels of pathogenicity as well as ecological factors and type of metabolism (aerobic or anaerobic) may help to understand differences in network organisation.

### 1.3. A Primer on Networks

#### 1.3.1. Mathematical descriptions of networks

Here we are primarily concerned with purely static interactions. That is, we consider the network fixed. Any changes the network might experience over time, e.g. over the life time of the organism or over evolutionary time scales, are not taken into account.

A graph  $G$  is the combination of a non-empty set of  $N$  nodes,  $\mathcal{V}$ , and a (generally but not necessarily non-empty) set of  $M$  edges,  $\mathcal{E}$ . In graph theory, nodes are often also called *vertices* and edges *arches*. Each edge  $e_s \in \mathcal{E}$  with  $1 \leq s \leq M$  is in turn associated with two nodes  $v_i, v_j \in \mathcal{V}$  and we write

$$e_s = (v_i, v_j) \quad \text{for } 1 \leq i \leq M \text{ and } 1 \leq i, j \leq N; \quad (1.1)$$

the edge  $e_s$  is then said to be *incident* on nodes  $v_i$  and  $v_j$ .

For a given set of nodes,  $\mathcal{V}$ , and a corresponding set of edges,  $\mathcal{E}$ , we write

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad (1.2)$$

to define the graph  $\mathcal{G}$ .

In general each edge may be associated with a *direction* and a *weight*,  $w_i \in \mathbb{R}$ . In a *directed graph* we attach a direction to each edge  $e_s^{(d)}$ .  $e_s^{(d)} = (v_i, v_j)$  means that the edge  $e_i$  starts at node  $v_i$  and ends at node  $v_j$ . In an *undirected graph* the order in which nodes are written does not matter and  $e_s^{(u)} = (v_i, v_j) = (v_i, v_j)$ . Quite generally we allow for  $v_i = v_j$ , that is an edge may originate and end on the same vertex; this edge is said to form a *one-edged loop* attached to node  $v_i$ . It is also possible to allow more than one edge between nodes  $v_i$  and  $v_j$ .

If a graph contains neither multiple edges between pairs of nodes nor loops, then the graph is called *simple*. For simple graphs a number of additional statements can be made. For example, the number of edges in a simple graph is at most

$$M^{\max} = \frac{N(N-1)}{2}, \quad (1.3)$$

in which case the network is called *fully connected*.

Figure 1.1 shows an example of an undirected simple network with  $N = 8$  nodes and  $M = 7$  edges, and a directed network. Note that node 4 is disjoint from the rest of the network. While genes or proteins which do not interact with other molecules inside their environment are biologically implausible, it is nevertheless possible that,

for instance, a protein's interaction partners are not included in the experimental setup.

### 1.3.1.1. Characteristics of a node

Biological networks are generally *labelled* with information. To each node  $v_i$  we have an associated vector of properties,  $V_i$ . These may include the biological name of the node, e.g. the name of the gene or protein, biological classifications and other experimental data.

One of the most prominent characteristics of a node in a network is its *degree*,  $d_i$ , the number of edges incident on a node. In a directed network we distinguish between the *in-degree* and the *out-degree*,  $d_i^{\text{in}}$  and  $d_i^{\text{out}}$ , i.e. the number of nodes ending on and starting from node  $v_i$ .

The degree of a node tells us how many neighbours it has in the network. We define the *neighbourhood*,  $\Gamma(v_i)$  of a node  $v_i$  through

$$\Gamma(v_i) := \{v_j | v_j \in V \text{ and } (v_i, v_j) \in E\}. \quad (1.4)$$

Trivially, the degree (in-degree) is also the size of the neighbourhood  $d_i := |\Gamma(v_i)|$ . In all networks we also have

$$\sum_i d_i = 2M \quad (1.5)$$

where  $M = |E|$  is the total number of edges in a graph. (For directed networks the sum is  $M$  and not  $2M$ .) From Eqn. (1.5) it follows straightforwardly that the total number of nodes with odd degrees must be an even number.

