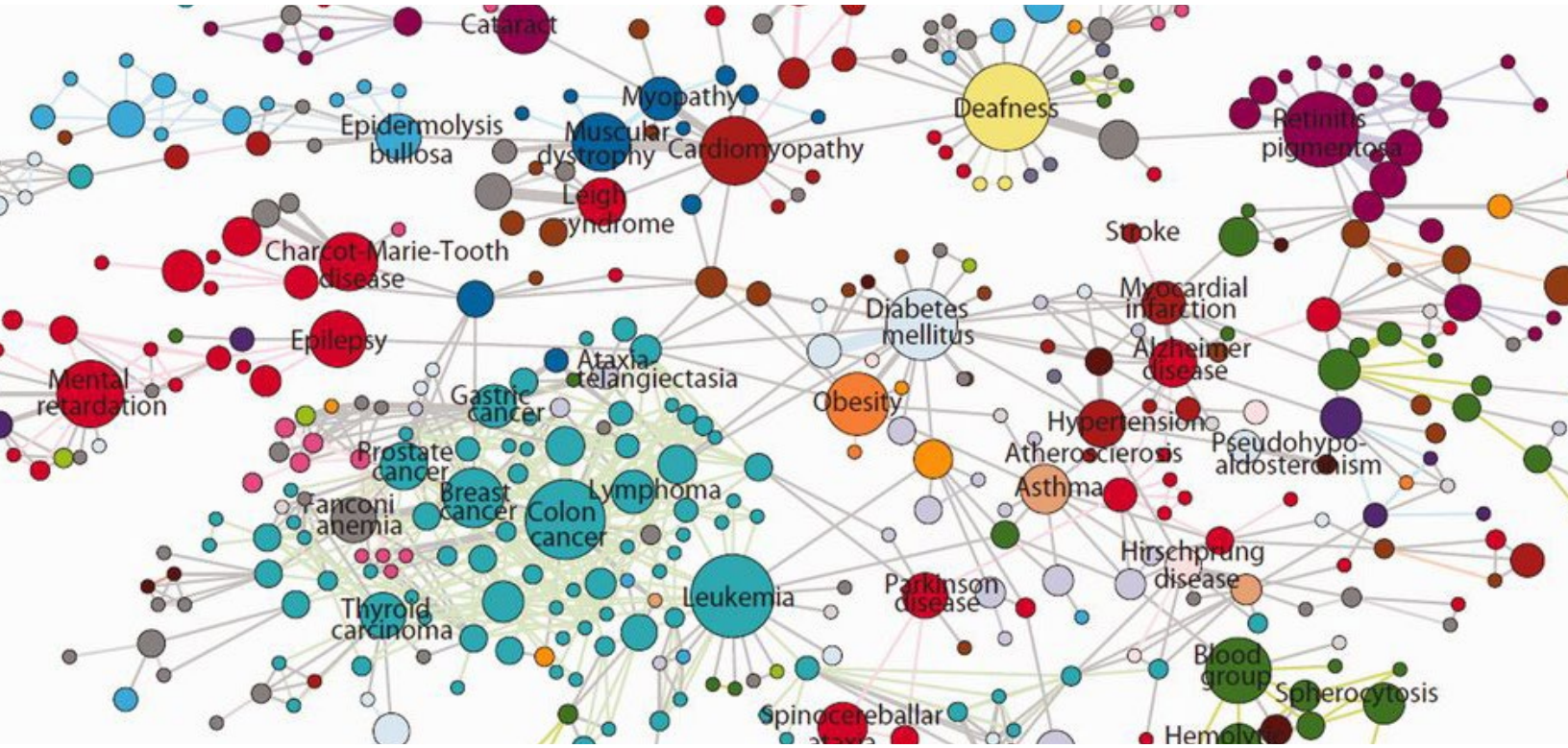
2

ALBERT-LÁSZLÓ BARABÁSI

NETWORK SCIENCE
GRAPH THEORY



ACKNOWLEDGEMENTS

MÁRTON PÓSFAI
GABRIELE MUSELLA
MAURO MARTINO
ROBERTA SINATRA
PHILIPP HOEVEL
SARAH MORRISON
AMAL HUSSEINI

Figure 2.0 (front cover)
Human Disease Network

The Human Disease Network, whose nodes are diseases connected if they have common genetic origin. Published as a supplement of the Proceedings of the National Academy of Sciences [1], the map was created to illustrate the genetic interconnectedness of apparently distinct diseases. With time it crossed disciplinary boundaries, taking up a life of its own. The New York Times created an interactive version of the map and the London-based Serpentine Gallery, one of the top contemporary art galleries in the world, have exhibited it part of their focus on networks and maps [2]. It is also featured in numerous books on design and maps [3, 4, 5].

PDF V27, 05.09.2014

THE BRIDGES OF KÖNIGSBERG

Few research fields can trace their birth to a single moment and place in history. Graph theory, the mathematical scaffold behind network science, can. Its roots go back to 1735 in Königsberg, the capital of Eastern Prussia, a thriving merchant city of its time. The trade supported by its busy fleet of ships allowed city officials to build seven bridges across the river Pregel that surrounded the town. Five of these connected to the mainland the elegant island Kneiphof, caught between the two branches of the Pregel. The remaining two crossed the two branches of the river (Figure 2.1). This peculiar arrangement gave birth to a contemporary puzzle: Can one walk across all seven bridges and never cross the same one twice? Despite many attempts, no one could find such path. The problem remained unsolved until 1735, when Leonard Euler, a Swiss born mathematician, offered a rigorous mathematical proof that such path does not exist [6, 7].

Euler represented each of the four land areas separated by the river with letters A, B, C, and D (Figure 2.1). Next he connected with lines each piece of land that had a bridge between them. He thus built a graph, whose nodes were pieces of land and links were the bridges. Then Euler made a simple observation: if there is a path crossing all bridges, but never the same bridge twice, then nodes with odd number of links must be either the starting or the end point of this path. Indeed, if you arrive to a node with an odd number of links, you may find yourself having no unused link for you to leave it.

A walking path that goes through all bridges can have only one starting and one end point. Thus such a path cannot exist on a graph that has more than two nodes with an odd number of links. The Königsberg graph had four nodes with an odd number of links, A, B, C, and D, so no path could satisfy the problem.

Euler's proof was the first time someone solved a mathematical problem using a graph. For us the proof has two important messages: The first is that some problems become simpler and more tractable if they are represented as a graph. The second is that the existence of the path does not

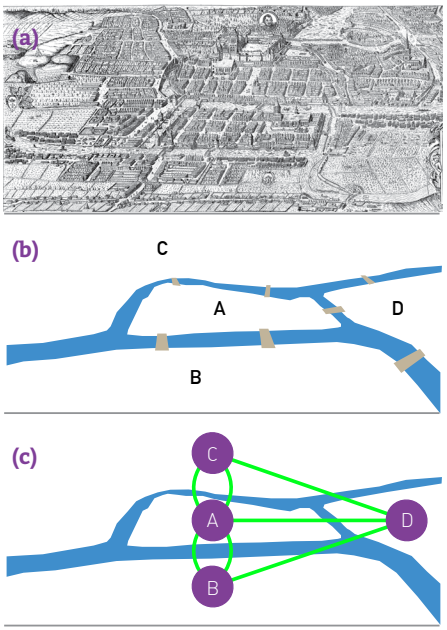


Figure 2.1
The Bridges of Königsberg

(a) A contemporary map of Königsberg (now Kaliningrad, Russia) during Euler's time.

(b) A schematic illustration of Königsberg's four land pieces and the seven bridges across them.

(c) Euler constructed a graph that has four nodes (A, B, C, D), each corresponding to a patch of land, and seven links, each corresponding to a bridge. He then showed that there is no continuous path that would cross the seven bridges while never crossing the same bridge twice. The people of Königsberg gave up their fruitless search and in 1875 built a new bridge between B and C, increasing the number of links of these two nodes to four. Now only one node was left with an odd number of links. Consequently we should be able to find the desired path. Can you find one yourself?

depend on our ingenuity to find it. Rather, it is a property of the graph. Indeed, given the structure of the Königsberg graph, no matter how smart we are, we will never find the desired path. In other words, networks have properties encoded in their structure that limit or enhance their behavior.

To understand the many ways networks can affect the properties of a system, we need to become familiar with graph theory, a branch of mathematics that grew out of Euler's proof. In this chapter we learn how to represent a network as a graph and introduce the elementary characteristics of networks, from degrees to degree distributions, from paths to distances and learn to distinguish weighted, directed and bipartite networks. We will introduce a graph-theoretic formalism and language that will be used throughout this book.

Online Resource 2.1

The Bridges of Königsberg

Watch a short video introducing the Königsberg problem and Euler's solution.

NETWORKS AND GRAPHS

If we want to understand a complex system, we first need to know how its components interact with each other. In other words we need a map of its wiring diagram. A network is a catalog of a system's components often called *nodes* or *vertices* and the direct interactions between them, called *links* or *edges* (BOX 2.1). This network representation offers a common language to study systems that may differ greatly in nature, appearance, or scope. Indeed, as shown in Figure 2.2, three rather different systems have exactly the same network representation.

Figure 2.2 introduces two basic network parameters:

Number of nodes, or N, represents the number of components in the system. We will often call N the *size of the network*. To distinguish the nodes, we label them with $i = 1, 2, \ldots, N$.

Number of links, which we denote with L, represents the total number of interactions between the nodes. Links are rarely labeled, as they can be identified through the nodes they connect. For example, the (2, 4) link connects nodes 2 and 4.

The networks shown in Figure 2.2 have $N = 4$ and $L = 4$.

The links of a network can be *directed* or *undirected*. Some systems have directed links, like the WWW, whose uniform resource locators (URL) point from one web document to the other, or phone calls, where one person calls the other. Other systems have undirected links, like romantic ties: if I date Janet, Janet also dates me, or like transmission lines on the power grid, on which the electric current can flow in both directions.

A network is called directed (or *digraph*) if all of its links are directed; it is called undirected if all of its links are undirected. Some networks simultaneously have directed and undirected links. For example in the metabolic network some reactions are reversible (i.e., bidirectional or undirected) and others are irreversible, taking place in only one direction (directed).

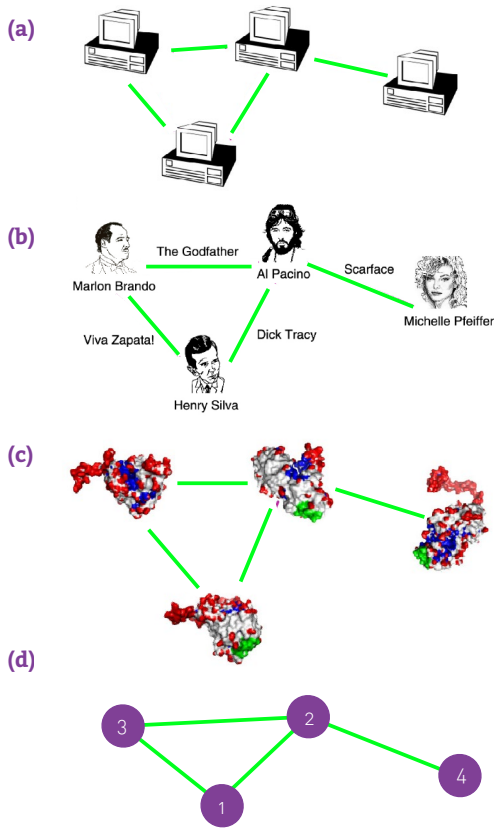


Figure 2.2
Different Networks, Same Graph

The figure shows a small subset of (a) the Internet, where routers (specialized computers) are connected to each other; (b) the Hollywood actor network, where two actors are connected if they played in the same movie; (c) a protein-protein interaction network, where two proteins are connected if there is experimental evidence that they can bind to each other in the cell. While the nature of the nodes and the links differs, these networks have the same graph representation, consisting of $N = 4$ nodes and $L = 4$ links, shown in (*d*).

