Networks in Nature:

Dynamics, Evolution, and Modularity

Sumeet Agarwal

Merton College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2012

To my entire social network; for no man is an island

Acknowledgements

Primary thanks go to my supervisors, Nick Jones, Charlotte Deane, and Mason Porter, whose ideas and guidance have of course played a major role in shaping this thesis. I would also like to acknowledge the very useful suggestions of my examiners, Mark Fricker and Jukka-Pekka Onnela, which have helped improve this work. I am very grateful to all the members of the three Oxford groups I have had the fortune to be associated with: Systems and Signals, Protein Informatics, and the Systems Biology Doctoral Training Centre. Their companionship has served greatly to educate and motivate me during the course of my time in Oxford. In particular, Anna Lewis and Ben Fulcher, both working on closely related D.Phil. projects, have been invaluable throughout, and have assisted and inspired my work in many different ways. Gabriel Villar and Samuel Johnson have been collaborators and co-authors who have helped me to develop some of the ideas and methods used here. There are several other people who have generously provided data, code, or information that has been directly useful for my work: Waqar Ali, Binh-Minh Bui-Xuan, Pao-Yang Chen, Dan Fenn, Katherine Huang, Patrick Kemmeren, Max Little, Aurélien Mazurie, Aziz Mithani, Peter Mucha, George Nicholson, Eli Owens, Stephen Reid, Nicolas Simonis, Dave Smith, Ian Taylor, Amanda Traud, and Jeffrey Wrana. I would also like to thank the Clarendon Fund for providing the scholarship that enabled me to come to Oxford, and Merton College and Oxford Physics for hosting me and enabling me to meet many wonderful people

and take advantage of many helpful resources. Finally, I must express my gratitude towards my parents and my family, without whom my existence would be neither possible nor meaningful.

Abstract

In this thesis we propose some new approaches to the study of complex networks, and apply them to multiple domains, focusing in particular on protein-protein interaction networks. We begin by examining the roles of individual proteins; specifically, the influential idea of 'date' and 'party' hubs. It was proposed that party hubs are local coordinators whereas date hubs are global connectors. We show that the observations underlying this proposal appear to have been largely illusory, and that topological properties of hubs do not in general correlate with interactor co-expression, thus undermining the primary basis for the categorisation. However, we find significant correlations between interaction centrality and the functional similarity of the interacting proteins, indicating that it might be useful to conceive of roles for protein-protein interactions, as opposed to individual proteins.

The observation that examining just one or a few network properties can be misleading motivates us to attempt to develop a more holistic methodology for network investigation. A wide variety of diagnostics of network structure exist, but studies typically employ only small, largely arbitrarily selected subsets of these. Here we simultaneously investigate many networks using many diagnostics in a data-driven fashion, and demonstrate how this approach serves to organise both networks and diagnostics, as well as to relate network structure to functionally relevant characteristics in a variety of settings. These include finding fast estimators for the solution of hard graph problems, discovering evolutionarily significant aspects of metabolic networks, detecting structural constraints on particular network types, and constructing summary statistics for efficient model-fitting to networks. We use the last mentioned to suggest that duplication-divergence is a feasible mechanism for protein-protein interaction evolution, and that interactions may rewire faster in yeast than in larger genomes like human and fruit fly.

Our results help to illuminate protein-protein interaction networks in multiple ways, as well as providing some insight into structure-function relationships in other types of networks. We believe the methodology outlined here can serve as a general-purpose, data-driven approach to aid in the understanding of networked systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Networks

Some citizens of Königsberg
Were walking on the strand
Beside the river Pregel
With its seven bridges spanned.

"O Euler, come and walk with us,"
Those burghers did beseech.
"We'll roam the seven bridges o'er,
And pass but once by each."

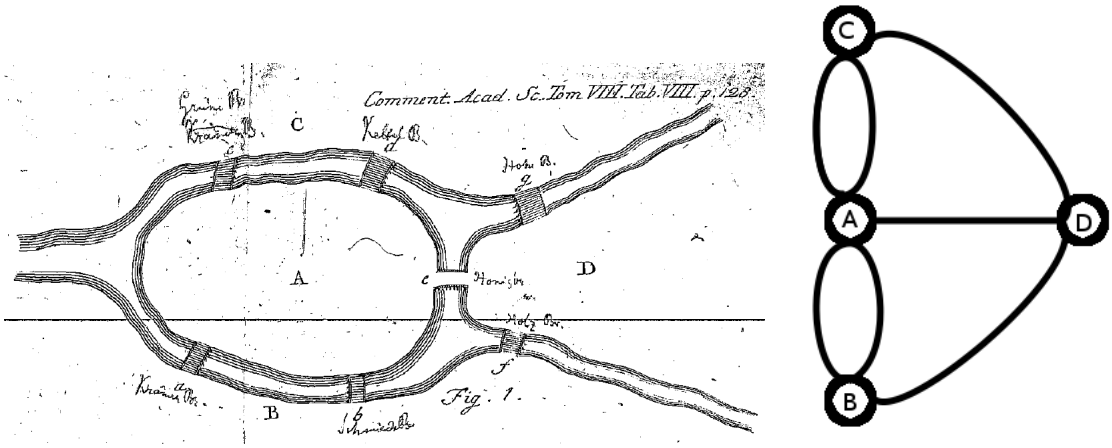
"It can't be done," thus Euler cried.
"Here comes the Q.E.D.
Your islands are but vertices
And four have odd degree."

William T. Tutte[1]

1.1.1 A brief history

A *graph* or *network* (we use the two terms interchangeably here, though in certain contexts more general non-graph networks may be defined) consists of a set of elements (called *nodes* or *vertices*) and a set of pairwise connections between those elements (called *links* or *edges* or *ties*). The mathematical study of graphs (*graph theory*) is

[1] As cited in C. Moore and S. Mertens, *The Nature of Computation*. Oxford University Press, 2011.



(a) Euler's original drawing [81] (b) Graph representation

Figure 1.1: **The Seven Bridges of Königsberg problem.**

Euler proved that in order for a path that passed through each link (bridge) precisely once (not necessarily returning to its starting point) to exist, there could be no more than two nodes with an odd number of links, whereas here all four nodes have an odd number of links.

thought to have originated in the 17th century, when Euler famously proved that the Seven Bridges of Königsberg problem was unsolvable [81] (Figure 1.1). Pure mathematicians and computer scientists have continued to study graph theory and related topics such as combinatorics, and the idea of representing real-world systems of various kinds as graphs or networks and studying their properties has become widespread in several different domains [196], such as engineering [217], sociology [272], physics, and biology [143].

To some extent, practitioners studying networks within different fields have developed their own hermetic methodology and terminology, and the fundamental unity of the mathematical abstractions being used to represent connections between machines or people or molecules or organisms has not always been fully appreciated. This has often led to repetition of work and reinvention of ideas: for instance, the *preferential attachment* model for network growth (Section 1.1.6.3), published by Barabási and

Albert in 1999 [28], which led to a surge of interest in network science within the statistical physics community, is closely related to growth mechanisms proposed several times previously.[2] Given a network representing an unexplored system, it may be difficult to decide what parts of the literature to draw on for ways of analysing it. In this thesis we will present an attempt to take a data-driven approach to comparing and organising both different kinds of networks and different ways of characterising networks.

In this introduction, we aim to provide an overview of the relevant literature and how it feeds into our work, including some technical details of concepts and methods that we will use later. This chapter is organised into four major parts: in the remainder of Section 1.1, we describe a selection of concepts and techniques related to networks; in Section 1.2, we focus on proteins and the networks formed by their interactions, which will be a running theme throughout this thesis; in Section 1.3, we provide a brief exposition of some topics in machine learning and introduce terms and tools that will be of use to us; and in Section 1.4, we give an overview of the thesis, outlining what we seek to do and attempting to situate it in the context of the existing literature.

1.1.2 Basic concepts and terminology

Here we provide definitions for some basic terms associated with networks that will be used throughout this thesis. The reader familiar with this terminology may wish to skip straight to Section 1.1.2.1.

- *Directed* and *Undirected*: The links in a network can in some cases be *directed*, i.e., going from one node to another (typically, such links will represent flows of

[2]Starting with Yule in 1925, who suggested a similar process to explain the number of species per genus of flowering plants [282]. Subsequently, Simon in 1955 devised a master equation method for preferential attachment, and used it to model distributions of the sizes of cities and other phenomena [243]. Price in 1976 was the first to apply preferential attachment to network growth [70].

some kind, for instance physical or informational ones). In this thesis, we deal largely with *undirected* networks, which comprise links that represent binary associations between nodes, without any directionality. In cases where we are dealing with data sets where there is a directionality to the links, we generally ignore that directionality in order to allow for comparison with undirected networks.

- *Weighted* and *Unweighted*: The links in a network can have weights associated with them, representing strength of association or some measure of distance; this is known as a *weighted* network. The protein interaction networks we examine in Chapter 2 are all *unweighted*, i.e., there are no weights assigned to the links; but subsequently we look at both weighted and unweighted networks. Because unweighted networks can be treated as a special case of weighted networks in which the only weights used are 0 (link absent) and 1 (link present), in general techniques and diagnostics applicable to weighted networks can also be applied to unweighted ones.

- *Degree* and *Strength*: The *degree* of a node in an undirected network is the number of links incident upon it. For weighted networks, we can extend this to define the *strength* of a node, which is the sum of the weights of all links incident upon it (in the unweighted case, degree and strength are equivalent).

- *Degree/Strength distribution*: This is the entire distribution of node degrees/strengths in the network.

- *Path*: In an undirected network, a *path* between any pair of nodes is a sequence of distinct nodes $v_1, v_2, ..., v_k$ (where v_1 and v_k are the nodes being connected) that can be followed to get from one node to the other, i.e., such that there is a link between v_i and v_{i+1}, for $i \in [1, k-1]$. The number of links on a path is known as its *length*.

- *Walk*: A walk is a sequence of linked nodes; unlike a path, it can contain multiple occurrences of the same node.

- *Geodesic distance*: In an undirected network, the *geodesic distance* between a pair of nodes is the length of the shortest path(s)—i.e., the one(s) with the fewest links—connecting those two nodes. If there is no path between two nodes in a network, the distance between them is defined to be infinite.

- *Weighted distance*: For a weighted network, the *weight* of a path is defined as the sum of the individual link weights along that path. The minimum total weight over all paths between a pair of nodes is their *weighted distance*. For unweighted networks, this is the same as the geodesic distance. For a general undirected network, we will call this the *distance* between pairs of nodes.

- *Connected component*: This is a set of nodes in a network such that there is a path between any two of them and to which no further nodes can be added whilst maintaining this property. The connected component with the most nodes is known as the *largest connected component (LCC)*; if this is equivalent to the entire network, then the network is said to be *connected*.

- *Clique*: A set of nodes where every node is linked to every other node. A *k-clique* is a clique containing k nodes.

- *Adjacency matrix*: A standard way of representing a network. An $n \times n$ matrix, where n is the number of nodes; if nodes i and j are linked, then $A_{ij} = 1$ for an unweighted link (also $A_{ji} = 1$, for an undirected link), or $A_{ij} = w_{ij}$, where w_{ij} is the weight of the link. If i and j are not linked, then $A_{ij} = 0$.

1.1.2.1 Hubs

In Chapter 2, we will focus primarily on the properties of certain network nodes called *hubs*. In general, a hub is a node with a large degree, relative to the degrees of other nodes in the network. In some cases an absolute degree threshold may be used to define hubs, but such a threshold is usually chosen so that the proportion of nodes that are hubs is not too large. In several real-world networks, it has been found that there are a small number of nodes with very high degrees; such a network is said to have a *heavy-tailed* degree distribution. This has sometimes been construed, often controversially, as evidence for *power laws* [28, 62], i.e., degree distributions where the probability of observing a given degree k scales as $p(k) \propto k^{-\gamma}$. Irrespective of this, the observation of a such a heavy tail is frequently thought to correspond to the existence of hubs as nodes of particular importance to a network [136, 266, 286]. There is, however, no standardised criterion for identifying hubs; in the context of protein-protein interaction networks (Section 1.2.3), they have been variously defined as nodes with degree greater than 5 [121], the top 20% of nodes by degree [41], and via other measures based on topological or functional properties of proteins [266]. Here we will largely stick to previously defined hubs for the protein interaction networks we study in Chapter 2, as our objective will be to scrutinise previous claims made regarding the roles played by such hubs.

1.1.3 Communities in networks

Many real-world networks display some sort of modular organisation, as they can be partitioned into cohesive groups of nodes such that there is a relatively high ratio of internal (within-group) to external (between-group) link density (the number of links as a fraction of the number of possible links). Such sub-networks, known as *communities*, are often construed to correspond to distinct functional units [90, 107, 213].

From an intuitive standpoint, communities should consist of groups of nodes such that there are many links between nodes in the same group but few links between nodes in different groups. To detect communities algorithmically, this notion must be formalised. A myriad of algorithms have been developed for detecting communities in networks [90,213]. We describe in some detail two of the most popular approaches, both of which are used in the work presented in this thesis.

1.1.3.1 Modularity

Perhaps the most widely used method for identifying community structure in networks is based on optimising a quality function known as Newman-Girvan *modularity* [194, 197]. Suppose that an undirected network with n nodes and m (weighted) links, with total weight S, is divided into N communities $C_1, C_2, \cdots, C_N$. Let s_i denote the strength of node i, c_i the community to which it belongs, and let A be the adjacency matrix. The Newman-Girvan modularity Q is then given by [194]

$$Q = \frac{1}{2S} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(A_{ij} - \frac{s_i s_j}{2S} \right) \delta(c_i, c_j) \,, \tag{1.1}$$

where $s_i s_j/(2S)$ is the expected weight of the link between nodes i and j in a network with the same expected sequence of node strengths but with link weights assigned at random; and $\delta(c_i, c_j) = 1$ if $c_i = c_j$ and 0 otherwise. This modularity measure thus captures how much more link weight there is within the specified communities than one would expect to see by chance in a network with no modular structure. Note, however, that (1.1) assumes a particular *null model*, $\frac{s_i s_j}{2S}$, that explicitly preserves the expected node strength distribution in the random setting. It is possible to employ other null models [213], though this is the most common choice.

1.1.3.2 Multi-resolution community detection

In general, a network can have community structure at multiple scales of organisation; there can be smaller communities nested inside bigger ones, for instance. The Newman-Girvan method only allows one to find communities at a single scale, but there are several methods that incorporate the concept of a *resolution parameter*, which allows one to probe structure at different scales; varying the value of the resolution parameter leads to communities of different sizes. Here we will use an extension of the Newman-Girvan method that is based on an analogy to the Potts model in statistical mechanics [220]. This incorporates a resolution parameter (denoted by γ) into the equation for modularity, leading to the quality function

$$H = \frac{1}{2S} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(A_{ij} - \gamma \frac{s_i s_j}{2S} \right) \delta(c_i, c_j) \,. \tag{1.2}$$

Setting $\gamma = 1$ leads to the standard modularity function (1.1). Values of γ greater than 1 in effect further reduce the strength or 'attractive force' associated with links in the network, i.e., they reduce the reward (in terms of increasing the value of the modularity H) obtained by making any given link an intra-community link rather than inter-community. Thus higher values of γ lead to smaller communities. In the limit, when $\gamma > 2SA_{ij}/(s_i s_j)$ for every pair of nodes (i, j), then maximising H just puts every node into its own community (giving $H = 0$), as any paired nodes will only make a negative contribution to the quality function. However, $\gamma < 1$ leads to larger communities, with the limit being reached at $\gamma = 0$, when the contribution of every node pair to H becomes non-negative, and thus the maximum is achieved by putting the entire network into a single community.

1.1.3.3 Optimisation

One can detect communities by maximising a quality function, such as Equation (1.1) or Equation (1.2), over all possible network partitions. Because this problem is known to be *NP-hard* [52], roughly meaning that the time taken to solve it scales very fast with the problem (network) size[3], reliably finding the global maximum is computationally intractable except for very small networks. Thankfully, there exist several good computational heuristics that can be used to obtain good local maxima [67, 90, 213]. Here we employ two different algorithms[4] to maximise Newman-Girvan modularity: recursive spectral bisection [193] (accompanied by the Kernighan-Lin algorithm [144] for fine-tuning), and a locally greedy algorithm known as the *Louvain* method [48]. The first method defines a *modularity matrix* B, whose elements are given by $B_{ij} = A_{ij} - s_i s_j/(2S)$, and relies on the fact that the modularity maximisation process can be formulated in terms of the spectrum of this matrix. The Kernighan-Lin fine-tuning step involves finding pairs of nodes that can be interchanged between communities, so as to lower the total weight of inter-community links. The Louvain method involves an iteration between local adjustments of communities and global aggregation of the obtained communities, and these two steps are repeated until modularity has converged to a maximum. The Louvain method also allows community detection at multiple settings of the resolution parameter γ in the Potts method [Equation (1.2)].

1.1.3.4 Infomap

Another well-known algorithm for community detection is the information-theoretic approach of Rosvall and Bergstrom (called *Infomap*) [226]. This is based on the

[3] If the network has n nodes, then there is no known algorithm whose runtime scales as a polynomial function of n. A typical algorithm might scale exponentially, e.g., the runtime might be proportional to 2^n.

[4] We obtained MATLAB code for the algorithms from Stephen Reid and Dan Fenn.

idea that a good network partition is one that allows a network to be compressed most effectively; one can think of this as finding communities where the nodes within them can be combined into a single node with minimal loss of information about global network structure. Let A be the adjacency matrix for a given (undirected, unweighted) network with n nodes, and suppose the assignment of nodes to one of N communities is represented by the vector $\mathbf{a} = (a_1, a_2, ..., a_n)$, where $a_i \in \{1, 2, ..., N\}$. Also, an $N \times N$ module matrix M is defined such that entry M_{ij} is the number of links between communities i and j. Then, the tuple $Y = \{\mathbf{a}, M\}$ represents the compressed version of the network. Infomap seeks to find an assignment $\mathbf{a}$ of nodes to N communities such that the mutual information between Y and the full network represented by A is maximised, and thus the information lost due to the compression is minimised. In one comparative study of a range of popular algorithms for community detection, Infomap was found to be one of the best performing methods [157].

1.1.3.5 Other methods

There are numerous other types of community detection algorithms we do not use in this thesis; two of the major approaches are briefly described here. *Local* community detection methods involve starting from individual nodes and finding communities around them, rather than attempting to partition the network globally. Examples include the *CFinder* method based on k-clique percolation [13,72] and the Lancichinetti *et al.* local method [158], which optimises a fitness function defined using the nodes within a given community plus immediate neighbours (rather than the entire network). Another possibility is to attempt to find cohesive groups of links, rather than nodes; this has been explored only quite recently [19, 82, 83]. Both these approaches have the feature of allowing for a node to belong to multiple communities, which is useful in some contexts.

1.1.4 Network diagnostics and summary statistics

As noted in Section 1.1.1, a variety of methods have been developed for characterising networks across various domains. One of the goals of this thesis is to attempt to consolidate a substantial number of these methods (the outputs of which we refer to as network *diagnostics*) and explore the connections between them, as well as to examine how they can collectively aid in comparing and highlighting interesting aspects of different sorts of networks.

Here we briefly review the main diagnostics we have used to study network structure. Many of these diagnostics are defined for unweighted networks; in these cases, we will ignore link weights when applying them to a weighted network (though the bulk of the networks examined in this thesis are unweighted to begin with). A complete list, along with references to original sources, can be found in Appendix A.

1.1.4.1 Connectivity measures

- *Density*: The number of links in the network as a fraction of the total number of possible links, which for an unweighted network is $\binom{n}{2}$, where n is the number of nodes.

- *Degree assortativity*: This is a measure of the extent to which nodes tend to connect to other nodes of similar degree. It is defined by the Pearson correlation coefficient (over all links in the network) of the degrees at either end of a link [191]. If m is the total number of links in a network, and the degrees of the nodes at the two ends of link i are denoted by j_i and k_i, then the degree assortativity r can be written as:

$$r = \frac{\frac{1}{m}\sum_{i=1}^{m} j_i k_i - [\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}(j_i + k_i)]^2}{\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}(j_i^2 + k_i^2) - [\frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}(j_i + k_i)]^2}. \tag{1.3}$$

- *k-cores*: The *k-core* of a network is obtained by removing all nodes of degree less

than k, iteratively (since after each round of removals the degrees of remaining nodes may be reduced), until no such nodes remain [239]. Thus, it is the maximal connected subnetwork such that all nodes have degree k or more. The fraction of the network contained in the k-core, for various values of k, can be seen as an indicator of cohesiveness in the network.

- *Rich-club coefficient*: The 'rich-club' phenomenon refers to the tendency of high-degree nodes to be well-connected to each other [64, 285]. The *rich-club coefficient* can be defined as a function of a node threshold by degree. In the standard notation, we let $n_{>k}$ denote the number of nodes with degree greater than k, and let $e_{>k}$, be the number links amongst these nodes, then the rich club coefficient ϕ is defined as a function of k:

$$\phi(k) = \frac{2e_{>k}}{n_{>k}(n_{>k} - 1)}. \tag{1.4}$$

For our purposes, we would like to re-write ϕ as a function of a node rank, rather than a degree threshold. Let m_i be the total number of links amongst the top i nodes by degree ($i \geq 2$). Then the rich-club coefficient ϕ can be written as:

$$\phi(i) = \frac{2m_i}{i(i-1)}. \tag{1.5}$$

In order to obtain a single number to estimate the amount of 'rich-clubness' in a given network, we compute i_{50}, the value of i for which $\phi(i)$ comes closest to 0.5; given that $\phi(i)$ is in practice nearly always observed to be a decreasing function with increasing i [64], this generally amounts to finding the largest value of i for which the rich-club coefficient is at least 0.5. We use the ratio i_{50}/n (n being the total number of nodes in the network) as a measure of the fraction of the network constituting the rich-club.

1.1.4.2 Node or link centrality

- *Degree centrality*[5]: This is defined for a node i in an unweighted network as the ratio of the node's degree k_i to the maximum possible degree, which (assuming no self-links) is $n-1$, where n is the number of nodes. Computing this measure for each node in a network gives us a distribution of degree centrality values. We use multiple summary statistics (such as mean and variance; see Appendix A for list) of this and other distributions as ways of characterising a given network.

  Group degree centrality is a measure of the dispersion or variation in degree centrality values [272]. This is defined as:

$$c_d^{group} = \frac{\sum_i (k_{max} - k_i)}{(n-1)(n-2)}, \tag{1.6}$$

  where k_{max} is the maximum degree in the given network.

- Geodesic *betweenness centrality*: There are two versions of this, for nodes [96] and links [107]. In either case, it is defined for an unweighted network as the number of pairwise shortest paths between (other) nodes in the network that pass through a given node/link, normalised by the total number of such pairs. A shortest path between a pair of nodes is one such that no other path has fewer links; if there are multiple shortest paths between a given node pair, then in adding them up for betweenness they are weighted down so that the total weight for the pair is 1 (e.g., if there are 3 shortest paths between a given pair, then each one of them will be counted as 1/3 of a path). Thus, the betweenness centrality c_b of a node v is given by:

$$c_b(v) = \sum_{s,t:s\neq t,s\neq v,t\neq v} \frac{2\sigma_{st}(v)}{(n-1)(n-2)\sigma_{st}}, \tag{1.7}$$

[5]In some contexts, the *degree centrality* may just refer to the degree. Here we use it to refer to the normalised degree as described.

where σ_{st} is the total number of shortest paths between nodes s and t and $\sigma_{st}(v)$ is the number of these paths that pass through the node v.

Group node betweenness centrality, which quantifies variation in the node betweenness values, is defined as [272]:

$$c_b^{group} = \frac{2\sum_{i=1}^{n}(c_b(v_{max}) - c_b(v_i))}{(n-1)^2(n-2)}, \tag{1.8}$$

where v_i is the i^{th} node and v_{max} denotes the node with the highest betweenness centrality.

- *Closeness centrality*: This is proportional to the inverse of the sum of geodesic distances from a node to all other nodes [233, 272]:

$$c_c(v_i) = \frac{n}{\sum_{j=1}^{n} d(v_i, v_j)}, \tag{1.9}$$

where $d(v_i, v_j)$ is the geodesic distance between nodes v_i and v_j.

Group closeness centrality is defined as [272]:

$$c_c^{group} = \frac{(2n-3)\sum_{i=1}^{n}(c_c(v_{max}) - c_c(v_i))}{(n-2)(n-3)}, \tag{1.10}$$

with v_{max} denoting the node with the highest closeness centrality.

- *Eigenvector centrality*: Unlike the degree centrality, which weights all links equally, the *eigenvector centrality* seeks to weight links based on the importance of the node being connected to; the idea being that links to nodes that are more influential or central will contribute more to one's own centrality. Thus, the eigenvector centrality of a node is defined as being proportional to the average of the eigenvector centralities of its neighbours; this leads to an eigenvalue problem, and the value of the centrality measure for a given node

turns out to be its weight in the leading eigenvector (the one corresponding to the largest eigenvalue) of the network's adjacency matrix. If the adjacency matrix is denoted by A and the largest magnitude eigenvalue by λ_1, then:

$$c_e(v_i) = x_i; A\mathbf{x} = \lambda_1 \mathbf{x}, \mathbf{x} = (x_1, x_2, ..., x_n)^T. \tag{1.11}$$

- *Information centrality*: This measure is designed to capture the *information* (defined just as the inverse of path length) contained in all paths originating at a given node [250, 272]. In essence, it computes the harmonic mean of the lengths of all these paths. Let $l_p(v_i, v_j)$ be the length of path p between nodes v_i and v_j. If there are a total of P paths between the two nodes, the total information between them is defined as

$$I(v_i, v_j) = \sum_{p=1}^{P} \frac{1}{l_p(v_i, v_j)}. \tag{1.12}$$

The information centrality of a given node v_i is then defined as

$$c_{inf}(v_i) = \frac{n}{\sum_{j=1}^{n} I(v_i, v_j)}. \tag{1.13}$$

The summary measure of *group information centrality* was proposed by Stephenson and Zelen [250] to be defined as simply the mean of the values for the individual nodes:

$$c_{inf}^{group} = \frac{1}{n} \sum_{i=1}^{n} c_{inf}(v_i). \tag{1.14}$$

- *Subgraph centrality*: This defines a node's importance by the number of subgraphs of the network that constitute a closed walk starting and ending at that node, with higher weights given to smaller subgraphs [65, 80]. It can be defined

in terms of the diagonal elements of the powers of the adjacency matrix A:

$$sc(v) = \sum_{k=0}^{\infty} \frac{(A^k)_{vv}}{k!}, \tag{1.15}$$

where $(A^k)_{vv}$ denotes the entry in row v and column v of the matrix obtained by raising A to the power k.

- *Bipartivity*: A *bipartite* graph is one in which there are two subsets of nodes such that all links are across the subsets and none within either subset. The definition of subgraph centrality can be used to define a measure of *bipartivity* (how close a network is to being bipartite), based on the observation that a bipartite graph will have no closed walks of odd length [79]. Thus, for each node, the fraction of its subgraph centrality contributed by even-length closed walks is a measure of how much it contributes to bipartivity; averaging over all nodes gives a bipartivity measure for the whole network. If $\lambda_1, \lambda_2, ..., \lambda_n$ are the n eigenvalues of the adjacency matrix for a given network, then the bipartivity measure β can be computed as [79]:

$$\beta = \frac{\sum_{i=1}^{n} \cosh(\lambda_i)}{\sum_{i=1}^{n} e^{\lambda_i}}. \tag{1.16}$$

1.1.4.3 Paths and distances

- *Characteristic path length*: This is the average of the lengths of all finite pairwise shortest paths in a given network.

- *Diameter*: The maximum geodesic distance between any pair of nodes in a network.

- *Radius*: The minimum *eccentricity* of any node in the network, where eccentricity of a node is the maximum distance to any other node.

- *Szeged index*: This measure was initially defined for the study of molecular graphs in chemistry [145]. For each link l, it defines a symmetry measure in terms of distances of other nodes from the two endpoints of the link. If $n_1(l)$ is the number of nodes whose distances from the first endpoint are less than that from the second endpoint, and $n_2(l)$ is the number of nodes for which it is vice versa, then the product of these two numbers is a measure of how similar the two endpoints are in terms of connectivity to other nodes. The overall Szeged index Sz is the sum of this quantity over all links in a network G:

$$Sz(G) = \sum_{l \in G} n_1(l) n_2(l). \tag{1.17}$$

- *Cyclic coefficient*: This measure was introduced by Kim and Kim [148] as a means of characterising cyclic topology in a graph. For node v_i, the cyclic coefficient *cyc* is defined as the mean inverse length of the smallest loops connecting v_i to all pairs of its neighbours:

$$cyc(v_i) = \frac{2}{k_i(k_i - 1)} \sum_{\{j,k\}: A_{ij}=1, A_{ik}=1, j \neq k} \frac{1}{L_c(v_i, v_j, v_k)}, \tag{1.18}$$

  where $L_c(v_i, v_j, v_k)$ denotes the length of the shortest closed or circular path that passes through all three nodes, and is defined to be infinity if no such path exists.

1.1.4.4 Clustering

- *Clustering coefficient*: Also known as the *local* clustering coefficient, this is defined for each node in an unweighted network as the number of links between its neighbours divided by the total number of such links possible (which is $\binom{k}{2}$, for a node with k neighbours) [273]. Another way to state this: if the number

of triangles in which a node v participates is denoted $N_{Tri}(v)$, and the number of connected triples (three-node sets) in which that node is the central node is denoted $N_3(v)$, then the clustering coefficient C is given by:

$$C(v) = \frac{N_{Tri}(v)}{N_3(v)}. \tag{1.19}$$

- *Transitivity*: Also sometimes known as the *global* clustering coefficient, though the latter term may also be used to denote the mean of the previous measure over the whole network. If N_{Tri} is the total number of triangles in a network, and N_3 is the number of connected triples, then transitivity T is defined as [272]:

$$T = \frac{3N_{Tri}}{N_3}. \tag{1.20}$$

1.1.4.5 Motifs

Motifs are small subgraphs, in practice typically of 3 or 4 nodes; it has been suggested that counting the number of different occurrences of these in a larger network can serve as a kind of signature of network structure [186]. For instance, in an undirected network, if one focuses on 3-node subgraphs, there are 4 possibilities: no links, a single link, two links, or all three links (i.e., a triangle). Examining the relative frequencies of these provides some characterisation of a given network. If one were to focus only on connected 3-node graphs, then the only two possibilities would be a triangle and a V (i.e., a triangle with one missing link); so in this case their relative frequencies would capture exactly the same information as the transitivity.

1.1.4.6 Graph complexity

- *Spanning trees*: A *spanning tree* of a connected n-node network is a subset of $n-1$ links such that they connect all the nodes into a *tree* (i.e., an acyclic connected

graph). The number of different spanning trees contained by a network is one possible way of trying to measure its complexity [149].

- *Off-diagonal complexity*: This is related to the degree assortativity of the network; the idea being that a network's links can be grouped into different types, based on the difference in the degrees of the two nodes being linked. Greater heterogeneity in the types of links occurring in a network is taken as a sign of greater complexity [149]. More precisely, one defines a matrix C, where C_{ij} is the total number of links between nodes with degree i and nodes with degree j. One then sums the diagonals of the upper triangle of this matrix (since it is symmetric, for an undirected network); if the number of unique degrees is M, then we have, for $i = 0, 1, ..., M-1$:

$$L_i = \sum_{j=1}^{M-i} C_{j(j+i)}, \tag{1.21}$$

where L_i is the sum of entries in the i^{th} diagonal. The entries in the principal (0^{th}) diagonal represent links between nodes with the same degree; the first diagonal above it counts links between nodes whose degrees differ by one; and so on. Thus, if there is no preference for nodes of certain types to attach to each other, then the sums of all the diagonals should be about the same. The entropy of the distribution of links (with m denoting the total number of links) across these sums gives the off-diagonal complexity O:

$$O = -\sum_{i=0}^{M-1} \frac{L_i}{m} \log\left(\frac{L_i}{m}\right). \tag{1.22}$$

- *Efficiency*: This is a measure of how well nodes can communicate with each other [160]; it is defined as the mean of the inverses of the geodesic distances between all pairs of nodes. If the distance between nodes i and j is denoted

$d(v_i, v_j)$, then efficiency E is given as:

$$E = \frac{1}{n(n-1)} \sum_{i,j:i \neq j} \frac{1}{d(v_i, v_j)}. \tag{1.23}$$

- *Efficiency complexity*: This measures the increase in efficiency relative to the least efficient network with the same number of nodes, which is the linear chain [149]. The efficiency of a chain of n nodes is:

$$E_{chain} = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \frac{n-i}{i}. \tag{1.24}$$

  In order to normalise the efficiency complexity E_{comp} to the range $[0, 1]$, it is defined for a network with efficiency E as:

$$E_{comp} = 4 \left(\frac{E - E_{chain}}{1 - E_{chain}} \right) \left(1 - \frac{E - E_{chain}}{1 - E_{chain}} \right). \tag{1.25}$$

- *Chromatic number*: Related to the *graph colouring* problem, which involves assigning a colour or category to each node in a graph such that no two linked nodes have the same colour. The problem is to find the minimum number of colours needed to do this for a given graph; this number is called the *chromatic number*. Finding the actual number is an instance of an NP-hard problem, and is generally intractable except for very small networks. However, heuristic algorithms can be used to obtain an estimate; here we make use of the Sage implementation (`http://www.sagemath.org`), which employs the Dancing Links algorithm [126, 152].