### 1.3.1.2. Paths, components and trees

A *path* from node  $v_i$  to  $v_j$  is a sequence of edges which can be traversed to reach  $v_j$  starting from  $v_i$ ; in directed networks paths cannot go against the direction of an edge. We say that node  $v_j$  is *connected* to node  $v_i$  if there is a path from node  $v_i$  to  $v_j$ , taking into account the directionality of edges in a directed network. Thus node 1 in the network shown in Fig. 1.1B is connected to node 4; equally node 4 is connected to node 1. Node 2, however, is not connected to node 1. In an undirected network, if there is a path from node  $v_i$  to node  $v_j$ , then there is also a path from  $v_j$  to  $v_i$ . If there is a path starting from and ending on a node  $v_i \in \mathcal{V}$ , then this is called a *loop*.

A set of  $k$  nodes  $\mathcal{C} = \{v_1, v_2, \dots, v_k\}$  where each node in  $\mathcal{C}$  can be reached from other nodes in  $\mathcal{C}$  but not from any node outside of  $\mathcal{C}$  is called a *connected component of size  $k$*  of the network. In a simple network the number of components  $K$  is given by

$$K \geq N - M \quad (1.6)$$

which is easily shown by induction.

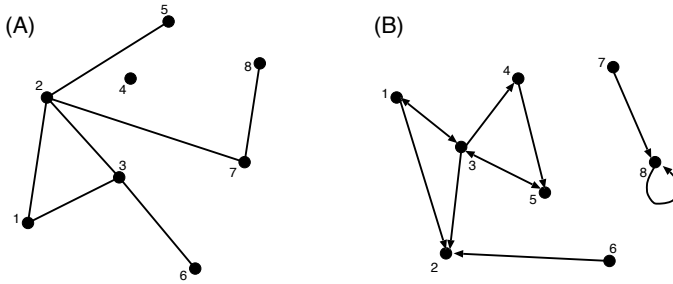


Fig. 1.1. Examples of a simple undirected network (A) and a directed network (B).

In many cases it may be preferable to study the largest connected component rather than the network as a whole. This may, for example, be the case when a large number of nodes occur in singletons, pairs or other small groups of nodes.

If there is more than one path between a pair of nodes  $v_i, v_j \in \mathcal{V}$ , then the graph contains *closed paths*, or loops. In an undirected simple graph, if there is precisely one path between each pair of nodes  $v_i, v_j \in \mathcal{V}$ , then there cannot be any loops and the graph is called a *tree*. If a graph consists of several components, each of which is a tree, the graph is sometimes referred to as a *forest*. The concept of a tree is very important and useful in the analysis of graphs and networks and we will sometimes borrow from the rich literature on trees.

Of particular interest is the *spanning tree*  $\mathcal{T}$  of a connected graph with nodes  $\mathcal{V}_{\mathcal{T}} = \mathcal{V}_{\mathcal{G}}$  and edges  $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}_{\mathcal{G}}$ , such that  $(\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}})$  is a tree. It is possible to show that a connected graph contains at least one spanning tree. Spanning trees can be used to traverse all nodes of a connected network.

### 1.3.1.3. Distance and diameter

If two nodes are connected by a sequence of nodes and edges, then the *distance*  $l_{ij}$  between them is defined as the number of edges that have to be traversed to reach node  $v_j$  from  $v_i$ ;

$$l_{ij} = \min\{X_{ij} | X_{ij} \text{ is a path from node } v_i \text{ to node } v_j \text{ along edges } e_s \in \mathcal{E}\}. \quad (1.7)$$

If there is no path by which node  $v_j$  can be reached from node  $v_i$  then we set

$$l_{ij} = \infty. \quad (1.8)$$

In directed networks, of course  $l_{ij}$  can be different from  $l_{ji}$ ; one of them can even be infinite as shown by nodes 1 and 2 in the network in Fig. 1.1 where  $l_{12} = 1$  and  $l_{21} = \infty$ .

The *diameter* of a network is defined as the maximum distance between two nodes in the network,

$$D = \max\{l_{ij} | v_i, v_j \in \mathcal{V}\}. \quad (1.9)$$

Thus by definition the diameter of the network which consists of more than one component is  $\infty$ . The definition for  $D$  is analogous to the definition of diameters in geometry and topology: the maximum distance between two points belonging to the same object.

Frequently, we therefore restrict analyses of biological networks to the nodes in the largest component. This is particularly relevant if the network exhibits a *giant connected component* (GCC) which is defined for growing networks only. A GCC is a component with non-zero relative size as the size of the network becomes large. The relative size of a component is defined as the number of nodes in the component divided by the total number of non-zero degree nodes. Because of the incomplete nature of many biological data sets, observed biological networks often appear fragmented and composed of several components. However, once a complete or truly integrated network, one which contains all physical, regulatory and small-molecule-mediated interactions has been established, we would expect all the nodes in the whole network to be connected.