The choices we make when we represent a system as a network will determine our ability to use network science successfully to solve a particular problem. For example, the way we define the links between two individuals dictates the nature of the questions we can explore:

(a) By connecting individuals that regularly interact with each other in the context of their work, we obtain the *organizational* or *professional network*, that plays a key role in the success of a company or an institution, and is of major interest to organizational research (Figure 1.7).

(b) By linking friends to each other, we obtain the *friendship network*, that plays an important role in the spread of ideas, products and habits and is of major interest to sociology, marketing and health sciences.

(c) By connecting individuals that have an intimate relationship, we obtain the *sexual network*, of key importance for the spread of sexually transmitted diseases, like AIDS, and of major interest for epidemiology.

(d) By using phone and email records to connect individuals that call or email each other, we obtain the *acquaintance network*, capturing a mixture of professional, friendship or intimate links, of importance to communications and marketing.

While many links in these four networks overlap (some coworkers may be friends or may have an intimate relationship), these networks have different uses and purposes.

We can also build networks that may be valid from a graph theoretic perspective, but may have little practical utility. For example, if we link all individuals with the same first name, Johns with Johns and Marys with Marys, we do obtain a well-defined graph, whose properties can be analyzed with the tools of network science. Its utility is questionable, however. Hence in order to apply network theory to a system, careful considerations must precede our choice of nodes and links, ensuring their significance to the problem we wish to explore.

Throughout this book we will use ten networks to illustrate the tools of network science. These *reference networks*, listed in Table 2.1, span social systems (mobile call graph or email network), collaboration and affiliation networks (science collaboration network, Hollywood actor network), information systems (WWW), technological and infrastructural systems (Internet and power grid), biological systems (protein interaction and metabolic network), and reference networks (citations). They differ widely in their sizes, from as few as N =1,039 nodes in the E. coli metabolism, to almost half million nodes in the citation network. They cover several areas where networks are actively applied, representing 'canonical' datasets frequently

BOX 2.1

NETWORKS OR GRAPHS?

In the scientific literature the terms *network* and *graph* are used interchangeably:

Network Science	Graph Theory
Network	Graph
Node	Vertex
Link	Edge

Yet, there is a subtle distinction between the two terminologies: the {*network, node, link*} combination often refers to real systems: The WWW is a network of web documents linked by URLs; society is a network of individuals linked by family, friendship or professional ties; the metabolic network is the sum of all chemical reactions that take place in a cell. In contrast, we use the terms {*graph, vertex, edge*} when we discuss the mathematical representation of these networks: We talk about the web graph, the social graph (a term made popular by Facebook), or the metabolic graph. Yet, this distinction is rarely made, so these two terminologies are often synonyms of each other.

used by researchers to illustrate key network properties. As we indicate in Table 2.1, some of them are directed, others are undirected. In the coming chapters we will discuss in detail the nature and the characteristics of each of these datasets, turning them into the guinea pigs of our journey to understand complex networks.

NETWORK	NODES	LINKS	DIRECTED UNDIRECTED	N	L	⟨k⟩
Internet	Routers	Internet connections	Undirected	192,244	609,066	6.34
WWW	Webpages	Links	Directed	325,729	1,497,134	4.60
Power Grid	Power plants, transformers	Cables	Undirected	4,941	6,594	2.67
Mobile Phone Calls	Subscribers	Calls	Directed	36,595	91,826	2.51
Email	Email addresses	Emails	Directed	57,194	103,731	1.81
Science Collaboration	Scientists	Co-authorship	Undirected	23,133	93,439	8.08
Actor Network	Actors	Co-acting	Undirected	702,388	29,397,908	83.71
Citation Network	Paper	Citations	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Metabolites	Chemical reactions	Directed	1,039	5,802	5.58
Protein Interactions	Proteins	Binding interactions	Undirected	2,018	2,930	2.90

Table 2.1
Canonical Network Maps

The basic characteristics of ten networks used throughout this book to illustrate the tools of network science. The table lists the nature of their nodes and links, indicating if links are directed or undirected, the number of nodes (N) and links (L), and the average degree for each network. For directed networks the average degree shown is the average in- or out-degrees $\langle k \rangle = \langle k_{in} \rangle = \langle k_{out} \rangle$ (see Equation (2.5)).

DEGREE, AVERAGE DEGREE, AND DEGREE DISTRIBUTION

A key property of each node is its *degree*, representing the number of links it has to other nodes. The degree can represent the number of mobile phone contacts an individual has in the call graph (i.e. the number of different individuals the person has talked to), or the number of citations a research paper gets in the citation network.

DEGREE

We denote with k_i the degree of the i^{th} node in the network. For example, for the undirected networks shown in Figure 2.2 we have k_1=2, k_2=3, k_3=2, k_4=1. In an undirected network the *total number of links*, L, can be expressed as the sum of the node degrees:

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i \,. \tag{2.1}$$

Here the 1/2 factor corrects for the fact that in the sum (2.1) each link is counted twice. For example, the link connecting the nodes 2 and 4 in Figure 2.2 will be counted once in the degree of node 1 and once in the degree of node 4.

AVERAGE DEGREE

An important property of a network is its *average degree* (BOX 2.2), which for an undirected network is

$$\langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2L}{N} \,. \tag{2.2}$$

In directed networks we distinguish between *incoming degree*, k_i^{in}, representing the number of links that point to node i, and *outgoing degree*, k_i^{out}, representing the number of links that point from node i to other nodes. Finally, a node's *total degree*, k_i, is given by

$$k_i = k_i^{\text{in}} + k_i^{\text{out}}. \tag{2.3}$$

For example, on the WWW the number of pages a given document points to represents its outgoing degree, k^{out}, and the number of documents that point to it represents its incoming degree, k^{in}. The total number

BOX 2.2

BRIEF STATISTICS REVIEW

Four key quantities characterize a sample of N values $x_1, \ldots, x_N$:

Average (mean):

$$\langle x \rangle = \frac{x_1 + x_2 + \ldots + x_N}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

The n^{th} moment:

$$\langle x^n \rangle = \frac{x_1^n + x_2^n + \ldots + x_N^n}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i^n$$

Standard deviation:

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \langle x \rangle\right)^2}$$

Distribution of x:

$$p_x = \frac{1}{N}\sum_i \delta_{x,x_i} \,.$$

where p_x follows

$$\sum_i p_x = 1 \left(\int p_x \, dx = 1\right)$$

of links in a directed network is

$$L = \sum_{i=1}^{N} k_i^{in} = \sum_{i=1}^{N} k_i^{out} . \tag{2.4}$$

The 1/2 factor seen in (2.1) is now absent, as for directed networks the two sums in (2.4) separately count the outgoing and the incoming degrees. The average degree of a directed network is

$$\langle k^{in} \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i^{in} = \langle k^{out} \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i^{out} = \frac{L}{N} \tag{2.5}$$

DEGREE DISTRIBUTION

The *degree distribution*, p_k, provides the probability that a randomly selected node in the network has degree k. Since p_k is a probability, it must be normalized, i.e.

$$\sum_{k=1}^{\infty} p_k = 1 . \tag{2.6}$$

For a network with N nodes the degree distribution is the normalized histogram (Figure 2.3) is given by

$$p_k = \frac{N_k}{N} , \tag{2.7}$$

where N_k is the number of degree-k nodes. Hence the number of degree-k nodes can be obtained from the degree distribution as $N_k = Np_k$.

The degree distribution has assumed a central role in network theory following the discovery of scale-free networks [8]. One reason is that the calculation of most network properties requires us to know p_k. For example, the average degree of a network can be written as

$$\langle k \rangle = \sum_{k=0}^{\infty} k p_k . \tag{2.8}$$

The other reason is that the precise functional form of p_k determines many network phenomena, from network robustness to the spread of viruses.

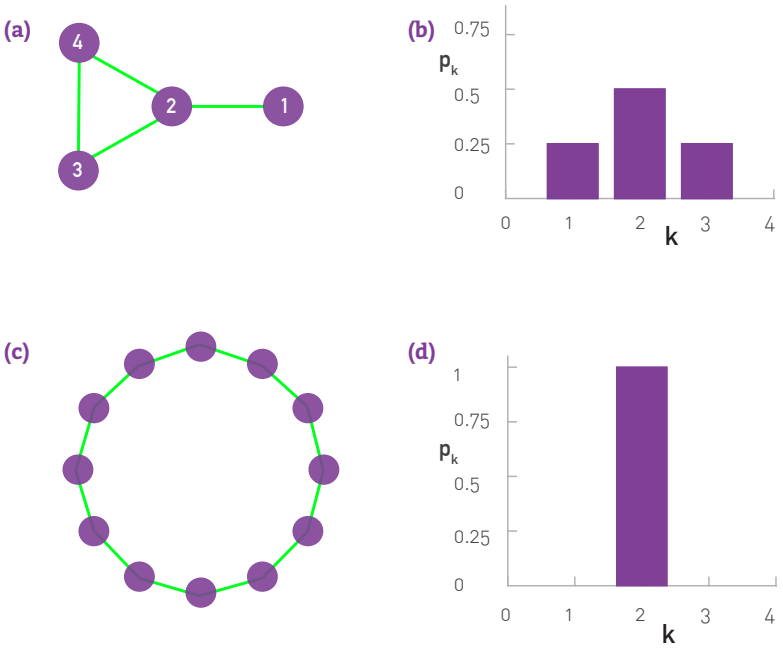


Figure 2.3
Degree Distribution

The degree distribution of a network is provided by the ratio (2.7).

(a) For the network in (a) with $N = 4$ the degree distribution is shown in (b).

(b) We have $p_1 = 1/4$ (one of the four nodes has degree $k_1 = 1$), $p_2 = 1/2$ (two nodes have $k_3 = k_4 = 2$), and $p_3 = 1/4$ (as $k_2 = 3$). As we lack nodes with degree $k > 3$, $p_k = 0$ for any $k > 3$.

(c) A one dimensional lattice for which each node has the same degree $k = 2$.

(d) The degree distribution of (c) is a Kronecker's delta function, $p_k = \delta(k - 2)$.