- *Travelling Salesman Problem*: Given a set of nodes with pairwise distances between them, this problem involves finding the optimal sequence in which to visit them so as to minimise the total distance travelled. Given a network, we

can convert it into an instance of this problem by using graph distances between node pairs. This is also intractable to solve for large instances; we use heuristics (including a cross-entropy method, a genetic algorithm and simulated annealing; details in Section 3.4.1) to solve it approximately and for a given network, use the time taken to get the solution and the total distance to be travelled for that solution as diagnostics.

1.1.4.7 Spectral diagnostics

- *Eigenvalues*: The eigendecomposition of an adjacency matrix provides information about the network's structural properties. As noted in Section 1.1.4.2, the components of the principal eigenvector can be interpreted as a set of measures of node centrality. The corresponding (largest) eigenvalue quantifies the importance of this eigenvector.

- *Spectral scaling*: This notion was introduced by Estrada [78] as a way of characterising unweighted network topologies. One examines how *odd subgraph centrality* (the subgraph centrality due to just odd-length closed walks) of nodes scales with their eigenvector centrality. It has been shown that there is a 'perfect scaling' that corresponds to the lack of topological bottlenecks, or to networks that are both sparsely populated and highly connected [77]; this is given by

$$[\gamma_1(v)]^2 \sinh(\lambda_1) \approx sc_{\text{odd}}(v) \tag{1.26}$$

for each node v, where $\gamma_1(v)$ is its eigenvector centrality, λ_1 is the largest eigenvalue, and sc_{odd} is the odd subgraph centrality. By examining how the actual $\gamma_1(v)$ values deviate from those predicted by this scaling relationship, one can place the network into a given topological category. Positive deviations (i.e., $\gamma_1(v)$ higher than perfect scaling) correspond to the presence of 'quasibipartites',

groups of nodes partitioned into disjoint subsets, whereas negative deviations correspond to 'quasicliques' or densely connected subgroups (i.e., communities) [78].

- *Laplacian*: The Laplacian matrix L of a graph is given by $L = D - A$, where A is the adjacency matrix, and D is a diagonal matrix with node degrees along the diagonal, i.e., $D_{ij} = \text{degree}(v_i)$ if $i = j$ and $D_{ij} = 0$ otherwise (v_i denotes the i^{th} node). The spectrum of this matrix can be used to compute several interesting graph properties; for instance, the second smallest eigenvalue gives the *algebraic connectivity*, a measure of how well connected the graph is (it is equal to 0 if the graph is not connected). The smallest non-zero eigenvalue is called the *spectral gap* or the *Fiedler value*, which is also a measure of connectivity (it is equal to the algebraic connectivity for connected graphs); it determines the rate of convergence of a random walker[6] on the network.

1.1.4.8 Community-based

- *Partition entropy*: This is the entropy of the community size distribution of a network partition into communities [203]. If one has a partition $\mathcal{P}$ with communities $C_1, C_2, \cdots, C_N$, and $|C_i|$ denotes the number of nodes in community i, then the entropy of this partition is given by

$$\text{entropy}(\mathcal{P}) = -\sum_{i=1}^{N} \frac{|C_i|}{n} \log \frac{|C_i|}{n}. \tag{1.27}$$

- *Functional cartography*: Guimerà and Amaral [114] devised a way of assigning functional roles to nodes in a network, given a partition into communities. Their classification of nodes used two diagnostics:

[6] A random walk on a graph is a stochastic process which starts at a particular node and then at each step jumps to a uniformly randomly chosen neighbour of the current node. It is said to have *converged* when the probability distribution of the walker over the graph's nodes becomes stable.

1. The *within-community degree* gives the number of connections a node has within its own community. It is normalised to a z-score, which for the i^{th} node is given by the formula

$$z_i = \frac{\kappa_i - \bar{\kappa}_{s_i}}{\sigma_{\kappa_{s_i}}}\,, \tag{1.28}$$

where s_i denotes the community label of node i, κ_i is the number of links of node i to other nodes in the same community s_i, the quantity $\bar{\kappa}_{s_i}$ is the mean of κ_i over all nodes in community s_i, and $\sigma_{\kappa_{s_i}}$ is the standard deviation of κ_i in community s_i.

2. The *participation coefficient* of node i measures how its links are distributed amongst different communities. It is defined as

$$P_i = 1 - \sum_{s=1}^{N} \left(\frac{\kappa_{is}}{k_i}\right)^2\,, \tag{1.29}$$

where N is the number of communities, κ_{is} is the number of links of node i to nodes in community s, and k_i is the total degree of node i. The participation coefficient approaches 1 if the links of node i are uniformly distributed amongst all communities (including its own) and is 0 if they are all within its own community.

Based on these quantities, the Guimerà-Amaral role classification first distinguishes between 'community hubs' and 'non-hubs'; the former are defined as those nodes with within-community degree $z \geq 2.5$. [7] In this context, the term 'hub' is applied to nodes with high within-community degree, so 'non-hubs' might have high total degree. One can further partition both 'community hubs' and 'non-hubs' on the basis of the participation coefficient P as follows [114]:

[7] The precise thresholds used by them are arbitrary, but provide a means of obtaining some indication of how nodes are distributed across different roles.

1. Non-hubs can be divided into 'ultra-peripheral nodes' ($P \leq 0.05$—virtually all links within their own community), 'peripheral nodes' ($0.05 < P \leq 0.62$—most links within their own community), 'non-hub connector nodes' ($0.62 < P \leq 0.80$—links to many other communities), and 'non-hub kinless nodes' ($P > 0.80$—links distributed roughly homogeneously amongst all communities).
2. Community hubs can be divided into 'provincial hubs' ($P \leq 0.30$—vast majority of links within own community), 'connector hubs' ($0.30 < P \leq 0.75$—many links to most other communities), and 'kinless hubs' ($P > 0.75$—links distributed roughly homogeneously amongst all communities).

- *Mesoscopic response functions*: These were defined by Onnela *et al.* [203] as a way of capturing a network's middle-level (community) structure across a range of resolutions. They use the Potts method (Section 1.1.3.2) to find communities, varying the resolution parameter γ such that it ranges from the whole network being in a single community to every node being in a separate community. Across this range of resolutions, they track three different properties of the network partition: the number of communities, the partition entropy, and the partition quality, as given by H in Equation (1.2). In this thesis, we will use this approach with community detection methods other than the Potts method as well. H cannot be defined in the same way for those, since the resolution parameter γ is specific to the Potts method; so here we will track only the number of communities and the partition entropy. We will also compute the first- and second- derivatives of these quantities (with respect to the resolution parameter) using finite-difference approximations (see Appendix A for details).

1.1.4.9 Network energy and entropy

In the statistical physics literature, networks have been studied with respect to various ensembles, such as that of all networks with a given degree distribution. Using such an ensemble, Bianconi [42] defined the *energy* of a given network as the logarithm of the number of indistinguishable networks that can be constructed with the same degree distribution. The expression for computing the energy (denoted $\mathcal{E}$) is:

$$\mathcal{E} = \log\left(\prod_k k!^{n_k}\right), \tag{1.30}$$

where n_k denotes the number of nodes with degree k. Network *entropy* with respect to the given ensemble may be defined as the logarithm of the number of ways the total number of links in the network (denoted by m) can be distributed into any degree sequence corresponding to the given distribution, with the nodes being regarded as unlabeled [42]. This entropy (denoted $\mathcal{S}$) is computed as follows:

$$\mathcal{S} = \log\left(\frac{(2m)!}{\prod_k (kn_k)!}\right). \tag{1.31}$$

We will discuss more general notions of network ensembles and entropy in Chapter 4.

1.1.4.10 Sampling

For many large networks, some of the diagnostics discussed are difficult to evaluate due to computational (time or memory) constraints. To attempt to partly address this issue, in this thesis we also compute several of the diagnostics on network subsamples. In particular, we will use two sampling methods: snowball sampling [110], a well-known method for obtaining network samples, and forest fire sampling [163], which can be thought of as a generalisation of snowball sampling.

Snowball sampling involves starting the sample from a random node, adding all of its neighbours to the sample, then adding all of their neighbours, and so on until a desired size has been reached. We use samples of 50 or 100 nodes, as these are sizes for which all our diagnostics can generally be computed quickly and easily.

Forest fire sampling adds a further stochastic element to snowball sampling, by randomly including at each step some fraction of the neighbours of the current nodes, as opposed to all of them. This model includes an additional parameter $p \in [0, 1]$, known as the *forward burning probability*. Starting the 'fire' from a random node v, one first generates a random number x that is geometrically distributed with mean $p/(1-p)$. Then x of the neighbours of v (or all of them, if there are fewer than x) are added to the sample ('burnt'). This process is then repeated recursively for all the newly added nodes, with only unburnt nodes being considered for addition at each step, until a desired sample size has been reached. If the fire 'dies out' before sufficient nodes have been sampled, then it is re-started at a new randomly chosen node. Note that setting $p = 1$ is equivalent to snowball sampling. At the other extreme, $p = 0$ corresponds to sampling the nodes uniformly at random, as each fire will die out at the very node it starts at.

We will make use of sampling in two ways. Firstly, whilst attempting to compute our full set of network diagnostics for large real-world networks (in Chapter 3), we will also compute 'sampled versions' of many of these diagnostics, i.e., we will take a single snowball sample of 100 nodes from the network and compute the diagnostic on that. We note that there is no presumption that these subsamples will in any way preserve the structural characteristics of the entire network; indeed it has been shown that this is not true for at least some, and perhaps most, types of networks [252]. However the sampled versions of the diagnostics are just added on in an experimental fashion to our list of several hundred network properties, and since they do not turn out to be of interest for the case studies we present, we choose not to examine them

in detail (see the discussion in Section 3.2).

Secondly, in Chapter 5 we will attempt to fit evolutionary models to protein interaction networks. We will do this by comparing ensembles of subsamples from the real networks and model-generated networks, in order to be able to feasibly compute the various network diagnostics used. In this case, we will look at the effect of different sample sizes (50 versus 100 nodes) and the two different sampling procedures described, to obtain an indication of the extent to which our results are robust to changes in these choices.

1.1.5 Types of real-world networks

One of our goals in this work is to examine relationships between different sorts of networks, and highlight both commonalities as well as structural aspects that typify particular classes of networks. We thus sought to obtain from various sources[8] a fairly diverse set of real-world network data sets (more details in Section 3.2). The various kinds of networks we study can be broadly classed into two types: interaction networks and similarity or correlation networks.

- *Interaction networks*: These capture interactions and information flows between the elements of a real-world system. Examples of such networks we look at include social networks of various kinds (including Facebook data), biological networks such as protein interaction and metabolic pathways, neuronal networks, and networks of fungal growth. In most cases, the networks we study in this category are treated as undirected and unweighted.

- *Similarity*: These networks represent similarities or correlations between different components of a system. Examples include networks of political cosponsorship amongst members of the US Congress, networks of stock price

[8]In particular, the major data set we use for comparing different types of real-world networks was obtained from Dan Fenn [203].

correlations in financial markets, and networks of co-expression of genes in a biological cell. Such networks are usually treated as undirected (including all of those used here) and weighted, with weights (representing magnitudes of correlation/similarity) typically in the range $[0, 1]$. They also tend to be nearly fully connected (i.e., almost every pair of nodes is linked), as it is rare for any two entities to be entirely uncorrelated.

1.1.6 Generative models for networks

In addition to studying properties of real-world networks, we also generate synthetic networks using multiple sorts of models, which take as inputs some parameter settings and output a randomly generated network (these are known as *generative models*). We use these in two ways: Firstly, by comparing synthetic and real-world networks one can hypothesise possible mechanisms for the emergence of certain kinds of empirically observed network structures (see Chapter 5). Secondly, synthetic networks can provide a controlled benchmark where one can constrain certain aspects of network structure and then examine the variations in, or relations between, other aspects (see Sections 3.4, 4.5, and 5.5.1). Here we describe briefly the different sorts of network models used in this thesis.

1.1.6.1 Erdős-Rényi

The most widely used models for generating random graphs are the *Erdős-Rényi model*, also known as the $G(n, p)$ model [106], and its closely related counterpart the $G(n, m)$ model [76]. The latter randomly generates a graph with n nodes and m links; all possible such graphs have equal probability of being generated. The $G(n, p)$ model generates a graph with n nodes where each possible link is independently present with probability p. The expected number of links is then $p\binom{n}{2}$; if one uses this value for m, then in the limit as $n \to \infty$ the two models become equivalent.

1.1.6.2 Random geometric graphs

The model of *random geometric graphs* [208] has been proposed as a means of generating spatially-embedded random graphs. It has two parameters: n, the number of nodes, and r, a distance threshold. A bounded geometric region is defined, typically the unit square, and n points are placed in it, independently uniformly at random. Subsequently, only pairs of points that are separated by a distance less than the threshold r are joined by (undirected, unweighted) links to obtain the graph structure.

1.1.6.3 Preferential attachment and 'scale-free' structure

It has been proposed that several real-world network growth processes proceed according to some form of (linear) *preferential attachment*, whereby the probability of a node acquiring new links is proportional to the number of links it already has (i.e., its current degree) [28,70,243,282]. This results in the node degrees being distributed according to a *power law*, such that the probability of a node having degree k is given by $p(k) \propto k^{-\gamma}$ for some exponent γ (which is determined by the precise preferential attachment mechanism chosen; for the widely-used Barabási-Albert version, by default $\gamma = 3$ [28]). It has been claimed that several real-world data sets show such a distribution (the corresponding networks have often been referred to as 'scale-free'); however these claims have often not stood up to scrutiny. For instance, Clauset *et al.* [62] used a principled statistical framework to show that a number of data sets claimed to follow power-law distributions were in fact better explained by other distributions such as the log-normal. Recently, Stumpf and Porter [251] have also argued against the claimed ubiquity and scientific utility of power laws.

In practice, variants of the *configuration model* [36,241] are often used to generate random networks that have a fixed (expected) degree distribution, such as a power law. Roughly, this approach proceeds by assigning to each node a number of 'stubs'

or half-links, with the number being drawn from the specified distribution and the constraint that the total number of stubs should be even. The stubs are then paired up randomly to form links between nodes.

1.1.6.4 Watts-Strogatz networks

Stanley Milgram's experiments [184, 264] established the idea that human social networks might be characterised by having a surprisingly small diameter: this resulted in the famous phrase 'six degrees of separation'. The term *small-world* networks was coined to refer to such structures, denoting the existence of a relatively short path between any pair of nodes, compared to the total number of nodes. More precisely, a small-world network family is defined as one where the typical distance d between a randomly chosen pair of nodes scales as the logarithm of the number of nodes: $d \propto \log n$. Watts and Strogatz [273] proposed a widely-used mechanism for generating such small-world families, which starts with nodes arranged on a circular lattice, where each node is linked to its k nearest neighbours ($k/2$ on either side). They showed that by then randomly rewiring a relatively small fraction of the links (i.e., detaching them from one endpoint and re-attaching to a random node in the network), networks with the small-world property were obtained. This model has three parameters: the number of nodes n, the mean degree k, and the probability with which any given link will be rewired p.

1.1.6.5 Community detection benchmark networks

These are networks that have a relatively clear-cut community structure, i.e. they contain densely connected subnetworks. One way of generating such networks is to use a *block model*, which involves dividing nodes into groups (or blocks) such that there are distinct link-formation probabilities for within-group and between-group links. This sort of approach is used by Lancichinetti *et al.* [159] to generate benchmark sets

of networks for testing community detection algorithms. Their model assumes that both node degrees (denoted by k) and community sizes (denoted by n_c) follow power-law distributions: $p(k) \propto k^{-\gamma}$ and $p(n_c) \propto n_c^{-\beta}$. The exponents γ and β are model parameters. The model proceeds by assigning degrees to n nodes from the degree distribution, fixing the minimum and maximum degrees such that the mean degree is $\langle k \rangle$ (another parameter). The nodes are connected similarly to the configuration model, and placed into communities such that a fraction $1 - \mu$ of links are within communities and a fraction μ (the *mixing parameter*) are between communities.

1.1.6.6 Exponential random graph models

These have been used largely in the study of social networks, as a way of generating random networks with certain structural features (typically to match those of some empirical network) [94, 223]. The idea behind exponential random graph models (ERGMs) is to define the maximum entropy probability distribution over networks that satisfy desired constraints. This distribution is of the following form:

$$P(A) = \frac{1}{\kappa} \exp\left\{ -\sum_{i=1}^{l} \beta_i s_i(A) \right\}, \tag{1.32}$$

where $P(A)$ is the probability of generating network A, κ is a normalising constant, there are l network statistics to be constrained (those specified by the s_i functions), and the β_i values give the corresponding weights, which are the model parameters (to be fitted to maximise the probability of observing a given network or set of networks). The simplest possible kind of model is given by choosing just one network statistic to constrain; this is typically the density of links; in this case, it becomes equivalent to the $G(n, p)$ model. In practice, these models have proven difficult to use for even remotely large networks (e.g., those with more than a couple of thousand nodes), especially when constraining less simple structural features like motif counts,

in particular because of the computational cost of evaluating the constant κ [111,260].

1.1.6.7 Duplication-divergence for gene/protein evolution

In the domain of subcellular biological networks, gene duplication and subsequent functional divergence has been proposed as one of the underlying evolutionary mechanisms [113, 201, 257, 284]. It is believed that new genes are often formed by duplication of existing ones and that such duplicates subsequently can take on novel functionality by rewiring their interactions. Statistical network growth models based on this idea have been formulated—particularly for protein-protein interaction networks [218, 219]— and we will make use of some of these in this thesis (see Sections 3.3.1 and 5.5).

1.2 Interactomics

One of the specific focus areas of this thesis is the study of networks of protein-protein interactions (also known as the *interactome*). Indeed, we begin in Chapter 2 with an examination of the possible biological roles played by hubs in these networks. Subsequently, our observations of certain shortcomings in the ways these networks have been previously studied motivates the development of a more general approach to the study of networks of various kinds, which is what we outline in Chapter 3. Then in Chapter 5 we return to looking at how our approach might assist in providing insights into the mechanisms of the interactome evolution. In this subsection we provide an introduction to proteins, their relevance in biology, and in particular to protein-protein interaction networks and why they are worth studying.

1.2.1 Proteins in biology

Proteins are one of the fundamental building blocks of living organisms, and are the major components of the machinery inside cells. In general, basic biological functions and processes at the subcellular level are carried out by groups of proteins acting in concert. This is why an understanding of how they work and interact with each other is so important.

Proteins consist of chains of amino acids (also known as polypeptides). The recipes for constructing proteins are contained in our genes, encoded in a language with a four-letter alphabet: A,C,G,T, corresponding to Adenine, Cytosine, Guanine, and Thymine, the four different nucleotides comprising DNA. Thus, one can think of a gene as a string that uses this alphabet. The flow of information from genes to proteins involves two steps: *transcription* and *translation*. (This is sometimes referred to as the 'central dogma' of molecular biology.) Transcription is essentially a copying process: the nucleotide sequence of the gene is copied to an RNA string, known as *messenger RNA* (mRNA). RNA also uses four nucleotides, though Thymine is replaced by Uracil. The mRNA, carrying the genetic sequence, is then transported to a ribosome, one of the protein-manufacturing factories of the cell. The second step of translation occurs here; this involves going from the four-letter alphabet of RNA to the twenty-letter alphabet of amino acids. This happens via the *genetic code*, which specifies a mapping from three-letter RNA strings (known as *codons*) to amino acids. The ribosome reads in the mRNA sequence and manufactures the corresponding amino acid chain; this chain then folds into a three-dimensional structure to form a protein.

Different cells have different requirements for proteins, and these also vary over time for any given cell. Thus, whilst every cell contains the entire *genome* (the complete recipe book for all proteins), it is critical to manufacture only the necessary proteins and to do so in appropriate quantities. The number of copies of a given protein present in a cell is known as its *expression level*; whilst this is controlled at

multiple stages, a key step is via regulating the number of copies of mRNA produced from the corresponding gene (generally known as the gene's expression level). Hence, it is easy to see why there is so much interest in studying gene expression across different types of cells and for different physiological conditions [75, 100, 142, 236, 253]; such expression levels are typically measured using the technology of *microarrays*. A microarray consists of a number of microscopic DNA/RNA spots, each one containing a small quantity of a given nucleic acid sequence (known as a *probe*), which can bind to its complementary sequence (the *target*); such binding is typically detected and quantified via labelling with fluorophores, chemicals which re-emit light upon excitation. The amount of fluorescence detected from a given spot serves as a measure of the relative expression level of the corresponding gene sequence.

Proteins have a wide range of functions, including metabolism, transport, signalling, etc.; they are also involved in the transcription and translation processes themselves, which are carried out by protein complexes. The cellular circuitry essentially consists of proteins and their interactions. Some proteins form large complexes with a specific function; for instance, the ribosomes comprise over 50 different proteins [21]. Other proteins such as kinases are part of sequential signalling cascades, which can serve to trigger events such as cell division. Mapping this circuitry is an important step in understanding not only how cells work but also the roles of different proteins within them [29]. It is also one stage in the broader biological project of understanding life at its many different scales of organisation. One can think of organisms as comprising complex interacting systems at several levels: organ systems, organs, tissues, cells, etc. Each level builds on the one below, and cells can be seen as the most fundamental biological building blocks, the lowest level at which we see a degree of autonomous 'life'. Understanding subcellular networks of control thus appears to be particularly important to unravelling life's mysteries.

1.2.2 Data sources

1.2.2.1 Protein-protein interaction data

Several experimental methods can be used to gather protein interaction data. These include high-throughput yeast two-hybrid (Y2H) screening [97, 98, 133, 265]; affinity purification of tagged proteins followed by mass spectrometry (AP/MS) to identify associated proteins [102,127]; curation of individual protein complexes reported in the literature [181]; and *in silico* predictions based on multiple kinds of gene data [270]. A more recent technique, known as the protein-fragment complementation assay [254], is even able to detect protein-protein interactions in their natural environment within the cell. However, to our knowledge only one large-scale study has used this technique thus far [254]. Each of these methods gives an incomplete picture of the interactome; for instance, a recent aggregation of high-quality Y2H data sets for *Saccharomyces cerevisiae* (yeast; the best-studied organism) was estimated to represent only about 20% of the whole yeast binary protein interaction network [280].

Each technique also suffers from particular biases. It has been suggested that Y2H is likely to report binary interactions more accurately, and (due to the multiple washing steps involved in affinity purification) it is also expected to be better at detecting weak or transient interactions [280]. Converting protein complex data into interaction data is an issue with AP/MS. This method entails using a 'bait' protein to 'capture' other proteins that subsequently bind to it to form complexes. Once one has obtained these complexes and identified their proteins using mass spectrometry, one generally assigns protein-protein interactions using either the spoke or the matrix model [120]. The *spoke model* only counts interactions between the bait and each of the proteins captured by it, whereas the *matrix model* counts all possible pairwise interactions in the complex. Unsurprisingly, the actual topology of the complex is generally different from either of these representations. As opposed to Y2H, AP/MS is expected

to be more reliable at finding permanent associations. Two-hybrid approaches also do not seem to be particularly suitable for characterising protein complexes, giving rise to the view that formation of complexes is not merely a summation of binary interactions [102]. Thus, the two major techniques appear to be disjoint and to cover different aspects of the interactome, and there is some evidence to suggest that the differences between data sets from these sources correspond largely to false negatives rather than false positives [280].

1.2.2.2 Gene expression data

Gene expression is generally measured at the transcript level—i.e., in terms of the number of mRNA copies produced from a given gene. DNA microarrays are a high-throughput technology that allow the measurement of the expression of thousands of genes simultaneously [21]. A large number of such expression profiles have been produced, which allow one to track how levels vary across a range of conditions or over time—for instance, when the yeast cell is subject to various sorts of stresses such as temperature shocks or chemical exposure [100] or as it goes through the different stages of the cell cycle [247]. Such data is generally recorded in the form of log ratios—i.e., the logarithm of the ratio of the expression level in a given condition to the expression level in some background condition. However, because different genes display different levels of variability in expression and there is also variation across experiments, there is a need to normalise microarray data to make it comparable across genes and conditions. There are multiple ways of doing this and no consensus on which one works best [167]. Two widely used methods that were applied to expression data used in Chapter 2 are the Affymetrix MAS5 algorithm [129] and the GCRMA algorithm [276].

As noted earlier, transcription to mRNA is only the first step of protein expression; subsequently there are also mechanisms of regulation at the post-transcriptional

level—e.g., RNA silencing carried out by microRNA [30,31], which can alter the number of mRNA copies that actually get translated by the ribosomes. Thus, even though mRNA levels have been widely used as proxies for protein expression (due to the difficulty of quantifying protein levels themselves on a large scale), there is not necessarily a strong correlation between the two. In fact, one recent study has suggested that the link might be very weak [91], indicating the need for a great deal of caution in interpreting the results of protein co-expression analyses based on microarray data (e.g., Han *et al.* [121], as discussed in Chapter 2).

1.2.2.3 The Gene Ontology

In order to obtain information on the functions of proteins, we use annotations from the Gene Ontology [25]. This provides a controlled and hierarchically structured vocabulary of terms that describe functionality of genes and gene products at different levels of specificity. These terms are linked into a tree structure, where each term has a parent that represents a supercategory of it. Within the ontology, there are three subontologies, which represent different sorts of annotations: `Biological Process`, `Cellular Component`, and `Molecular Function`. For each subontology, the top-level term or root of the corresponding tree is the name of the subontology itself, which serves as a catch-all description. As one goes further down the tree, the terms become more specific; for instance, in the Biological Process tree, a high-level term (a direct child of the root) is `biological regulation`; a lot of genes/proteins can be annotated with something this generic. An example of a lower-level term is `regulation of carbohydrate utilisation`; this is two levels down from (i.e., a 'grandchild' of) `biological regulation`, and only 5 genes or proteins are annotated with it.[9] The functional similarity of two proteins can be assessed based on the overlap in their annotations; we discuss how to do this in Chapter 2.

[9] `http://www.geneontology.org/`, accessed 20-11-2011

1.2.3 Protein interaction networks

With increasing availability of data on protein-protein interactions [97, 102, 133, 229, 254, 265, 280], there has recently been a focus on studying these networks and attempting to relate their structure to their functionality, for instance by studying how network topology might serve to identify the roles of particular proteins [141, 176, 187, 240, 278] or examining the nature of modules in the interactome [101, 222, 248]. It has been reported that these networks generally show substantial community structure, with many communities being much more functionally homogeneous than might be expected by chance [13,58,74,165,171]. Thus the organisation of the interactome appears to be partly modular, with modules often seemingly corresponding to protein complexes. It has also been noted that interactome hubs tend to include essential proteins (i.e., those critical to the organism's survival under standard lab conditions) [136, 266, 286], although there is not a clear correspondence between high degree and essentiality; it has been suggested that many hubs might be important only locally (i.e., within their own modules) [119, 121, 176].

An issue that has perhaps received insufficient consideration is that protein interaction networks, as constructed from data obtained via the standard techniques (see Section 2.2.1), are unable to capture the dependence upon prevailing physiological conditions of the actual interactions occurring *in vivo*. For instance, actively expressed proteins vary amongst the tissues in an organism and also change over time. Thus, the specific parts of the interactome that are active, as well as their organisational form, might depend a great deal on where and when one examines the network [121,125,256,275]. One way to attempt to incorporate such information is to use gene expression data. Han *et al.* [121] examined the extent to which hubs (defined by them as proteins with degree at least 5) in the yeast interactome are co-expressed with their interaction partners. They did this by defining a quantity referred to as the *averaged Pearson correlation coefficient* (avPCC). Suppose the expression profile of

protein a is denoted by the vector $[x_a^1, x_a^2, ..., x_a^T]$, where the values are dimensionless log ratios (see Section 1.2.2.2), and T denotes the number of different conditions or time points for which measurements have been made; let the mean of the values in this vector be denoted by $\bar{x}_a$. Let this protein have degree k, i.e., it has k interaction partners which we denote $a_1, a_2, ..., a_k$; let $x_{a_i}^t$ denote the expression level of the i^{th} partner at time point t, and $\bar{x}_{a_i}$ the average expression level of the partner across all time points. Then the avPCC can be computed as follows:

$$\text{avPCC}(a) = \frac{1}{k} \sum_{i=1}^{k} \frac{\sum_{t=1}^{T} (x_a^t - \bar{x}_a)(x_{a_i}^t - \bar{x}_{a_i})}{\sqrt{\sum_{t=1}^{T} (x_a^t - \bar{x}_a)^2} \sqrt{\sum_{t=1}^{T} (x_{a_i}^t - \bar{x}_{a_i})^2}}. \tag{1.33}$$

Based on the observed avPCC distributions, Han *et al.* concluded that hubs fall into two distinct classes: those with a low avPCC (which they called *date* hubs) and those with a high avPCC (so-called *party* hubs; see Figure 1.2). They inferred that these two types of hubs play different roles in the modular organisation of the yeast protein interaction network: Party hubs were construed to coordinate single functions performed by a group of proteins that are all expressed at the same time, whereas date hubs were described as higher-level connectors between groups that perform varying functions and are active at different times or under different conditions.

The validity of such a date/party hub distinction has since been debated in a sequence of papers [33,34,41,275], and there appears to be no consensus on the issue. Despite the controversy, essentially the same idea was also posited for human protein interaction data, with the names *intermodular* and *intramodular* hubs being used in place of date and party hubs [256]; despite the terminology, these too were defined purely in terms of partner co-expression, without taking topological properties into account. Extensions of the idea, such as a three-way categorisation also including *family* hubs (i.e., hubs that show little variation in their expression across conditions and invariably interact with their partners), have also been proposed [153]. In Chapter

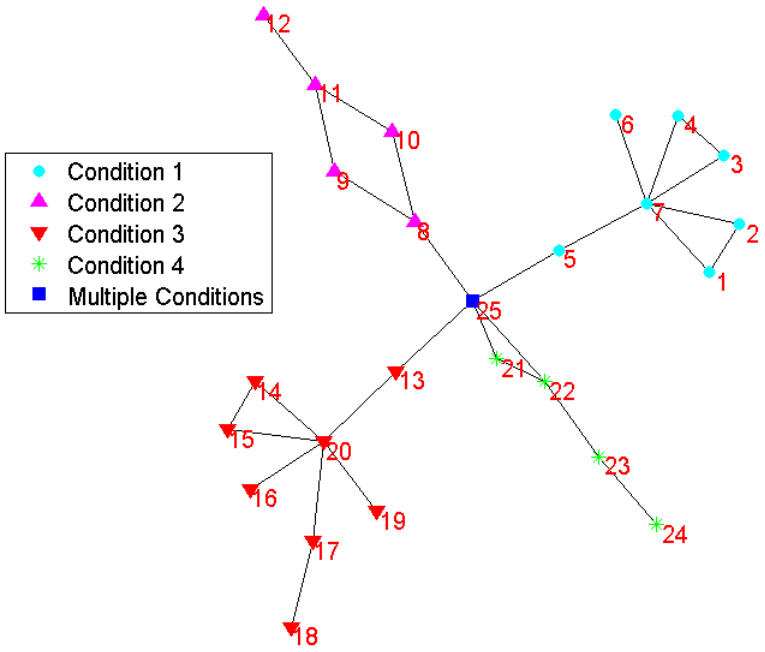


Figure 1.2: **Date and party hubs.**

Different nodes represent proteins that are expressed under different conditions (time and/or space). Here nodes 7 and 20 would be described by Han *et al.* [121] as party hubs, as they are expressed under the same conditions as their interaction partners. Node 25 would be described as a date hub, because its interaction partners are not all co-expressed but show up under a variety of different conditions.