### 1.3.2. Network properties

Some of the quantities introduced above can be used to characterise aspects of networks. Here we will introduce some of the common statistics that have been used to describe them.

#### 1.3.2.1. The degree distribution

We have already discussed the degree of a node  $v_i$ , here denoted by  $d_i$ . The average degree,  $\bar{d}$ , of a network is given by

$$\bar{d} = \frac{1}{N} \sum_{i=1}^N d_i. \quad (1.10)$$

We note that in a directed network the average in- and out-degrees of a node must be equal,

$$\frac{1}{N} \sum_{i=1}^N d_i^{\text{in}} = \frac{1}{N} \sum_{i=1}^N d_i^{\text{out}}. \quad (1.11)$$

Surprisingly, this simple fact is frequently ignored and any analysis which contains reports of unequal in- and out-degrees should be treated with considerable caution.

The degree is analogous to the coordination number of a site in a regular lattice. Unlike coordination numbers, however, the degrees of nodes in a network will generally take on many different values. Thus the average degree is not very informative about a network and what is generally considered instead, is the degree distribution  $n(k)$ , the probability of a node to have degree  $d_i = k$ ,  $k = 0, 1, 2, \dots$

The degree distribution is defined by

$$n(k) = \frac{1}{N} \sum_{i=1}^N \delta_{d_i,k} \quad \text{for } k = 0, 1, 2, \dots \quad (1.12)$$

where  $\delta_{i,j}$  is the Kronecker delta function

$$\delta_{i,j} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} \quad (1.13)$$

defined for integers  $i, j$ . The degree distribution summarises information about the local environments in a network. It has to be kept in mind, though, that the degree distribution is highly degenerate, i.e. there are many different networks which have the same degree distribution. While the average in- and out-degrees in networks have to be identical, the corresponding degree distributions,

$$n^{\text{in}}(k) = \frac{1}{N} \sum_{i=1}^N \delta_{d_i^{\text{in}},k} \quad (1.14)$$

and

$$n^{\text{out}}(k) = \frac{1}{N} \sum_{i=1}^N \delta_{d_i^{\text{out}},k}, \quad (1.15)$$

respectively, can be very different indeed.

### 1.3.2.2. Clustering

A further statistic which describes the local environment, but also including next-nearest neighbours, is given by the so-called clustering coefficient. The clustering coefficient measures the probability that two nodes  $v_j$  and  $v_k$ , which are both neighbours of  $v_i$  (i.e.  $(v_i, v_j), (v_i, v_k) \in \mathcal{E}$  in an undirected graph), are themselves connected by an edge  $(v_j, v_k) \in \mathcal{E}$ . For node  $v_i$  the clustering coefficient is defined by

$$c_i = \frac{2\eta_i}{d_i(d_i - 1)} \quad \text{for } d_i \geq 2 \quad (1.16)$$

where  $\eta_i$  is the number of edges among the nodes connected to  $v_i$ . The average clustering coefficient of the network is then given by

$$\bar{c} = \frac{1}{N} \sum_{i=1}^N c_i. \quad (1.17)$$

In a social network the clustering coefficient could for instance measure the extent to which my friends are also friends themselves.

Just like the average degree fails to capture the diversity of degrees observed in most natural networks, the average clustering coefficient fails to describe the

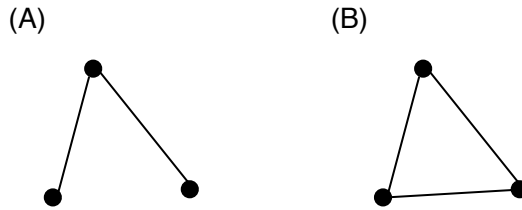


Fig. 1.2. Three connected nodes in an undirected network can either form an open (A) or a closed triangle (B). A network's transitivity is defined as the probability of a triangle to be closed on all three sides.

network's local inhomogeneity. It is therefore often useful to study the distribution of clustering coefficients, e.g. using the cumulative distribution defined by

$$C(c) = \sum_{i=1}^N \int_0^c \delta(c_i - c') dc' \quad (1.18)$$

where  $\delta(x)$  is the Dirac delta function, defined by  $\delta(x) = 1$  for  $x = 0$  and  $\delta(x) = 0$  otherwise.