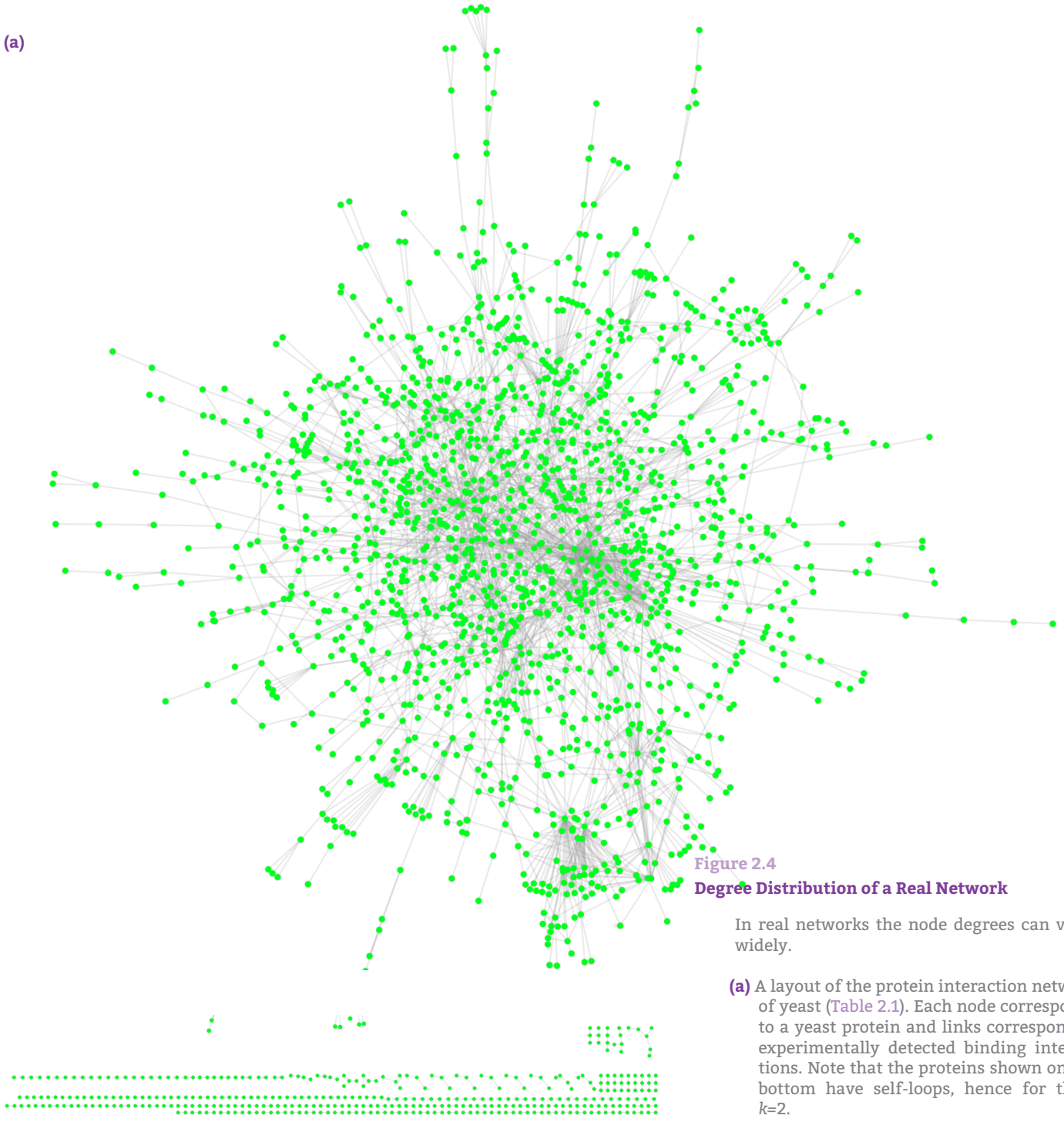


Figure 2.4
Degree Distribution of a Real Network

In real networks the node degrees can vary-widely.

(a) A layout of the protein interaction network of yeast (Table 2.1). Each node corresponds to a yeast protein and links correspond to experimentally detected binding interactions. Note that the proteins shown on the bottom have self-loops, hence for them k=2.

(b) The degree distribution of the protein interaction network shown in (a). The observed degrees vary between k=0 (isolated nodes) and k=92, which is the degree of the most connected node, called a *hub*. There are also wide differences in the number of nodes with different degrees: Almost half of the nodes have degree one (i.e. p_1=0.48), while we have only one copy of the biggest node (i.e. $p_{92} = 1/N$=0.0005).

(c) The degree distribution is often shown on a log-log plot, in which we either plot log p_k in function of ln k, or, as we do in (c), or we use logarithmic axes. The advantages of this representation are discussed in Chapter 4.

ADJACENCY MATRIX

A complete description of a network requires us to keep track of its links. The simplest way to achieve this is to provide a complete list of the links. For example, the network of Figure 2.2 is uniquely described by listing its four links: {(1, 2), (1, 3), (2, 3), (2, 4)}. For mathematical purposes we often represent a network through its adjacency matrix. The *adjacency matrix* of a directed network of N nodes has N rows and N columns, its elements being:

$A_{ij} = 1$ if there is a link pointing from node j to node i
$A_{ij} = 0$ if nodes i and j are not connected to each other

The adjacency matrix of an undirected network has two entries for each link, e.g. link (1, 2) is represented as $A_{12} = 1$ and $A_{21} = 1$. Hence, the adjacency matrix of an undirected network is symmetric, $A_{ij} = A_{ji}$ (Figure 2.5b).

The degree k_i of node i can be directly obtained from the elements of the adjacency matrix. For undirected networks a node's degree is a sum over either the rows or the columns of the matrix, i.e.

$$k_i = \sum_{j=1}^{N} A_{ji} = \sum_{i=1}^{N} A_{ji} \ . \tag{2.9}$$

For directed networks the sums over the adjacency matrix' rows and columns provide the incoming and outgoing degrees, respectively

$$k_i^{\text{in}} = \sum_{j=1}^{N} A_{ij} \ , \qquad k_i^{\text{out}} = \sum_{j=1}^{N} A_{ji} \ . \tag{2.10}$$

Given that in an undirected network the number of outgoing links equals the number of incoming links, we have

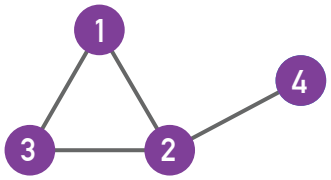
$$2L = \sum_{i=1}^{N} k_i^{\text{in}} = \sum_{i=1}^{N} k_i^{\text{out}} = \sum_{ij}^{N} A_{ij} \ . \tag{2.11}$$

The number of nonzero elements of the adjacency matrix is $2L$, or twice the number of links. Indeed, an undirected link connecting nodes i and j appears in two entries: $A_{ij} = 1$, a link pointing from node j to node i, and $A_{ji} = 1$, a link pointing from i to j (Figure 2.5b).

(a) Adjacency matrix

$$A_{ij} = \begin{matrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{matrix}$$

(b) Undirected network



$$A_{ij} = \begin{matrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

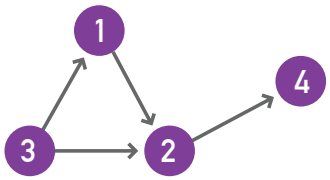
$$k_2 = \sum_{j=1}^{4} A_{2j} = \sum_{i=1}^{4} A_{i2} = 3$$

$$A_{ij} = A_{ji} \qquad A_{ii} = 0$$

$$L = \frac{1}{2}\sum_{i}\sum_{i=1}^{N} A_{ij}$$

$$\langle k \rangle = \frac{2L}{N}$$

(c) Directed network



$$A_{ij} = \begin{matrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

$$k_2^{\text{in}} = \sum_{j=1}^{4} A_{2j} = 2,\ k_2^{\text{out}} = \sum_{i=1}^{4} A_{i2} = 1$$

$$A_{ij} \neq A_{ji} \qquad A_{ii} = 0$$

$$L = \sum_{i,j=1}^{N} A_{ij}$$

$$\langle k^{\text{in}} \rangle = \langle k^{\text{out}} \rangle = \frac{L}{N}$$

Figure 2.5

The Adjacency Matrix

(a) The labeling of the elements of the adjacency matrix.

(b) The adjacency matrix of an *undirected network*. The figure shows that the degree of a node (in this case node 2) can be expressed as the sum over the appropriate column or the row of the adjacency matrix. It also shows a few basic network characteristics, like the total number of links, L, and average degree, $\langle k \rangle$, expressed in terms of the elements of the adjacency matrix.

(c) The same as in **(b)** but for a *directed network*.

REAL NETWORKS ARE SPARSE

In real networks the number of nodes (N) and links (L) can vary widely. For example, the neural network of the worm C. elegans, the only fully mapped nervous system of a living organism, has $N = 302$ neurons (nodes). In contrast the human brain is estimated to have about a hundred billion ($N \approx 10^{11}$) neurons. The genetic network of a human cell has about 20,000 genes as nodes; the social network consists of seven billion individuals ($N \approx 7\times10^{9}$) and the WWW is estimated to have over a trillion web documents ($N > 10^{12}$).

These wide differences in size are noticeable in Table 2.1, which lists N and L for several network maps. Some of these maps offer a complete wiring diagram of the system they describe (like the actor network or the E. coli metabolism), while others are only samples, representing a subset of the full network (like the WWW or the mobile call graph).

Table 2.1 indicates that the number of links also varies widely. In a network of N nodes the number of links can change between $L = 0$ and L_{max}, where

$$L_{max} = \binom{N}{2} = \frac{N(N-1)}{2} \qquad (2.12)$$

is the total number of links present in a *complete graph* of size N (Figure 2.6). In a complete graph each node is connected to every other node.

In real networks L is much smaller than L_{max}, reflecting the fact that most real networks are sparse. We call a network *sparse* if $L << L_{max}$. For example, the WWW graph in Table 2.1 has about 1.5 million links. Yet, if the WWW were to be a complete graph, it should have $L_{max} \approx 5\times10^{10}$ links according to (2.12). Consequently the web graph has only a 3×10^{-5} fraction of the links it could have. This is true for all of the networks in Table 2.1: One can check that their number of links is only a tiny fraction of the expected number of links for a complete graph of the same number of nodes.

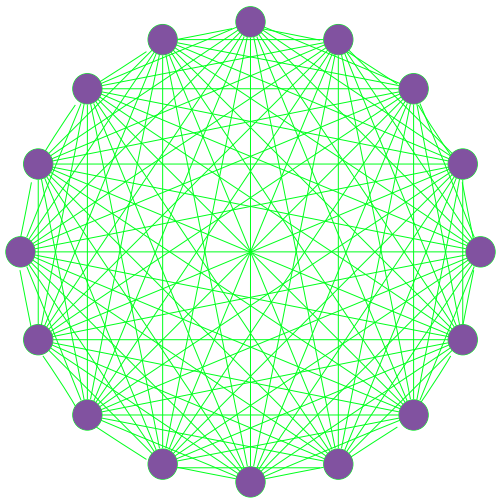


Figure 2.6
Complete Graph

A complete graph with $N = 16$ nodes and $L_{max} = 120$ links, as predicted by (2.12). The adjacency matrix of a complete graph is $A_{ij} = 1$ for all $i, j = 1, \ldots N$ and $A_{ii} = 0$. The average degree of a complete graph is $\langle k \rangle = N - 1$. A complete graph is often called a *clique*, a term frequently used in community identification, a problem discussed in CHAPTER 9.

The sparsity of real networks implies that the adjacency matrices are also sparse. Indeed, a complete network has $A_{ij} = 1$, for all (i, j), i.e. each of its matrix elements are equal to one. In contrast in real networks only a tiny fraction of the matrix elements are nonzero. This is illustrated in Figure 2.7, which shows the adjacency matrix of the protein-protein interaction network listed in Table 2.1 and shown in Figure 2.4a. One can see that the matrix is nearly empty.