2, we will focus on revisiting this categorisation and discuss it at length.

It should also be noted that interactome data is widely believed to be highly noisy. Several papers have discussed concerns about the completeness and reliability (or lack thereof) of existing protein interaction data sets and their implications for how much biological understanding these networks can really provide [26,27,53,119,234,237,267]. We will look at this issue in Chapter 2 as well; in particular, we seek to examine how much any conclusions about the existence of date and party hubs might be weakened by data uncertainty.

1.3 Machine learning

Machine learning [47] is concerned with algorithmically finding patterns and relationships in data, and using these to perform tasks such as classification and prediction in various domains. In this thesis, we are interested in using machine learning techniques to categorise different sorts of networks and to find relationships between network structure and function (see Chapters 3–5). We now introduce some relevant terminology and provide an overview of the different sorts of machine learning approaches employed by us.

1.3.1 Basics

- *Feature vector*: A typical setting for machine learning is to be given a collection of objects (or data points), each of which is characterised by several different *features*. Features can be of different sorts: e.g., they might be continuous (say, real- or integer-valued) or categorical (for instance, a feature for colour can have values like `green,blue,red`). We will be concerned only with continuous features, in particular, features of network structure. These features correspond to the outputs of the various sorts of diagnostics listed in Section 1.1.4 (and

Table 1.1: **Example design matrix.**

Object	Weight (g)	Colour (0=Green, 1=Red)
Red Apple 1	147	0.90
Red Apple 2	159	0.70
Red Apple 3	170	0.77
Green Apple 1	163	0.17
Green Apple 2	151	0.13
Banana 1	104	0.10
Banana 2	119	0.15
Banana 3	113	0.34
Banana 4	122	0.23
Banana 5	125	0.30

Design matrix for 10 objects and 2 numerical features. The colour spectrum from green to red is mapped to a 0–1 scale (see also Figure 1.3).

additional ones in Appendix A). A vector containing all of the feature values for a given data point is called the *feature vector*; if this is a vector of length d, then one can think of each data point as being mapped to a d-dimensional vector space (in the case of real-valued features, this is $\mathbb{R}^d$), called the *feature space*.

- *Design matrix*: A collection of feature vectors for different data points constitutes a *design matrix*. Each row of the matrix is one data point (i.e., one feature vector), and each column represents the values of a given feature across all of the data points (Table 1.1). The design matrix is the basic data object on which machine learning algorithms operate.

1.3.2 Supervised learning

The task of supervised learning is to learn an association between features and external labels of some kind. A label is typically either one of a finite set of categories (in which case it becomes a classification problem), or continuous-valued (in which case

one has a regression problem). We discuss both of these settings next.

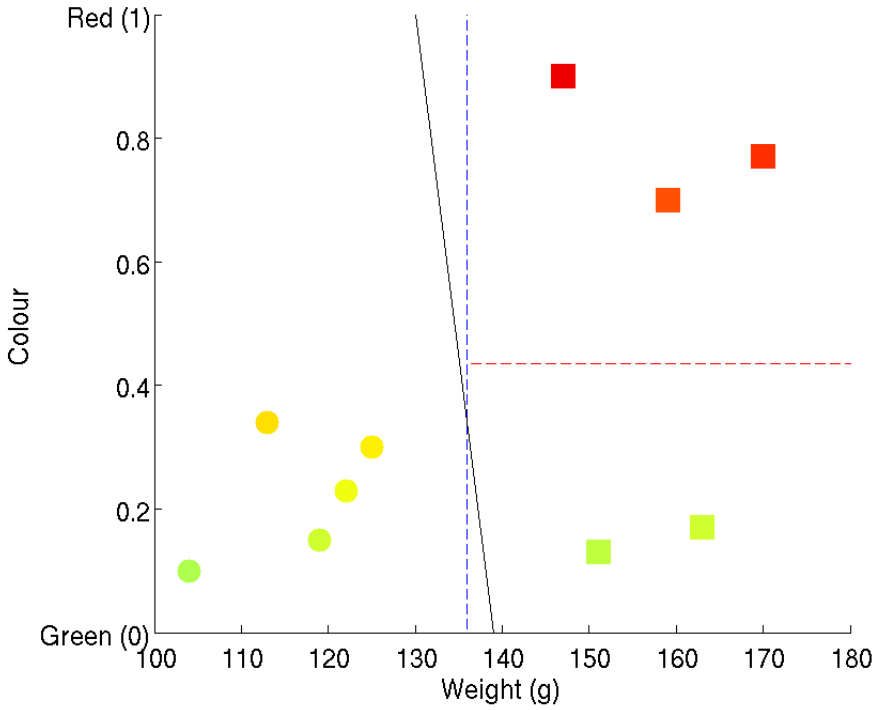
1.3.2.1 Classification

Given a set of objects represented as feature vectors and an associated class label for each object, one would like to learn a model (known as a *classifier*) that can predict the class given the features. The model itself can take on many different forms: linear classifiers, decision trees, neural networks, and support vector machines are a few popular examples [47]. Here we will briefly discuss the first two, as they will be useful later.

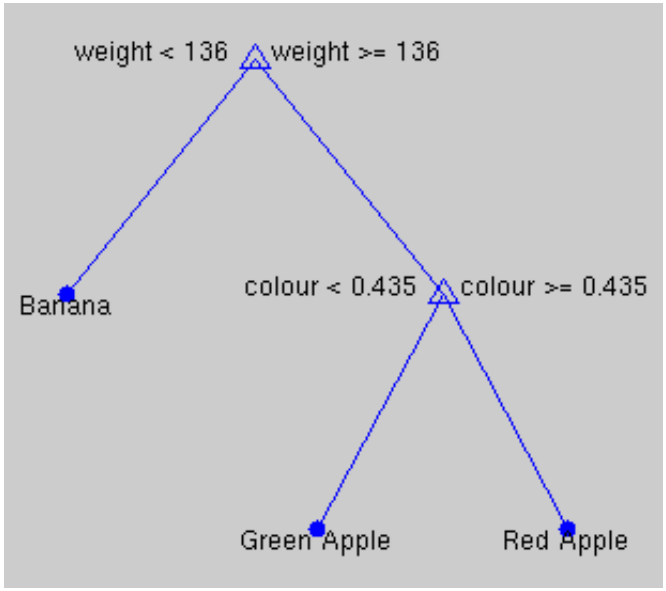
A *linear classifier* uses some linear combination of the features as its criterion for distinguishing between classes [47, 124]. This corresponds to drawing a separating hyperplane in the feature space; in two dimensions, this is a line, as in Figure 1.3(a). Thus, linear classifiers by default are defined for binary classification problems—i.e., those in which there are only two classes. When there are more than two classes, it is typical to use multiple linear classifiers; two possible approaches are *all-vs-all*, in which a binary classifier is learnt for every pair of classes, and *one-vs-all*, in which a binary classifier is learnt to discriminate each class from the combination of all of the other classes.

Decision trees consist of a set of rules based on feature values [54]; they are arranged in the form of a binary tree, as in Figure 1.3(b). Following these rules down the tree specifies increasingly restricted regions of feature space until at some point a leaf node is reached and all points in the corresponding region get assigned to a single class.

Having chosen a particular form of model, one then needs to use the data to learn a specific classifier—typically one that optimises some performance measure. An obvious choice for this measure might be what fraction of the given data points the classifier is able to place in the correct class (known as classification *accuracy*).



(a)



(b)

Figure 1.3: **Example classifiers.**

(a) Data points from Table 1.1 in feature space (colour represented both visually and numerically on the y-axis): one can split them into two classes, bananas (circles) and apples (squares). The black line is a linear classifier separating the data. The apples can be further split into red and green varieties; the dashed lines show the partitions imposed by a decision-tree classifier for the three-class problem. (b) The decision tree.

However, this suffers from the problem of *overfitting*, in that one would like to use the classifier to make predictions on novel data points, and the data set at hand will in general not be representative of the full underlying distribution of points in feature space [47, 124]. Thus, the learnt classifier might tend to fit peculiarities of the data set, thereby worsening its performance on unseen examples. In order to avoid this, it is usual to evaluate a classifier not on the data set used to learn it (called the *training set*), but rather on a separate set (the *test set*); this is known as *out-of-sample* evaluation [47, 124]. Some fraction of the available data (10% or 20% are typical choices) would be designated as the test set and would not be used to train the classifier (but instead to evaluate it). One popular variant of this approach, which allows the use of all data for training, is known as *cross-validation* [189]. In this approach, the data is split into k equal parts, known as *folds* (a common choice is $k = 10$, in which case it is called 10-fold cross-validation).[10] Subsequently, k different classifiers are trained—each time with one of the k parts used as the test set and the rest used as the training set. Thus, the combined test results of these k classifiers allow one to estimate out-of-sample accuracy on the entire data set. This can then be used as a criterion for classifier choice—for instance, via setting model parameters.

1.3.2.2 Regression

When the dependent variable—i.e., the one we would like to predict, given the features we have for the data points—is not a categorical class label but instead a continuous (typically real-valued) quantity, then learning a predictive model for this can be seen as a regression problem. One is required to find a function f that maps from a feature vector $\mathbf{x}$ to an output y: ideally, $y = f(\mathbf{x})$. The simplest form is *linear regression*,

[10] Larger values of k lead to more robust error estimates, as a larger number of classifiers are averaged over and each classifier is trained on a larger number of data points. Thus in this sense the optimal value for k is equal to the size of the data set, which corresponds to having just one point in the test set each time; this is known as *leave-one-out* cross-validation. However, larger values of k also mean greater computational cost in re-running the training algorithm for each fold; thus in practice relatively small values of k are preferred, with $k = 5$ or $k = 10$ being widely used choices.

analogous to linear classification, in which f is a linear combination of the features (plus a possible constant term or offset): $f(\mathbf{x}) = \mathbf{w}.\mathbf{x} + b$ [47, 124]. Here $\mathbf{w}$ is a *weight vector* that represents the coefficients of the different features in f. Thus, $\mathbf{w}$ and b are the parameters in a linear regression model that are to be learnt from the data.

The concepts of training, testing, and cross-validation can be extended to regression once one has defined an appropriate accuracy measure. Suppose one is given a data set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_r, y_r)$, where r is the number of data points. We use this to learn a regression function f, such that $\hat{y}_i = f(\mathbf{x}_i)$. Thus, $\hat{y}_i$ would be the predicted value of the dependent variable for the i^{th} data point. The difference $y_i - \hat{y}_i$ is known as the *residual* for the i^{th} point; this is a measure of the error the learnt function makes in predicting for this instance. In general, lower residuals correspond to higher accuracy; the most common way of evaluating accuracy over a set of points is to take the sum of the squares of the residuals: $\sum_{i=1}^{r}(y_i - \hat{y}_i)^2$. This is sometimes known as the *deviance* [175] or the *squared error* [47, 124].

1.3.3 Unsupervised learning

The task of unsupervised learning is to find patterns in data without any external labelling; most commonly, the patterns of interest are clusters, in which case it is also described as *clustering*. Community detection (Section 1.1.3) is a specific example of clustering in the context of networks. There are a large number of approaches for unsupervised learning [47, 124]; we will now discuss a few that we use in subsequent chapters.

Single-linkage clustering is a distance-based method that involves initially defining a distance measure between pairs of points [242]. If the points lie in a vector space, as in Figure 1.3(a), then this can be a standard measure like Euclidean distance. Having computed the distance between every pair of points, this method then proceeds by initially assigning each point to a separate cluster, and then iteratively finding and

merging the closest pair of clusters until all of the points have been lumped into a single cluster. The distance between a pair of clusters is defined as the minimum of all pairwise distances between points across the two clusters.[11] This leads to a hierarchical clustering: at each iteration of the process, one moves up the hierarchy (or decreases the resolution, as in community detection). One can use a threshold to specify the minimum distance between clusters to terminate the iteration at a particular point and obtain a single set of clusters.

In order to detect patterns in data, it is often useful to map it to a low-dimensional space, where the number of dimensions is typically chosen to be as low as possible whilst capturing the bulk of the variability in the given data set. The canonical way of doing this is via *principal component analysis* (PCA) [207]. The essential idea is to find directions in feature space along which the spread of the data is the greatest; each direction is given by some linear combination of the features. This can be done via *singular value decomposition* (SVD), which is, for non-square matrices, the analog of eigendecomposition [47, 214]. Suppose we denote the $r \times d$ design matrix by X. According to the SVD theorem, this can then be factorised as $X = V\Sigma W^T$, where V is an $r \times r$ orthogonal matrix of eigenvectors of XX^T, W is a $d \times d$ orthogonal matrix of eigenvectors of X^TX, and Σ is an $r \times d$ matrix with nonnegative numbers along the diagonal (with all other entries equal to 0). The PCA transformation is given by $Y = XW$; the matrix Y is also an $r \times d$ matrix; it represents the design matrix in the transformed feature space (the features of which are the principal components). These features will be in decreasing order of the amount of data variance they capture.

[11] Other common choices for this distance measure include the average of all pairwise distances, which leads to *average-linkage* clustering, and the maximum of all pairwise distances, which leads to *complete-linkage* clustering. One drawback of using single-linkage clustering is that it may lead to clusters where some elements are very far apart, as clusters are merged based only on the distance between the closest elements. More generally, this sort of agglomerative clustering is a standard, simple method but the results may not always be easy to interpret, as it gives a hierarchy of clusters at different levels, rather than a single partition. However, we will use it here just to obtain a meaningful ordering of data points for visualisation purposes, rather than attempting an actual partitioning.

In order to obtain a reduced representation (say, in l-dimensional space), one can take the first l columns of W. If we denote this by W_l, then the design matrix in the l dimensions is given by $Y_l = XW_l$. In practice, it is often useful to choose $l = 2$ in order to produce a two-dimensional plot of the data; this allows for visual inspection and can aid in the detection of intuitive clusters or patterns (we make use of this in Chapter 3). However such dimensionality reduction of course also involves throwing away information, and one has to be cautious in interpreting the results, particularly if the reduced dimensions leave a substantial proportion of the variance in the data uncaptured.

One limitation of PCA is that the reduced dimensions must be linear combinations of the given features. It can sometimes be useful to select "directions" that are not straight lines in feature space; for instance, if all of the data points lie along a circle, then one actually needs only a single dimension to capture the variation between them, but PCA will not be able to detect this. To account for this, several methods have been developed in recent years for non-linear dimensionality reduction [161, 162, 227, 258]; the one we will use is known as *Isomap* [258]. The idea behind Isomap is to capture the local geometry of the surface on which the points sit in feature space. A weighted network using these points as nodes is defined as follows: each data point is connected to its k nearest Euclidean neighbours in the space with links of weight equal to the Euclidean distance, with the parameter k to be specified by the user.[12] A distance matrix D between points is then defined by using weighted distances (see Section 1.1.2) in this network. One obtains the eigendecomposition of D (which is analogous to X^TX above), and the top l eigenvectors (analogous to W_l) then define the coordinates for an l-dimensional embedding. The amount of data variability captured in the reduced space can be quantified via the *residual variance*,

[12]One would like k to be relatively small, as the objective is to approximate local geometry. However, if k is too small then it might lead to a sparse or disconnected graph. In our usage we choose k to be the smallest number which leads to a connected graph containing all the data points in the set under consideration.

which can be computed as $1 - R^2(D, D_l)$, where R denotes the linear correlation coefficient, and D_l is the matrix of pairwise Euclidean distances between points in the l-dimensional embedding.

1.4 Overview

The work in this thesis begins with an examination of a specific issue concerning the study of protein interaction networks: the classification of hubs into date and party types (see Section 1.2.3). In Chapter 2, we re-examine protein interaction data from multiple angles. In particular, we attempt to obtain a network-based categorisation of hubs (via their relationship to community structure) into the sorts of roles proposed for date/party hubs, and we investigate how such categorisations match up to definitions based on gene-expression dynamics. We find a lack of correspondence between the two definitions, indicating that date and party hubs do not have the network characteristics that were originally attributed to them. We also try an alternative link-centric (as opposed to node-centric) approach to thinking about roles in protein interaction networks, finding that geodesic link betweenness centrality (see Section 1.1.4.2) appears to have a strong negative correlation with the functional similarity of the proteins being linked. This provides one path to associating network structure with function, and it is also reminiscent of the weak/strong tie distinction observed in social networks [112, 216], whereby social links between people with relatively low levels of interaction ('weak' links) tend to be the most important for global network connectivity and information flow, given certain assumptions about information transfer [204].

In general, one purpose of modelling real-world systems—for instance as networks—is to understand better how their structure relates to their functionality. Whilst people have attempted to address this issue in many different ways across

many different domains, the example of date and party hubs highlights how appearances of simplistic correlations between the two can be misleading. Networks can be characterised in many different ways, and *a priori* it is hard to know what aspects of network structure are of interest in a specific context. Accordingly, subsequent to Chapter 2 we expand the scope of this thesis towards an attempt to develop a more comprehensive approach to the analysis of networks than has traditionally been adopted. We call this a *high-throughput* approach, as it involves simultaneously examining and comparing both a large variety of networks and a large variety of network diagnostics or features. This also represents a step towards consolidating the many different strands of existing literature and techniques developed in different disciplines, thereby potentially helping to establish cross-disciplinary links. The observation that a phenomenon first observed in social networks can also be relevant to understanding protein interaction networks provides additional motivation for expanding our horizons in this fashion.

In Chapter 3, we describe our high-throughput methodology. We start with a large and diverse data set of networks and compute a large number of properties—i.e., features—of these networks (thus creating a design matrix). The objective of our work is to attempt to leverage this data to help improve understanding of the systems and diagnostics being studied. Due to the scale involved, some automation is necessary to highlight directions which may be worth pursuing in greater detail. We thus take a machine learning approach to finding relationships and patterns of interest, motivated by specific scientific questions. We demonstrate how our approach can be used to organise and classify networks, as well as to obtain insights into how network structure relates to functionally relevant characteristics in a variety of settings. These include finding fast estimators for the solution of hard graph problems (we use Travelling Salesman as an example) and detecting structural features of metabolic pathway networks that correlate with biological evolution. With the sort of large-scale data-

driven approach we employ, caution is certainly necessary in interpreting the results, as we are dealing with simplified, abstracted representations of real-world systems where significant detail has been discarded. Thus the results and hypotheses thrown up should not be seen as definitive in any sense, but rather as pointers to guide more detailed investigation; pointers which would have been hard to obtain via conventional small-scale studies (we discuss this further in subsequent chapters).

Chapters 4 and 5 build on the approach of Chapter 3. They are motivated primarily by the broad questions: how can one uncover the structural peculiarities or constraints that characterise particular types of networks, and how do these arise? Chapter 4 examines how correlations between different network features arise within specific network families and what these features might indicate about structural constraints that those networks have to satisfy. This is related to the notion of network entropy, and we explore the connection between the two. The key idea is that constraints can be seen as entropy-lowering rules, in the sense that a constraint on what structures a particular type of network can have reduces the amount of uncertainty associated with possible observations of that type, because it reduces the size of the statistical ensemble from which those observations are drawn. In practice, we would like to be able to use structural constraints to make inferences about the generative mechanisms responsible for a particular class of networks. In Chapter 5, we examine this by developing a statistical procedure (making use of our feature vector representations) that attempts to match mechanisms to ensembles of networks. We focus in particular on the evolution of protein interaction networks, showing how our methodology can provide pointers to possible evolutionary models and also to differences between species.

Finally, Chapter 6 summarises the outcomes of this research and identifies directions for improvements and future work.

Chapter 2

Roles in Protein Interaction Networks

The bulk of the work presented in this chapter has been published in reference [15].

2.1 Background and motivation

A key challenge in biology is to understand how complex functionality emerges from systems composed of many relatively simple interacting components. At the cellular level, proteins are the major building blocks and functional units of life. Protein interaction networks have been widely studied in recent years (see Section 1.2.3); one prominent observation has been that hubs, nodes with high degree (see Section 1.1.2.1), appear to play key roles in network organisation, though the precise nature of these roles has been subject to some controversy. Jeong *et al.* suggested that hubs tended to be essential nodes, proteins whose removal would be lethal to the organism [136]. Han *et al.* proposed that hubs actually fall into two classes, date and party, with only the latter being critical for global network connectivity; this claim has proved controversial [33,34,41,121,256]. It has also been suggested that there may exist hubs with different numbers of binding interfaces: single- versus multi-interface

hubs, with this distinction mirroring the date/party one, and putatively being relevant to understanding the roles of protein-protein interactions in cancer [141, 150]. There has also been discussion and debate about the nature of the relationship between the topological properties of hubs and their functional roles [116], the reason why hubs are essential [286], and what defines a hub in the first place [266].

In particular, we focus on the debate over date and party hubs. Two points of contention are: (1) Is the distribution of hubs truly bimodal (as opposed to exhibiting a continual variation without clear-cut groupings) and (2) is the date/party distinction that was originally observed a general property of the interactome or an artefact of the data set employed and choices in analysis? Different statistical tests have suggested seemingly different answers. However, despite (or in some cases due to) this ongoing debate, the hypothesis has been highly prominent in the literature [116, 141, 150, 153, 187, 256, 266, 277, 280, 281].

Here (and in Ref. [15]), following up on the work of Batada *et al.* [33, 34], we revisit the initial data and suggest additional problems with the statistical methodology that was employed in originally proposing date and party hubs [121]. In accordance with the results of Refs. [33, 34], we find in Section 2.3 that the differing behaviour observed on the deletion of date and party hubs, which seemed to suggest that date hubs were more essential to global connectivity, was largely due to a very small number of key hubs rather than being a generic property of the entire set of date hubs. More generally, we use a complementary perspective to Batada *et al.* to define structural roles for hubs in the context of the modular organisation of protein interaction networks. Our results indicate that there is little correspondence between average expression correlation with partners (as measured by avPCC, defined in Section 1.2.3) and structural roles. In light of this, the more refined categorisation of date, party, and family hubs [153] (see Section 1.2.3) also appears inappropriate.

A recent study by Taylor *et al.* [256] argued for the existence of intermodular and

intramodular hubs—a categorisation essentially identical to date and party hubs—in the human interactome. We show that their observation of a binary hub classification is susceptible to changes in the algorithm used to normalise microarray expression data or in the procedure used to smooth the histogram of the avPCC distribution. The data does not in fact display any statistically significant deviation from unimodality as per the DIP test [122, 123], as has already been observed by Batada *et al.* [33, 34] for yeast data. We seek to revisit the bimodality question because it was a key part of the original paper [121], and in particular because it made a reappearance in Taylor *et al.* [256] for human data. However, it is possible that a continuum between date and party hubs (i.e., a relationship between hub expression correlation with partners and the hub's topological role) might exist even in the absence of a bimodal distribution, and this is why we will also attempt to examine the more general question of whether the network roles of hub proteins really are related to their co-expression properties with interaction partners.

Several studies in recent years have considered the existence of community structure in protein-protein interaction networks (see Section 1.2.3) [13, 58, 74, 101, 165, 171, 222, 248, 278]. In Section 2.4, we will use the idea of community structure to take a new approach to the problem of hub classification by attempting to assign roles to hubs purely on the basis of network topology rather than on the basis of expression data. Our rationale is that the biological roles of date and party hubs, as hypothesised by Han *et al.* [121], are essentially topological in nature and should thus be identifiable from the network alone (rather than having to be inferred from additional information). Once we have partitioned the network into a set of meaningful communities, it is possible to compute statistics to measure the connectivity of each hub both within its own community and to other communities. One method for assigning relevant roles to nodes is the "functional cartography" of Guimerà and Amaral [114] (see Section 1.1.4.8), and here we follow an analogous procedure for hubs in protein

interaction networks.

One might also wonder about the extent to which observed interactome properties are dependent on the particular instantiation of the network being analysed. In a recent paper, Yu *et al.* [280] examined the properties of interaction networks that were derived from different sources such as Y2H and AP/MS (see Section 1.2.2.1), suggesting that experimental bias might play a key role in determining which properties are observed in a given data set. In particular, their findings suggest that Y2H tends to detect key interactions between protein complexes—so Y2H data sets might contain a high proportion of date hubs (i.e., hubs with low partner co-expression)—whereas AP/MS tends to detect interactions within complexes, so hubs in AP/MS-derived networks are predominantly highly co-expressed with their partners (i.e., these networks will contain party hubs). This indicates that a possible reason for observing the bimodal hub avPCC distribution [121] is that the interaction data sets used information that was combined from both of these sources. In Section 2.5, we compare several yeast interaction data sets and find both widely differing structural properties and a low level of overlap.

Finally, as an alternative to the node-based date/party categorisation, we suggest thinking about topological roles in networks by defining measures on links rather than on nodes. In other words, one can attempt to categorise interactions between proteins rather than the proteins themselves. In Section 2.6, we use geodesic betweenness centrality as a measure of link significance and examine its relation to phenomena such as protein co-expression and functional overlap.

To summarise, in this chapter, we examine the proposed division of hubs in a protein interaction network into date and party categories from several different angles, demonstrating that prior arguments in favour of a date/party dichotomy appear to be susceptible to various kinds of changes in data and methods. Observed differences in network vulnerability to attacks on the two hub types seem to arise from only a

small number of particularly important hubs. These results strengthen the existing evidence against the existence of date and party hubs. Furthermore, a detailed investigation of network topology, employing the perspective of community structure and the roles of hubs within this context, suggests that the picture is more complicated than a simple dichotomy. Proteins in the interactome exhibit a variety of topological characteristics that appear to lie along a continuum—and there does not exist a clear correlation between their location on this continuum and the avPCC of expression of their interaction partners. However, investigating link (interaction) betweenness centralities reveals an interesting relation to the functional linkage of proteins, suggesting that a framework incorporating a more nuanced notion of roles for both nodes and links might provide a better framework for understanding the organisation of the interactome.

2.2 Materials and methods

2.2.1 Protein interaction data sets

Given the factors discussed in Section 1.2.2.1, choosing which data sets to use for building and investigating a network is itself a significant issue (see also the discussion in Section 2.5). For our investigation, we chose to work predominantly with networks consisting of multiply-verified interactions, which are constructed from evidence attained using at least two distinct sources. Such data sets are unlikely to contain many false positives, but might include many false negatives (i.e., missing interactions). Table 2.1 summarises the data sets that we employed, and additional details about how they were compiled are provided below:

- Online Predicted Human Interaction Database (OPHID): This data was sent to us by Ian Taylor; it is an updated version of the interaction data used in Ref. [256]. It is based on their curation of the online OPHID repository [56];

Table 2.1: **Protein interaction data sets.**

Data set name	**Species**	**Nodes**		**Links**		**Source**
		Total	LCC	Total	LCC	
Online Predicted Human Interaction Database (OPHID)	*H. sapiens*	8,199	7,984	37,968	37,900	Brown & Jurisica [56] (curated by Taylor *et al.* [256])
Filtered yeast interactome (FYI)	*S. cerevisiae*	1,379	778	2,493	1,798	Han *et al.* [121]
Filtered high-confidence (FHC)	*S. cerevisiae*	2,559	2,233	5,991	5,750	Bertin *et al.* [41]
Database of Interacting Proteins core (DIPc)	*S. cerevisiae*	2,808	2,587	6,212	6,094	DIP website [6] (October 2007 version)
Center for Cancer Systems Biology Human Interactome version 1 (CCSB-HI1)	*H. sapiens*	1,549	1,307	2,611	2,483	Rual *et al.* [229]
Protein-fragment complementation assay (PCA)	*S. cerevisiae*	1,124	889	2,770	2,407	Tarassov *et al.* [254]

Protein interaction data sets used in this chapter. LCC refers to the largest connected component.

they have mapped proteins to their corresponding NCBI (National Center for Biotechnology Information) gene IDs. Additionally, we removed genes that did not have expression data in GeneAtlas [253] (comparable avPCC values cannot be calculated for these, as GeneAtlas is the only expression data set used by Taylor *et al.* [256]), leaving a network with 8,199 human gene IDs and 37,968 interactions between them.

- Filtered Yeast Interactome (FYI): Compiled by Han *et al.* [121]. This was created from the intersecting data generated by several methods, including Y2H, AP/MS, literature curation, in silico predictions, and the MIPS (`http://mips.gsf.de/`) physical interactions list. It contains 1,379 proteins and 2,493 interactions that were observed by at least two different methods.

- Filtered High-Confidence (FHC): This data set was generated by Bertin *et al.* [41]. They filtered the high-confidence (HC) data set compiled by Batada *et al.* [34]. HC consists of 9,258 interactions between 2,998 proteins, taken from (published) literature-curated and high-throughput data sets, and they were supposed to be multi-validated. However, Bertin *et al.* [41] claimed that many interactions in HC had in fact been derived from a single experiment that was re-

ported in multiple publications. To conduct the filtration, Bertin *et al.* applied criteria similar to those used for FYI and obtained 5,991 independently-verified interactions between 2,559 proteins.

- Database of Interacting Proteins core (DIPc): We obtained this data set from the DIP website [6]. DIP is a large database of protein interactions compiled from several sources. The 'core' subset of DIP consists of only the "most reliable" interactions, as judged manually by expert curators and also automatically using computational approaches [71]. We used the version dated 7 October 2007 that contains 2,808 proteins and 6,212 interactions.

- Protein-fragment Complementation Assay (PCA): This experimental technique was used by Tarassov *et al.* [254] to obtain an *in vivo* map of the yeast interactome that consists of 1,124 proteins and 2,770 interactions. An attractive feature of this data set is that it measures interactions between proteins in their natural cellular context, in contrast to other prominent methods, such as Y2H (which requires transportation to the cell nucleus) and AP/MS (which requires multiple rounds of *in vitro* purification).

- Center for Cancer Systems Biology Human Interactome version 1 (CCSB-HI1): This data set was constructed by Rual *et al.* [229] using a high-throughput Y2H system, which they employed to test pairwise interactions amongst the products of about 8100 human open reading frames. The data set, which contains 2611 interactions amongst 1549 proteins, achieved a verification rate of 78% in an independent co-affinity purification assay (that is, from a representative sample of interactions in the data set, 78% could be detected in the independent experiment).

2.2.2 Functional homogeneity of communities

To assess how well the obtained topological communities reflect functional organisation, we used annotations from the Gene Ontology (GO) database [25] (see Section 1.2.2.3) to define their *Information Content* (IC). For each community, we computed the p-value of the most *enriched* GO annotation term, i.e., the term with the highest frequency within that community relative to its background frequency in the entire network. To do this, we used the hypergeometric distribution, which corresponds to random sampling without replacement. The extent of enrichment can then be gauged using IC [221], which is the negative logarithm of the p-value; higher IC reflects greater enrichment:

$$IC = -\log_{10}(p)\,, \tag{2.1}$$

where p denotes the p-value.

2.2.3 Jaccard distance

If one has two partitions of a given set of nodes, and a node i is part of subset (or community) C_i^1 of nodes in one partition and part of subset C_i^2 in the other partition, then the Jaccard distance [134] for node i across the two partitions is defined as

$$J(i) = 1 - |C_i^1 \cap C_i^2| / |C_i^1 \cup C_i^2|\,. \tag{2.2}$$

The symbols $\cap$ and $\cup$ correspond, respectively, to set intersection and union, and $|C|$ denotes the number of elements in set C. A Jaccard distance of 0 corresponds to identical partitions, whereas the distance approaches 1 for very different ones. By averaging $J(i)$ over all nodes in the set, we can get an estimate of the similarity of the two partitions. Here we will use this measure to compare node partitions obtained via community detection on networks that contain the same proteins as nodes but

not necessarily the same links between them.

2.2.4 Functional similarity

In order to compute the functional similarity of two interacting proteins, we first define the set information content (SIC) [221] of each term in our ontology for a given data set. Suppose the complete set of proteins is denoted by S, and the subset annotated by term i is denoted by S_i. The SIC of the term i is then defined as

$$\mathrm{SIC}(i) = -\log_{10}\left(\frac{|S_i|}{|S|}\right). \tag{2.3}$$

Now suppose that we have two interacting proteins called P and Q. Let S_P and S_Q, respectively, denote their complete sets of annotations (consisting of not only their leaf terms but also all of their ancestors) from the ontology. The functional similarity of the proteins is then given by

$$f(P,Q) = \frac{\sum_{i\in(S_P\cap S_Q)} \mathrm{SIC}(i)}{\sum_{i\in(S_P\cup S_Q)} \mathrm{SIC}(i)}. \tag{2.4}$$

2.3 Revisiting date and party hubs

The notions of date and party hubs (see Section 1.2.3) are based on the expression correlations of hubs with their interactors in a protein interaction network. Specifically, Han *et al.* [121] computed the avPCC for each hub (defined by them as a protein with at least 5 interactions), and observed that the avPCC distribution was bimodal in some cases. A date/party threshold value of avPCC (for a given expression data set) was defined to separate the two types of hubs; for bimodal distributions, this was the estimated value at the minimum of the distribution between the two modes [121].