Related but not identical to the clustering coefficient is the transitivity. This is defined by

$$T = \frac{\# \text{ of closed triangles}}{\# \text{ of connected triplets of nodes}}. \quad (1.19)$$

For trees we necessarily have  $\bar{c} = 0$ ; the same is also true for the square (or cubic or hypercubic lattices). Thus small values of  $C$  are not indicative of the absence of loops or closed paths. In fact, as we shall see later, most naturally occurring lattices, including those in systems biology, are locally tree-like. For this reason we prefer the distribution of clustering coefficients rather than the average clustering coefficient.

### 1.3.2.3. Average path length

The average path length of a network follows from all pairwise distances in a network and is given by

$$\bar{l} = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^N l_{ij}. \quad (1.20)$$

By definition  $l_{ii} = 0$ .

Analogous to the degree and clustering distributions, it is also possible to define a distribution of network distances. One convenient definition is given by

$$\lambda(l) = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^N \delta_{l_{ij}, l} \quad \text{for } l = 1, 2, \dots, \quad (1.21)$$

which counts the number of distances of length  $l$ .

Because the distance of two unconnected nodes is  $\infty$ , the average path length (and the diameter) will diverge in networks which consist of more than one component. Therefore one often considers only the largest connected component when analysing network distances. We note that the diameter  $D$  and the average path length in a network may be very different.

### 1.3.3. Mathematical representation of networks

There are three basic methods to represent or store a graph. Here we will define these different representations before giving some guidelines on when to use which representation.

#### 1.3.3.1. The adjacency matrix

The *adjacency matrix*  $\mathbb{A}$  of a graph is an  $N \times N$  matrix and is defined by

$$A_{ij} = \begin{cases} w_{ij}, & \text{if nodes } i \text{ and } j \text{ are connected by an edge with weight } w_{ij} \\ 0, & \text{otherwise.} \end{cases} \quad (1.22)$$

This is the most general case but we will often consider special cases of Eqn. (1.22). For an unweighted graph, for example,  $w_{ij} = n_{ij} \in \mathbb{Z}_0$  is the number of (directed) edges between nodes  $v_i$  and  $v_j$ . For an undirected graph we have

$$A_{ij} = A_{ji}, \quad (1.23)$$

i.e. the adjacency matrix is symmetrical. The adjacency matrix of a *simple* graph is given by

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between node } i \text{ and } j \text{ and } j \neq i \\ 0 & \text{otherwise.} \end{cases} \quad (1.24)$$

For real networks, as we will see below, the actual number of edges is much lower than the maximum number of edges possible, Eqn. (1.3), and the adjacency matrix will be a *sparse matrix*.

The adjacency matrix of the simple undirected graph in Fig. 1.1, for example, is given by

$$\mathbb{A} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (1.25)$$

Table 1.1. Computational complexity of some elementary graph operations in terms of the number of nodes,  $N$ , and number of edges,  $M$ . Costs also include a constant factor which has been ignored here.

Property	Adjacency matrix	Adjacency list	Edge list
Memory requirement	$N^2$	$N + M$	$M$
Initialisation	$N^2$	$N$	1
Copying a node	$N^2$	$M$	$M$
Deleting an edge	$N$	$M$	1
Finding an edge	1	$N$	$M$
Is a node isolated	$N$	1	$M$
Testing for a path between two nodes	$N^2$	$M \log(N)$	$N + M$

where the nodes and columns correspond to the node labels in Fig. 1.1. The labelling of the nodes can of course be changed and the corresponding new adjacency matrix can be obtained from the adjacency matrix in Eqn. (1.25) by rearranging the rows and columns.

### 1.3.3.2. *The adjacency list*

We see in Eqn. (1.25) that the adjacency matrix is sparse. This is typical for many real networks and the adjacency matrix will typically have only a small fraction of non-zero entries. An alternative and slightly less wasteful way of storing the structure of the network is through the *adjacency list*. This list contains all nodes connected to a node; the adjacency list corresponding to the matrix in Eqn. (1.25) is

$$\begin{aligned}
 1 & : 2, 3 \\
 2 & : 1, 3, 5 \\
 3 & : 1, 2, 6 \\
 4 & : \\
 5 & : 2 \\
 7 & : 2, 8 \\
 8 & : 7
 \end{aligned} \tag{1.26}$$

Computationally this is generally implemented by defining an array of lists such that the nodes connected to a given node can be accessed immediately.