Sparseness has important consequences on the way we explore and store real networks. For example, when we store a large network in our computer, it is better to store only the list of links (i.e. elements for which $A_{ij} \neq 0$), rather than the full adjacency matrix, as an overwhelming fraction of the A_{ij} elements are zero. Hence the matrix representation will block a huge chunk of memory, filled mainly with zeros (Figure 2.7).

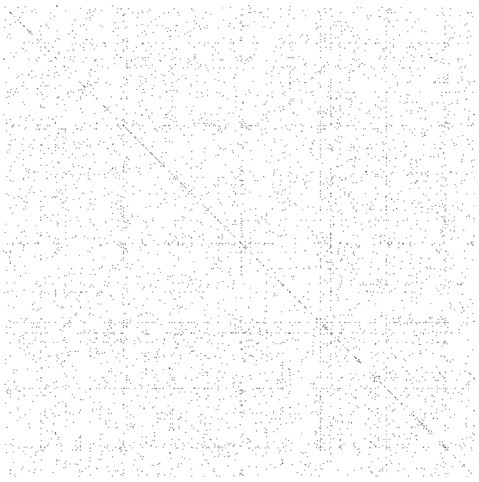


Figure 2.7

The Adjacency Matrix is Sparse

The adjacency matrix of the yeast protein-protein interaction network, consisting of 2,018 nodes, each representing a yeast protein (Table 2.1). A dot is placed on each position of the adjacent matrix for which $A_{ij} = 1$, indicating the presence of an interaction. There are no dots for $A_{ij} = 0$. The small fraction of dots illustrates the sparse nature of the protein-protein interaction network.

WEIGHTED NETWORKS

So far we discussed only networks for which all links have the same weight, i.e. $A_{ij} = 1$. In many applications we need to study *weighted networks*, where each link (i, j) has a unique weight w_{ij}. In mobile call networks the weight can represent the total number of minutes two individuals talk with each other on the phone; on the power grid the weight is the amount of current flowing through a transmission line.

For *weighted networks* the elements of the adjacency matrix carry the weight of the link as

$$A_{ij} = w_{ij} \, . \tag{2.13}$$

Most networks of scientific interest are weighted, but we can not always measure the appropriate weights. Consequently we often approximate these networks with an unweighted graph. In this book we predominantly focus on unweighted networks, but whenever appropriate, we discuss how the weights alter the corresponding network property (BOX 2.3).

BOX 2.3

METCALFE'S LAW: THE VALUE OF A NETWORK

Metcalfe's law states that the *value of a network* is proportional to the square of the number of its nodes, i.e. N^2. Formulated around 1980 in the context of communication devices by Robert M. Metcalfe [9], the idea behind Metcalfe's law is that the more individuals use a network, the more valuable it becomes. Indeed, the more of your friends use email, the more valuable the service is to you.

During the Internet boom of the late 1990s Metcalfe's law was frequently used to offer a quantitative valuation for Internet companies. It suggested that the value of a service is proportional to the number of connections it can create, which is the square of the number of its users. In contrast the cost grows only linearly with N. Hence if the service attracts sufficient number of users, it will inevitably become profitable, as N^2 will surpass N at some large N (Figure 2.8). Metcalfe's Law therefore supported a "build it and they will come" mentality [10], offering credibility to growth over profits.

Metcalfe's law is based on (2.12), telling us that if *all links* of a communication network with N users are equally valuable, the total value of the network is proportional to $N(N-1)/2$, that is, roughly, N^2. If a network has $N = 10$ consumers, there are $L_{max} = 45$ different possible connections between them. If the network doubles in size to $N = 20$, the number of connections doesn't merely double but roughly quadruples to 190, a phenomenon called *network externality* in economics.

Two issues limit the validity of Metcalfe's law:

(a) Most real networks are sparse, which means that only a very small fraction of the links are present. Hence the value of the network does not grow like N^2, but increases only linearly with N.

(b) As the links have weights, not all links are of equal value. Some links are used heavily while the vast majority of links are rarely utilized.

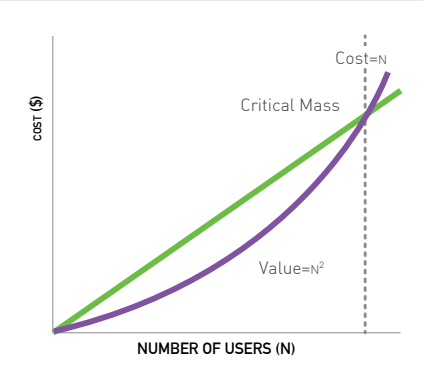


Figure 2.8
Metcalfe's Law

According to Metcalfe's law the *cost* of network based services increases linearly with the number of nodes (users or devices). In contrast the *benefits* or *income* are driven by the number of links L_{max} the technology makes possible, which grows like N^2 according to (2.12). Hence once the number of users or devices exceeds some *critical mass*, the technology becomes profitable.

BIPARTITE NETWORKS

A *bipartite graph* (or *bigraph*) is a network whose nodes can be divided into two disjoint sets U and V such that each link connects a U-node to a V-node. In other words, if we color the U-nodes green and the V-nodes purple, then each link must connect nodes of different colors (Figure 2.9).

We can generate two *projections* for each bipartite network. The first projection connects two U-nodes by a link if they are linked to the same V-node in the bipartite representation. The second projection connects the V-nodes by a link if they connect to the same U-node (Figure 2.9).

In network theory we encounter numerous bipartite networks. A well-known example is the Hollywood actor network, in which one set of nodes corresponds to movies (U), and the other to actors (V). A movie is connected to an actor if the actor plays in that movie. One projection of this bipartite network is the *actor network*, in which two nodes are connected to each other if they played in the same movie. This is the network listed in Table 2.1. The other projection is the *movie network*, in which two movies are connected if they share at least one actor in their cast.

Medicine offers another prominent example of a bipartite network: The *Human Disease Network* connects diseases to the genes whose mutations are known to cause or effect the corresponding disease (Figure 2.10).

Finally, one can also define *multipartite networks*, like the *tripartite* recipe-ingredient-compound network shown in Figure 2.11.

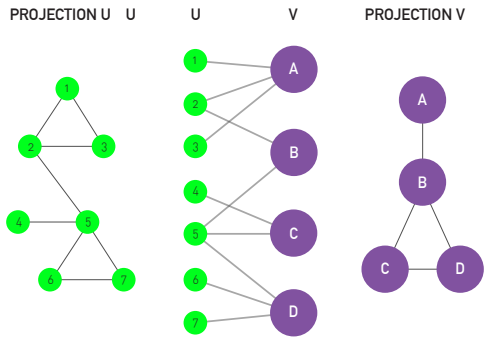
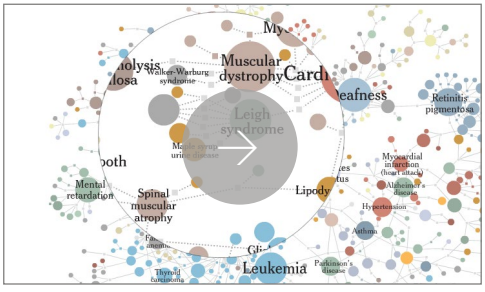


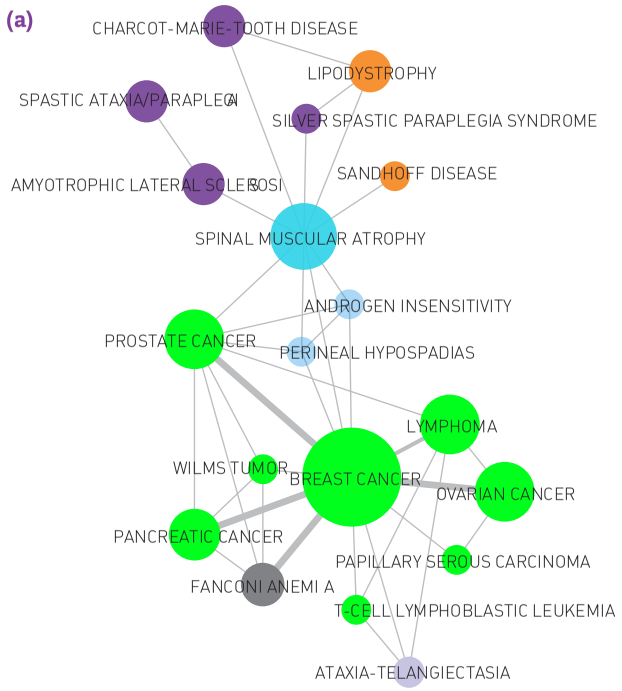
Figure 2.9
Bipartite Network

A bipartite network has two sets of nodes, U and V. Nodes in the U-set connect directly only to nodes in the V-set. Hence there are no direct U-U or V-V links. The figure shows the two projections we can generate from any bipartite network. Projection U is obtained by connecting two U-nodes to each other if they link to the same V-node in the bipartite representation. Projection V is obtained by connecting two V-nodes to each other if they link to the same U-node in the bipartite network.

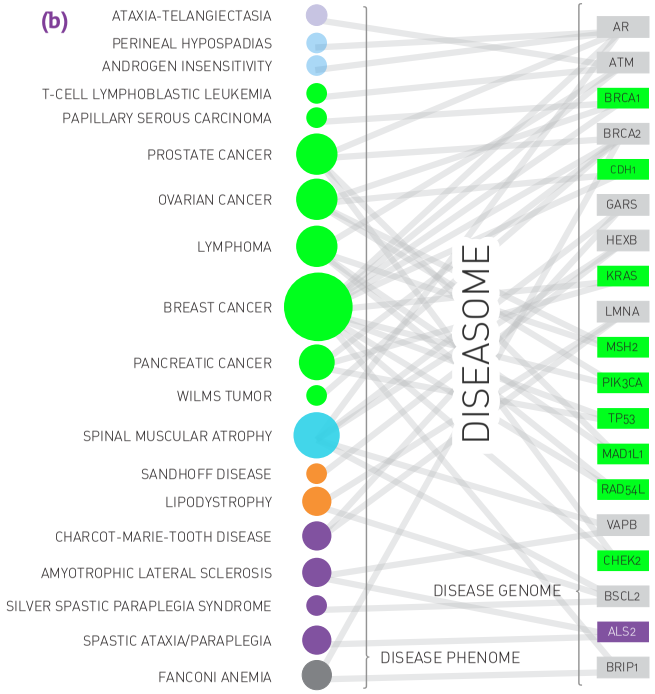


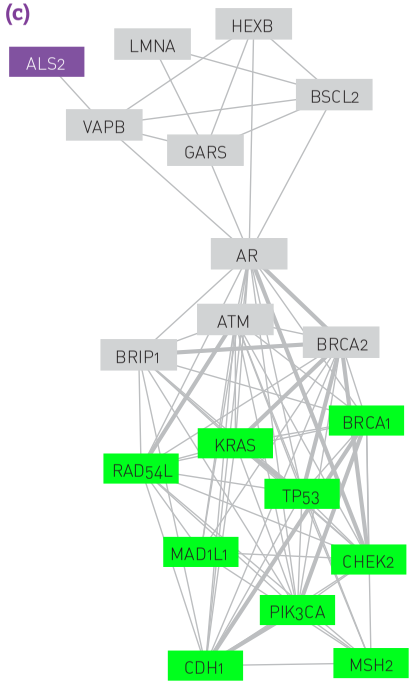
Online Resource 2.2
Human Disease Network

Download the high resolution version of the Human Disease Network [1], or explore it using the online interface built by the New York Times.