We have re-examined the data sets and computations that were used to propose

the existence and dichotomy of date versus party hubs. In the original studies on yeast data [41, 121], any hub that exhibited a sufficiently high avPCC (i.e., any hub lying above the date/party threshold) on *any one* expression data set was identified as a party hub. Batada *et al.* [33] noted that this definition causes the date/party assignment to be overly conservative, in that a hub's status is unlikely to change as a result of additional expression data. In fact, some of the original expression data sets were quite small, containing fewer than 10 data points per gene. This suggests that classification of proteins as 'party' hubs was based on high co-expression with partners for just a small number of conditions in a single microarray experiment, even though such co-expression need not have been observed in other conditions and experiments. For instance, Han *et al.* found 108 party hubs in their initial study [121]. However, calculating avPCC across their entire expression compendium (rather than separately for the five constituent microarray data sets) and using the date/party threshold specified by the authors for this compendium avPCC distribution yields just 59 party hubs. Using only the "stress response" data set [100], which comprises over half of the data points in their compendium and is substantially larger than the other 4 sets, yields 74 party hubs. Thus, the results of applying this method to categorise hubs depend heavily on the expression data sets that one employs and is vulnerable to variability in smaller microarray experiments.

Recent support for the idea of date and party hubs appeared in a paper by Taylor *et al.* [256] that considered data relating to the human interactome; the authors found multimodal distributions of avPCC values, seemingly supporting a binary hub classification. We used an interaction data set provided by them (an updated version of the one used in their paper, sourced from the Online Predicted Human Interaction Database (OPHID) [56]), and found that the form of the distribution of hub avPCC that they observed is not robust to methodological changes. For instance, raw intensity data from microarray probes has to be processed and normalised in

order to obtain comparable expression values for each gene (see Section 1.2.2.2). The expression data used by Taylor *et al.* [256] (taken from the human GeneAtlas [253]) was normalised using the MAS5 algorithm [129]; when we repeated the analysis using the same data normalised by the GCRMA algorithm [276] instead of by MAS5, we obtained significantly different results.[1] Additionally, in order to smooth the discrete sequence of avPCC values into a continuous distribution, a Gaussian smoothing kernel is generally used. This in effect blurs each data point into a Gaussian spread; all points are summed to give the aggregate probability density. The width of the Gaussian used is a parameter that determines the amount of smoothing; we see that the observed bimodal nature of the avPCC distributions is very sensitive to the value of this parameter.

Figure 2.1 depicts the avPCC distributions for hubs (defined as the top 15% of nodes by degree [256], corresponding in this case to degree 15 or greater) in the two cases. We obtained probability density plots for varying smoothing kernel widths. The GCRMA-processed data does not appear to lead to a substantially bimodal distribution at any kernel width, whereas the MAS5-processed data appears to give bimodality for only a relatively narrow range of widths and could just as easily be regarded as trimodal. We used Hartigan's DIP test [5, 122, 123] to check whether either of the two versions of the expression data gives a distribution of avPCC values that exhibit significant evidence of bimodality; the results suggest that the apparent bimodal or trimodal nature of some of the curves in Figure 2.1 is illusory and not statistically robust. The DIP value is a measure of how far an observed distribution deviates from the best-fit unimodal distribution, with a value of 0 corresponding to no deviation. We used a bootstrap sample of 10,000 to obtain p-values for the DIP

[1] It has been noted that GCRMA tends to produce higher correlations than MAS5 between gene expression profiles obtained via replicated experiments; however, GCRMA also leads to significant correlations between random expression profiles, indicating that it produces some spurious correlation [167]. It was thus suggested that MAS5 is the more reliable method for inferring interactions between genes; but GCRMA has been argued to be better at detecting differentially expressed genes [276].

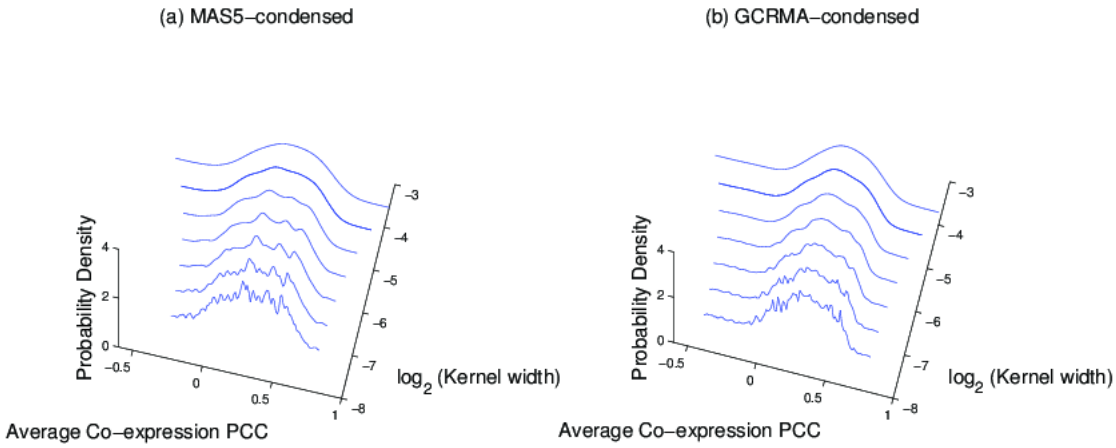


Figure 2.1: **Variation in hub avPCC distribution.**

Probability density plots of the distribution of hub avPCC values for human interaction data from OPHID (provided by Taylor *et al.* [256]). Gene expression data from GeneAtlas [253], normalised using (a) MAS5 and (b) GCRMA [167]. We obtain the curves using a normal smoothing kernel function at varying window widths. Hartigan's DIP test for unimodality [122, 123] returns values of 0.0087 (p-value ≈ 0.821) for (a) and 0.0062 (p-value ≈ 0.998) for (b), indicating that there is no significant deviation from unimodality in either case.

statistic. We found no significant deviation from unimodality: for MAS5, the DIP value is 0.0087 (p-value ≈ 0.821), and for GCRMA the DIP value is 0.0062 (p-value ≈ 0.998).

We also find variability across different interaction data sets. For instance, we analysed the recent protein-fragment complementation assay (PCA) data set [254] and found no clear evidence of a bimodal distribution of hubs along date/party lines. Even in cases in which multimodality is observed, it might have arisen as a consequence, or artefact, of combining different types of interaction data; there are believed to be significant and systematic biases in what types of interactions each data-gathering method is able to obtain [165, 254, 280] (See Section 1.2.2.1). For instance, analysing avPCC values on the stress-response expression data set [100] for hubs in networks obtained from Y2H or AP/MS alone [280], we find that 100% (259/259) are date

hubs in the former but that only about 30% (56/186) are date hubs in the latter. At the moment, it is reasonable to entertain the possibility that new kinds of interaction tests might smear the observed bimodality; this appears to be the case with the PCA data set.

One of the key pieces of evidence used to argue that date and party hubs have distinct topological properties was the apparent observation of different effects when they are deleted from a network [41, 121]. Removing date hubs seemed to lead to very rapid disintegration into multiple components, whereas removal of party hubs had much less effect on global connectivity. However, it has been observed that removing just the top 2% of hubs by degree from the comparison of deletion effects obviates this difference, suggesting that the observation is actually due to just a few extreme date hubs [33]. To study this in greater detail and to isolate the extreme hubs, we used geodesic node betweenness centrality (defined in Section 1.1.4.2), a standard diagnostic of a node's importance to network connectivity (which need not be strongly correlated with degree; the correlation coefficient between the two is just 0.28 for the FYI network). We found that in the original FYI data set [121], date hubs have on average somewhat higher node betweenness centralities (1.79×10^4 for 91 date hubs versus 1.07×10^4 for 108 party hubs; a two-sample t-test gives p-value ≈ 0.08). However, there is one date hub (SPC24/UniProtKB:Q04477, a highly connected protein involved in chromosome segregation [259]) that has an exceptionally high node betweenness (2.45×10^5) in this network. When the set of date hubs except for this one hub is targeted for deletion, we find that the observed difference between date and party hubs is greatly reduced [Figure 2.2(a)].

It was subsequently shown that the FYI network was particularly incomplete; as more data became available, the updated FHC data set was similarly analysed [41] (we also looked at the Y2H-only and AP/MS-only networks [280]; see Figure 2.3). In the case of FHC, the network did not break down on removing date hubs but

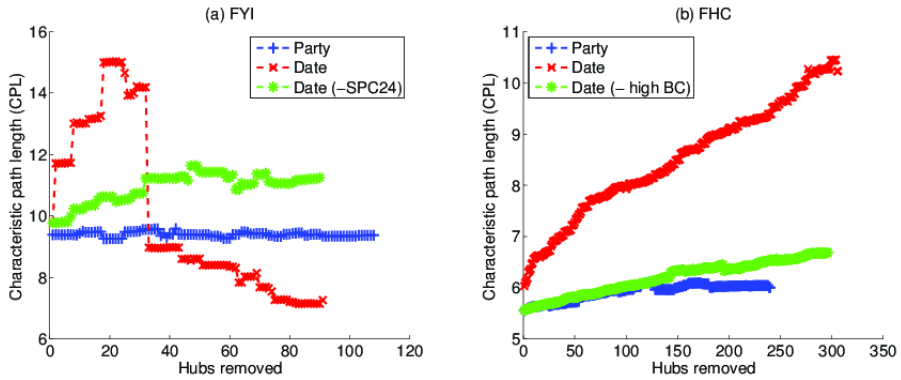


Figure 2.2: **Effects of hub deletion on network connectivity.**

(a) FYI network [121]. 'Date (− SPC24)' refers to the set of date hubs except for the protein SPC24. In each case, we used the complete network consisting of 1379 nodes as the starting point and then deleted all hubs in the given set from the network in order of decreasing degree. The characteristic path length is the mean of the lengths of all finite-length paths between two nodes in the network. (b) FHC network [41]. 'Date (− high BC)' refers to the set of date hubs except for the 10 hubs with the highest node betweenness centrality (BC) values (listed in Table 2.2). We used the upper bound on the BC for party hubs as a threshold to define these 10 'high BC' date hubs. (Note: Results similar to those presented here are obtained if the hubs are divided into bottleneck/non-bottleneck categories [281] instead of date/party categories.)

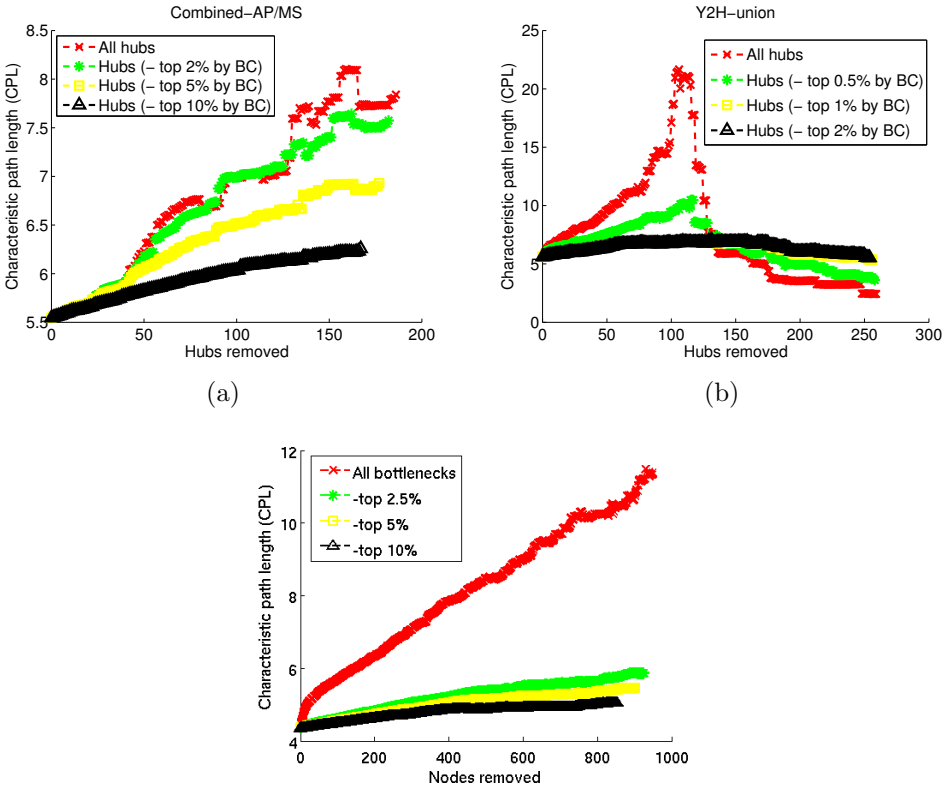


Figure 2.3: **Hub deletion effects for AP/MS-only, Y2H-only, and bottlenecks data sets.**

Change in characteristic path length (CPL, the mean length of all finite pairwise shortest paths) on removal of hubs in decreasing order of degree from the (a) 'Combined-AP/MS' and (b) 'Y2H-union' data sets [280], and (c) the 'bottlenecks' data set [281]. The 'top X%' labels refer to deletion of all hubs except the X% with the highest geodesic node betweenness centrality values. Note that when deleting the full sets of hubs, the Y2H network exhibits a much more dramatic increase in CPL, which might suggest that date hubs are more crucial to network connectivity than party hubs (the Y2H hubs are predominantly date hubs, whereas the AP/MS hubs are mostly party hubs [280]). However, only a very tiny fraction of Y2H-union hubs seem to be responsible for the huge CPL increase on deletion, and protecting these few high-betweenness hubs greatly reduces the impact of hub deletion on network connectivity. Similarly, only about 0.5% of the bottlenecks are responsible for the vast majority of the CPL increase in that case. These results show that the vast majority of so-called 'date hubs' are on average no more critical to global connectivity than party hubs.

nevertheless displayed a substantially greater increase in characteristic path length (see Section 1.1.4.3) than seen for party hub deletion, suggesting that deletion of the date hubs has a larger impact on network connectivity. For FHC too, date hubs have, on average, higher betweenness values than party hubs (3.7×10^4 for 306 date hubs versus 2.15×10^4 for 240 party hubs, p-value ≈ 0.06). However, the larger average is due almost entirely to a small number of hubs with unusually high betweennesses, as removing the top 10 date hubs by betweenness (which all had values higher than any party hub) greatly reduced the difference between the distributions (p-value ≈ 0.29). Furthermore, the removal of just these 10 hubs from the set of targeted date hubs is sufficient to virtually obviate the difference with party hubs, as shown in Figure 2.2(b). Notably, the set of 10 high-betweenness hubs includes prominent proteins such as Actin (ACT1/UniProtKB:P60010), Calmodulin (CMD1/UniProtKB:P06787), and the TATA binding protein (SPT15/UniProtKB:P13393), which are known to be key to important cellular processes (see Table 2.2).

Thus, we can account for the critical nodes for network connectivity using just a few major hubs, and most of the proteins that are classified as date hubs appear to be no more important in this respect than the party hubs. High betweenness centrality nodes have previously been called *bottlenecks* [281], and it has been suggested that these tend to correspond to date hubs. However, the same sort of analysis on the bottlenecks data set [281] once again reveals that only the top 0.5% or so of nodes by betweenness are truly critical for connectivity [see Figure 2.3(c)]. Additionally, the 10 key hubs in the FHC network exhibit a wide range of avPCC values (see Table 2.2): high betweenness does not necessitate low avPCC. Similarly, we found that there is not a strong correspondence between bottleneck/non-bottleneck and date/party distinctions across multiple data sets. These observations further weaken the claim that there is an inverse relation between a hub's avPCC and its importance in connecting different parts of a network.

Table 2.2: **High-betweenness hubs in the FHC network.**

Protein	UniProtKB	Degree	AvPCC	BC($/10^5$)	Functions
CDC28	P00546	202	0.06	19.99	Essential for the completion of the start, the controlling event, in the cell cycle
RPO21	P04050	58	0.05	3.56	Catalyses the transcription of DNA into RNA
SMT3	Q12306	42	0.08	3.07	Not known; suppressor of MIF2 (UniProtKB:P35201) mutations
ACT1	P60010	35	0.13	2.83	Cell motility
HSP82	P02829	37	0.19	2.51	Maturation, maintenance, and regulation of proteins involved in cell-cycle control and signal transduction
SPT15	P13393	50	0.12	2.45	Regulation of gene expression by RNA polymerase II
CMD1	P06787	46	0.05	2.11	Mediates the control of a large number of enzymes and other proteins
PAB1	P04147	25	0.28	1.92	Important mediator of the roles of the poly(A) tail in mRNA biogenesis, stability, and translation
PSE1	P32337	24	0.28	1.73	Nuclear import of ribosomal proteins and protein secretion
GLC7	P32598	35	−0.01	1.55	Glycogen metabolism, meiosis, translation, chromosome segregation, cell polarity, and cell cycle progression

List of the 10 hubs with highest node betweenness in the FHC network [41], with UniProtKB accessions [259], degrees, avPCC values (as computed using the 'Compendium' expression data set [121, 142]), betweenness centrality (BC) values, and selected functional annotations from UniProtKB.

2.4 Topological properties and node roles

If the hypothesised categorisation of hubs into a date/party dichotomy is correct, then one should be able to observe this directly in the network structure, as the two kinds of hubs were inferred to have different neighbourhood topologies. We thus leave gene expression data to one side for the moment and focus on what can be inferred about node roles purely from network topology. Guimerà and Amaral [114] have introduced a scheme for classifying nodes into topological roles in a modular network according to the pattern of intramodule and intermodule connections. To study our networks in this fashion, we first seek to partition them into modules or communities. We optimise the standard Newman-Girvan modularity function (see Section 1.1.3.1) using recursive spectral bisection [193] to obtain the communities used for the results in Figure 2.5. Maximising graph modularity [Equation (1.1)] is expected to give a partition in which the density of links within each community is significantly higher than the density of links between communities. In Figure 2.4,

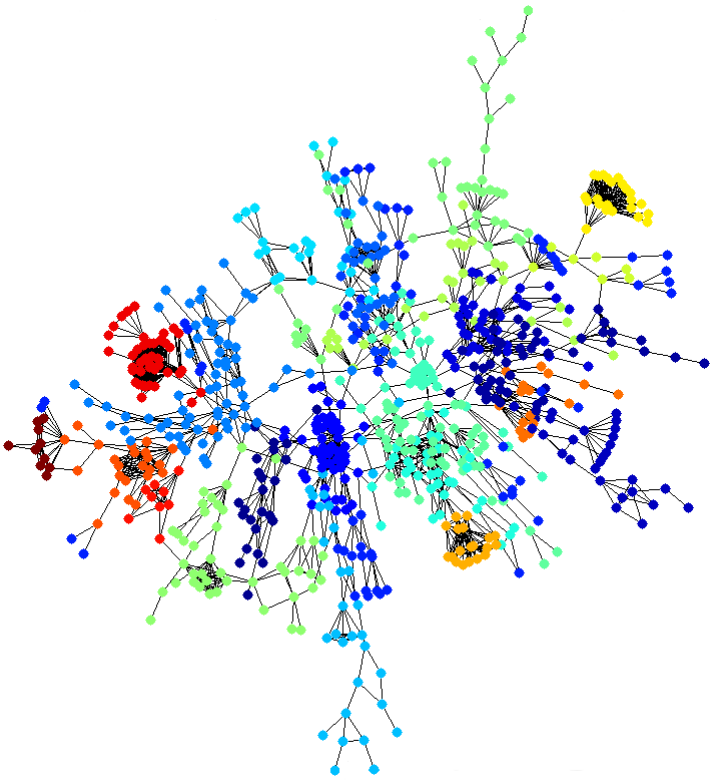


Figure 2.4: **Community structure in the largest connected component of the FYI network.**

Data from Han *et al.* [121]. The different colours correspond to different communities (25 in all). The graph modularity value for this partition is 0.8784. We generated this visualisation using the Kamada-Kawai algorithm [139] (MATLAB code obtained from Amanda Traud [262]).

we show the network partition (with nodes coloured according to community) that results from applying such an optimisation to the largest connected component of the filtered yeast interactome (FYI) data set [121].

We wish to assess how well the network communities obtained in this way correspond to groupings of functionally similar proteins, as per Gene Ontology [25] annotations. We do this using the Information Content (IC) measure [Equation (2.1)]. In Table 2.3, we summarise the results of calculating this measure for communities detected (for resolution parameter value $\gamma = 1$) on two of the yeast interaction data

sets: FYI and the more recent filtered high-confidence (FHC [41]). Although the IC value itself is a measure of the unlikelihood of a given event occurring by chance, for additional comparison we also examine a uniformly random partition of FYI into communities with the same size distribution as the actual ones. It is clear that on average the detected communities are far more functionally homogeneous than could be expected by chance. This is in accordance with previous studies on communities in protein interaction networks [13, 58, 74, 165, 171]. It is also evident that IC varies widely over communities and that not all of them are equally enriched. There are some relatively heterogeneous communities (which are not aptly described by a single, specific GO term) and others that show a very high functional coherence. In particular, a more detailed inspection of the community composition reveals that proteins that are part of the large and small ribosomal subunit complexes are almost perfectly grouped together, and several other communities consist exclusively of proteins that are known to be part of a given complex.

Thus, the topology of the interaction network provides a great deal of information about the functional organisation of the proteome. Our particular partitioning is of course not unique; it is only a means to an end, as our aim is to examine the implications of community structure for individual protein roles, with particular reference to the notion of date and party hubs. We have also used the locally greedy algorithm described by Blondel *et al.* [48] (see Section 1.1.3.3) as an alternative method for optimising modularity, and this makes no difference to the salient observations presented below.

Having obtained meaningful network partitions, we can proceed to categorise nodes into roles, such as suggested by Guimerà and Amaral [114]. Their classification uses two diagnostics for each node—within-community degree and participation coefficient—and divides the plane that they define into regions encompassing seven possible roles (see Section 1.1.4.8). In Figures 2.5 and 2.6, we plot all nodes in the

Table 2.3: **Evaluating community partitions.**

Data set	Communities	MF *IC*			CC *IC*			BP *IC*			Best *IC*		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
FYI	25	2.05	43.09	14.36	4.28	51.60	17.18	2.99	35.74	15.72	4.81	51.60	20.15
FYI	25 (random)	1.28	2.78	1.88	1.25	3.00	2.07	1.46	3.04	2.13	1.46	3.04	2.36
FHC	63	1.47	51.37	11.22	0.11	68.18	16.40	1.73	98.51	17.08	1.97	98.51	20.08

Information Content (IC) of the most enriched term for each of the three GO ontologies (MF – Molecular Function; CC – Cellular Component; and BP – Biological Process) and over all three ontologies combined ('Best IC'). We give the minimum, maximum, and average IC over all of the communities (at the default resolution value $\gamma = 1$) that we detected in two data sets: FYI [121] and FHC [41]. We generated the random communities for FYI using the size distribution of the actual ones. In other words, we remove the actual community labels of all proteins and then randomly re-assign them (using one label per protein).

network in a two-dimensional space using coordinates determined by the two statistics, and we divide the space into regions that correspond to different node roles. The boundaries between regions are of course arbitrary, so for simplicity we have used the demarcations employed by Guimerà and Amaral [114]. We depict the 7 roles defined by them as demarcated regions in the plots in Figures 2.5 and 2.6.

Figure 2.5 shows the node roles for yeast (FHC [41]) and human (Center for Cancer Systems Biology Human Interactome version 1 (CCSB-HI1) [229]) data sets, based on communities obtained via modularity maximisation at default resolution. We also use the Potts model [Equation (1.2)] (Section 1.1.3.2) as an alternative way of partitioning the network; this allows one to adjust the resolution parameter γ to get more or fewer communities [220]. We present results for two alternative settings of the resolution parameter ($\gamma = 0.5$ and $\gamma = 2$) in Figure 2.6, indicating that whilst the number of communities changes substantially as we decrease or increase the resolution, the pattern of role assignments to the nodes remains similar to that shown in Figure 2.5 (using the default choice of $\gamma = 1$), and the conclusions below are valid across the multiple resolutions examined.

Some of the topological roles defined by this method appear to correspond to those ascribed to date/party hubs. For instance, one might argue that party hubs ought to be 'provincial hubs', which have many links within their community but few or

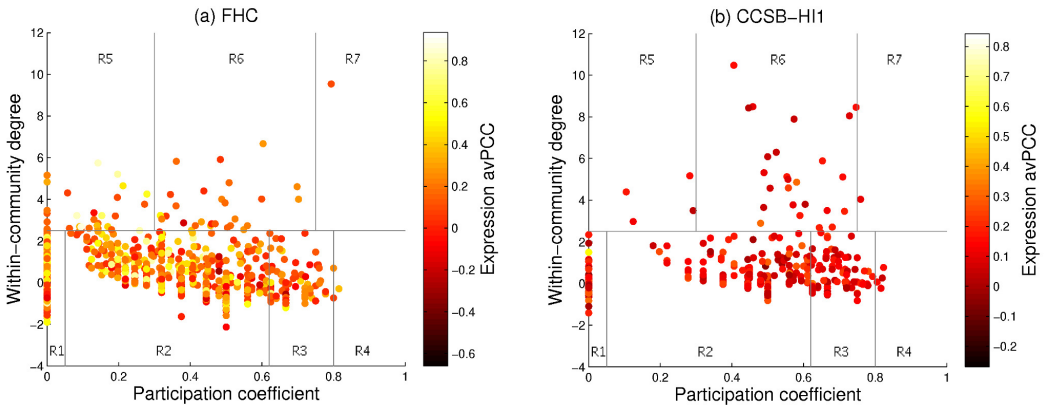
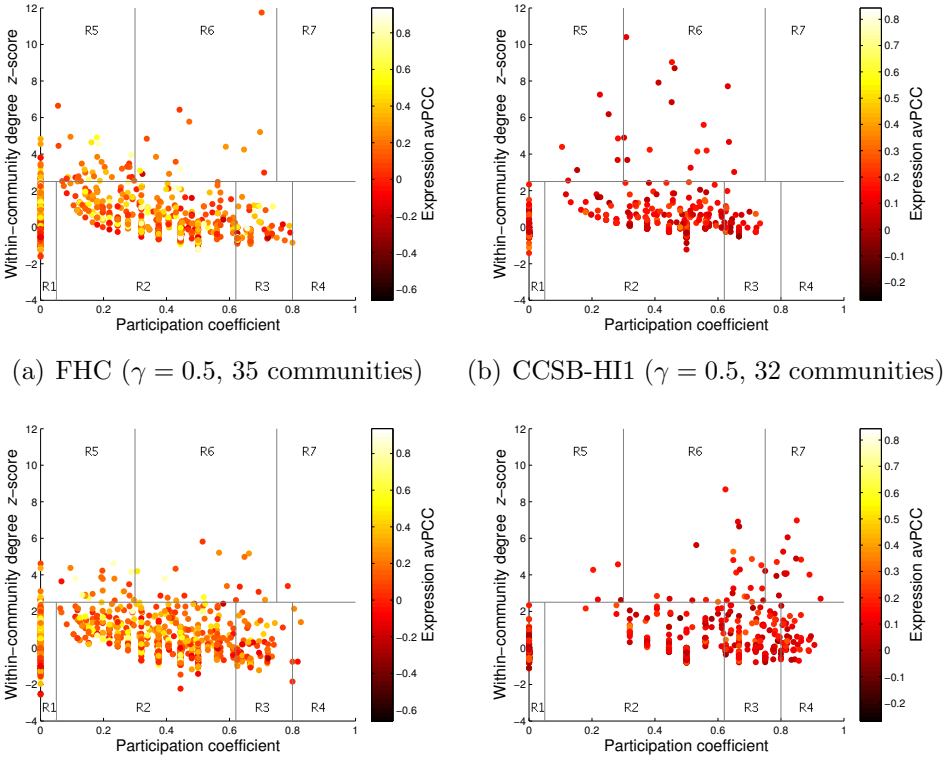


Figure 2.5: **Topological node role assignments and relation with avPCC.**

Plots for (a) yeast network (FHC [41]—2,233 nodes, 63 communities) and (b) human network (CCSB-HI1 [229]—1,307 nodes, 38 communities). Following Guimerà and Amaral [114], we designate the roles as follows: R1 – Ultra-peripheral; R2 – Peripheral; R3 – Non-hub connector; R4 – Non-hub kinless; R5 – Provincial hub; R6 – Connector hub; and R7 – Kinless hub. We colour proteins according to the avPCC of expression with their interaction partners. We computed expression avPCC using the stress response data set [100] (which was the largest, by a considerable margin, of the expression data sets used in the original study [121]) for FHC and COXPRESdb [200] for CCSB-HI1. No partner expression data was available for a few proteins (25 in FHC, 1 in CCSB-HI1), so these are not shown on the plots.



(a) FHC ($\gamma = 0.5$, 35 communities) (b) CCSB-HI1 ($\gamma = 0.5$, 32 communities)

(c) FHC ($\gamma = 2$, 81 communities) (d) CCSB-HI1 ($\gamma = 2$, 71 communities)

Figure 2.6: **Topological node role assignments and relation with avPCC.**

Plots for (a),(c) yeast network (FHC [41]—2,233 nodes) and (b),(d) human network (CCSB-HI1 [229]—1,307 nodes). As in Figure 2.5, but with different resolution parameter (γ) values used for community detection.

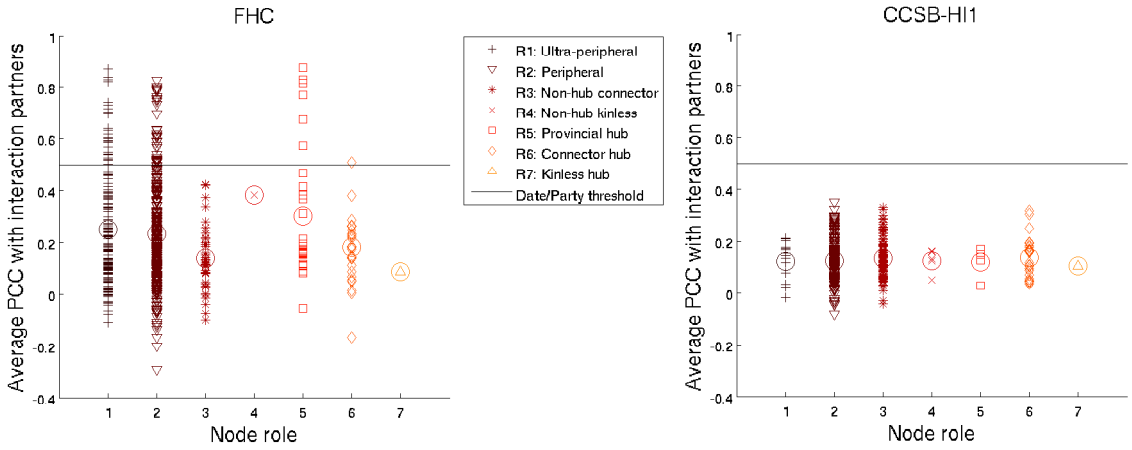


Figure 2.7: **Rolewise hub avPCC distributions.**

Plots show node role versus average expression correlation with partners for hubs in yeast (FHC [41]—553 hubs with a minimum degree of 7) and human (CCSB-HI1 [229]—326 hubs with a minimum degree of 4) networks. Larger circles represent means over all nodes in a given role. Note that 'hub' as used in the role names refers only to within-community hubs, but all of the depicted nodes are hubs in the sense that they have high degree. In each case, we determined the degree threshold so that approximately the top 20% highest-degree nodes are considered to be hubs. We also fixed the date/party avPCC threshold at 0.5, in accordance with Bertin *et al.* [41].

none outside. Date hubs might be construed as 'non-hub connectors' or 'connector hubs', both of which have links to several different modules; they could also fall into the 'kinless' roles (though very few nodes are actually classified as such). We thus sought to examine the relationship between the date/party classification and this topological role classification. In Figure 2.5, we colour proteins according to their avPCC. In Figure 2.7, we only show the hubs (defined as the top 20% of nodes ranked by degree [41]) in the two interaction networks and plot them according to node role and avPCC. The horizontal lines correspond to an avPCC of 0.5, which was the threshold used to distinguish date and party hubs in the yeast interactome [41].