### 1.3.3.3. *The edge list*

The two representations introduced above focus on nodes. In some instances it may be more interesting to describe the edges, e.g. when we want to study if two

interacting biological molecules share the certain characteristics. In this case we can use the *edge list* notation. This, for the above example, takes the form

$$\{(1, 2), (1, 3), (2, 3), (2, 5), (2, 7), (3, 6), (7, 8)\}. \quad (1.27)$$

Thus we store a list containing each edge that exists in the graph, keeping in mind that for an undirected graph  $(v_i, v_j) = (v_j, v_i)$ . In many circumstances the edge list is the most memory-efficient way to store network information.

#### 1.3.3.4. *Some remarks on complexity*

Here, *complexity* refers to the computational effort required to evaluate a property of the graph. The effort of performing simple computational tasks such as setting up a network or testing if two nodes are connected depends on the way in which network information is represented. The complexities of a number of different tasks for the three network representations outlined above are given in Table 1.1. Strictly speaking, the true cost of each task is proportional to the factor in Table 1.1 multiplied by a constant factor.

All real networks are finite sized and, as far as biological networks are concerned, *mesoscopic systems*. The number of nodes is typically of the order of several thousand to tens of thousands. This implies that (i) in principle, it is possible to analyse networks computationally and (ii) the size of the network is sometimes of the same order as the proportionality constant by which the complexities in Table 1.1 are multiplied.

The computational complexity of several important and interesting problems in the analysis of networks belong, however, to classes of problems which are considerably more cumbersome. Briefly, problems are often divided into the following classes

*P*: A problem that can be solved in polynomial time.

*NP*: (Non-deterministic polynomial) A problem that has a solution that can be verified (by a non-deterministic Turing machine) in polynomial time. All problems in *P* are also in *NP*; the reverse is not necessarily true.

*NP-hard*: A problem that can be solved by an algorithm which can be translated into one for solving any other *NP* problem. *NP-hard* problems are at least as hard to solve as any other problem in *NP*.

*NP-complete*: A problems that is both in *NP* and *NP-hard*.

Issues of computational complexity are frequently encountered in the analysis of networks. Especially when trying to understand properties of theoretical network models or when assessing statistical significance of network properties, we will often have to repeatedly calculate the same network property.

## 1.4. Comparing Biological Networks

In the previous section we have discussed some basic mathematical properties of networks. Unfortunately, as will be discussed later, networks with identical/similar properties are not necessarily identical/similar. Moreover it has so far been impossible to come up with a useful definition of *distance* between networks. Here, we therefore only briefly discuss basic notions of network identity as far as these are required in order to compare biological networks.

Comparative analysis is a cornerstone of evolutionary analysis and at the sequence level has provided us with detailed insights into the evolutionary history of life. Thus the biological analysis of networks must necessarily involve comparison of networks from different species. For example there has been considerable interest as to whether evolutionary inferences from protein interaction network data provide similar information in different organisms. But while the vagaries of the highly stochastic evolutionary process are already hard enough to understand at the level of DNA and protein sequences, these problems are exacerbated at a spectacular scale once we enter the system level. Here we therefore focus only on the basics of the underlying theoretical framework that may aid in comparing biological networks.

An important lesson that can be learned from sequence-based (or even traditional morphological-trait-based) comparative biology is the need to compare species over the broadest range of evolutionary divergences possible. Our understanding of sequence evolution (including the evolution of e.g. transcription factor binding sites) has benefited enormously from the abundance of data from several closely related species. For many biological networks, the evolutionary separation between model organisms is simply too large for meaningful comparisons to be made. We therefore need to map interactomes, gene regulatory and metabolic networks in those species that are sufficiently closely related to model species such as *S. cerevisiae* and *E. coli*.

### 1.4.1. Identity of networks

Two networks  $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$  and  $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$  are called *isomorphic* if there is a one-to-one correspondence between the nodes,  $\mathcal{V}_1$  and  $\mathcal{V}_2$ , and edges,  $\mathcal{E}_1$  and  $\mathcal{E}_2$ , which preserves the assignment of nodes to edges and *vice versa*. That is, if  $e_s \in \mathcal{E}_1$  is associated with  $e_t \in \mathcal{E}_2$ , and if  $e_s = (v_i, v_j)$  and  $e_t = (v_k, v_l)$ , then  $v_i$  must be associated with  $v_k$  and  $v_j$  with  $v_l$ .