HUMAN DISEASE NETWORK





DISEASE GENE NETWORK

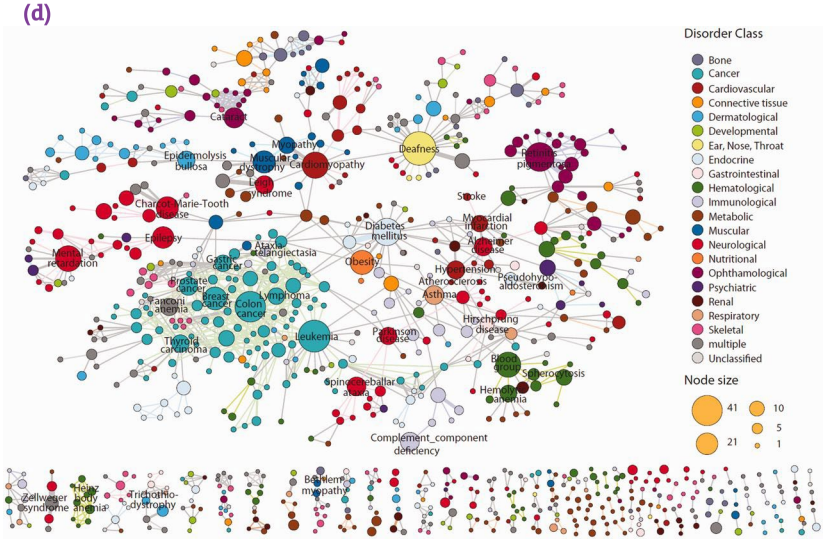


Figure 2.10
Human Disease Network

(a) One projection of the diseaseome is the *gene network*, whose nodes are genes, and where two genes are connected if they are associated with the same disease.

(b) The Human Disease Network (or *diseaseome*) is a bipartite network, whose nodes are diseases (U) and genes (V). A disease is connected to a gene if mutations in that gene are known to affect the particular disease [4].

(c) The second projection is the *disease network*, whose nodes are diseases. Two diseases are connected if the same genes are associated with them, indicating that the two diseases have common genetic origin. Figures (a)-(c) shows a subset of the diseaseome, focusing on cancers.

(d) The full diseaseome, connecting 1,283 disorders via 1,777 shared disease genes. After [1]. See **Online Resource 2.2** for the detailed map.

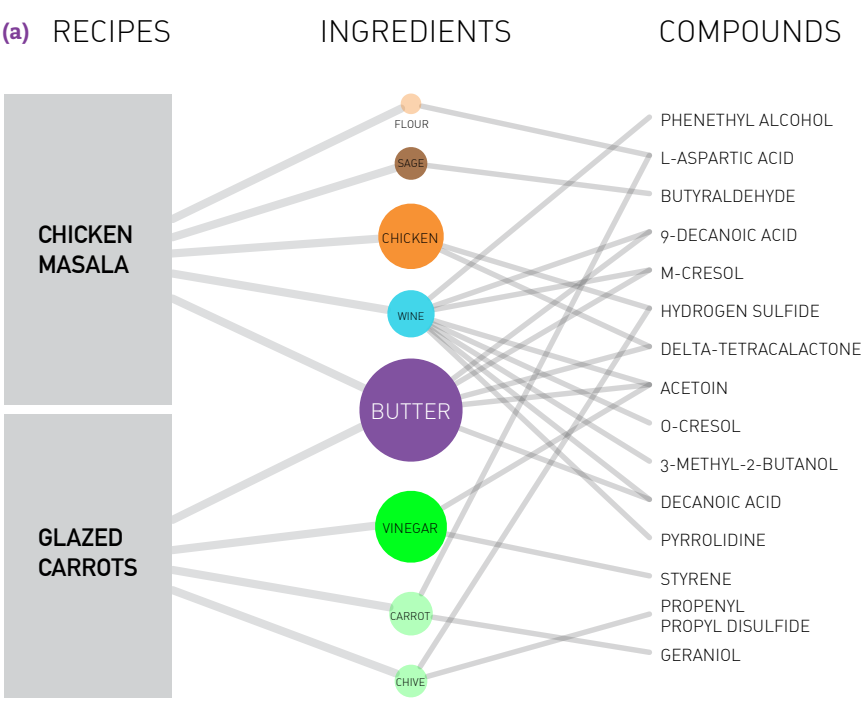
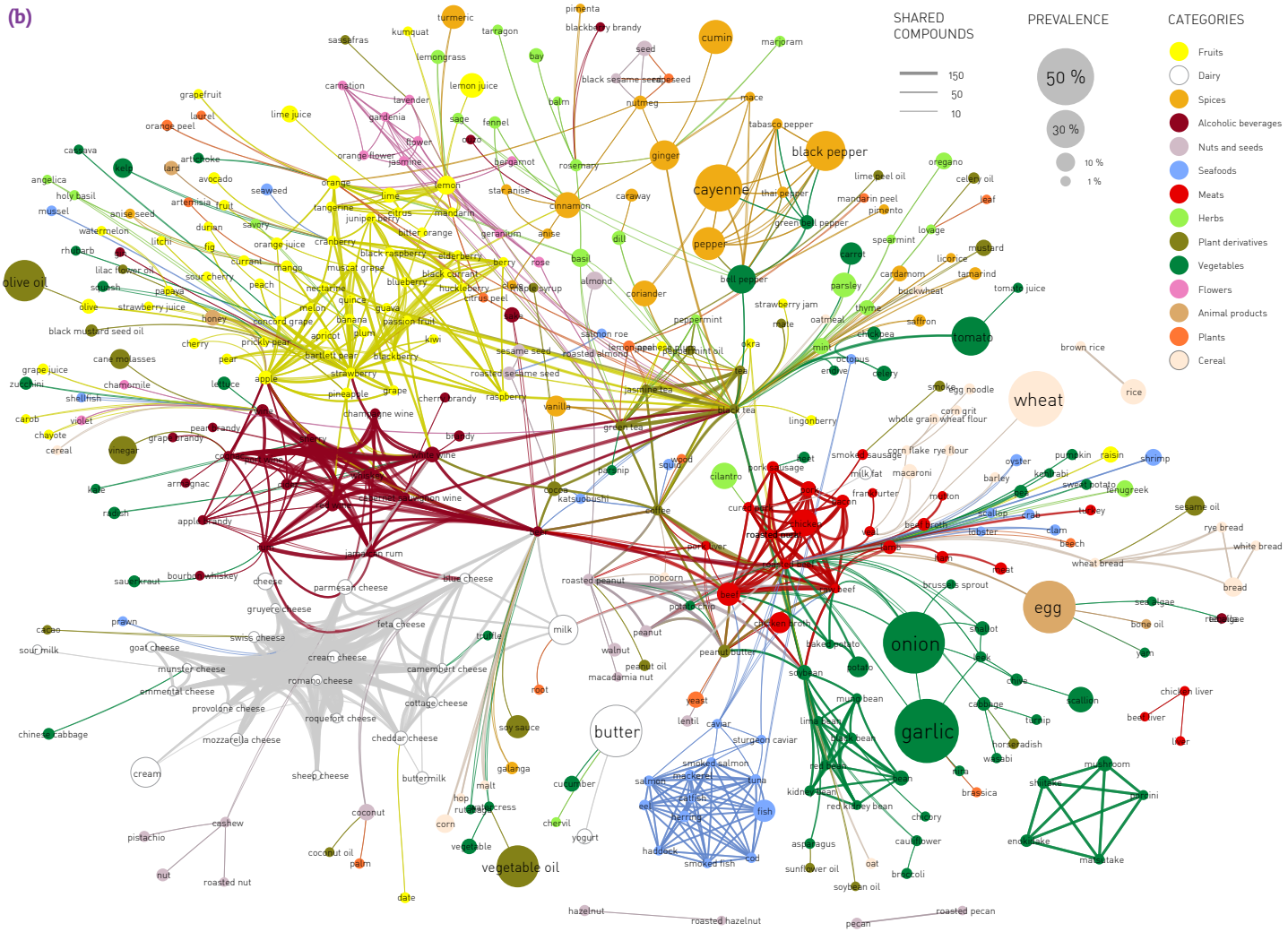


Figure 2.11
Tripartite Network

(a) The construction of the tripartite recipe-ingredient-compound network, in which one set of nodes are recipes, like Chicken Marsala; the second set corresponds to the ingredients each recipe has (like flour, sage, chicken, wine, and butter for Chicken Marsala); the third set captures the flavor compounds, or chemicals that contribute to the taste of each ingredient.

(b) The *ingredient* or the *flavor network* represents a projection of the tripartite network. Each node denotes an ingredient; the node color indicating the food category and node size indicates the ingredient's prevalence in recipes. Two ingredients are connected if they share a significant number of flavor compounds. Link thickness represents the number of shared compounds.

After [11].



PATHS AND DISTANCES

Physical distance plays a key role in determining the interactions between the components of physical systems. For example the distance between two atoms in a crystal or between two galaxies in the universe determine the forces that act between them.

In networks distance is a challenging concept. Indeed, what is the distance between two webpages, or between two individuals who do not know each other? The physical distance is not relevant here: Two webpages could be sitting on computers on the opposite sides of the globe, yet, have a link to each other. At the same time two individuals that live in the same building may not know each other.

In networks physical distance is replaced by *path length*. A *path* is a route that runs along the links of the network. A path's *length* represents the number of links the path contains (Figure 2.12a). Note that some texts require that each node a path visits is distinct.

In network science paths play a central role. Next we discuss some of their most important properties, many more being summarized in Figure 2.13.

SHORTEST PATH

The shortest path between nodes i and j is the path with the fewest number of links (Figure 2.12b). The shortest path is often called the distance between nodes i and j, and is denoted by d_{ij}, or simply d. We can have multiple shortest paths of the same length d between a pair of nodes (Figure 2.12b). The shortest path never contains loops or intersects itself.

In an undirected network $d_{ij} = d_{ji}$, i.e. the distance between node i and j is the same as the distance between node j and i. In a directed network often $d_{ij} \neq d_{ji}$. Furthermore, in a directed network the existence of a path from node i to node j does not guarantee the existence of a path from j to i.

In real networks we often need to determine the distance between two

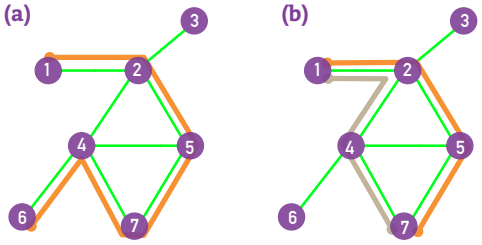
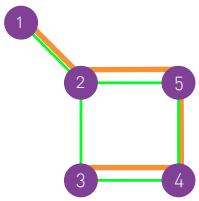


Figure 2.12
Paths

(a) A path between nodes i_0 and i_n is an ordered list of n links P = {(i_0, i_1), (i_1, i_2), (i_2, i_3), ... ,(i_n-1, i_n)}. The length of this path is n. The path shown in orange in (a) follows the route 1→2→5→7→4→6, hence its length is n = 5.