One immediate observation from these results is that the avPCC threshold clearly does not carry over to the human data. In fact, all of the hubs in the latter have

an avPCC of well below 0.5. Even if we utilise a different threshold in the human network, we find that there is little difference in the avPCC distribution across the topological roles, suggesting that no meaningful date/party categorisation can be made (at least for this data set). This might be the case because the human data set represents only a small fraction of the actual interactome. Additionally, it is derived from only one technique (Y2H) and is thus not multiply-verified like the yeast data set.

For yeast, we see that hubs below the threshold line (i.e., the supposed date hubs) include not only virtually all of those that fall into the 'connector' roles but also many of the 'provincial hubs'. Those that lie above the line (i.e., the supposed party hubs) include mainly the provincial hub and peripheral categories. Although one can discern a difference in role distributions above and below the threshold, it is not clear-cut and the so-called date hubs fall into all 7 roles. It would thus appear that even for yeast, the distribution of hubs does not clearly fall into two types (the original statistical analysis was already disputed by Batada *et al.* [33,34]), and the properties attributed to date and party hubs [121] do not seem to correspond very well with the actual topological roles that we estimate. Indeed, these roles are more diverse than what can be explained using a simple dichotomy.

2.5 Data incompleteness and experimental limitations

It has been proposed that date and party hubs play different roles with respect to the modular structure of protein interaction data [121,256]. As there are diverse examples of such data, one might ask to what extent entities like date and party hubs can be consistently defined across them. It has been noted previously that many of the existing large-scale data sets show little overlap and are highly inconsistent [119, 270]. In

Table 2.4: **Comparisons of yeast data sets.**

Data sets (number of nodes)	**Common nodes**$^{(a)}$	**Links in overlap**$^{(b)}$	**Between-community Jaccard distance**$^{(c)}$	**Role**$^{(c)}$ **overlap**$^{(d)}$
FYI (778) vs. FHC (2233)	714	FYI–1444; FHC–2027; Both–1195	0.76	332 (47%)
FYI (778) vs. DIPc (2587)	660	FYI–1310; DIPc–1698; Both–956	0.77	265 (40%)
FHC (2233) vs. DIPc (2587)	1661	FHC–4395; DIPc–4141; Both–2665	0.85	854 (51%)
FYI (778) vs. PCA (889)	165	FYI–154; PCA–180; Both–65	0.74	109 (66%)
FHC (2233) vs. PCA (889)	460	FHC–512; PCA–667; Both–187	0.86	214 (47%)
DIPc (2587) vs. PCA (889)	492	DIPc–568; PCA–782; Both–183	0.86	206 (42%)

Pairwise comparisons of the largest connected components of different yeast protein interaction data sets. Notes: (*a*) **Proteins occurring in both networks.** (*b*) **Links amongst the common nodes as counted in the previous column: individually in either network and common to both networks.** (*c*) **Communities and node roles computed over entire data sets; for pairwise comparison, we then narrow down communities in each case to only those nodes also present in the data set to which the comparison is being made.** (*d*) **The number of nodes with the same role classification (as per Guimerà and Amaral [114]) in both networks and their percentage as a share of the entire set of common nodes.**

order to further investigate the extent of network overlap and in particular the preservation of the interactome's structural properties (such as community structure and node roles) for different data sets and data-gathering techniques, we compared statistics and results for four different yeast interaction data sets: FYI, FHC, Database of Interacting Proteins core (DIPc), and PCA (see Table 2.1 and Section 2.2.1 for details of these). Our motivation for these choices of data sets (aside from PCA) was that they all encompass multiply-verified or high-confidence interactions. We also used PCA data because it is from the first large-scale screen with a new technique that records interactions in their natural cellular environment [254]. For each data set, we counted the number of nodes and links in common using pairwise comparisons in the largest connected component of the network. For the overlapping portions, we then computed the extent of overlap in node roles and communities. For the latter, we employed the Jaccard distance [134], which ranges from 0 for identical partitions to 1 for entirely distinct ones (see Section 2.2.3). Whilst there exist a number of measures for comparing partitions [130], for our purposes it is sufficient to get some indication of whether communities in different networks have substantial overlap or not; thus we choose to apply just the simple Jaccard measure. In Table 2.4, we present the results of our binary comparisons of the yeast data sets.

Table 2.4 reveals that there are large variations amongst the different networks reported in the literature. FYI, FHC, and DIPc are all regarded as high-quality data sets, yet they contain numerous disparate interactions. PCA has a very low overlap with both FYI and DIPc (considered separately), suggesting that it provides data that is not captured by either Y2H or AP/MS screens. Such differences unsurprisingly lead to nodes belonging to highly varying communities across data sets. We compare the networks pairwise; for comparison purposes, communities are computed over the complete network in each case, and then we prune each community to retain only those nodes also present in the other network. The Jaccard distance for each pairwise comparison amongst the 4 networks is about 0.8, so on average the intersection of communities for the same node covers only about a fifth of their union. As we compute topological node roles relative to assignment of nodes to communities, it is not surprising that the role overlap is also not very high in any of the cases.

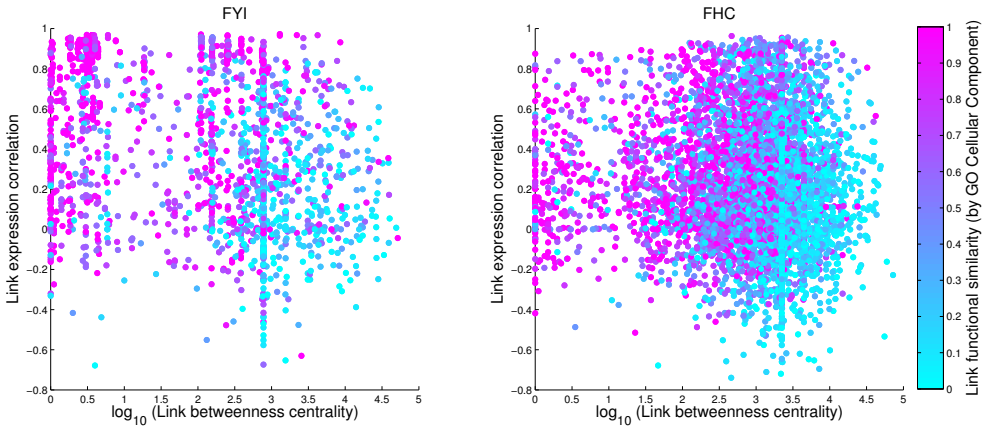
Given the above, it is difficult to make any general inferences regarding proteome organisation from results on existing protein interaction networks. They depend a great deal on the explored data set, which in each case represents only part of the total interactome and likely also contains substantial noise (as discussed in Sections 1.2.2.1 and 1.2.3).

2.6 The roles of interactions

Most research on interactome properties has focused on node-centric diagnostics, which draws on the perspective of individual proteins (e.g., [121, 136, 153, 286]). Here we try an alternative approach that instead uses link-centric diagnostics in order to examine how the topological properties of interactions in a network relate to their function. In order to quantify the importance of a given link to global network connectivity, we use geodesic link betweenness centrality [107] (see Section 1.1.4.2). We

investigate the relationship between link betweenness and the expression correlation for a given interaction. If date and party hubs genuinely exist, then one might expect a similar sort of dichotomy for interactions, with interactions that are more important for global network connectivity having lower expression correlations and vice versa. That is, given the hypothesised functional roles of date and party hubs, most intermodular interactions would connect to a date hub, whereas most intramodular interactions would connect to a party hub. In Figure 2.8, we depict all of the interactions in two yeast data sets (FYI and FHC), which we position on a plane based on the values of their link betweenness and interactor expression PCC (calculated using the stress response data set, as before). Additionally, we colour each point according to the level of functional similarity between the interacting proteins, as determined by overlap in the three types of GO annotations (see Section 2.2.4).

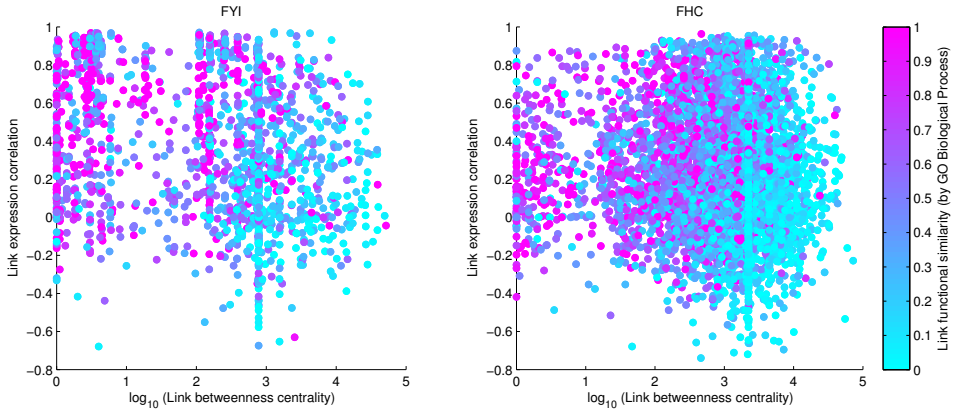
For the FHC data set, we find no substantial relation between expression PCC and the logarithm of link betweenness (linear Pearson correlation coefficient $\rho \approx -0.04$, z-score ≈ -3.1, and p-value ≈ 0.0022). For the FYI data set, there is a larger correlation ($\rho \approx -0.31$, z-score ≈ -13.6, and p-value $\approx 4.5 \times 10^{-42}$). Correspondingly, we observe a dense cluster of interactions in the top left (i.e., they have low betweennesses and high expression correlations), but most of these are interactions within ribosomal complexes. If one removes such interactions from the data set, then here too one finds only a small correlation ($\rho \approx -0.12$, z-score ≈ -4.5, and p-value $\approx 5.8 \times 10^{-6}$) between expression PCC and (log of) link betweenness. (Note that ribosomal proteins were already removed from FHC [41].) However, we find a fairly strong correlation between link betweenness (on a log-scale) and similarity in cellular component annotations (which can be used as a measure of co-localisation): the PCC values are $\rho \approx -0.51$ (z-score ≈ -23.9, p-value $\approx 1.4 \times 10^{-126}$) for FYI and $\rho \approx -0.46$ (z-score ≈ -37.2, p-value $\approx 1.6 \times 10^{-303}$) for FHC (we obtain very similar values for the Spearman rank correlation coefficient: $\rho \approx -0.52$ for FYI and



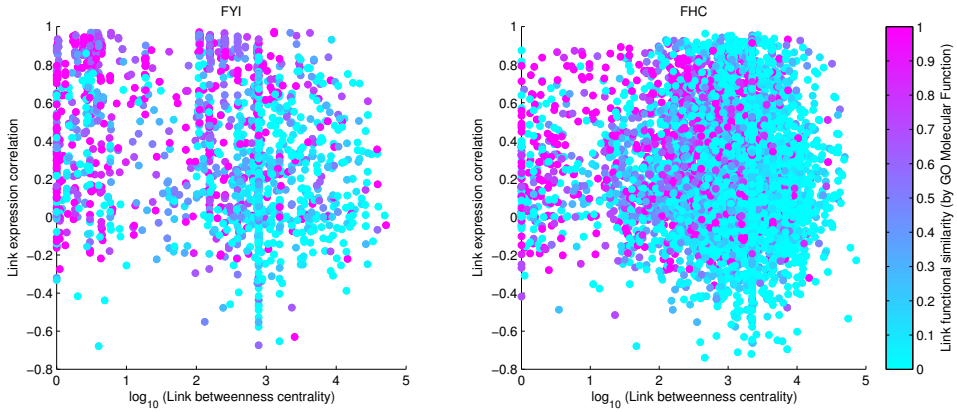
(a) Cellular Component

Figure 2.8: **Relating interaction betweenness, co-expression, and functional similarity.**

The plots show link betweenness centralities versus expression correlations, with points coloured according to mean similarity of interactors' GO (Cellular Component) annotations, for two protein interaction data sets: FYI [121] (778 nodes, 1,798 links) and FHC [41] (2,233 nodes, 5,750 links). The PCC values of log(link betweenness) with functional similarity are -0.51 (z-score ≈ -23.9, p-value $\approx 1.4 \times 10^{-126}$) for FYI and -0.46 (z-score ≈ -37.2, p-value $\approx 1.6 \times 10^{-303}$) for FHC.



(b) Biological Process



(c) Molecular Function

Figure 2.8 (continued): The plots show link betweenness centralities versus expression correlations, with points coloured according to mean similarity of interactors' (b) GO (Biological Process) and (c) GO (Molecular Function) annotations for two protein interaction data sets: FYI [121] (778 nodes, 1,798 links) and FHC [41] (2,233 nodes, 5,750 links). The PCC values of log(link betweenness) with functional similarity are (b) -0.41 (z-score ≈ -18.6, p-value $\approx 3.9 \times 10^{-77}$) for FYI and -0.42 (z-score ≈ -33.9, p-value $\approx 4.7 \times 10^{-252}$) for FHC; and (c) -0.39 (z-score ≈ -17.3, p-value $\approx 4.5 \times 10^{-67}$) for FYI and -0.31 (z-score ≈ -24.7, p-value $\approx 1.6 \times 10^{-134}$) for FHC.

$\rho \approx -0.47$ for FHC). In particular, there appears to be a natural threshold at the modal value of betweenness; this is a finite-size effect.[2] This is somewhat reminiscent of the weak/strong tie distinction in social networks [112, 216], as the 'weak' (high-betweenness) interactions serve to connect and transmit information between distinct cellular modules, which are composed predominantly of 'strong' (low-betweenness) interactions. For instance, we found that interactions involving kinases fall largely into the 'weak' category. Additionally, GO terms such as intracellular protein transport, GTP binding, and nucleotide binding were significantly overrepresented in proteins involved in high-betweenness interactions.

2.7 Discussion

In this chapter, we have analysed modular organisation and the roles of hubs in protein interaction networks. We revisited the proposed date/party hub dichotomy and found substantial areas of concern. In particular, claims of bimodality in hub avPCC distributions do not appear to be robust across available interaction and expression data sets, and tests for the differences observed on deletion of the two hub types have not considered important outlier effects. Moreover, there is considerable evidence to suggest that the observed date/party distinction is at least partly an artefact, or even a consequence, of the different properties of the Y2H and AP/MS data sets.

In order to study the topological properties of hub nodes in greater detail, we partitioned protein interaction networks into communities and examined the statistics

[2]For finite, relatively sparse, unweighted networks such as the ones we study, the distribution of link betweenness centrality is almost normal, with the exception of a large spike at a value well above the mean (see the long vertical bar of points in the plots in Figure 2.8). This results from the large number of nodes with degree 1. The link that connects such a node to the rest of the network must have a betweenness of $n-1$, where n is the total number of nodes in the network. This link must lie on the $n-1$ shortest paths that connect the degree-1 node to all of the other nodes, and it cannot lie on any other shortest paths. Thus, for our networks, the link betweenness centrality distribution shows a strong spike at a value of precisely $n-1$.

of the distributions of hub links. Our results show that hubs can exhibit an entire spectrum of structural roles and that, from this perspective, there is little evidence to suggest a definitive date/party classification. We find, moreover, that expression avPCC of a hub with its partners is not a strong predictor of its topological role, and that the extent of interacting protein co-expression varies considerably across the data sets that we examined.

Additionally, a key issue with existing interaction networks is that they are incomplete. We have compared some of the available 'high-quality' yeast data sets and shown that they overlap very little with each other. One can obtain protein interaction data using several different experimental techniques, and each method appears to preferentially pick up different types of interactions [165, 280]. The only published interactome map of which we are aware that examines proteins in their natural cellular environment [254] is largely disjoint with other data sets and shows little evidence of a date/party dichotomy. We find similar issues in human interaction data sets. A general conclusion about interactome properties is thus difficult to reach, as it would require robust results for several different species. This is unattainable at present due to the limited quantity and questionable quality of protein interaction and expression data.

As an alternative way of defining roles in the interactome, we have also investigated a link-centric approach, in which we study the topological properties of links (interactions) as opposed to nodes (proteins). In particular, we examined link betweenness centrality as an indicator of a link's importance to network connectivity. We found that this too does not correlate significantly with expression PCC of the interacting proteins. For certain data sets, however, it does appear to correlate fairly strongly with the functional similarity of the proteins. Additionally, there appears to be a threshold value of link betweenness centrality beyond which one observes a sudden drop in functional similarity. We also found that kinase bindings and other

kinds of interactions involved in signalling and transportation functions are significantly overrepresented in the high-betweenness interactions. This suggests that a notion of intramodular versus intermodular interactions, somewhat analogous to the weak/strong tie dichotomy in social networks, might be more useful. However, further work would be required to establish such a framework of elementary biological roles in protein interaction networks. As the quantity, quality, and diversity of protein interaction and expression data sets increases, we hope that this perspective will enhance understanding of the organisational principles of the interactome.

More broadly, the story of date and party hubs shows that simplistic attempts to relate structural properties of networks to their functionality can be very misleading, and that examining network structure from multiple perspectives can yield a wider range of insights. Motivated by this observation, we now seek a holistic approach by attempting to consolidate many different ways of diagnosing and characterising networks extant in the literature and seeking the data-driven discovery of patterns of relationships between structure and function. This is what we describe in the next chapter.

Chapter 3

High-Throughput Analysis of Networks

The main ideas presented in this chapter have appeared as extended abstracts in two workshop proceedings [18, 268], and the work is also included in a manuscript currently in preparation [17].

3.1 Motivation

Our study of date and party hubs demonstrated that it is important to take into account multiple measures of network structure in order to attempt to understand how structure correlates with functionality. We also saw that ideas from the study of social networks, such as betweenness [96] and weak and strong ties [112], can be insightful when analysing biological networks. Motivated by these observations, we sought to develop a more comprehensive methodology for studying networks, that might help consolidate different strands of the literature and highlight cross-disciplinary connections.

A large number of approaches for the study and analysis of networks have been developed across multiple disciplines (see Section 1.1); however, for a given task or

question on a network, it is often hard to determine appropriate methods of characterisation, as networks are intrinsically high-dimensional objects and there need not be any universally useful set of diagnostics. Studies focused on a particular network tend to employ a small subset of existing diagnostics, and choices are typically motivated by intuition and familiarity (often influenced by one's disciplinary background). Particularly when studying new, unfamiliar kinds of networks, we would like to examine them from as many different perspectives as possible to get a handle on how they relate to other networks we have already observed and studied. Additionally, when a new diagnostic meant to capture some aspect of network structure is introduced, it is generally compared and contextualised with only a few of the existing ones, which tend to be the ones prevalent in the authors' discipline(s). This can lead to reinvention of the wheel, as the relations between diagnostics originating from different academic communities are often left unexplored.

Here we seek to address these issues empirically by carrying out a large-scale collation and investigation of both different types of networks and different ways of characterising networks. Limited efforts of this sort have recently been made [88] and applied, for instance, to the comparison of metabolic networks from different species [228] and the evaluation of models for protein-protein interaction networks [155, 183]. However, there is still no "systematic program for characterising network structure" (a phrase from Newman [192]) that can be used to compare both networks themselves as well as network diagnostics. We attempt to move in this direction by setting up a large, diverse database of networks, along with a library of algorithms (drawn from a variety of disciplinary literatures) for computing different characteristics of these networks. The end result can be represented as a matrix whose rows correspond to networks and whose columns correspond to features—i.e., a design matrix (see Section 1.3.1). Each entry in the design matrix represents the value of one feature for one network—if we denote the matrix by D, then we have $D_{ij} = f_j(G_i)$, where G_i is

the i^{th} network and f_j denotes a function which computes the j^{th} feature. We refer to this approach as *high-throughput* analysis of networks, in analogy to high-throughput experimental methods in biology that involve simultaneously making a large number of measurements on a large number of objects [e.g., gene expression microarrays (see Section 1.2.2.2)]; here the design matrix is analogous to the microarray. Once this matrix has been computed, it becomes possible to use machine learning techniques (see Section 1.3) to discover interesting patterns and relationships in the data; indeed, it is difficult to deal with data on scale we seek to look at here (hundreds of network and hundreds of features) without the use of such statistical techniques.

In this and subsequent chapters, we explore how this high-throughput approach can be leveraged to aid the understanding of networks. We first describe the different network data sets and diagnostics that we collected (in Section 3.2). We then show how our feature-based representation of networks enables a data-driven comparison and organisation of both networks and network diagnostics (in Section 3.3). Subsequently, via two types of case studies, we demonstrate its utility for inferring connections between network structure and *functionality*, i.e., what processes are or could be carried out via that network, and/or how efficiently; the precise specification depends on the context and will be determined in our discussion of particular examples. Our methodology involves regressing functional properties of interest on network features: for instance, we show that certain features can serve as fast estimates for the solution and runtime of hard graph-theoretic computational problems, such as the Travelling Salesman Problem (in Section 3.4). In biological networks, we show how regression of evolutionary distances on network features allows us to detect phylogenetic signals, thereby suggesting which aspects of network structure correlate with biological evolution (in Section 3.5). Finally, we summarise our conclusions from this chapter and how they lead into the subsequent ones (in Section 3.6). We suggest that the approach described here allows one to unearth structure-function relation-

Table 3.1: **Sets of networks used.**

Network type	No. of networks	No. of features[(a)]	Source	Section
Varied; real (see Appendix B) and synthetic	192 real, 120 synthetic	338[(b)]/347[(c)]	Real: Onnela *et al.* [203]; Synthetic: From models (see Section 3.3.1)	3.3
Preferential Attachment Poisson (PAP)	500	438	From PAP model [218]	3.4
Community detection benchmark	250	436	Generated from Lancichinetti *et al.* model [159]	3.4
Metabolic networks of interacting pathways	620	222	Mazurie *et al.* [178]	3.5
Pseudomonas metabolic pathways	$17 \times 6^{(d)}$	804[(d)]	Mithani *et al.* [188]	3.5

Notes: (*a*) **These numbers vary because not all features are defined or feasibly computable for all networks. The numbers listed are the total features used in each case.** (*b*) **This was the number of features retained after removing those for which values could be obtained for fewer than** 80% **of the** 312 **networks. We also tried variations of the** 80% **threshold, which led to different numbers of features being retained (see Section 3.3.2).** (*c*) **When using the** 80% **threshold as above but examining only the** 192 **real networks, as in Section 3.3.4.** (*d*) **There were networks representing** 6 **different pathways for each species; thus for each species we computed** 6 **versions of each network feature, one per pathway. Out of all of these,** 804 **was the total number of features retained after removing those with missing or ill-defined values.**

ships for networks in a more comprehensive fashion than has been possible when relying on intuitive network diagnostics alone; and it can thus serve as a powerful tool to aid scientific discovery on networks.

3.2 Data sets and algorithms

We used several hundred real and synthetic networks for our different case studies; we drew them from a variety of sources (see the summary in Table 3.1). The real networks include several kinds of biological networks (such as brain connectivity, protein interaction, and metabolic networks), social networks, and miscellaneous others (such as word adjacency, fungal growth, and financial correlation networks; details in Appendix B). The networks ranged in size from a few tens of nodes up to tens of thousands of nodes.

In addition to the real networks, we generated synthetic networks using several different kinds of models, including the community-detection benchmark of Lancichinetti et al. [159], Erdős-Rényi, preferential attachment, and duplication-divergence

(see Section 1.1.6 for descriptions of these). We used these network families for comparison to various sorts of real networks and for studies of hardness scaling (see Sections 3.3 and 3.4).

We used approximately 70 different network algorithms or diagnostics (listed in Appendix A) taken from the literature (see Section 1.1.4). Each of them takes a network as input and computes some property(s) of it.[1] For each network to correspond to one row and each feature to correspond to one column in our design matrix, we would ideally like each diagnostic to return only one real number when applied to a network. Whilst some of our diagnostics, such as the diameter (see Section 1.1.4.3), are of this sort, many others return a vector of real numbers: for example, the degree distribution (see Section 1.1.4.1) returns a sequence of node degrees. In order to obtain features from diagnostics that return distributions, we computed several summary statistics of these distributions—e.g., mean, variance, and other measures of central tendency and spread. We give the full list of summary statistics used in Appendix A.

Our collection of diagnostics also includes some community-detection algorithms, which return a partition of the network into subnetworks (see Section 1.1.3). We then compute several summary statistics of these partitions, such as the number of communities, partition entropy, and the fractions of nodes falling into different functional cartography roles (see Section 1.1.4.8), to serve as network features (a full list of these summaries is also in Appendix A). Certain community-detection methods contain a resolution parameter (Section 1.1.3.2), which allows one to examine community structure in a network at different scales; in particular, we use the Potts method [220] to compute discretised versions of mesoscopic response functions [203] (see Section 1.1.4.8), which also become features in our design matrix.

[1] In some cases, the code for these was publicly available (sources listed in Appendix A), but in other cases it had to be written. MATLAB code for some diagnostics was obtained from Gabriel Villar.

We attempted to apply all of the diagnostics to all of the networks, though given that many of them are computationally intensive, it is not always feasible to compute their values on larger networks. In order to partly compensate for this problem, we also compute sub-sampled versions of many of our diagnostics: we use a snowball sampling procedure [110] to draw 100-node samples from larger networks and then compute the given diagnostic on these. As noted in Section 1.1.4.10, these samples are not expected to preserve the characteristics of the full network; we thus add these features to our design matrix on a purely experimental basis, in keeping with the philosophy of attempting to probe network structure in as many different ways as possible. Since the sampled versions of the diagnostics do not show up in any of the specific results we present here, we have not studied them in depth. While no reliable conclusions about a full network could be drawn based on a single sample using a single sampling procedure, we do not attempt to make any such inferences here and indeed the actual results presented in this chapter are essentially unaffected by the presence of these sample-based features. A proper investigation of the differences between samples and the full network, and also of the effects of using different sampling strategies, remains a topic for future investigation. In Chapter 5, where we will use subsampling to fit generative models to protein interaction networks, we will examine the effects of changing the sampling procedure and sample sizes.

In addition to computational constraints, some diagnostics are also undefined for certain networks, for instance those which are not connected. Thus, when using the full set of features, our design matrix usually has some missing entries. We handle this either by removing certain columns (features) if they have too many missing entries or by imputing missing values in some way; details are provided along with each data set that we consider. In order to compare meaningfully the values of different features, we would generally like to put them on a common scale. We do this by normalising the design matrix as follows: for each feature, its values for all networks are standardised

to have zero mean and unit standard deviation; these are then mapped to the $[0, 1]$ interval via the logistic function $f(z) = 1/(1 + \exp(-z))$, which is a commonly used normalisation procedure [47].

Alongside the values of the features themselves, we also record the time taken to compute each feature for each network to which it is applied. This computation time itself can be used as a network feature—e.g., as a measure of the hardness of solving certain network problems (as demonstrated in Section 3.4).

3.3 Organisation of networks and features

As an illustration of our high-throughput approach, we show here how we can gain an overview of the relationships between different kinds of networks, as well as between different kinds of features. We map high-dimensional feature-space representations (see Section 1.3.1) of networks to lower-dimensional spaces and show that just a few (2–4) dimensions are sufficient to capture the bulk of variation between commonly studied types of empirical and model-generated networks. For this purpose, we computed the design matrix for a set of 312 networks (192 real, 120 synthetic) from a wide range of disciplines and models. We attempted to compute a total of 438 features for each of these networks, drawn from about 70 different diagnostics covering many kinds of structural properties: measures of the degree distribution, clustering of links, different notions of node centralities, frequencies of small motifs, mesoscopic structure via partitioning into communities, spectral properties of the adjacency matrix, and several others; a full list of diagnostics and features is in Appendix A.

3.3.1 Network data

Our set of 312 networks includes 192 real-world ones, obtained from Onnela *et al.* [203], which they classified into 12 different categories (these are listed in Appendix

B). In addition to these, we include 20 networks[2] for each of 6 different generative models (the networks in the last 3 categories were generated using code written by Samuel Johnson):

- *Duplication-Divergence-Attachment and Preferential Attachment (DDA+PA)*: This model was proposed in the context of the biological evolution of protein interaction networks [219] (the study of which is a particular theme of this thesis); it incorporates both duplication-divergence (Section 1.1.6.7) and preferential attachment (Section 1.1.6.3) mechanisms. At each step, with probability α a new node is linearly preferentially attached to one node of the existing network. With probability $1 - \alpha$, an existing node is chosen uniformly at random (the parent) and all of its links are duplicated for the new node. However, for each parental link, both it and its duplicate link from the child then have (independently) a probability δ_{Div} of being lost, but at least one of the links is retained; and also neither parent nor child is allowed to lose all of its links (this is the divergence step). Finally, the parent is attached to its child with probability δ_{Att}. We generate a set of 20 networks, of 50 nodes each, drawing parameters uniformly at randomly from the ranges $\alpha \in [0, 1]; \delta_{Div} \in [0, 0.3]; \delta_{Att} \in [0, 1]$. (These are as per the settings used when generating networks for model-fitting; see the discussion in Section 5.5. A random subset of the ensemble generated for that purpose was used here.)

- *Preferential Attachment Poisson (PAP)*: This version of preferential attachment was also proposed in the context of protein interaction evolution [218]. At each step of this model, one new node is added to the network via linear preferential attachment (see Section 1.1.6.3). The number of attachments formed by the

[2]The number 20 was chosen as it is close to the numbers of networks in the 12 real-world categories. Since we are including many different sorts of networks in this data set and our objective is primarily to examine the structural variations between categories, we chose to use a relatively small number of networks per category for ease of computation and visualisation.

new node being added is chosen from a Poisson distribution with mean m (the number drawn is incremented by 1, to ensure that at least one attachment is formed). Thus, m and the number of nodes n are the sole parameters for this model. We generate 20 networks, 50 nodes each, drawing the parameter m uniformly at random from $[0, 30]$ (as per Ref. [218]).

- *Erdős-Rényi*: 20 networks chosen uniformly at random from all possible networks with 50 nodes and 100 links, i.e., the $G(n = 50, m = 100)$ model (see Section 1.1.6.1).

- *Modular*: 20 networks of 100 nodes each, generated as follows: we start with 10 modules of 10 nodes each. Initially each module is a fully connected network (i.e., a *clique*), and there are no links between modules. Then, we iterate through all of the links in the network and for each one, with probability 0.5, we disconnect it from one of its nodes and rewire it to a new randomly chosen node.

- *Power-law degree distribution*: 20 networks of 100 nodes each, generated using the configuration model (see Section 1.1.6.3), which imposes a fixed degree sequence $(k_1, k_2, ..., k_{100})$ and then the expected value of nodes i and j being connected is $k_i k_j/(100\langle k\rangle)$, where $\langle k\rangle$ is the mean degree. We set $\langle k\rangle = 4$ and choose the degrees from a power-law distribution with exponent $\gamma = 3.1$—i.e., $p(k) \propto k^{-3.1}$. (These parameter settings were initially used to generate part of an ensemble of networks we intended to use for the entropy comparisons discussed in Section 4.5. A random subset of that ensemble was included here. We chose $\gamma = 3.1$ as that is nearly in the middle of the range of exponent values observed for real-world networks [28].)

- *Small world*: 20 networks of 100 nodes each, generated from the Watts-Strogatz model (see Section 1.1.6.4), with $\langle k\rangle = 4$ and $p = 0.5$. (This is a random subset

of the ensemble used in Section 4.5; we chose $p = 0.5$ as it is in the middle of the $[0, 1]$ range for that parameter.)

3.3.2 Isomap and network clustering

For each of the 312 networks, we compute the full set of features, which amounts to 438. This yields a 312×438 design matrix. To deal with missing values (features that are undefined or intractable for some networks), we filter out columns (features) that are less than 80% full—i.e., the corresponding features could only be computed for less than 80% of the networks.[3] This leaves 338 columns. Computing pairwise network/feature correlation distances (the correlation distance between two vectors is $1 - |\rho|$, where ρ is their linear correlation coefficient) and reordering the design matrix using single-linkage clustering (see Section 1.3.3), such that similar rows or columns are adjacent, allows similarities to be seen. See Figure 3.1.

To examine in more detail the nature of these similarities between these networks as mapped to our feature space and to see if they correspond to meaningful divisions or categories, we carried out a non-linear dimensionality reduction using the Isomap algorithm [258] (see Section 1.3.3). This allows us to map the high-dimensional feature space onto a low-dimensional projection, with dimensions chosen so as to maximise the variance captured. We run Isomap on our 312×338 design matrix (setting the parameter $k = 11$, i.e., each data point is connected to its 11 nearest neighbours, this being the lowest that gives a connected network for our data), having replaced missing values (which amount to 6.5% of all entries in the matrix) with the average

[3] We repeated the procedure with this threshold set at 70% and 90%, and this did not significantly change the results. In either case, the first reduced dimension correlated very strongly with density ($\rho \approx 0.94$ for the 70% threshold and $\rho \approx 0.96$ for the 90% threshold), whilst the second correlated quite strongly with energy ($\rho \approx 0.73$ and $\rho \approx 0.61$, respectively) and with the number of nodes ($\rho \approx 0.66$ and $\rho \approx 0.79$, respectively). Whilst these two dimensions captured over 95% of the variance, we additionally observed that in both cases 2 of the next 3 reduced dimensions were ones with strong correlations to group degree centrality (see Section 1.1.4.2) and the fraction of the network covered by the 2-core (see Section 1.1.4.1). Thus, the groupings of networks obtained in the results presented here (see Figure 3.2) are essentially preserved in these two settings.