If  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are isomorphic we write

$$\mathcal{G}_1 \simeq \mathcal{G}_2 \tag{1.28}$$

rather than  $\mathcal{G}_1 = \mathcal{G}_2$  to indicate that  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are instances of the same (abstract) graph; they may still have different graphical or mathematical representations: for

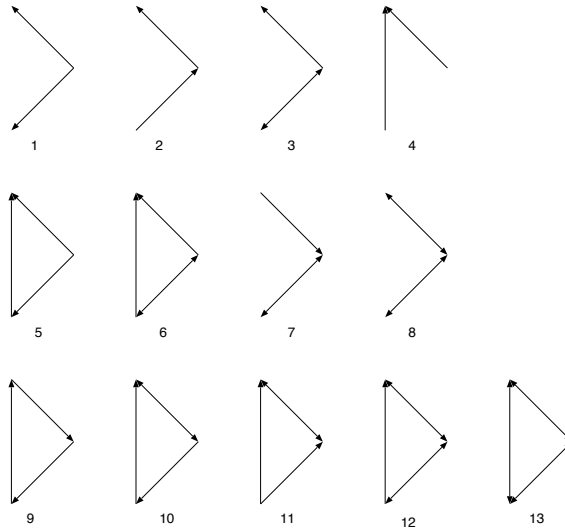


Fig. 1.3. The 13 patterns possible to observe for three connected nodes in a directed networks.

example, the rows or columns of their respective adjacency matrices may be interchanged.

Each network can be drawn in many different ways. We also say that a graphical representation of a network is an *instance* of a network and we will seek to define under what circumstances two networks are identical, in the sense that their network structure is the same.

Determining if two graphs are isomorphic has been shown not to be in  $P$  but so far there has been no proof that it is *NP-complete*. Some people prefer to assign it to its own class of *graph isomorphism problems*. In practice, these issues may pose severe limitations on the exhaustive analysis of biological networks. For example, a human protein-interaction network which covers the 20,000 or so different proteins (ignoring splice variants) cannot easily be analysed in a comprehensive statistical manner. For computational reasons the search for suitable heuristics for network investigation will therefore increase in importance.

### 1.4.2. Subnets and patterns

A *subnet*  $\mathcal{S}$  of a network  $\mathcal{N}$  is defined by  $\mathcal{S} := (\mathcal{V}^*, \mathcal{E}^*)$  with

$$\mathcal{V}^* \subset \mathcal{V}$$

$$\mathcal{E}^* \subset \mathcal{E}$$

$$\text{If } e_s = (v_i, v_j) \in \mathcal{E}^* \text{ then } v_i, v_j \in \mathcal{V}^*$$

$$\text{If } v_i, v_j \in \mathcal{V}^* \text{ and } (v_i, v_j) \in \mathcal{E} \text{ then } e_s = (v_i, v_j) \in \mathcal{E}^* \tag{1.29}$$

Thus a subnet is itself a network consisting of a subset of nodes of the *global network*  $\mathcal{G}$  and all the edges connecting pairs of nodes in the subnet. Equally, we could define the subnet through the set of edges and the associated nodes.

The way subgraphs are set up can influence the inferences to be gained from an analysis of  $\mathcal{S}$ . We may, for example, study a particular biochemical pathway as a subset of an organism's metabolism; or we may seek to test for interactions among the known proteins in an organism.

Closely related to subnets is the notion of a *pattern* which we define through a connected graph  $\mathcal{P} := (\mathcal{V}_{\mathcal{P}}, \mathcal{E}_{\mathcal{P}})$ ; we define the size of the pattern as the number of nodes needed to define it,  $s = |\mathcal{V}_{\mathcal{P}}|$ . For example, nodes 1, 2 and 3 in Fig. 1.1A form a closed triangle which is a pattern of size 3. In many cases we will be interested in determining the frequencies of a set of patterns in a network. The sets of all patterns formed by three nodes in a directed network are shown in Fig. 1.3; the corresponding patterns of size 3 in an undirected network are in Fig. 1.2. These patterns may represent important functional or logical units of organisation; of particular interest are those patterns in a network which have more internal edges than would be expected to occur by chance, given the rest of the network.