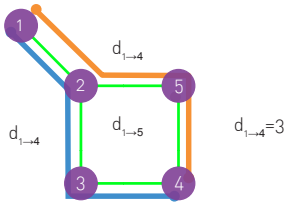
(b) The shortest paths between nodes 1 and 7, or the distance d_{17}, correspond to the path with the fewest number of links that connect nodes 1 to 7. There can be multiple paths of the same length, as illustrated by the two paths shown in orange and grey. The network diameter is the largest distance in the network, being d_{max} = 3 here.

(a)



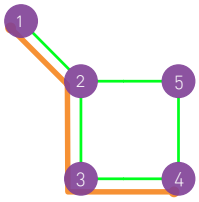
Path
A sequence of nodes such that each node is connected to the next node along the path by a link. Each path consists of n+1 nodes and n links. The length of a path is the number of its links, counting multiple links multiple times. For example, the orange line 1 → 2 → 5 → 4 → 3 covers a path of length four.

(b)



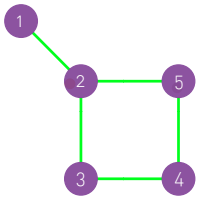
Shortest Path (Geodesic Path, d)
The path with the shortest distance d between two nodes. We also call d the distance between two nodes. Note that the shortest path does not need to be unique: between nodes 1 and 4 we have two shortest paths, 1→ 2→ 3→ 4 (blue) and 1→ 2→ 5→ 4 (orange), having the same length $d_{1,4}=3$.

(c)



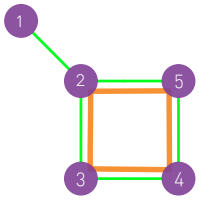
Diameter (d_{max})
The longest shortest path in a graph, or the distance between the two furthest nodes. In the graph shown here the diameter is between nodes 1 and 4, hence $d_{max}=3$.

(d)



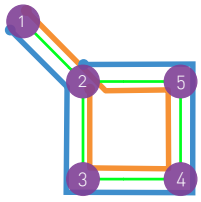
Average Path Length ($\langle d\rangle$)
The average of the shortest paths between all pairs of nodes. For the graph shown on the left we have $\langle d\rangle=1.6$, whose calculation is shown next to the figure.

(e)



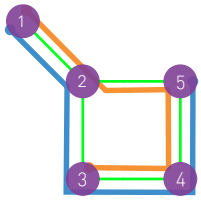
Cycle
A path with the same start and end node. In the graph shown on the left we have only one cycle, as shown by the orange line.

(f)



Eulerian Path
A path that traverses each link exactly once. The image shows two such Eulerian paths, one in orange and the other in blue.

(g)



Hamiltonian Path
A path that visits each node exactly once. We show two Hamiltonian paths in orange and in blue.

BOX 2.4

NUMBER OF SHORTEST PATHS BETWEEN TWO NODES

The number of shortest paths, N_{ij}, and the distance d_{ij} between nodes i and j can be calculated directly from the adjacency matrix A_{ij} .

$d_{ij} = 1$: If there is a direct link between i and j, then $A_{ij} = 1$ ($A_{ij} = 0$ otherwise).

$d_{ij} = 2$: If there is a path of length two between i and j, then $A_{ik}A_{kj}=1$ ($A_{ik}\ A_{kj} = 0$ otherwise). The number of $d_{ij} = 2$ paths between i and j is

$$N_{ij}^{(2)} = \sum_{k=1}^{N} A_{ik}A_{kj} = \left[A^2\right]_{ij}$$

where $[...]_{ij}$ denotes the $(ij)^{\text{th}}$ element of a matrix.

$d_{ij} = d$: If there is a path of length d between i and j, then $A_{ik} \ldots A_{lj} = 1$ ($A_{ik} \ldots A_{lj} = 0$ otherwise). The number of paths of length d between i and j is

$$N_{ij}^{(d)} = \left[A^d\right]_{ij}.$$

These equations hold for directed and undirected networks. The *distance* between nodes i and j is the path with the smallest d for which $N_{ij}^{(d)} > 0$. Despite the elegancy of this approach, faced with a large network, it is more efficient to use the breadth-first-search algorithm described in BOX 2.5.

nodes. For a small network, like the one shown in Figure 2.12, this is an easy task. For a network with millions of nodes finding the shortest path between two nodes can be rather time consuming. The length of the shortest path and the number of such paths can be formally obtained from the adjacency matrix (BOX 2.4). In practice we use the breadth first search (BFS) algorithm discussed in BOX 2.5 for this purpose.

NETWORK DIAMETER

The *diameter* of a network, denoted by $d_{\max}$, is the maximum shortest path in the network. In other words, it is the largest distance recorded between *any* pair of nodes. One can verify that the diameter of the network shown in Figure 2.13 is $d_{\max} = 3$. For larger networks the diameter can be determined using the BFS algorithm described in BOX 2.5.

AVERAGE PATH LENGTH

The *average path length*, denoted by $\langle d \rangle$, is the average distance between all pairs of nodes in the network. For a directed network of N nodes, $\langle d \rangle$ is

$$d = \frac{1}{N(N-1)} \sum_{\substack{i,j=1,N \\ i \neq j}} d_{i,j} \, . \tag{2.14}$$

Note that (2.14) is measured only for node pairs that are in the same component (SECTION 2.9). We can use the BFS algorithm to determine the average path length for a large network. For this we first determine the distances between the first node and all other nodes in the network using the algorithm described in BOX 2.5. We then determine the distances between the second node and all other nodes but the first one (if the network is undirected). We then repeat this procedure for all nodes. The sum

BOX 2.5

BREADTH-FIRST SEARCH (BFS) ALGORITHM

BFS is a frequently used algorithms in network science. Similar to throwing a pebble in a pond and watching the ripples spread from it, BFS starts from a node and labels its neighbors, then the neighbors' neighbors, until it reaches the target node. The number of "ripples" needed to reach the target provides the distance.

The identification of the shortest path between node i and j follows the following steps (Figure 2.14):

1. Start at node i, that we label with "0".

2. Find the nodes directly linked to i. Label them distance "1" and put them in a queue.

3. Take the first node, labeled n, out of the queue ($n = 1$ in the first step). Find the unlabeled nodes adjacent to it in the graph. Label them with $n + 1$ and put them in the queue.

4. Repeat step 3 until you find the target node j or there are no more nodes in the queue.

5. The distance between i and j is the label of j. If j does not have a label, then $d_{ij} = \infty$.

The computational complexity of the BFS algorithm, representing the approximate number of steps the computer needs to find d_{ij} on a network of N nodes and L links, is $O(N + L)$. It is linear in N and L as each node needs to be entered and removed from the queue at most once, and each link has to be tested only once.

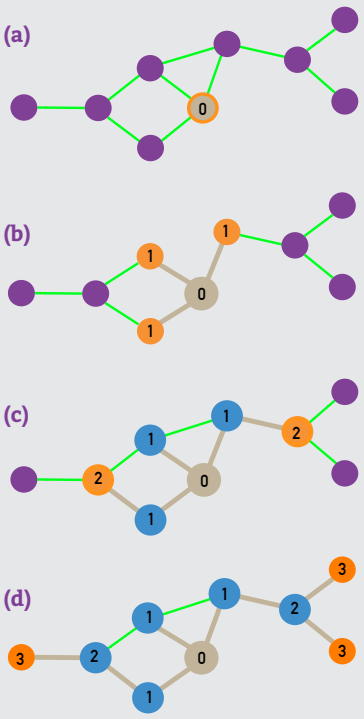


Figure 2.14
Applying the BFS Algorithm

(a) Starting from the orange node, labeled "0", we identify all its neighbors, labeling them "1".

(b)-(d) Next we label "2" the unlabeled neighbors of all nodes labeled "1", and so on, in each iteration increasing the label number, until no node is left unlabeled. The length of the shortest path or the distance d_{0i} between node 0 and any other node i in the network is given by the label of node i. For example, the distance between node 0 and the leftmost node is $d = 3$.

CONNECTEDNESS

A phone would be of limited use as a communication device if we could not call any valid phone number; email would be rather useless if we could send emails to only certain email addresses, and not to others. From a network perspective this means that the network behind the phone or the Internet must be capable of establishing a path between *any* two nodes. This is in fact the key utility of most networks: they ensure *connectedness*. In this section we discuss the graph-theoretic formulation of connectedness.

In an undirected network nodes i and j are *connected* if there is a path between them. They are *disconnected* if such a path does not exist, in which case we have $d_{ij} = \infty$. This is illustrated in Figure 2.15a, which shows a network consisting of two disconnected clusters. While there are paths between any two nodes on the same cluster (for example nodes 4 and 6), there are no paths between nodes that belong to different clusters (nodes 1 and 6).

A *network is connected* if all pairs of nodes in the network are connected. A *network is disconnected* if there is at least one pair with $d_{ij} = \infty$. Clearly the network shown in Figure 2.15a is disconnected, and we call its two subnetworks *components* or *clusters*. A *component* is a subset of nodes in a network, so that there is a path between any two nodes that belong to the component, but one cannot add any more nodes to it that would have the same property.

If a network consists of two components, a properly placed single link can connect them, making the network connected (Figure 2.15b). Such a link is called a *bridge*. In general a bridge is any link that, if cut, disconnects the network.

While for a small network visual inspection can help us decide if it is connected or disconnected, for a network consisting of millions of nodes connectedness is a challenging question. Mathematical and algorithmic tools can help us identify the connected components of a graph. For example, for a disconnected network the adjacency matrix can be rearranged into a block diagonal form, such that all nonzero elements in the matrix

are contained in square blocks along the matrix' diagonal and all other elements are zero (Figure 2.15a). Each square block corresponds to a component. We can use the tools of linear algebra to decide if the adjacency matrix is block diagonal, helping us to identify the connected components.

In practice, for large networks the components are more efficiently identified using the BFS algorithm (BOX 2.6).

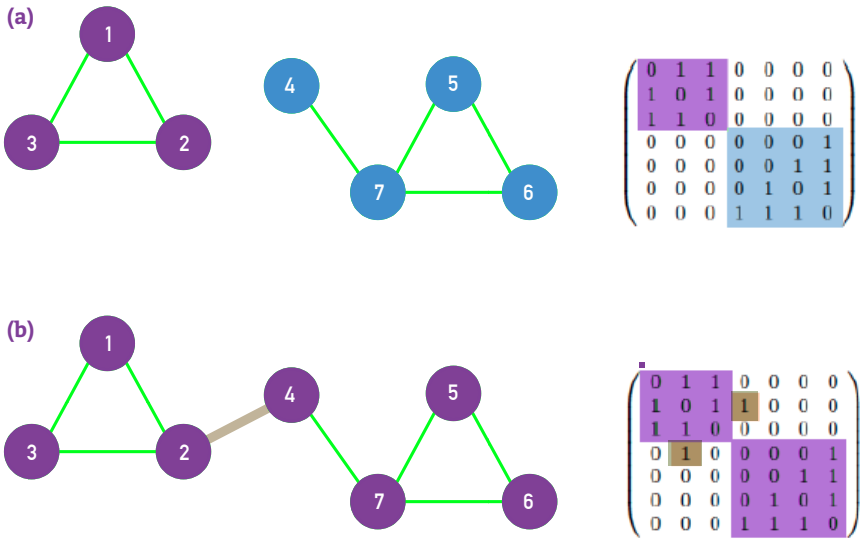


Figure 2.15
Connected and Disconnected Networks

(a) A small network consisting of two disconnected components. Indeed, there is a path between any pair of nodes in the (1,2,3) component, as well in the (4,5,6,7) component. However, there are no paths between nodes that belong to the different components.