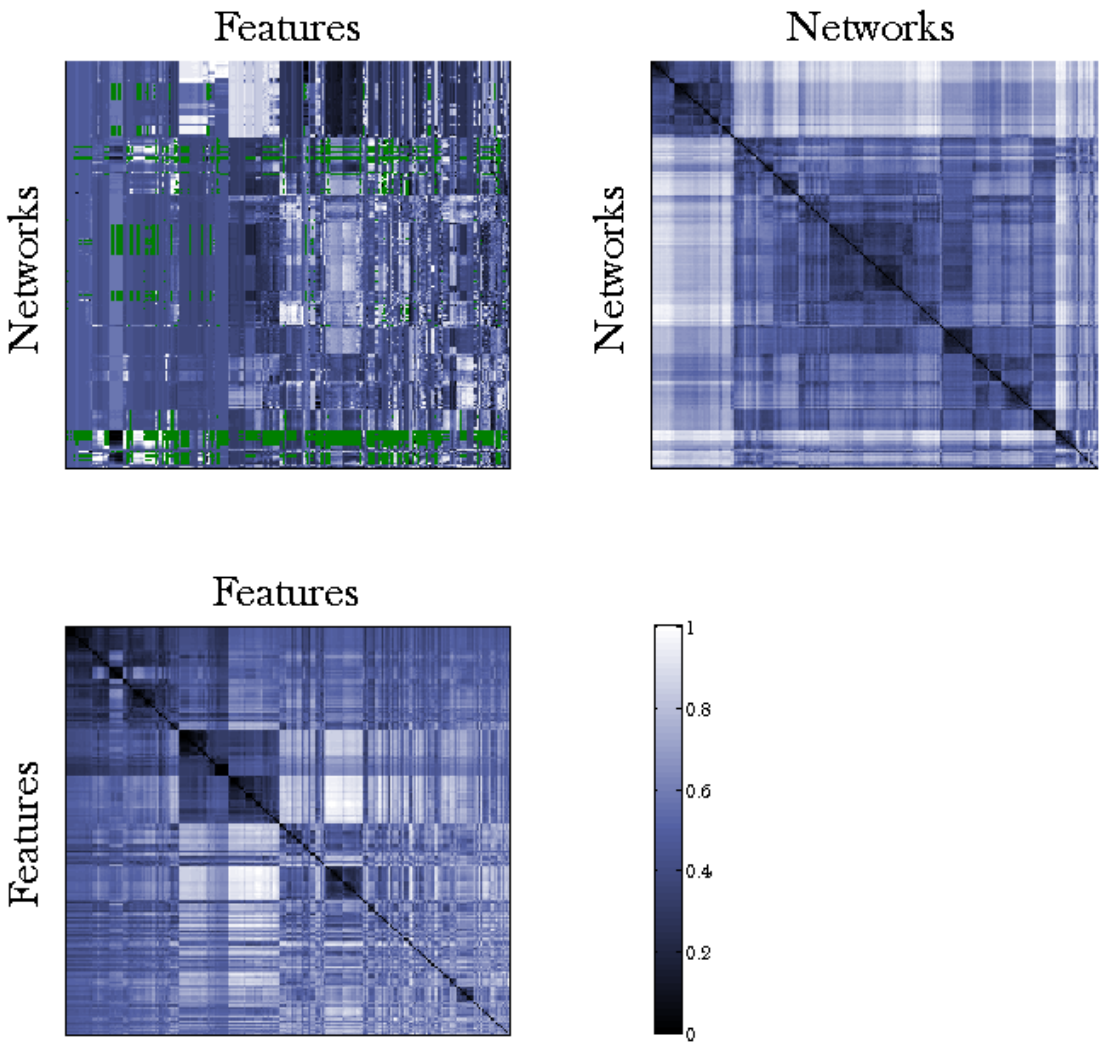
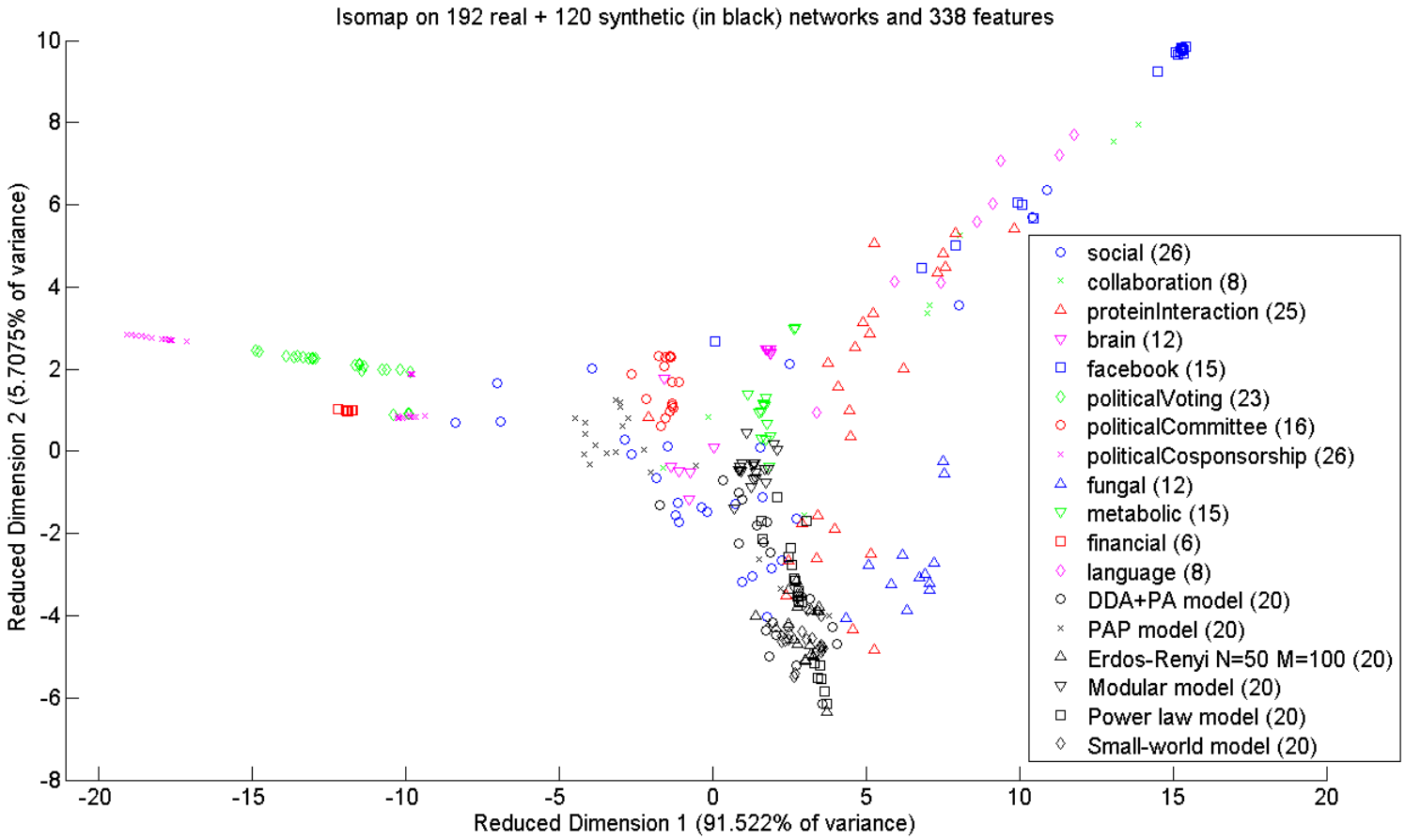


Figure 3.1: **Network-feature matrices.**

(Top left) Design matrix, with networks along the rows and features along the columns, ordered using single-linkage clustering (see Section 1.3.3). Green entries denote features that were undefined or were not computed for the given network due to time constraints. (Top right) Matrix of pairwise correlation distances ($1 - |\rho|$, where ρ is the linear correlation coefficient) between networks. The roughly block-diagonal structure indicates the presence of sets of closely related networks. (Bottom) Matrix of pairwise correlation distances between features.

of all entries in that column.[4] We show the resultant low-dimensional embeddings in Figure 3.2. Each data point represents a network's position along the top 4 reduced dimensions and different symbols depict the different domains from which the networks are drawn. We see that even in these 2-dimensional mappings, certain kinds of networks form highly cohesive groupings, including financial, fungal, and metabolic networks. Other network types, such as protein interaction, collaboration, and social networks, are less clear-cut, though still confined to relatively restricted regions of the space. (This likely indicates that these latter categories are less well-defined and include networks from a wider range of sources. Similar clustering patterns were observed by Onnela *et al.* [203], who constructed a taxonomy of these networks using a distance measure based on the mesoscopic response functions discussed in Section 1.1.4.8.)

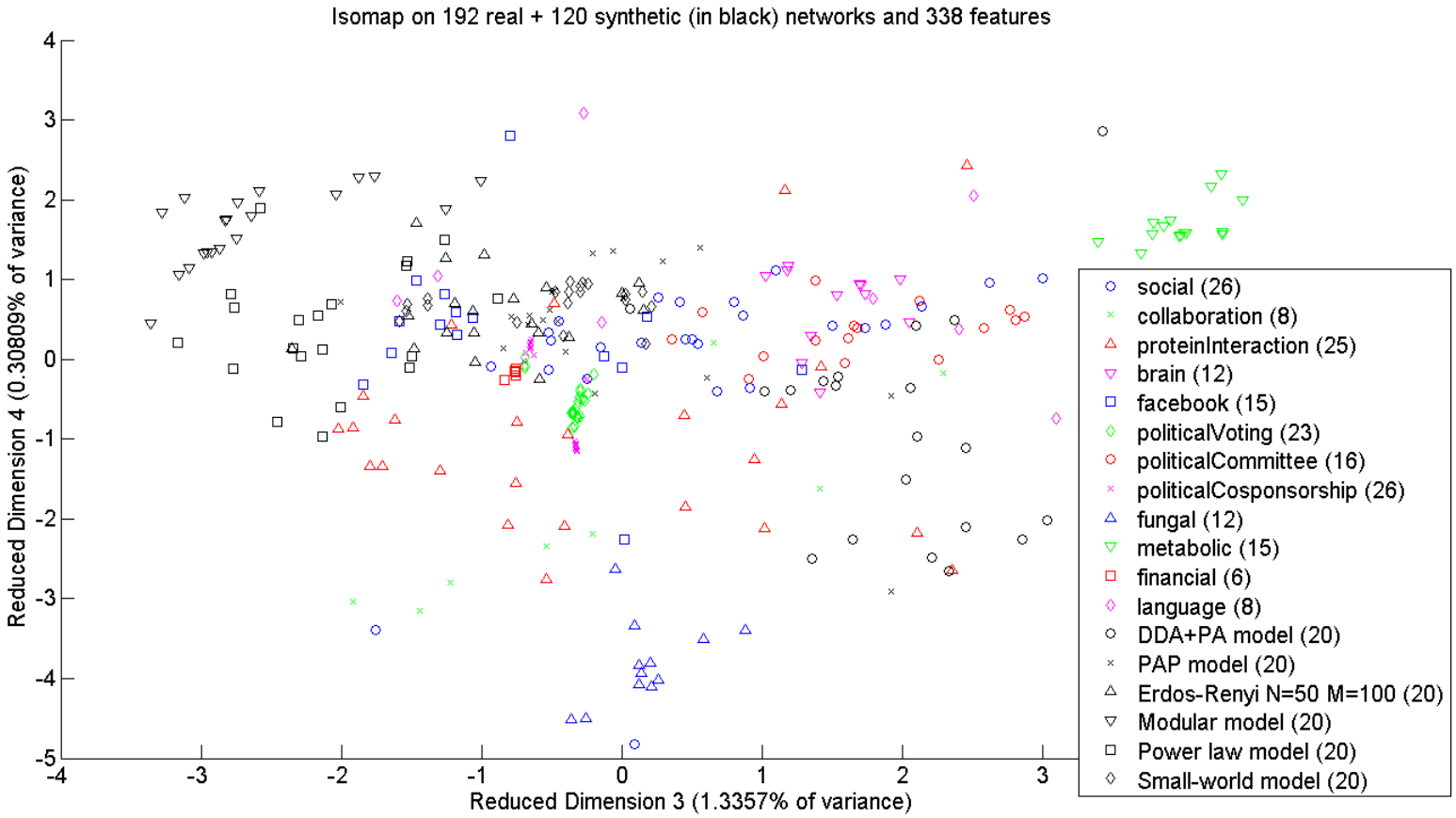
[4]We also tried adding zero-mean Gaussian noise to these imputed missing values, with the spread (variance) of the Gaussian being set to the empirical variance of the values in the column. Repeating the computations with the noise added did not significantly change the results. The first reduced dimension was still found to have a high correlation with network density ($\rho \approx 0.95$), and the second with energy ($\rho \approx 0.72$) and the number of nodes ($\rho \approx 0.73$). Whilst these two dimensions captured 96% of the variance, we additionally found that the third reduced dimension was substantially correlated with group degree centrality ($\rho \approx 0.63$) and the fifth one with the fraction of the network covered by the 2-core ($\rho \approx 0.55$), thus recapitulating the axes of structural variability depicted in Figure 3.2.



(a) First 2 reduced dimensions

Figure 3.2: **Network clustering via Isomap dimensionality reduction.**

We note that many of the network types form tight clusters in this reduced feature space. Numbers in parentheses indicate the number of networks in each category. The first reduced dimension is strongly correlated with the density of links in the network ($\rho \approx -0.91$); and the second is strongly correlated with network energy (see Section 1.1.4.9) ($\rho \approx 0.87$) and also with the number of nodes ($\rho \approx 0.73$).



(b) Third and fourth reduced dimensions

Figure 3.2 (continued): Several network types show substantial clustering even along dimensions capturing aspects of network structure other than size and density, indicating that these categories are marked by distinctive structural characteristics. The third dimension has $\rho \approx 0.6$ with group degree centrality, a measure of variation in the node degrees (see Section 1.1.4.2) and the fourth has $\rho \approx 0.67$ with the fraction of the network covered by the 2-core, a measure of network cohesion (see Section 1.1.4.1).

The first four Isomap dimensions combined capture nearly 99% of the total variance, with the first one alone accounting for over 91%. In Figure 3.3 we depict the residual variance, i.e., the proportion of the variance in the data not captured in a low-dimensional embedding (see Section 1.3.3), on the set of 312 networks as we increase the dimensionality of the Isomap. These numbers indicate that differences between several of the types of real-world and model-generated networks commonly studied can be captured substantively by a very small number of quantities. This would appear to be at odds with our earlier statement that networks are 'intrinsically high-dimensional objects'. There are at least two factors which are likely to be part of the explanation for this. One is that we consider only a selected set of network types, and they are of many very differing sorts; so the differences between these may predominantly be accounted for by just a few structural features, as a large part of the space of possible network structures is probably not covered by our data. However, it may also be that it is not just an issue of limited data, but that the types of network structures that are observed at all in the real world or obtained via commonly used generative models are substantially constrained, such that they occupy only a restricted part or parts of network structure space. The second factor is that by mapping networks to a space of features we are clearly discarding some information; and whilst the dimensionality of our feature space is several hundreds, it is also the case that there are many correlations between these features (as discussed further in Section 3.3.4) and the number of effectively independent structural properties they can capture may be far fewer. We will explore further the issue of constraints on particular network structures and how they might relate to correlations between our network features, in Chapter 4.

To examine which network features these reduced dimensions are capturing, we computed linear correlation coefficients of the dimensions with the original set of features. The first reduced dimension is maximally correlated with the density of

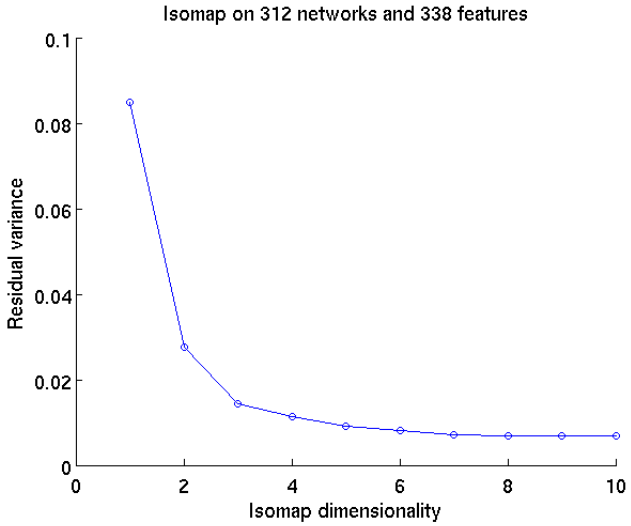


Figure 3.3: **Residual variance as the number of Isomap dimensions is increased.**

links in the network ($\rho \approx -0.91$). The second is most correlated ($\rho \approx 0.87$) with network energy (a measure of the number of indistinguishable networks with the same degree distribution; see Section 1.1.4.9). This dimension also shows a substantial association with the number of nodes in the network ($\rho \approx 0.73$). Thus, unsurprisingly (given the wide variety of networks examined), the two basic measures of network size and density are sufficient to capture a large amount of the variability between the different types of networks. The third and fourth reduced dimensions are not strongly correlated with any such basic features; however, the features that show relatively large correlations with the third dimension include group degree centrality ($\rho \approx 0.60$) and group closeness centrality ($\rho \approx 0.55$) (see Section 1.1.4.2 for definitions), whereas the fourth dimension correlates substantially with the fraction of the network's nodes covered by the 2-core (see Section 1.1.4.1) ($\rho \approx 0.67$) and the network diameter (see Section 1.1.4.3) ($\rho \approx -0.64$). Thus, roughly speaking the third dimension is providing an indication of how much variability there is in the centralities (which can be defined in multiple ways) of nodes in the network; whilst the fourth dimension is associated with how cohesively the network is connected up (a bigger 2-core is indicative of

greater cohesion, as is a smaller diameter, other things being equal). These two lower dimensions allow us to see that certain kinds of networks, such as fungal and metabolic, form tight clusters even with regard to these more complicated structural characteristics.

3.3.3 Network classification

We also attempted to learn a supervised classification tree (see Section 1.3.2.1) for the same set of networks, using MATLAB's `classregtree()` function, to examine how accurately different sets of network features can categorise the 312 networks into the 18 classes (12 real, 6 model-generated) we specified. Because the low-dimensional mapping of Figure 3.2(a) suggests that network size and density alone seem to account for a lot of the differences between the network categories in our data set, we first did the classification using these two features alone. This achieves a 10-fold cross-validation accuracy (see Section 1.3.2.1 of $71.15 \pm 2.10\%$ (mean and standard error over 10 folds). We then constructed trees using a third feature in addition to size and density, iterating through all other features and picking one at at time. In each case, we evaluated the classification accuracy in the same way. The highest average accuracy obtained was $79.49 \pm 1.84\%$, for the feature $pottsModel_{numComm\_auc}$, which represents the mean number of communities obtained when a given network is partitioned at 10 different resolutions (equally spaced between the limits in which the whole network is in one community and in which every node is a separate community) via the Potts model [220], using the Louvain optimisation algorithm [48] (see Section 1.1.3). We show a three-dimensional scatter plot of all 312 networks in the space of these three features in Figure 3.4. The next best features in terms of classification performance are $szegedIndex$ (see Section 1.1.4.3), with an accuracy of $78.85 \pm 1.96\%$, and $geodesicDistanceMean$ (see Section 1.1.2) with $78.85 \pm 1.95\%$. It is notable that both these are measures of how well or uniformly pairs of nodes in the graph are

connected. The Szeged index was in fact originally proposed for characterising graphs of molecular structure [145]; it was suggested as an alternative to the *Wiener index*, which is directly proportional to $geodesicDistanceMean$.

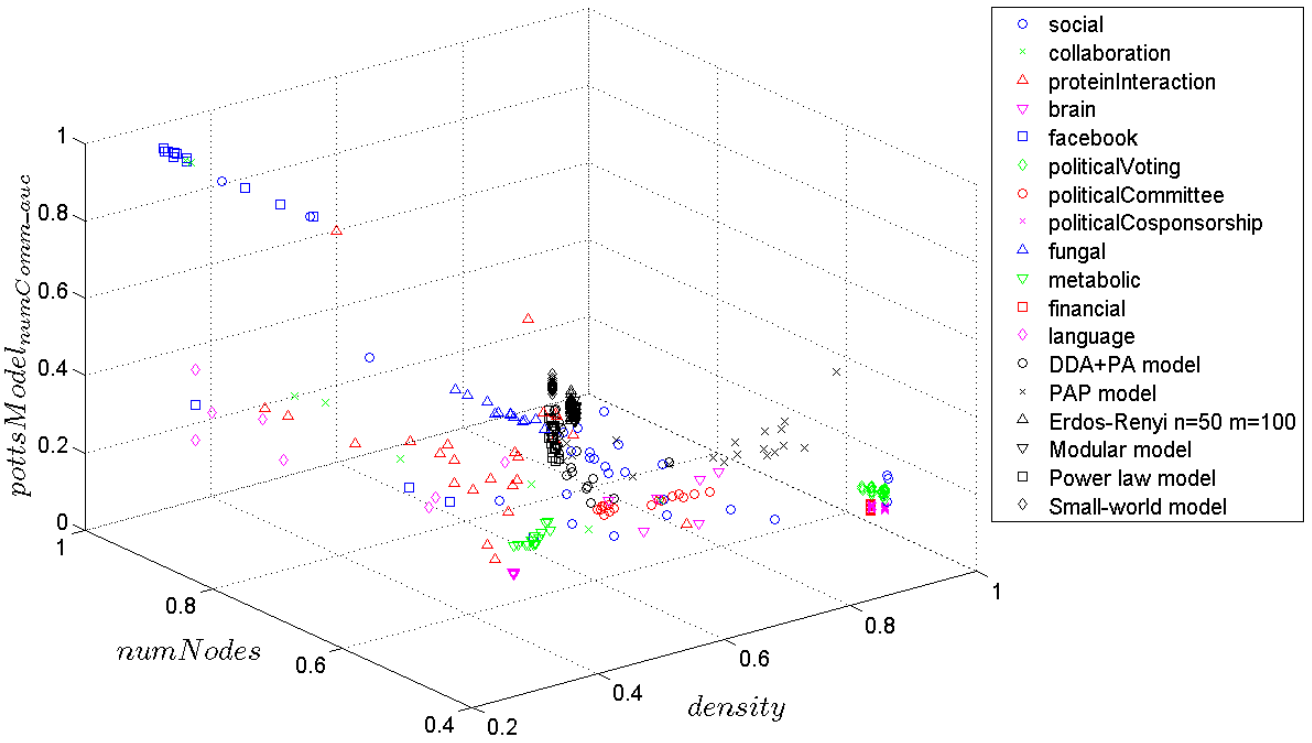


Figure 3.4: **Scatter plot in the space of 3 selected features.**

Set of 312 networks (including 120 synthetic, in black) plotted in the space of 3 features: the number of nodes, the density, and the average number of communities (as a fraction of the number of nodes) returned over 10 resolutions by the Potts model method, using Louvain optimisation (see Section 1.1.3). All features are normalised to the $[0, 1]$ range via the logistic function (see Section 3.2). Using these three features to construct a classification tree for these 18 network types leads to a cross-validation accuracy of $79.49 \pm 1.84\%$, whereas if only size and density are used the corresponding accuracy is $71.15 \pm 2.10\%$, indicating that the feature based on multi-resolution community structure captures a significant distinguishing factor between the different network types (see the main text discussion).

We also built a classification tree allowing for the use of all of the features; the optimal tree obtained used a subset of 16 features and achieved an average 10-fold cross-validation accuracy of $80.45 \pm 1.90\%$. These results suggest that adding further features beyond 3 does not lead to any significant improvement in classification accuracy for this varied set of networks; thus whilst some information about structural characteristics (in addition to the basic measures of size and density) is useful for distinguishing between these network types, it is possible to essentially capture this in just a single feature that our scan of a large feature space allows us to pick out. In particular, the usefulness of the Potts model feature is in agreement with the observations of Onnela *et al.* [203], who found that community structure at multiple resolutions is informative in organising different kinds of networks. As noted there, this suggests that the community-level or mesoscopic organisation exhibited by a network is functionally relevant, and thus networks of a certain kind tend to display certain specified or constrained kinds of community structures. However, as noted we find that the Szeged index and the mean geodesic distance, measures of global connectivity, are also nearly equally useful in distinguishing between the different sorts of networks considered here. This suggests that whilst mesoscopic structure may be relevant to the functionality of a particular network type, this correspondence may to a large extent be captured by a simpler measure of global connectivity. Clearly these aspects of network structure are not independent; indeed, in general the network features we use display substantial correlations, which we seek to examine in more detail in the next section as well as in Chapter 4.

3.3.4 Communities of features

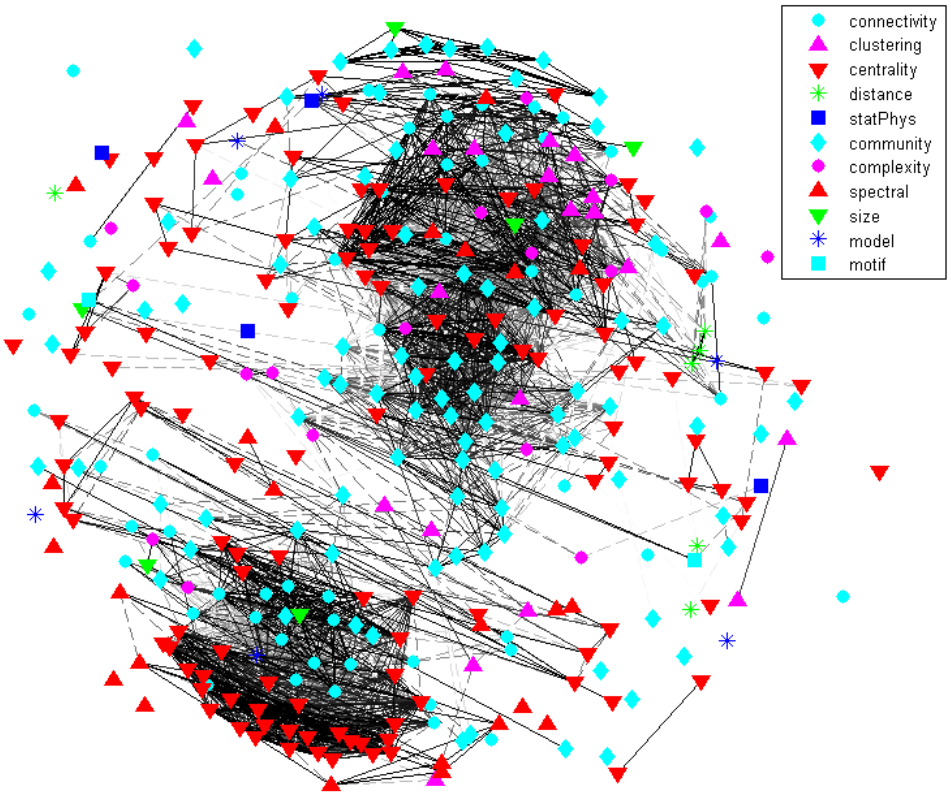
Figure 3.1 suggests that the features also show substantial clustering. To examine this further, we computed linear correlation coefficients (over the set of 192 real networks) between pairs of features and represented this as a weighted network of features, where

the weight of the link joining two features is their absolute linear correlation. This network contains the 347 features that remain after having accounted for missing values by filtering out columns that are less than 80% full and then replacing by column averages (see the discussion in Section 3.3.2). We show this in Figure 3.5(a), with node symbols representing a crude categorisation of features. We also carried out community detection on this network to detect clusters of highly correlated features. For this we used the C++ implementation of the Louvain optimisation algorithm by Blondel *et al.* [48] to detect communities via the Potts method [220] (see Section 1.1.3). We tried 10 different evenly-spaced values for the resolution parameter (as described in Section 3.3.3), after Onnela *et al.* [203]; for visualisation purposes, we chose the resolution where the average community size is nearest to the square root of the total number of nodes, yielding 21 communities with an average of about 16.5 nodes per community.

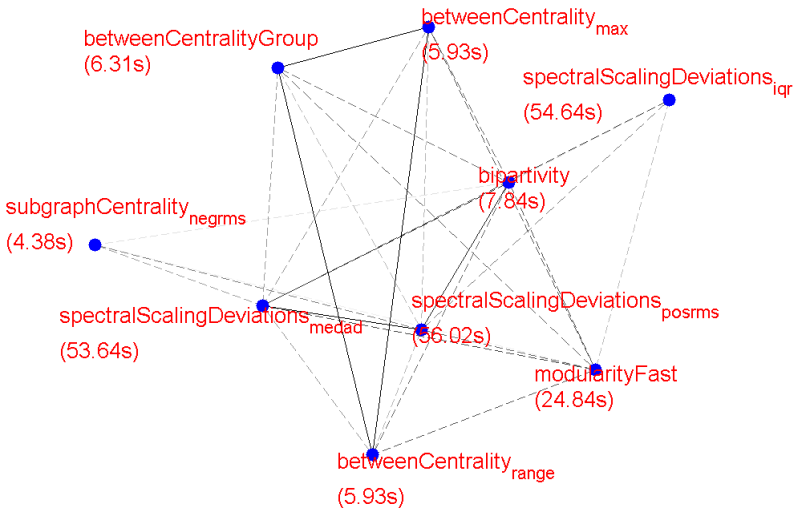
We depict one of these 21 communities in Figure 3.5(b) as an example of how our approach can uncover interesting feature associations. This shows, for instance, that summary statistics of the distribution of spectral scaling deviations, which were proposed as a way of classifying network topologies [78] (see Section 1.1.4.7), are (for the data considered) substantially captured by more directly interpretable measures like bipartivity (see Section 1.1.4.2) and Newman-Girvan modularity (maximised here via the Louvain method [48]; see Section 1.1.3), which are also quicker to compute.[5] This is sensible in light of the fact that the spectral scaling deviations were proposed to provide a measure of how close a network is to being bipartite or being composed of cliques [78].

Here we have examined a single community from a single network constructed on the basis of linear correlation coefficients between features on our particular data

[5]Even though exactly maximising the Newman-Girvan modularity is NP-hard [52] and thus not tractable except for very small networks, the value from the Louvain heuristic used here (denoted $modularityFast$) is nearly twice as fast to compute on average as features based on spectral scaling deviations (see Figure 3.5(b)).



(a)



(b)

Figure 3.5: **Feature correlations on a set of 192 real-world networks.**

(a) Network of features grouped by broad categories (see Appendix A for details). Darker links represent stronger correlations; absolute correlations of at least 0.9 are depicted by solid lines, and those in the range $[0.8, 0.9)$ we depict by dashed lines. (b) Magnified view of one community from the above network, showing feature names and the average time taken to compute each one (on an Intel Core 2 Quad Q9550 2.83GHz CPU).

set. To assess the robustness of the associations suggested one would like to look at the effects of varying these particular choices—i.e., using other distance measures between features or other network data sets. Whilst we have not done so due to time constraints, the results presented here provide an example of how the sort of large-scale comparison we carry out can be useful in identifying the most direct and computationally tractable ways of capturing relevant aspects of network structure.

As an extension of classifying networks or features into discrete groupings, we can attempt continuous regression of network features against functional properties. This will help us to understand better how structural characteristics are correlated with functionality, and might assist in enabling appropriate design choices in situations where we are seeking to construct/modify a networked system towards a specific end. In the remainder of this chapter, we demonstrate the potential utility of this approach via two case studies. We first look at how structural features can be used to estimate the hardness (i.e., computational difficulty) of a graph-theoretic problem and thus provide insight into what kinds of networks are amenable to the performance of useful computations. We then return to the domain of biology and seek to ask in what ways evolution has constrained the structure of metabolic networks and what this might tell us about the biological significance of the way these networks are organised. Thus, by means of the following examples, we show how our methodology can help provide insights into the functional relevance of different aspects of network structure.