### 1.4.3. *The challenges of the data*

We have already mentioned the complexity of evolutionary processes, especially when trying to go beyond the sequence level. The analysis of this highly stochastic and contingent process is exacerbated when one considers the often woeful quality of the data: for protein interaction networks (PIN) the rates for false-positive and false-negative results are estimated to be around 40%. Bioinformatics and statistics may help to clean the data to some extent but improvements in experimental techniques offer the only real solution to this problem. Although important and interesting we will here not be concerned with such issues of quality control. Rather we will discuss what should be included in theoretical descriptions of complex networks in a biological setting.

It has to be kept in mind, though, that present network data are highly averaged and artificial constructs: the language of graph theory may simply be too static to usefully describe complex biological networks. We may in approximation seek to understand networks as entities that change over three different time scales: (i) they will change over evolutionary time scales between species (millions of years), (ii) they will change during the course of an organism's development (years), and finally, (iii) connections will be formed and lost in response to physiological change and external stimuli (sub-second to minutes). Already we are seeing the first attempts to map biological networks *in vivo* and future experimental developments will, no doubt, enable us to probe the dynamics on the biologically relevant time and spatial scale. For protein interaction networks, experimental methods can at the moment only resolve the changes in PIN structure accumulated between species,<sup>16–18</sup> but the data are not yet sufficiently reliable to make meaningful comparisons.

## References

1. P. Uetz, L. Giot, G. Cagney, T. Mansfield, R. Judson, V.D.L. Narayan, M. Srinivasan, P. Pochart, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamar, M. Yang, M. Johnston, S. Fields and J. Rothberg A comprehensive analysis of protein-protein interaction networks in *Saccharomyces cerevisiae*. *Nature*, 403:623–627, 2000.
2. S. Maslov and K. Sneppen Specificity and stability in topology of protein networks. *Science*, 296(5569):910–3, 2002.
3. I. Agrafioti, J. Swire, J. Abbott, D. Huntley, S. Butcher and M.P.H. Stumpf Comparative analysis of the *Saccharomyces cerevisiae* and *Caenorhabditis elegans* protein interaction networks. *BMC Evolutionary Biology*, 5:23, 2005.
4. H. Ma and A.P. Zeng Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, 19:270–277, 2003.
5. M. Ronen, R. Rosenberg, B. Shraiman and U. Alon Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl. Acad. Sci. USA*, 99(16):10555–10560, 2002.
6. A. Evangelisti and A. Wagner Molecular evolution in the yeast transcriptional regulation network. *Journal of Experimental Zoology Part B-Molecular and Developmental Evolution*, 302B(4):392–411, 2004.
7. R. Albert and A.L. Barabasi Statistical mechanics of complex networks. *Rev.Mod.Phys.*, 74(1):47–97, 2002.
8. M. Newman The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.
9. T. Evans Complex networks. *Contemporary Physics*, 45(6):455–474, 2004.
10. S. Dorogovtsev and J. Mendes *Evolution of Networks*. Oxford University Press, 2003.
11. M.P.H. Stumpf, W.P. Kelly, T. Thorne and C. Wiuf Evolution at the system level: the natural history of protein interaction networks. *Trends Ecol.Evol.*, 22:366–373, 2007.
12. A.P. Coates, S.H. Muggleton and M.J.E. Sternberg The identification of similarities between biological networks: Application to the metabolome and interactome. *Journal of Molecular Biology*, 369:1126–1139, 2007.
13. S. Proulx, D. Promislov and P. Phillips Network thinking in ecology and evolution. *Trends.Ecol.Evol.*, 20(6):345–353, 2005.
14. R.M. May Network structure and the biology of populations. *Trends.Ecol.Evol.*, 21:394–399, 2006.
15. G. Robins and P. Pattison Random graph models for temporal processes in social networks. *J.Math.Soc.*, 25:4–21, 2001.
16. H.B. Fraser, A.E. Hirsh, L.M. Steinmetz, C. Scharfe and M.W. Feldman Evolutionary rate in the protein interaction network. *Science*, 296(5568):750–2, 2002.
17. I.K. Jordan, Y.I. Wolf and E.V. Koonin No simple dependence between protein evolution rate and the number of protein-protein interactions: only the most prolific interactors tend to evolve slowly. *BMC Evol Biol*, 3(1):1, 2003.
18. H. Qin, H.H.S. Lu, W.B. Wu and W.H. Li Evolution of the yeast protein interaction network. *Proc. Natl. Acad. Sci. USA*, 100(22):12820–4, 2003.