The right panel shows the adjacently matrix of the network. If the network has disconnected components, the adjacency matrix can be rearranged into a block diagonal form, such that all nonzero elements of the matrix are contained in square blocks along the diagonal of the matrix and all other elements are zero.

(b) The addition of a single link, called a *bridge*, shown in grey, turns a disconnected network into a single connected component. Now there is a path between every pair of nodes in the network. Consequently the adjacency matrix cannot be written in a block diagonal form.

BOX 2.6

FINDING THE CONNECTED COMPONENTS OF A NETWORK

1. Start from a randomly chosen node i and perform a BFS (BOX 2.5). Label all nodes reached this way with $n = 1$.

2. If the total number of labeled nodes equals N, then the network is connected. If the number of labeled nodes is smaller than N, the network consists of several components. To identify them, proceed to step 3.

3. Increase the label $n \rightarrow n + 1$. Choose an unmarked node j, label it with n. Use BFS to find all nodes reachable from j, label them all with n. Return to step 2.

CLUSTERING COEFFICIENT

The clustering coefficient captures the degree to which the neighbors of a given node link to each other. For a node i with degree k_i the *local clustering coefficient* is defined as [12]

$$C_i = \frac{2L_i}{k_i(k_i-1)} \tag{2.15}$$

where L_i represents the number of links between the k_i neighbors of node i. Note that C_i is between 0 and 1 (Figure 2.16a):

- $C_i = 0$ if none of the neighbors of node i link to each other.
- $C_i = 1$ if the neighbors of node i form a complete graph, i.e. they all link to each other.
- C_i is the probability that two neighbors of a node link to each other. Consequently $C = 0.5$ implies that there is a 50% chance that two neighbors of a node are linked.

In summary C_i measures the network's local link density: The more densely interconnected the neighborhood of node i, the higher is its local clustering coefficient.

The degree of clustering of a whole network is captured by the *average clustering coefficient*, $\langle C \rangle$, representing the average of C_i over all nodes $i = 1, \ldots, N$ [12],

$$\langle C \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i \,. \tag{2.16}$$

In line with the probabilistic interpretation $\langle C \rangle$ is the probability that two neighbors of a randomly selected node link to each other.

While (2.16) is defined for undirected networks, the clustering coefficient can be generalized to directed and weighted [13, 14, 15, 16] networks as well. In the network literature we may encounter the *global clustering coefficient* as well, discussed in ADVANCED TOPICS 2.A.

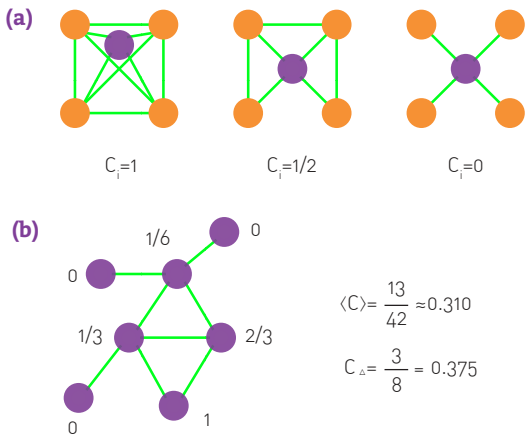


Figure 2.16
Clustering Coefficient

(a) The local clustering coefficient, C_i, of the central node with degree $k_i = 4$ for three different configurations of its neighborhood. The local clustering coefficient measures the local density of links in a node's vicinity.

(b) A small network, with the local clustering coefficient of each nodes shown next to it. We also list the network's average clustering coefficient $\langle C \rangle$, according to (2.16), and its global clustering coefficient C_Δ, defined in SECTION 2.12, Eq. (2.17). Note that for nodes with degrees $k_i = 0,1$, the clustering coefficient is zero.

SUMMARY

The crash course offered in this chapter introduced some of the basic graph theoretical concepts and tools used in network science. The set of elementary network characteristics, summarized in Figure 2.17, offer a formal language through which we can explore networks.

Many of the networks we study in network science consist of thousands or even millions of nodes and links (Table 2.1). To explore them, we need to go beyond the small graphs shown in Figure 2.17. A glimpse of what we are about to encounter is offered by the protein-protein interaction network of yeast (Figure 2.4a). The network is too complex to understand its properties through a visual inspection of its wiring diagram. We therefore need to turn to the tools of network science to characterize its topology.

Let us use the measures we introduced so far to explore some basic characteristics of this network. The undirected network, shown in Figure 2.4a, has N = 2,018 proteins as nodes and L=2,930 binding interactions as links. Hence its average degree, according to (2.2), is $\langle k \rangle$ = 2.90, suggesting that a typical protein interacts with approximately two to three other proteins. Yet, this number is somewhat misleading. Indeed, the degree distribution p_k shown in Figure 2.4b,c, indicates that the vast majority of nodes have only a few links. To be precise, in this network 69% of nodes have fewer than three links, i.e. for these $k < \langle k \rangle$. These numerous nodes with few links coexist with a few highly connected nodes, or hubs, the largest having as many as 92 links. Such wide differences in node degrees is a consequence of the network's scale-free property, discussed in CHAPTER 4. We will see that the shape of the degree distribution determines a wide range of network properties, from the network's robustness to the spread of viruses.

The breadth-first-search algorithm (BOX 2.5) helps us determine the network's diameter, finding $d_{\max}$ = 14. We might be tempted to expect wide variations in d, as some nodes are close to each other, others, however, may be quite far. The distance distribution (Figure 2.18a) indicates otherwise: p_d has a prominent peak between 5 and 6, telling us that most distances are rather short, being in the vicinity of $\langle d \rangle$ =5.61. Also, p_d decays fast for

large d, suggesting that large distances are absent. Indeed, the variance of the distances is $\sigma_d = 1.64$, indicating that most path lengths are in the close vicinity of $\langle d \rangle$. These are manifestations of the small world property discussed in CHAPTER 3.

The breadth first search algorithm also tells us that the protein interaction network is not connected, but consists of 185 components, shown as isolated clusters and nodes in Figure 2.4a. The largest, called the giant component, contains 1,647 of the 2,018 nodes; all other components are tiny. As we will see in the coming chapters, such fragmentation is common in real networks.

The average clustering coefficient of the protein interaction network is $\langle C \rangle = 0.12$, which, as we will come to appreciate in the coming chapters, indicates a significant degree of local clustering. A further caveat is provided by the dependence of the clustering coefficient on the node's degree, or the $C(k)$ function (Figure 2.18b). The fact that $C(k)$ decreases for large k indicates that the local clustering coefficient of the small nodes is significantly higher than the local clustering coefficient of the hubs. Hence the small degree nodes are located in dense local network neighborhoods, while the neighborhood of the hubs is much sparser. This is a consequence of *hierarchy*, a network property discussed in CHAPTER 9.

Finally, a visual inspection reveals an interesting pattern: hubs have a tendency to connect to small nodes, giving the network a hub and spoke character (Figure 2.4a). This is a consequence of degree correlations, discussed in CHAPTER 7. Such correlations influence a number of network based processes, from spreading phenomena to the number of driver nodes needed to control a network.

Taken together, Figures 2.4 and 2.18 illustrate that the quantities we introduced in this chapter can help us diagnose several key properties of real networks. The purpose of the coming chapters is to study systematically these network characteristics and understand what they tell us about a particular complex system.

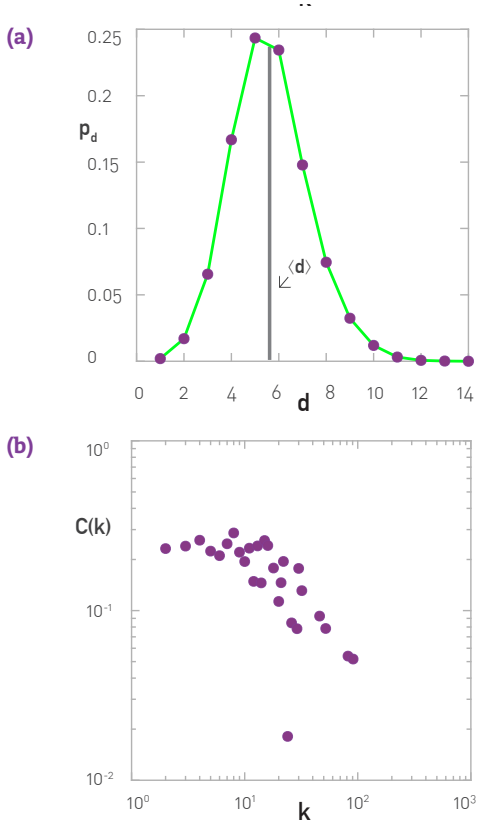


Figure 2.18
Characterizing a Real Network

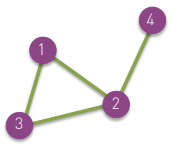
The protein-protein interaction (PPI) network of yeast is frequently studied by biologists and network scientists. The detailed wiring diagram of the network is shown in Figure 2.4a. The figure indicates that the network, consisting of N=2,018 nodes and L=2,930 links, has a large component that connects 81% of the proteins. We also have several smaller components and numerous isolated proteins that do not interact with any other node.

(a) The distance distribution, p_d, for the PPI network, providing the probability that two randomly chosen nodes have a distance d between them (shortest path). The grey vertical line shows the average path length, which is $\langle d \rangle = 5.61$.

(b) The dependence of the average local clustering coefficient on the node's degree, k. The $C(k)$ function is obtained by averaging over the local clustering coefficient of all nodes with the same degree k.

In network science we often distinguish networks by some elementary property of the underlying graph. Here we summarize the most commonly encountered network types. We also list real systems that share the particular property. Note that many real networks combine several of these elementary network characteristics. For example the WWW is a directed multi-graph with self-interactions; the mobile call network is directed and weighted, without self-loops.

(a) **Undirected**



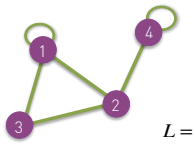
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$A_{ii} = 0 \qquad A_{ij} = A_{ji}$

$L = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad < k >= \frac{2L}{N}$

Undirected Network
A network whose links do not have a defined direction.
Examples: Internet, power grid, science collaboration networks.

(b) **Self-loops**



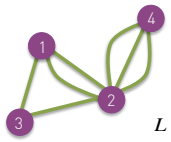
$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$\exists i, A_{ii} \neq 0 \qquad A_{ij} = A_{ji}$

$L = \frac{1}{2}\sum_{i,j=1, i\neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii} \qquad ?$

Self-loops
In many networks nodes do not interact with themselves, so the diagonal elements of the adjacency matrix are zero, $A_{ii} = 0$, $i = 1,..., N$. In some systems self-interactions are allowed; in such networks, self-loops represent the fact that node i interacts with itself.
Examples: WWW, protein interactions.

(c) **Multigraph**
(undirected)



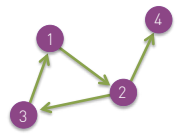
$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

$A_{ii} = 0 \qquad A_{ij} = A_{ji}$

$L = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad < k >= \frac{2L}{N}$

Multigraph/Simple Graphs
In a multigraph nodes are permitted to have multiple links (or parallel links) between them. Hence A_{ij} can be any positive integer. Networks that do not allow multiple links are called *simple*.
Multigraph Examples: Social networks, where we distinguish friendship, family and professional ties.