3.4 Hardness regression

Many problems defined on graphs are motivated by real-world tasks: a classic example is the Travelling Salesman Problem (TSP), which involves finding the shortest tour that traverses a given set of cities, visiting each exactly once [117]. Like many

interesting graph problems, this is known to be NP-hard in general [99] (meaning that there is no known algorithm that can solve it in time that grows no faster than a polynomial function of the graph size), though there exist various heuristic and approximate approaches to solving it. Here we show via regression against network features how it is possible to identify informative predictors of quantities like the TSP solution length and computation time for a given graph instance; in general, such predictors can be computed much more quickly than actually solving the problem itself.

In recent years, work in the area of parameterized complexity [73, 89, 198] has sought to discover algorithms that can solve such NP-hard problems in polynomial time for certain kinds of graphs: graphs that in some sense have a sufficiently 'simple' structure so as to make the problem easier than on arbitrary structures. However, the major difficulty with this approach is often finding an appropriate structural parameter of the graph that can capture this notion of 'simplicity' in the context of a given problem and which is itself easy to compute. Because our framework allows the computation and comparison of a large number of graph or network characteristics, this might help filter out specific features that correlate in some way with the hardness of solving the problem at hand. Hardness classification of TSP instances based on a small predetermined set of features specific to the problem has been attempted recently [245]; here we are able to utilise a much larger and more wide-ranging set of network characteristics and automate the selection of the most informative ones. Network features can also be regressed against some measures of the problem solution (e.g., length of the optimal TSP tour) for known instances (i.e., a training set). This allows identification of features that can be used as predictors of the measures of interest for novel instances (i.e., test sets) and are significantly faster to compute than solving the problem itself. To demonstrate this, we take the specific example of solving TSP on the distance matrix of pairwise shortest-paths between nodes for a given graph, using the heuristics described in the next section.

3.4.1 TSP solvers

We examine the relation of network structure to the TSP solving performance of three different types of randomised heuristic methods, all of which have been widely used for combinatorial optimisation in general: cross-entropy, genetic algorithms, and simulated annealing. The cross-entropy method [68, 232] was originally proposed as a way of doing rare-event simulation [231] and was later extended to optimisation [230]. It is an iterative Monte Carlo method that explores the search space of possible solutions by random sampling but uses a technique known as *importance sampling* to improve efficiency: this involves sampling from a distribution that increases the probability of the occurrence of a rare event (here, finding the optimal solution). The basic nature of the iteration is as follows:

1. Generate a random sample of possible solutions (i.e., for TSP, possible node sequences) using some specified mechanism (probability distribution over instances).

2. Use the sample to update the parameters of the mechanism for the next iteration to improve efficiency (get closer to the optimal solution).

The key feature of the cross-entropy method is that it provides a precise way to carry out step 2 based on minimising the *Kullback-Leibler (K-L) divergence* (or cross-entropy) between the current sampling distribution and the theoretically most efficient one for importance sampling. The K-L divergence [denoted $D_{KL}(p||q)$] is a standard measure of the distance between two probability distributions (p and q, defined over domain $\mathcal{D}$) and is defined as

$$D_{KL}(p||q) = \int_{\mathcal{D}} p(x) \log \frac{p(x)}{q(x)} dx. \tag{3.1}$$

The cross-entropy method has been applied to a range of combinatorial optimisation problems; we used the MATLAB TSP solver implemented as part of the Cross-Entropy Toolbox [3].

In order to compare the results obtained using cross-entropy with other popular optimisation methods for solving TSP, we also utilised a genetic algorithm [128] implemented by Joseph Kirk [2][6], and a simulated annealing [57] implementation by Aravind Seshadri [1][7].

3.4.2 Network feature correlations

We generate a set of 500 synthetic networks of 50 nodes each, using the Preferential Attachment Poisson (PAP) model (as described in Section 3.3.1); the model parameter m is chosen uniformly at randomly from the range $[0, 30]$, following Ref. [218]. We chose to use this model as an example for this purpose because it generates networks with heavy-tailed degree distributions (see Section 1.1.6.3), a property that has been observed in several kinds of real-world networks [28]. For each of these networks, we compute a feature vector, along with the length of the best TSP solution returned by the cross-entropy algorithm (which we denote *tspl*). This solution length is equal to the sum of lengths of all of the pairwise shortest-paths included in the solution, normalised by the shortest possible length for the given graph, which is equal to the number of nodes since the graphs are unweighted. We also record the time taken to compute each best solution, allowing us to rank the different properties based on their correlation with either TSP length or runtime. We show scatter plots of three of the most correlated features found in each case in Figure 3.6.

Several types of features show strong correlations to the solution length: in particular, modularity (which we optimise via the spectral method; see Section 1.1.3) and

[6] We used the default settings of a population size of 100 and $10,000$ iterations.

[7] Following example suggestions, we set the initial temperature to 30, and the cooling rate to 0.5 (the initial temperature was multiplied by this factor every 10 iterations). The number of iterations used was $1,000$ and a maximum of 4 node pairs were swapped in each iteration.

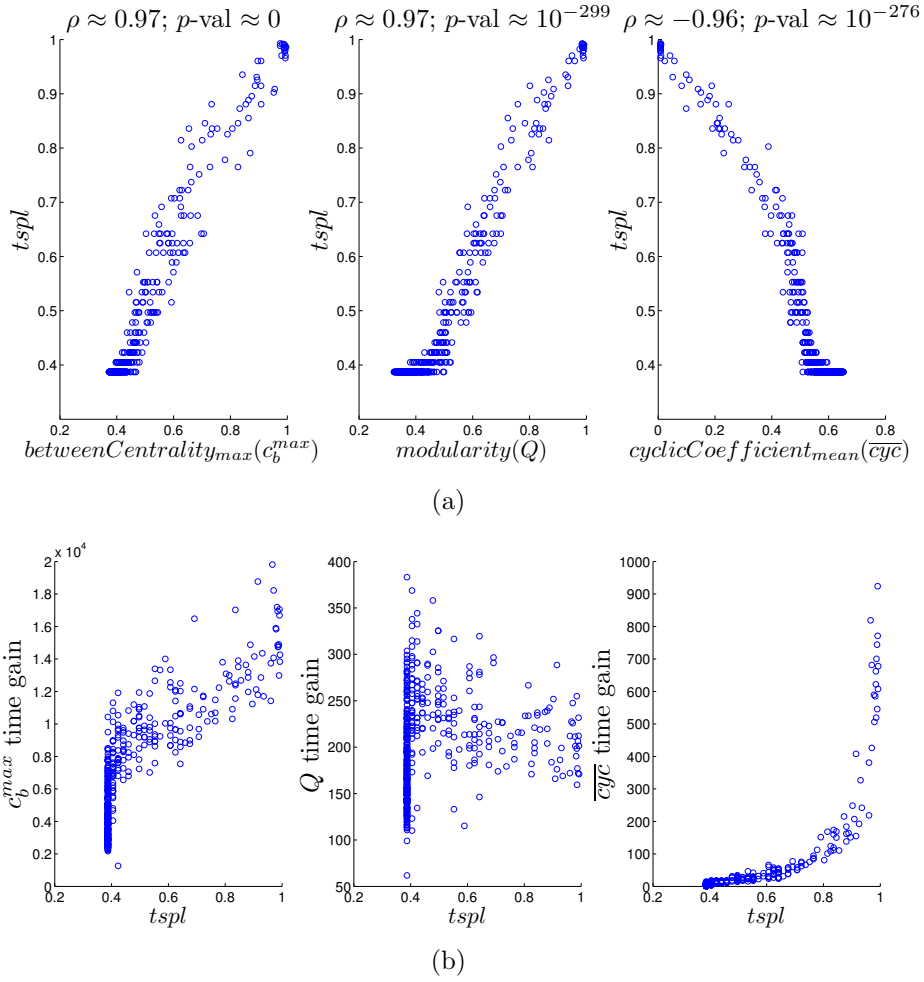
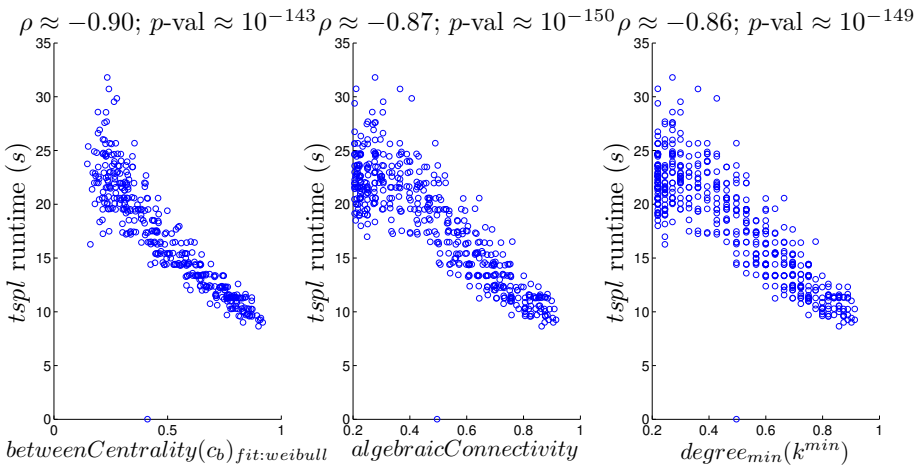


Figure 3.6: **Network features correlate significantly with outputs from a cross-entropy TSP solver.**

(a) Example features with high correlation (denoted by ρ) to solution length from the cross-entropy TSP solver, over a set of 500 networks from the PAP model (see Section 3.3.1). The p-values are for the null hypothesis of zero correlation and are *Bonferroni-corrected* to account for the multiple tests (i.e., they are multiplied by the number of features tested). (b) Time gain ratio: TSP solver runtime divided by diagnostic runtime. This indicates the putative gain in computational time by using these features to estimate solution length on novel TSP instances, as opposed to actually solving them.



(c)

Figure 3.6 (continued): (c) Example features highly correlated to TSP runtime. The *algebraicConnectivity* and $degree_{min}$ features indicate that this solver is faster for networks with more uniform connectivity (see the discussion in the main text). See Appendix A for feature details.

the maximum of geodesic node betweenness centrality (see Section 1.1.4.2). Networks that are more modular or have very high-betweenness nodes tend to have longer TSP solutions. This appears to be reasonable because of the need to take long paths to get between sparsely connected modules or to cycle back frequently to a high-betweenness node in order to get from one part of the network to another. It also suggests that these features, which can in general be computed much more quickly than actually solving TSP[8], are useful in obtaining quick estimates of TSP solution length for a given graph. Figure 3.6(b) shows that the maximum betweenness centrality, for instance, is on average about $10,000$ times faster to compute than running the TSP solver on the corresponding graph. In general, the time required for computing this feature scales as a polynomial function of n, the number of nodes in the network (it is proportional to n^3 using a version of the standard Floyd-Warshall algorithm),

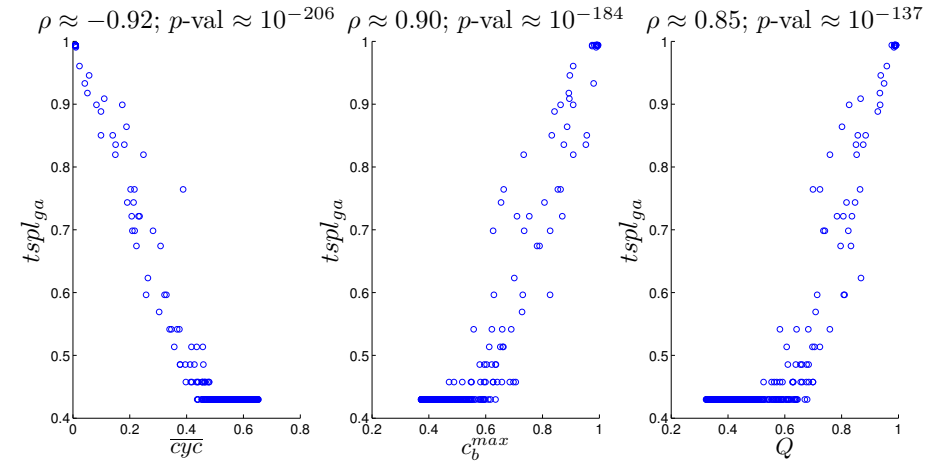
[8]This is based on the algorithms that we actually employ. For instance, maximising modularity exactly is also NP-hard [52]. However, a comparison of the time taken to solve TSP via the cross-entropy heuristic versus the time taken to maximise modularity via the spectral heuristic (both heuristics implemented in MATLAB) shows that the latter is 100 to 300 times faster for the data considered here (see Figure 3.6(b)).

whereas solving TSP as noted earlier is NP-hard.

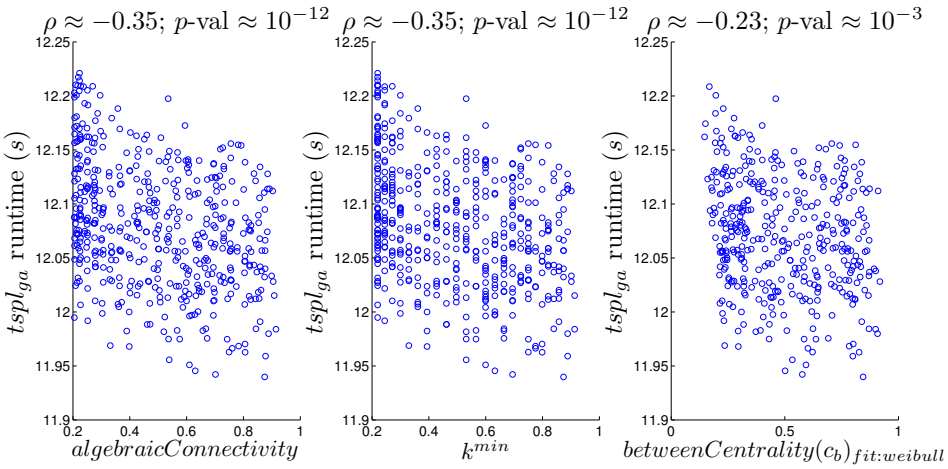
There are also significant feature correlations with the runtime of the cross-entropy TSP solver (examples depicted in Figure 3.6(c)): algebraic connectivity [60] (see Section 1.1.4.7) and minimum degree appear to be amongst the best predictors in this case. This also seems sensible: networks that are more uniformly connected (something also likely to co-occur with a relatively high minimum degree) should be ones in which finding a good TSP solution is faster, as a randomly chosen node sequence will tend to be not too far off the optimal path. It is worth noting that the features highlighted by this computation depend not just on the particular problem being solved, but also on the algorithm used to solve it.

For the two other TSP solvers we tried, based on a genetic algorithm and simulated annealing, we found that the correlations of their solution lengths (denoted by $tspl_{ga}$ and $tspl_{sa}$ respectively) with these network parameters are quite similar, but the correlations of runtimes are substantially different, as depicted in Figures 3.7 and 3.8. This indicates that when observing correlations with runtime, we are in large part picking out aspects of network structure specifically relevant to the performance of each algorithm (in fact, for these two algorithms, there is not much variation in runtime across the set of networks employed here).

The detection of features like modularity and maximum node betweenness centrality, which show very high correlations with TSP solution length across our 3 solvers, suggests that directly constraining these features when constructing synthetic networks (at least for the class of preferential attachment networks considered here) can be a means of obtaining networks with a specified property such as a relatively short TSP solution. Also, correlations with runtime (like the observation that the cross-entropy solver is faster for more uniformly connected networks) can be used to choose an appropriate solver for a network with particular structural characteristics, and can also motivate a deeper investigation of whether the sorts of features identified here can



(d)



(e)

Figure 3.7: **Network feature correlations with genetic algorithm TSP solver.**

(a),(b) Correlations of network features (as in Figure 3.6) with solution length over a data set of 500 networks from the PAP model. (c),(d) Correlations to TSP runtime.

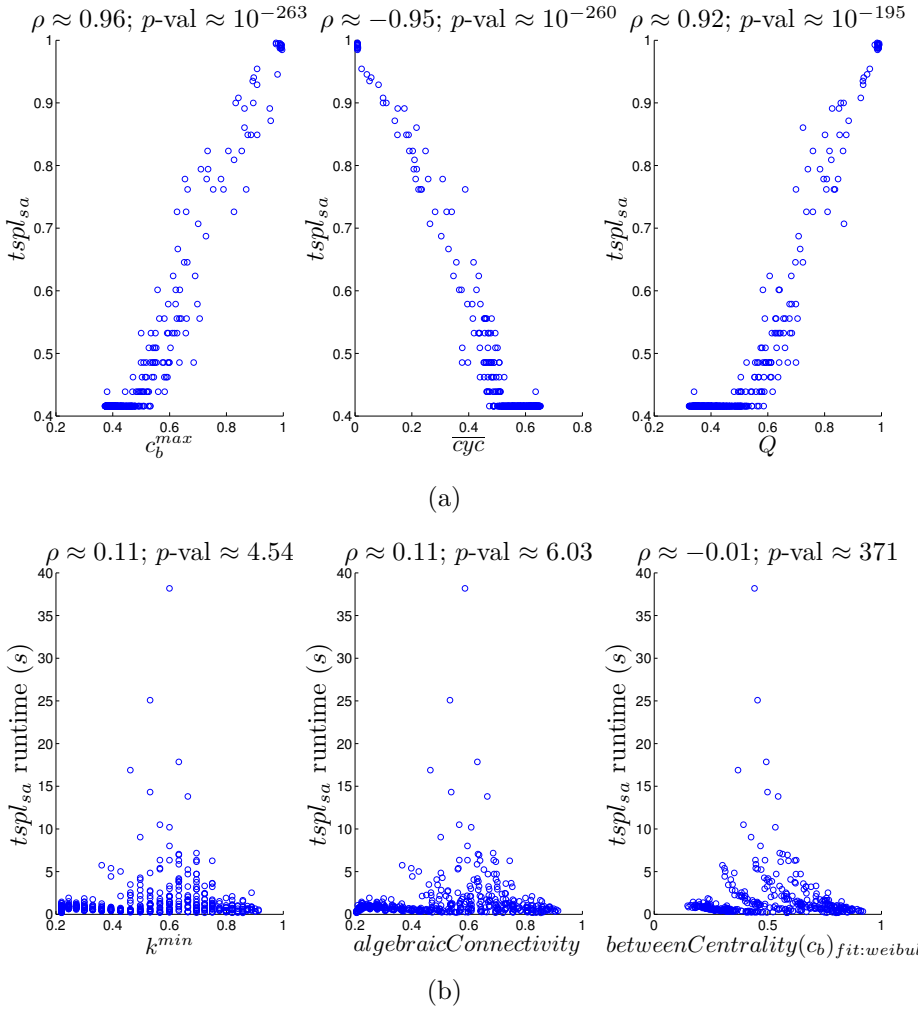


Figure 3.8: **Network feature correlations with simulated annealing TSP solver.**

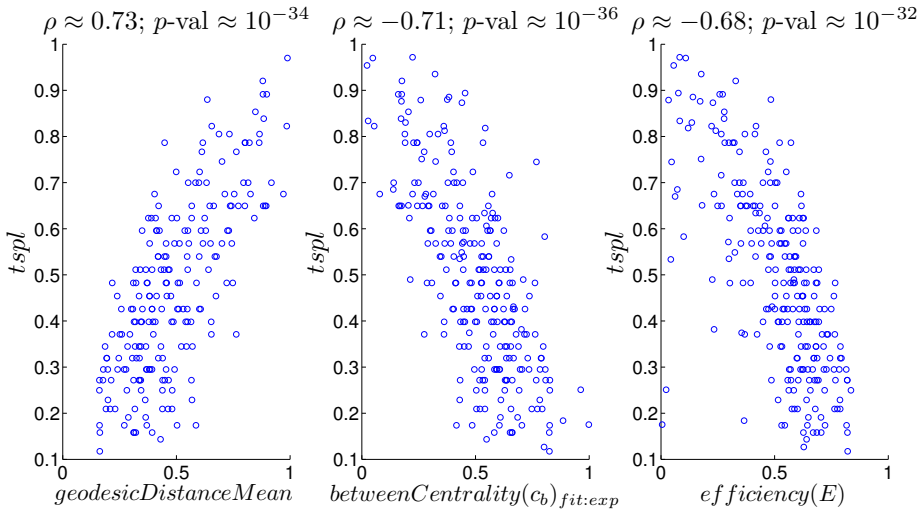
(a),(b) Correlations of network features (as in Figure 3.6) with solution length over a data set of 500 networks from the PAP model. (c),(d) Correlations to TSP runtime.

serve as markers of 'simplicity' with respect to a particular TSP heuristic, for graphs in general as opposed to the particular class considered here. Whilst our observations here are contingent upon the particular choices made, such as the graphs considered and the solvers used, and further work is needed to obtain more robust conclusions, the examples presented demonstrate how our approach can help to highlight specific diagnostics and aspects of network structure which are relevant to a particular task and are worth examining in greater detail.

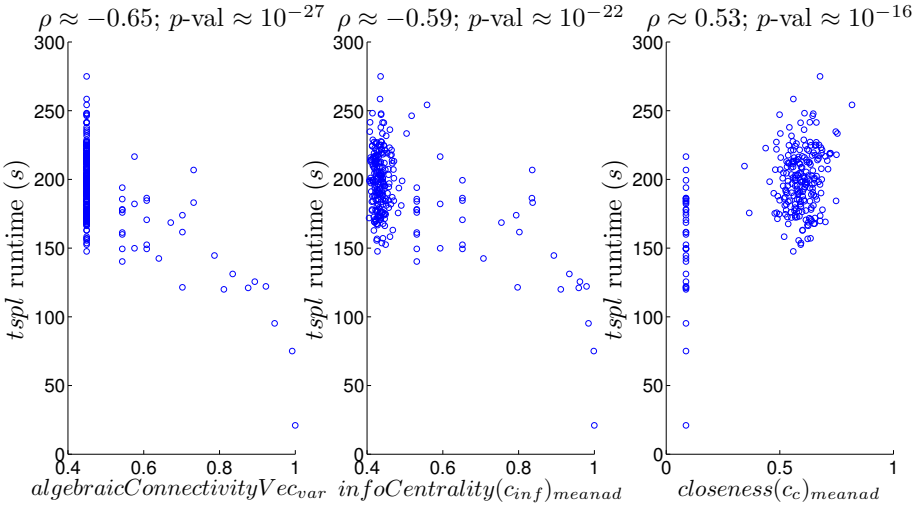
3.4.3 Effect of network density

We also generated another set of 250 synthetic networks using the Lancichinetti *et al.* community detection benchmark model [159] (see Section 1.1.6.5). For these networks, we fixed the number of nodes and mean degree but varied the other model parameters. On this more restricted set, in which all networks have about the same density of links, the correlations between TSP solution length or runtime (using the cross-entropy algorithm) and other network features become weaker (see Figure 3.9), suggesting that density variation is a significant factor in producing graph structures with varying complexity[9], with respect to this algorithm (the linear correlation of network density for the PAP data set is 0.78 with solution length and 0.83 with runtime). However, density is not the only relevant factor; for instance, Figure 3.9(a) shows that the mean geodesic distance between node pairs is still a substantial predictor of TSP solution length, even with density held fixed: networks that are better connected (have lower mean geodesic distance) tend to have shorter TSP solutions.

[9]This is probably because constraining the density also limits the scope for variation in many other aspects of graph structure. This is why many network features are found to co-vary substantially with density, as can be seen from its very strong correlation with the Isomap dimension capturing over 90% of the total variance in the low-dimensional mapping shown in Figure 3.2(a).



(a)



(b)

Figure 3.9: **Network feature correlations with cross-entropy TSP solver, when density is fixed.**

(a) Examples of features that display high correlation with solution length over a data set of 250 networks randomly generated using the Lancichinetti et al. benchmark [159]. (b) Example features highly correlated to TSP runtime. See Appendix A for descriptions of the feature names.

3.5 Phylogeny regression

We now consider some biological networks and demonstrate how regression against network features can also be used to find structural diagnostics that are potentially functionally relevant. Networks are often used to model aspects of biological organisms, such as their metabolic pathways (sequences of biochemical reactions involved in metabolism); in this context, it is natural for a comparative study of such networks to use, as a benchmark, evolutionary relationships between the corresponding species, which can be represented by means of a phylogenetic tree. We would like to ask the following question: are there aspects of network structure that appear to have been constrained by biological evolution, such that they show a pattern of correlated evolution with the phylogeny (also known as a *phylogenetic signal*)? An observation of such characteristics might help to motivate more realistic models for network evolution in a given context by placing constraints on the conservation and drift rates of different sorts of network properties.

Here we examine this using ideas from the area of the phylogenetic comparative method [86, 169, 175]: one can make the very strong (but common in evolutionary studies) assumption of a certain stochastic process (e.g., Brownian motion) underlying the variation in network characteristics along the branches of a phylogeny and then estimate the extent to which different characteristics have been evolutionarily conserved and are constrained by the structure of the phylogeny. We depict this in Figure 3.10) which shows the simulation of a random drift or Brownian motion process on a toy phylogeny created by us. We suppose that there is some time-varying characteristic or trait (let $z(t)$ denote its value at time t), which has a value of 0 at the root of the phylogeny (i.e., $z(0) = 0$). Subsequently, we allow the trait to evolve in discrete time along the tree according to a one-dimensional Brownian motion process with *drift parameter* β; this means that we have $z(t+1) = z(t) + \omega$, where $\omega \sim \mathcal{N}(0, \beta)$, the normal distribution with mean 0 and variance β. At each branching

point in the tree, the process splits into two processes which then begin to evolve independently. The final trait values at the leaves are depicted via the colour bar in Figure 3.10. The utility of this model is that leaves (species) which split more recently are likely to exhibit closer trait values; thus it simulates a trait that co-evolves with the phylogeny, i.e., a phylogenetic signal.

3.5.1 Data

We obtained a set of nearly $1,000$ metabolic networks (a total of 620 of which are used here), represented as networks of interacting pathways (NIPs), from Aurélien Mazurie [178]. These are networks in which each node represents an entire metabolic pathway, and two nodes are linked if the corresponding pathways have shared metabolites[10], with link weight equal to the number of metabolites shared.[11] Obtaining an independent high-quality phylogeny for this range of species is difficult, so we chose to use the Tree of Life [170], which provides an unweighted evolutionary tree for a large number of organisms. For a subset of these, a weighted Tree of Life is also available [61, 164], with the weights or lengths of the branches corresponding to estimated evolution times between species (as in Figure 3.10). In the unweighted tree, there is no information about evolution times; in effect all branches are taken to be the same length, clearly an incorrect and very simplistic assumption. Thus, we will use the unweighted tree only for purposes of comparison, as a sort of negative control.

In the unweighted Tree of Life, we found matches for 158 of the genuses for which we had at least one NIP; the total number of networks matched was 450 (networks were grouped at the genus level in order to have multiple samples for each instance and make the model fit less susceptible to noise in the data). In the weighted Tree of

[10]Only water was excluded; other widely-occurring metabolites, such as adenosine triphosphate (ATP), were included by Mazurie *et al.* [178] as these were found to provide useful structural information [177]. None of the metabolites was overly ubiquitous; all the networks used had a link density of below 60%.

[11]As mentioned in Section 1.1.4, many of our network diagnostics are defined for unweighted networks and thus link weights are ignored when computing these.

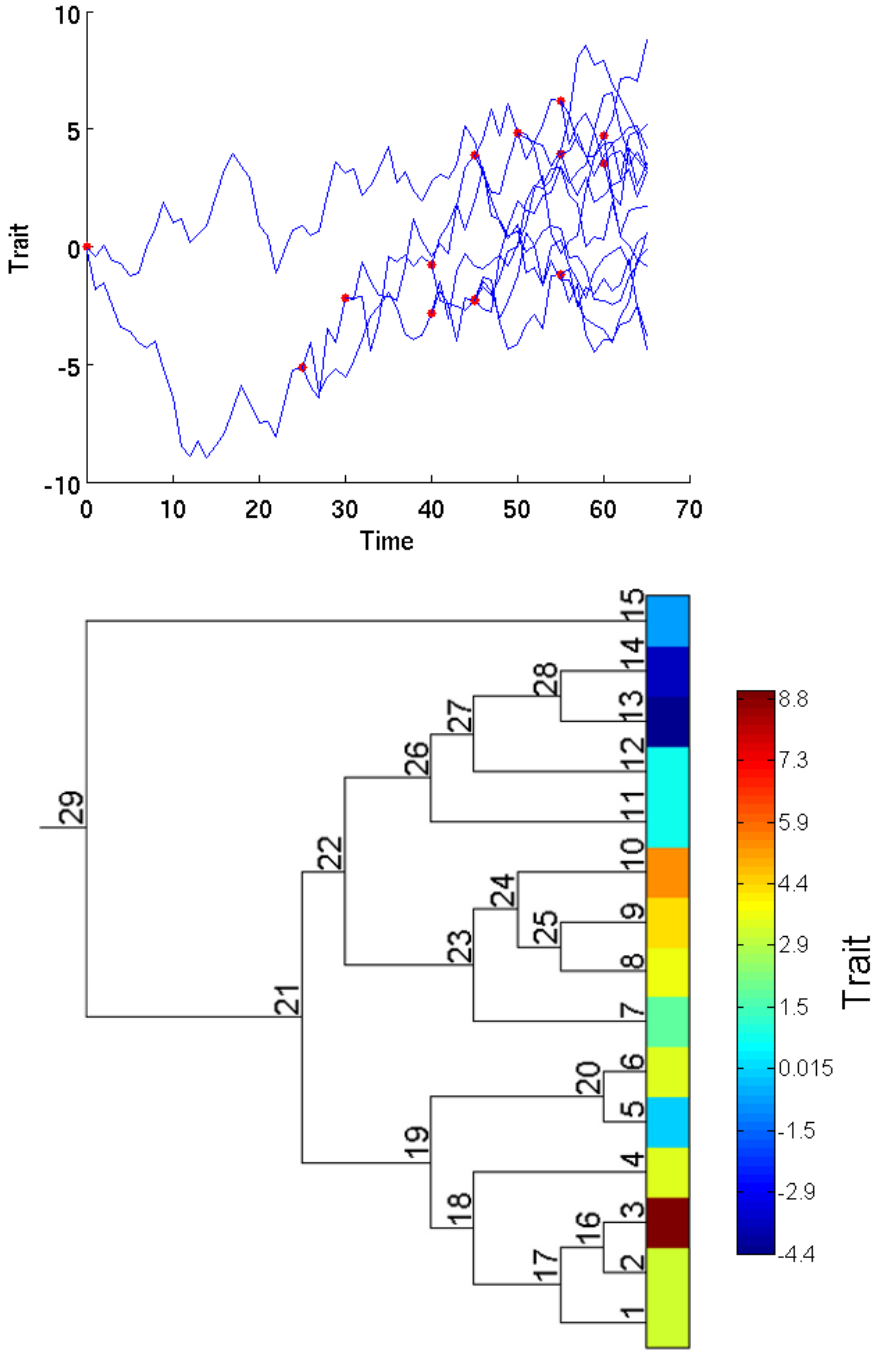


Figure 3.10: **Example of Brownian motion process on a toy phylogeny.**

The upper plot depicts how the trait value evolves with time (with drift parameter $\beta = 1$); red dots depict branching points. The lower plot shows the underlying tree structure, with the colours at the leaves depicting the final trait values. The relevant property of this model is that species that branched off from each other more recently have had less time to diverge and are thus more likely to exhibit similar trait values.

Life, we found 145 matching species to our set of metabolic NIPs, with a total of 341 networks being matched (for this tree, we do not group networks at the genus level, because there are fewer species and also because the weighted branching structures below genus level are more informative).

3.5.2 Model fitting

In order to fit the data to the standard random drift or Brownian motion model of trait evolution to try and detect phylogenetic signals in network features, we do need the branch lengths on the tree—i.e., the lengths of time between different speciation events. We used the unweighted tree for purposes of comparison, assigning equal lengths to all branches (and thus discarding all information about actual branching times). Comparing how well the model fits the data on this unrealistic tree with the fit on an actual weighted tree with meaningful branch lengths will help provide an indication of the strength of the phylogenetic signal in the observed network traits.

To fit the model we would first like to compute the differences in the value of a particular feature or trait between pairs of species in the phylogeny. However, if we just take the differences between the observed species (i.e., the leaves of the phylogeny), then these are not all independent; for instance, in Figure 3.10, the difference between the trait values for species 1 and species 2 is not independent of the difference between species 1 and species 3, because the paths between these pairs of species in the tree are common between species 1 and species 16. Thus, we use the method of *independent contrasts* [86], which instead considers pairs including both current and ancestral species, such that all pairwise differences (contrasts) in trait values are independent of each other. For the tree in Figure 3.10 this proceeds as follows: we start from the root (species 29), and the first contrast computed is between its two children, i.e., species 21 and species 15 (of course, we have not observed the trait value at internal nodes like 21; these have to be estimated, as described below).