(d) **Directed**



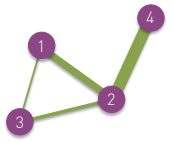
$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$A_{ij} \neq A_{ji}$

$L = \sum_{i,j=1}^{N} A_{ij} \qquad < k >= \frac{L}{N}$

Directed Network
A network whose links have selected directions.
Examples: WWW, mobile phone calls, citation network.

(e) **Weighted**
(undirected)



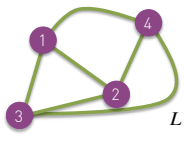
$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

$A_{ii} = 0 \qquad A_{ij} = A_{ji}$

$< k >= \frac{2L}{N}$

Weighted Network
A network whose links have a defined weight, strength or flow parameter. The elements of the adjacency matrix are $A_{ij} = w_{ij}$ if there is a link with weight w_{ij} between them. For unweighted (binary) networks, the adjacency matrix only indicates the presence ($A_{ij} = 1$) or the absence ($A_{ij} = 0$) of a link.
Examples: Mobile phone calls, email network.

(f) **Complete Graph**
(undirected)



$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$A_{ii} = 0 \qquad A_{i\neq j} = 1$

$L = L_{\max} = \frac{N(N-1)}{2} \qquad < k >= N - 1$

Complete Graph (Clique)
In a complete graph, or a clique, all nodes are connected to each other.
Examples: Actors in the cast of the same movie, as they are all linked to each other in the actor network.

HOMEWORK

2.1. Königsberg Problem

Which of the icons in Figure 2.19 can be drawn without raising your pencil from the paper, and without drawing any line more than once? Why?

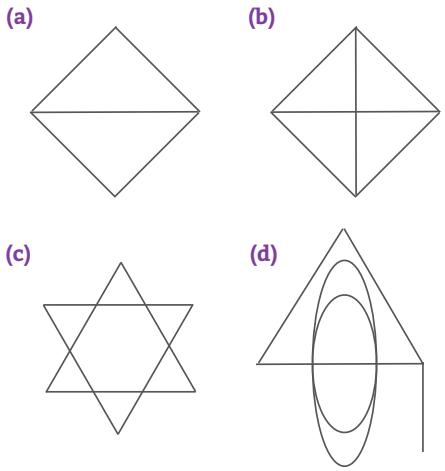


Figure 2.19
Königsberg Problem

2.2. Matrix Formalism

Let A be the $N \times N$ adjacency matrix of an undirected unweighted network, without self-loops. Let **1** be a column vector of N elements, all equal to 1. In other words $\mathbf{1} = (1, 1, ..., 1)^T$, where the superscript T indicates the *transpose* operation. Use the matrix formalism (multiplicative constants, multiplication row by column, matrix operations like transpose and trace, etc, but avoid the sum symbol ∑) to write expressions for:

(a) The vector $\mathbf{k}$ whose elements are the degrees k_i of all nodes $i = 1, 2, ..., N$.

(b) The total number of links, L, in the network.

(c) The number of triangles T present in the network, where a triangle means three nodes, each connected by links to the other two (Hint: you can use the trace of a matrix).

(d) The vector $\mathbf{k}_{nn}$ whose element i is the sum of the degrees of node i's neighbors.

(e) The vector $\mathbf{k}_{nnn}$ whose element i is the sum of the degrees of node i's second neighbors.

2.3. Graph Representation

The adjacency matrix is a useful graph representation for many analytical calculations. However, when we need to store a network in a computer, we can save computer memory by offering the list of links in a Lx2 matrix, whose rows contain the starting and end point i and j of each link.

Construct for the networks (a) and (b) in Figure 2.20:

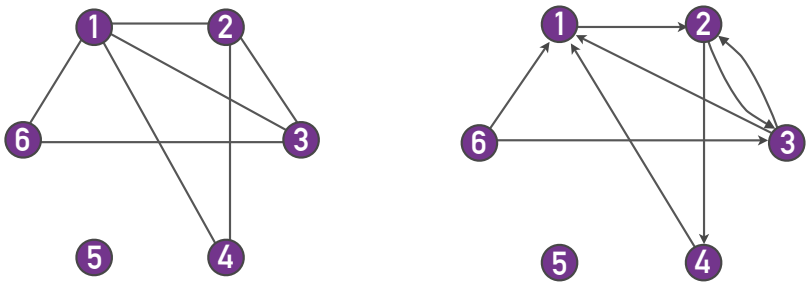


Figure 2.20
Graph Representation

(a) Undirected graph of 6 nodes and 7 links.
(b) Directed graph of 6 nodes and 8 directed links.

(a) The corresponding adjacency matrices.

(b) The corresponding link lists.

(c) Determine the average clustering coefficient of the network shown in Figure 2.20a.

(d) If you switch the labels of nodes 5 and 6 in Figure 2.20a, how does that move change the adjacency matrix? And the link list?

(e) What kind of information can you not infer from the link list representation of the network that you can infer from the adjacency matrix?

(f) In the (a) network, how many paths (with possible repetition of nodes and links) of length 3 exist starting from node 1 and ending at node 3? And in (b)?

(g) With the help of a computer, count the number of cycles of length 4 in both networks.

2.4. Degree, Clustering Coefficient and Components

(a) Consider an undirected network of size N in which each node has degree $k = 1$. Which condition does N have to satisfy? What is the degree distribution of this network? How many components does the network have?

(b) Consider now a network in which each node has degree $k = 2$ and clustering coefficient $C = 1$. How does the network look like? What condition does N satisfy in this case?

2.5. Bipartite Networks

Consider the bipartite network of Figure 2.21

(a) Construct its adjacency matrix. Why is it a block-diagonal matrix?

(b) Construct the adjacency matrix of its two projections, on the purple and on the green nodes, respectively.

(c) Calculate the average degree of the purple nodes and the average degree of the green nodes in the bipartite network.

(d) Calculate the average degree in each of the two network projections. Is it surprising that the values are different from those obtained in point (c)?

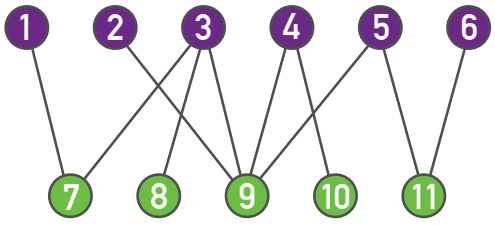


Figure 2.21
Bipartite network

Bipartite network with 6 nodes in one set and 5 nodes in the other, connected by 10 links.

2.6. Bipartite Networks - General Considerations

Consider a bipartite network with N_1 and N_2 nodes in the two sets.

(a) What is the maximum number of links L_{max} the network can have?

(b) How many links cannot occur compared to a non-bipartite network of size $N = N_1 + N_2$?

(c) If $N_1 \ll N_2$, what can you say about the network density, that is the total number of links over the maximum number of links, L_{max}?

(d) Find an expression connecting N_1, N_2 and the average degree for the two sets in the bipartite network, $\langle k_1 \rangle$ and $\langle k_2 \rangle$.

ADVANCED TOPICS 2.A
GLOBAL CLUSTERING COEFFICIENT

In the network literature we ocassionally encounter the *global clustering coefficient*, which measures the total number of closed triangles in a network. Indeed, L_i in (2.15) is the number of triangles that node i participates in, as each link between two neighbors of node i closes a triangle (Figure 2.17). Hence the degree of a network's global clustering can be also captured by the *global clustering coefficient*, defined as

$$C_{\Delta}=\frac{3\times Number\ Of\ Triangles}{Number\ Of\ Connected\ Triples}\ , \tag{2.17}$$

where a *connected triplet* is an ordered set of three nodes ABC such that A connects to B and B connects to C. For example, an A, B, C triangle is made of three triplets, ABC, BCA and CAB. In contrast a chain of connected nodes A, B, C, in which B connects to A and C, but A does not link to C, forms a single open triplet ABC. The factor three in the numerator of (2.17) is due to the fact that each triangle is counted three times in the triplet count. The roots of the global clustering coefficient go back to the social network literature of the 1940s [17, 18], where C_{Δ} is often called the *ratio of transitive triplets*.

Note that the average clustering coefficient $<C>$ defined in (2.16) and the global clustering coefficient (2.17) are not equivalent. Indeed, take a network that is a double star, consisting of N nodes, where nodes 1 and 2 are joined to each other and to all other nodes, and there are no other links. Then the local clustering coefficient C_i is 1 for $i \geq 3$ and $2/(N-1)$ for $i = 1, 2$. It follows that the average clustering coefficient of the network is $<C> = 1–O(1)$, while the global clustering coefficient is $C_{\Delta} \sim 2/N$. In less extreme networks the two definitions will give more comparable values, but they still differ from each other [19]. For example, for the network of in Figure 2.16b we have $<C> = 0.31$ and $C_{\Delta} = 0.375$.

BIBLIOGRAPHY

[1] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási. The human disease network. PNAS, 104:8685–8690, 2007.

[2] H.U. Obrist. *Mapping it out: An alternative atlas of contemporary cartographies*. Thames and Hudson, London, 2014.

[3] I. Meirelles. *Design for Information*. Rockport, 2013.

[4] K. Börner. *Atlas of Science: Visualizing What We Know*. The MIT Press, 2010.

[5] L. B. Larsen. *Networks: Documents of Contemporary Art*. MIT Press. 2014.

[6] L. Euler, Solutio Problemat is ad Geometriam Situs Pertinentis. Commentarii Academiae Scientiarum Imperialis Petropolitanae 8:128-140, 1741.

[7] G. Alexanderson. Euler and Königsberg's bridges: a historical view. *Bulletin of the American Mathematical Society* 43: 567, 2006.

[8] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. Science, 286:509–512, 1999.

[9] G. Gilder. Metcalfe's law and legacy. Forbes ASAP, 1993.

[10] B. Briscoe, A. Odlyzko, and B. Tilly. Metcalfe's law is wrong. IEEE Spectrum, 43:34–39, 2006.

[11] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási. Flavor network and the principles of food pairing, Scientific Reports, 196, 2011.

[12] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. Nature, 393:440–442, 1998.

[13] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. PNAS, 101:3747–3752, 2004.

[14] J. P. Onnela, J. Saramäki, J. Kertész, and K. Kaski. Intensity and coherence of motifs in weighted complex networks. Physical Review E, 71:065103, 2005.

[15] B. Zhang and S. Horvath. A general framework for weighted gene coexpression network analysis. Statistical Applications in Genetics and Molecular Biology, 4:17, 2005.

[16] P. Holme, S. M. Park, J. B. Kim, and C. R. Edling. Korean university life in a network perspective: Dynamics of a large affiliation network. Physica A, 373:821–830, 2007.

[17] R. D. Luce and A. D. Perry. A method of matrix analysis of group structure. Psychometrika, 14:95–116, 1949.

[18] S. Wasserman and K Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[19] B. Bollobás and O. M. Riordan. Mathematical results on scale-free random graphs, in Stefan Bornholdt, Hans Georg Schuster, *Handbook of Graphs and Networks: From the Genome to the Internet* (2003 Wiley-VCH Verlag GmbH & Co. KGaA).