The next contrast taken is that between the two children of species 21: species 19 and species 22. This is followed by the pair of children of species 19: species 18 and species 20. We proceed in this way, at each step choosing the children of a node from a previous step, until we are down to all of the leaf nodes. Because the number of internal nodes is one less than the number of leaf nodes, we obtain precisely $l - 1$ independent contrasts for a phylogeny with l leaves.

We then fit a linear regression model to these contrasts and the corresponding divergence times [175]:

$$\mathbf{V} = \beta\mathbf{t} + \epsilon. \tag{3.2}$$

This is a random drift model: $\mathbf{V}$ is the vector of contrasts for a given feature/trait, $\mathbf{t}$ is the vector of mean times for which the respective species pairs have evolved mutually independently, β is the model parameter representing the rate of evolutionary drift, and ϵ is the vector of noise terms or residuals. Thus, for each feature, we independently obtain a fitted estimate of β, and ϵ gives a measure of how far the observed trait values deviate from model predictions. We can take the sum of squares of these residuals to get the deviance, a standard measure of goodness-of-fit (see Section 1.3.2.2). Lower deviances can be taken to indicate stronger phylogenetic signals.

In practice, computing the independent contrasts requires knowing the β parameter (which is what one wants to estimate), as this has to be used to estimate the trait values at the internal nodes (i.e., the unobserved ancestral species). Thus, we use an iterative procedure described by Martins [175]. The essential idea behind it is to obtain initial estimates of the ancestral trait values by taking an unweighted average of the two daughter species:

$$z_a = \frac{z_i + z_j}{2}. \tag{3.3}$$

Here z_a denotes the trait value at an ancestral species a, and i and j are the daughter species of species a. For instance, for the phylogeny in Figure 3.10, we would have

$z_{16} = (z_2 + z_3)/2$ (where z_2 and z_3 are the measured trait values for species 2 and 3). We can then propagate this value of z_{16} to obtain estimates further up the tree: $z_{17} = (z_1 + z_{16})/2$, and so on. Having thus obtained an initial value for all nodes in the phylogeny, we compute the independent contrasts and fit the regression model of Equation (3.2) to obtain an initial estimate of β. This is then used to obtain updated estimates of internal node values using an equation of the following form [86, 175]:

$$z_a = \frac{z_i/(\beta t_i) + z_j/(\beta t_j)}{1/(\beta t_i) + 1/(\beta t_j)}. \tag{3.4}$$

Here t_i is one half of the time interval between species i and species a. Having obtained these updated estimates for the ancestral trait values, the whole process is then iterated, with the values for β and the trait being alternately re-estimated. This is continued until the values have converged to a steady state.[12] The algorithm is not guaranteed to converge, but we did not face convergence problems in any of our computations, and in general convergence is usually quite rapid, occurring within 20 iterations [175].

To assess the extent to which our simplistic model is able to explain the variation in the individual network features across species, we create two additional models to compare against. One is what we refer to as the *positive null model* (i.e., it represents a situation that should be observed if the features have indeed co-evolved with the phylogeny): we run a Brownian motion process forwards on the tree starting from the root (and having chosen a particular value of β). This gives a set of values at the leaves, which we then normalise (using a logistic function, so that they are comparable with our actual features, which have been normalised in the same way) and fit to the model using the procedure described above. By doing this many times for a range of β values (we chose 10 different values of β, evenly spaced on a logarithmic scale

[12] We use a convergence criterion of change in the deviance being less than 0.001% from one iteration to the next.

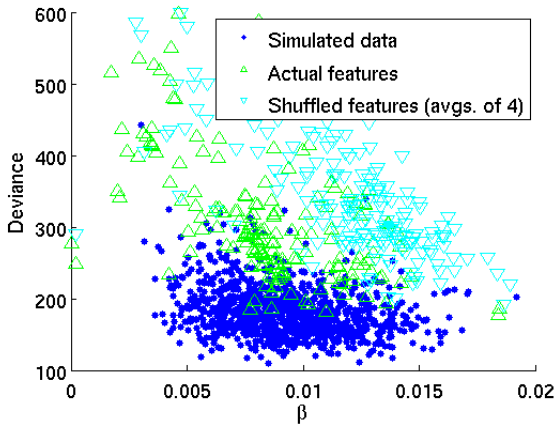
running from $2^0 \times 10^{-3}$ to $2^9 \times 10^{-3}$, and ran 100 simulations for each value), we obtain an estimate of the range in which our fitted β and deviance values lie when the data has actually been generated via the process we are assuming in the model.[13]

The other model is what we call the *negative null model* (i.e., it represents a situation in which there is no phylogenetic signal in the feature values): we shuffle our actual values for any given network feature, so that each value is re-assigned to a randomly chosen species, and then fit the model to this shuffled version of the data. If there is any phylogenetic signal in the actual features that we are measuring, then we would expect it to be lost on shuffling. We do 4 shuffles for each of our network features[14] and average the β and deviance values obtained over the 4 shuffled fits.

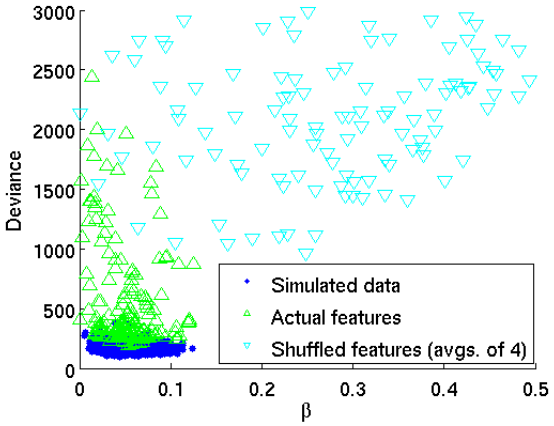
We fit the Brownian motion model to 222 different network features (some diagnostics had to be excluded because they could not be feasibly computed or were not defined for all networks, as discussed in Section 3.2), and each one was fitted separately. The values of the features were normalised to lie between 0 and 1 via the logistic function, as described earlier. In Figure 3.11, we show the results of the model fitting procedure to the actual data and to the two null models just described. The two plots correspond to the two versions of the phylogeny mentioned earlier: unweighted and weighted. We see that there is no clear separation between the fits to the actual and shuffled versions of the data in the unweighted case; but a much stronger signal is apparent in the data in the weighted case, where the range of values observed for β and the deviance also overlap substantially with those from the positive null model (simulated data). The unweighted phylogeny can be regarded as extremely

[13]Note that the β values recovered from the fitting are not the same as those originally used to generate the data, due to the intermediate normalisation step involved (we checked that the original values were indeed accurately recovered if no normalisation was done). However the same normalisation is used for the actual data and the null models, so that the fitted values obtained are comparable.

[14]We chose to do no more than 4 shuffles per feature due to time constraints; as we are examining a large number of features, in sum we obtain a sufficiently large number of samples from the shuffled feature distribution to be able to demonstrate a significant difference between the shuffled and actual features, as discussed shortly.



(a) Unweighted phylogeny



(b) Weighted phylogeny

Figure 3.11: **Phylogenetic signal in networks of interacting metabolic pathways.**

Plots of β versus deviance for Brownian motion model fits to network features on the two phylogenies. Green triangles: actual data; Cyan inverted triangles: negative null model—i.e., shuffled data (each point is an average over fits to 4 independent shuffles of a single feature); Blue dots: positive null model—i.e., simulated data. Fits with exceptionally high values of β or deviance (i.e., those lying outside the range of the axes) have been excluded to make the plots easier to view (for the unweighted phylogeny, 32 of 222 actual feature fits and 34 of the mean shuffled feature fits are not shown; for the weighted phylogeny, 8 actual feature fits and 117 mean shuffled ones are not shown). It is evident that for the appropriate weighted phylogeny, most actual network characteristics are much better fit by the model than would be expected at random, indicating a significant (see main text for quantification and discussion of this) phylogenetic signal in the structural properties of these networks.

naïve (it makes no attempt to estimate branch lengths in a meaningful way), whilst the weighted phylogeny is much more realistic as it contains actual information about estimated divergence times. Thus the presence of a significant phylogenetic signal on the more meaningful tree suggests that many of these network features are correlated with aspects of biological function which are relevant for evolution.

In an attempt to quantify the significance of the signal, we compared the distributions of the deviance values for the actual features and their shuffled versions; because deviance is a measure of how well the model fits the data, significantly lower deviances for the actual features would indicate that they are better explained by the model than could be expected at random. For the unweighted phylogeny, the mean and standard deviation of the deviances are 302.75 ± 81.69 for the actual features depicted in Figure 3.11(a) and 495.56 ± 3146.7[15] for their shuffled versions (4 shuffles each). We compared the two distributions using a Mann-Whitney U test, a non-parametric test for assessing whether the values in one sample tend to be larger than in another. This gives a p-value of 9.59×10^{-8}, for the null hypothesis that both samples are from identical distributions. For the weighted phylogeny, the mean and standard deviation of the deviances are 522.20 ± 434.89 for the actual features depicted in Figure 3.11(b), and 2949.4 ± 2204.1 for their shuffled versions. The Mann-Whitney U test gives a p-value of 2.91×10^{-91}. Whilst these p-values cannot be taken as directly meaningful, because the different network features are not independent, they nevertheless provide an indication of how much more significant the feature signals on the weighted phylogeny are, complementing the visualisations in Figure 3.11. Thus this is further evidence that many aspects of the structure of these metabolic pathway networks are biologically significant.

We can also examine the deviance values for the fits obtained for individual network features to see which particular network characteristics are most correlated with

[15]The large standard deviation is due to a few extreme values.

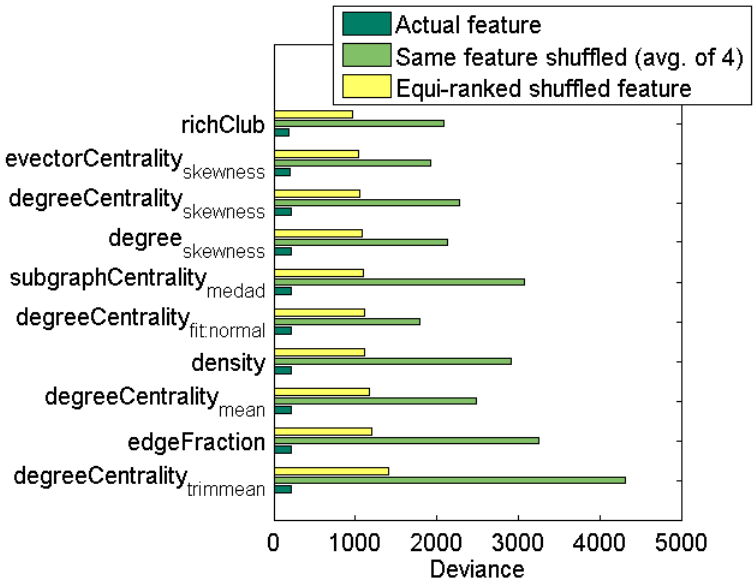


Figure 3.12: **Network features with the strongest phylogenetic signals.**

We show the lowest deviances from the Brownian motion fit on the weighted phylogeny. Dark green (bottom) bars are deviance values for actual feature data; light green (middle) bars are averaged deviances over 4 fits to random shuffles of that feature's values amongst the tree leaves; and yellow (top) bars are the equi-ranked (i.e., 10 lowest, in order) mean deviances obtained amongst fits to shuffled versions of all 222 network features examined.

a given phylogeny under our model. Figure 3.12 shows the 10 features with the lowest deviances for the weighted phylogeny. For comparison, we also show the deviances obtained after shuffling the values of each those features (average of fits to 4 shuffles, depicted by the light green middle bars) and, as a more stringent null model, also the 10 lowest deviances obtained amongst mean fits to the shuffled versions of all network features considered (yellow top bars). We note again that the actual feature data is significantly better fit than the null models. In particular, the rich-club coefficient, which is a measure of the fraction of a network's nodes (ordered by decreasing degree) which show high mutual connectivity (see Section 1.1.4.1 for the precise definition) is found to be most strongly correlated with the weighted phylogeny; see Figure 3.13. This suggests that the proportionate size of the 'rich club' of hubs, or high-degree metabolic pathways in the NIPs, might be a significant differentiating factor between different types of species. The rich-club coefficient was originally proposed in the context of social and Internet networks [64, 285], and to our knowledge has not previously been examined for such metabolic pathway networks. Thus, whilst this particular feature is not unique in exhibiting a strong phylogenetic signal, and is in fact strongly correlated for this data with other simpler features that exhibit similar signals, such as density (see Figure 3.12), it does provide an example of how our high-throughput approach can highlight the potential relevance of particular aspects of network structure in contexts where they would have been unlikely to crop up otherwise.

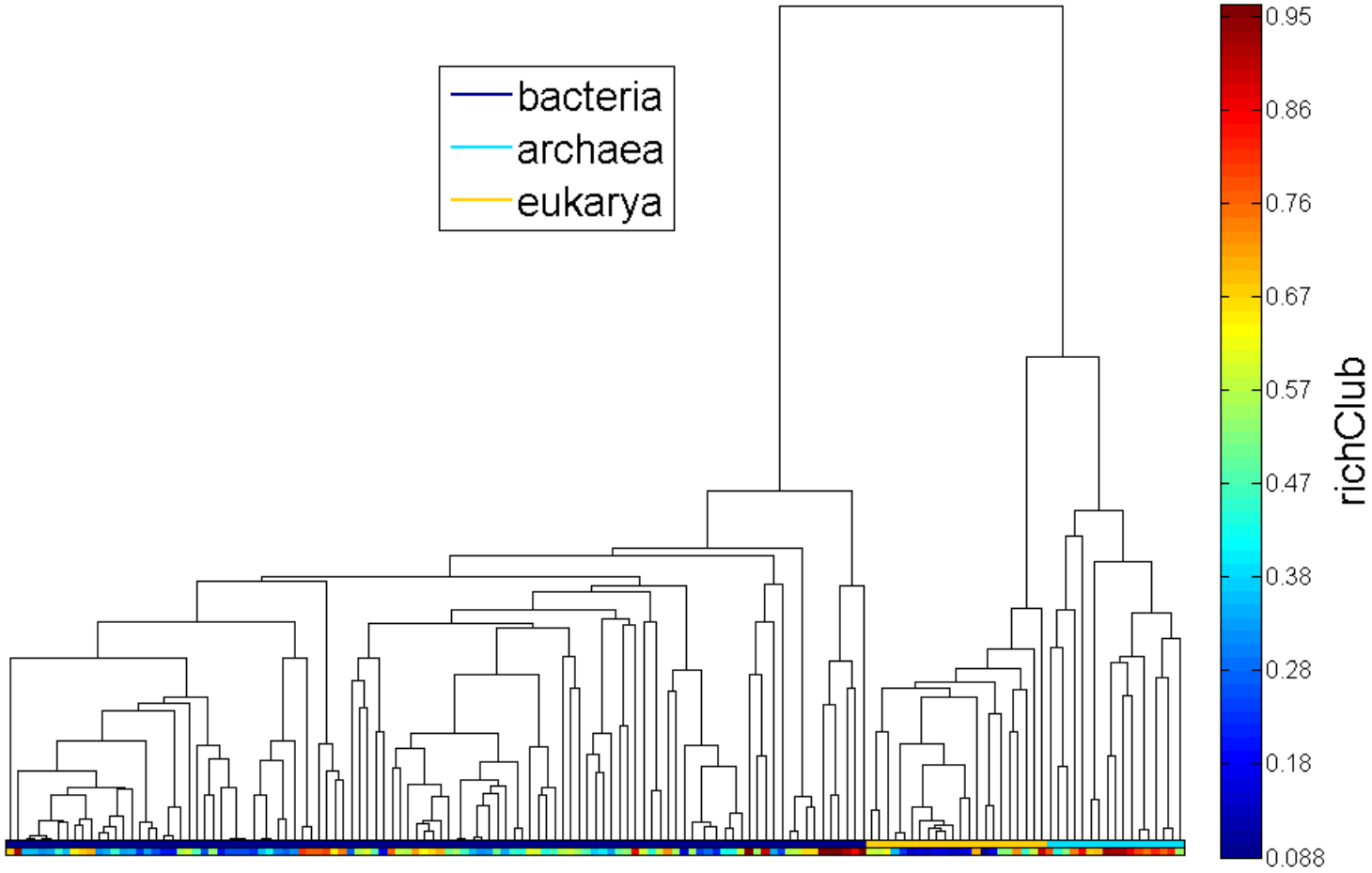


Figure 3.13: **Variation of rich-club coefficient in networks of interacting pathways (NIPs), across the weighted Tree of Life.**

Upper colour bar depicts the type of organism at each leaf; lower colour bar depicts the observed rich-club coefficient of the network(s) corresponding to that organism in the NIPs data set [178]. Visualisation produced using a modified version of MATLAB code originally written by Dan Fenn.

To test our method on a more specific and higher-confidence phylogeny, we repeated this procedure for a set of more detailed metabolic networks from 17 bacterial species belonging to the genus *Pseudomonas*; this data set contains 6 networks for each species, corresponding to 6 distinct metabolic pathways [188]. In this case, too, we obtained a better fit with actual network features compared to the negative null model, suggesting that aspects of the structure of these detailed metabolic networks are also relevant to biological function, though the much smaller number of data points makes it harder to get a strong signal (see Figure 3.14). We also find that the largest values of the drift parameter β are obtained predominantly for features corresponding to 2 of the 6 pathways: lysine degradation (4 of the top 10) and phenylalanine metabolism (3 of the top 10). This is in accord with the much more detailed evolutionary model fit by Mithani *et al.* [188], which suggested a higher probability of random rewiring for these 2 pathways relative to the others; as noted there, this appears to be in agreement with experimental data suggesting that some of the *Pseudomonas* bacteria are losing their ability to assimilate lysine and phenylalanine.

We show the best-fitted features in Figure 3.15. Once again real feature fits are substantially better than the shuffled ones. Summary statistics of the distributions of certain node centralities such as subgraph centrality and eigenvector centrality (see Section 1.1.4.2), alongside measures like modularity and the rich-club coefficient mentioned earlier, show up prominently in the set of best-fit features. This suggests that the corresponding aspects of the structure of those particular *Pseudomonas* metabolic pathways are biologically interesting and may be used to guide more detailed investigation of the correspondence between structure and function in these networks.

One limitation of our methodology is that it is generally able to indicate only associations, not causal connections. It also involves the use of a random drift model which cannot capture actual biological evolution. However, the associations thrown up may serve as starting points for motivating more realistic evolutionary mechanisms.

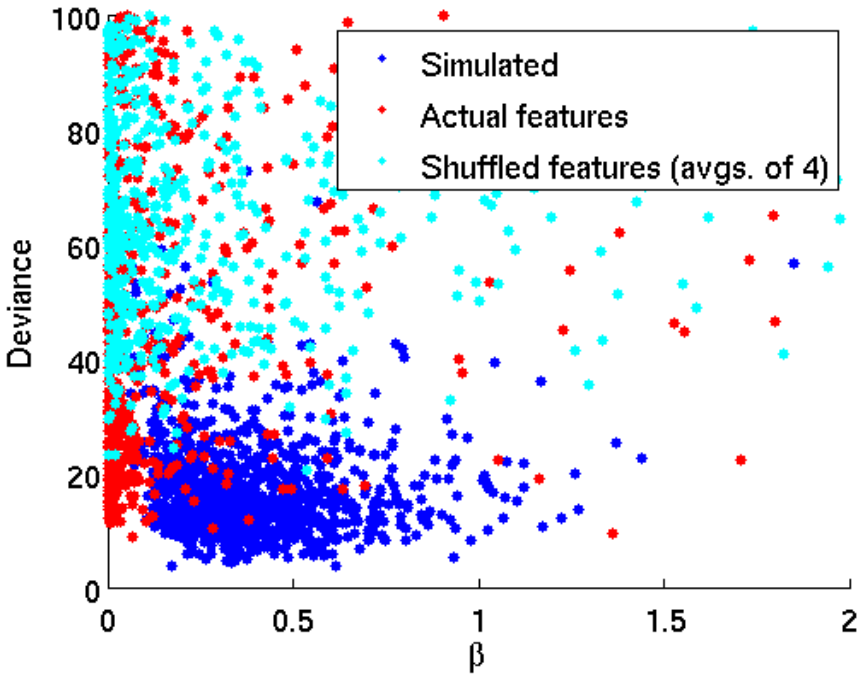


Figure 3.14: **Phylogenetic signal in** *Pseudomonas* **metabolic networks.**

Plot of β versus deviance for Brownian motion model fits to metabolic network features on the *Pseudomonas* phylogeny [188]. Red: actual data (fits to 804 features, distributed over 6 pathways—we obtained 6 separate networks for each species from Aziz Mithani); Cyan: negative null model—i.e., shuffled data (804 mean fits—each point is an average over fits to 4 independent shuffles of a single feature); Blue: positive null model—i.e., simulated data (1000 fits). Fits with values of β or deviance lying outside the axis ranges have been excluded to make the plots easier to view (240 of the 804 actual feature fits and 279 of the mean shuffled feature fits are not shown).

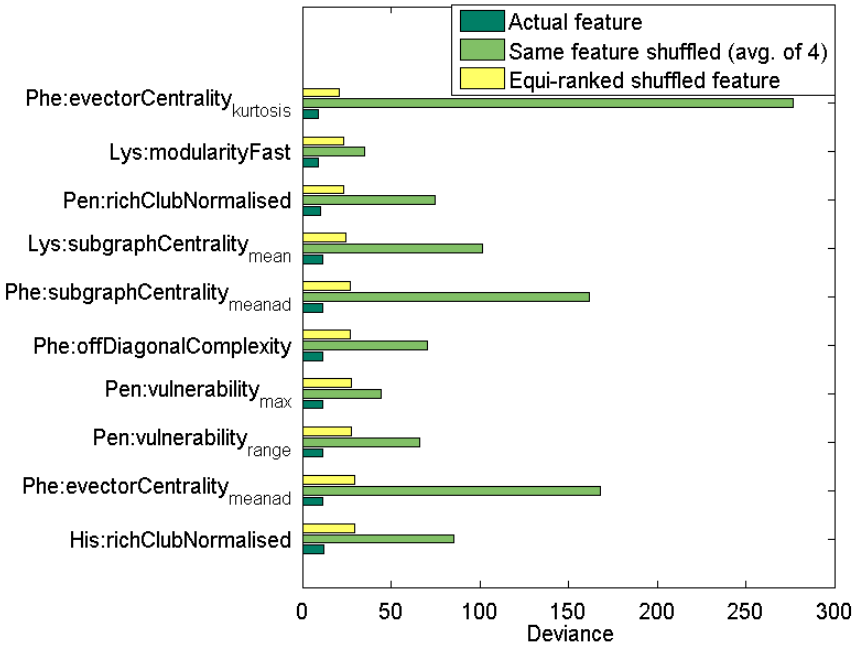


Figure 3.15: *Pseudomonas* **metabolic network features with the strongest phylogenetic signals.**

We show the lowest deviances from the Brownian motion fit on the *Pseudomonas* phylogeny [188]. Dark green bars are deviance values for actual feature data; light green bars are averaged deviances over 4 fits to random shuffles of that feature's values amongst the tree leaves; and yellow bars are the equi-ranked (i.e., 10 lowest, in order) mean deviances obtained amongst fits to shuffled versions of all 808 features examined. Features come from networks of 6 different metabolic pathways [188]; the 3-letter prefix in each feature label denotes the pathway it relates to (Pen: pentose phosphate pathway, Lys: lysine degradation, His: histidine metabolism, Phe: phenylalanine metabolism).

For instance, one may simulate models of proposed mechanisms such as duplication-divergence (see Section 1.1.6.7), and examine whether particular network features such as the rich-club coefficient tend to co-evolve with those mechanisms, and what particular parameter settings or other tweaks in such a model might lead to behaviour consistent with the phylogenetic signals we observe. If one wishes to reproduce the conservation of rich-club behaviour in NIPs for instance, then something like a model where links between high-degree nodes have a lower probability of being lost may be suggested. Whilst it is true that our methodology, due to its operating at a high level of abstraction[16], leads only to quite preliminary suggestions and hypotheses of this sort, we believe these can help to guide more focused investigation in the right directions and also sometimes illuminate cross-disciplinary connections that might have been difficult to arrive at otherwise. In Chapter 5, we provide a further example of how our feature-based network representation can be utilised to fit generative models to biological networks and obtain some insight into evolutionary mechanisms.

3.6 Discussion

In this chapter, we have laid out a data-driven approach to organising and utilising many different ways of characterising networks. Our framework can serve as a general-purpose tool for exploratory investigation of networks, and can suggest which aspects of network structure are relevant in a given context. The approach we take is, in a sense, complementary to standard perspectives in network science. When a new diagnostic for studying networks is proposed, it is typically motivated by the need to capture a particular structural aspect or by a desire for more efficient computation relative to existing methods. Here, we instead seek to apply a large and wide-ranging

[16]Networks themselves represent abstractions of real-world systems where much information has been discarded; and our mapping of networks to a feature vector representation involves a further level of abstraction. Whilst this involves the loss of a lot of detail, it is also what enables us to examine and compare such a large number of objects simultaneously.

set of diagnostics to a set of networks, and allow the resulting information (the design matrix) to direct our attention to aspects of network structure that are of interest in a given context.

We demonstrate via examples ways in which our methodology can be applied to the study of networks, in particular for the inference of relationships between structural properties and functional outcomes: detecting phylogenetic signals in biological networks, and detecting structural features of graphs that can be used to estimate the hardness of computational problems defined on them. In each case, we show that the approach adopted here can highlight connections between network structure and functionality that might have been hard to identify by conventional means, which typically involve studying a small number of networks and examining a small number of diagnostics. For instance, we find that the rich-club coefficient of the metabolic pathway interaction networks we study appears to capture a structural aspect which is evolutionarily significant; given that the rich-club phenomenon was originally posited for social networks, this diagnostic may not normally have been used to characterise metabolic networks. As discussed in the previous section, such an observation may help to motivate certain kinds of evolutionary mechanisms, which can then be modelled and simulated more rigorously to examine how well they reproduce trends in the evolution of particular network features that have been noted here. Similarly, the finding that network features like the maximum node betweenness and average cyclic coefficient (see Section 1.1.4.3) are strongly correlated with the TSP solution length across a range of solvers, at least for a class of networks generated via preferential attachment, would be hard to obtain without a large-scale study of this sort.

As mentioned in the previous section, our methodology operates at a high level of abstraction and as a consequence of this, the conclusions we obtain tend to be preliminary and suggestive in nature. Another limitation of the work described here is the choice of specific data sets and generative models for the networks we examine. The

choice of the set of network diagnostics, whilst covering a fairly wide range of the existing literature, is also necessarily subjective and non-comprehensive. Thus, though the high-throughput approach enables the simultaneous investigation of a larger number of networks and diagnostics than has been attempted previously, it is important to keep in mind the specific choices of these that have been made when interpreting any of our results. In our case studies here and in the subsequent chapters, we attempt to study the effects of varying at least some of these choices, but given more time one could build further on the sets of networks and diagnostics compiled here. Overall, keeping in mind these caveats, we believe our examples demonstrate how the approach adopted here might serve to aid tasks involving the characterisation and comparison of networks in a variety of settings, by uncovering in a semi-automated fashion connections and aspects of network structure which can serve as pointers for more detailed and rigorous human investigation.

We have shown here how a simplistic evolutionary model (random drift) can be used to identify network characteristics that correlate with some known functional or phenotypic property. To take this further, in many instances one would like to develop realistic models of network evolution as a whole: models that generate synthetic networks which look as similar as possible to the real network. Thus, it is desirable to be able to compare and match models to data, in order to quantify the structural complexity of observed real-world networks and identify generative mechanisms that reproduce it. In the following two chapters, we show how our high-throughput approach, combined with existing ideas from statistical physics and Bayesian inference, can be used to develop an efficient methodology for doing this.

Chapter 4

Feature Degeneracies and Network Entropies

In this chapter, we examine patterns of correlations between different network features and how these correlations vary depending on the set of networks being analysed. We attempt to relate this to notions of network entropy. This chapter is exploratory in nature, and might normally be placed at the end; however we have put it here as some of the ideas described here help to set the scene for the next chapter. Part of the work presented in this chapter will be included in a manuscript currently in preparation [16].

4.1 Background

In Chapter 3, we described how our library of network diagnostics allows us to obtain a feature vector representation of a given network, which in effect maps it to a high-dimensional vector space. Our focus thus far has been on looking at how different dimensions of this space relate to functional characteristics of certain kinds of networks, and we have demonstrated with specific case studies how regression on structural features can help identify functionally relevant aspects of network structure.

However, one can also examine the correlations within the set of features themselves, rather than regressing against an external variable. We have already seen in Figures 3.1 and 3.5 that there are substantial correlations amongst the features we are considering; clearly the dimensions of our vector space are not all measuring independent quantities.

One can discern two separate causes for the correlations we observe between network features. The first is that for a given pair of features, there might be a general relationship between the aspects of network structure they are measuring: for instance, the density of links is related to nearly every other structural diagnostic, as networks that either have very few or very many links are much less likely to show complex structure than networks with intermediate densities. Such relations should be observed for nearly any set of networks one might examine, as long as it contains sufficient variation in the characteristics being examined to allow for correlations to be discernible. The second type of cause could be a relationship that is imposed by the constraints particular to a specific family of networks (or to the observations one is able to make from that family). For instance, in Watts-Strogatz small-world networks (Section 1.1.6.4), if the re-wiring is low then the clustering coefficient [see Equation (1.19)] will be largely determined by the node degrees, something which is not the case in general. Thus, correlations of this second type can inform us about the nature of the structural constraints imposed on a particular family of networks, and can be seen as an indicator of how 'complex' a given kind of network is, if we take more constrained structure to generally correspond to greater complexity, as per Bianconi [43]. In this chapter, we make these ideas more precise, and also relate them to notions of quantifying network complexity via thermodynamic entropy in statistical physics [22–24, 43–45]. We demonstrate how our vector space representation can be used to define a novel notion of network entropy, and explore its relation to the thermodynamic one.

4.2 Network feature degeneracies

We start by observing how different network features correlate across different sets of real-world networks. Figure 4.1 depicts the absolute values (i.e., magnitudes) of linear correlations between pairs of features across multiple network data sets: here we use only the interaction networks contained in the set of 192 networks used previously (Appendix B), leaving out the similarity networks as many features that capture topology, or unweighted structure, are constant for these fully connected networks, so many of the feature-feature correlations are not meaningful for these. There are a total of 137 such interaction networks in 9 categories; the figure depicts correlations across the entire set as well as for the subsets thereof corresponding to Facebook, brain, and protein interaction networks. These plots all depict a reduced set of 53 features, to aid visualisation. We note that both the Facebook and the brain networks (see Appendix B for details) also show a much higher level of correlation on average between these features, an observation replicated across our full set of network features (see Table 4.1 and the discussion below). The protein interaction networks too show more feature pairs with high correlations than the full set of 137, though the difference is less marked than for the Facebook and brain networks, which is in accord with earlier observations that this particular set of networks is structurally incoherent (Ref. [203] and Section 3.3.2) and that these data sets in general have high levels of noise and unreliability [26, 27, 53, 119, 234, 237, 267] (see Section 2.5). However, on the whole, these enhanced correlations suggest the kinds of type-specific constraints on structure mentioned earlier; they also tie in with Figure 3.2, which shows that several of the different kinds of networks appear to lie in restricted regions of our feature space. Table 4.1 shows the mean and median pairwise feature correlations across a larger set of features, for all 9 categories of interaction networks. This table again shows that for each of the 9 types, features on average correlate substantially more than they do for the full set of networks. The two categories for which the difference is the least are

Table 4.1: **Mean and median feature correlations across different network types.**

Category (samples)	Mean absolute correlation	Median absolute correlation
All interaction networks (137)	0.30	0.26
Brain (12)	0.59	0.66
Collaboration (8)	0.49	0.50
Facebook (15)	0.60	0.65
Fungal (12)	0.45	0.43
Language (8)	0.54	0.57
Metabolic (15)	0.48	0.45
Political committee (16)	0.54	0.57
Protein interaction (25)	0.38	0.33
Social (26)	0.37	0.32

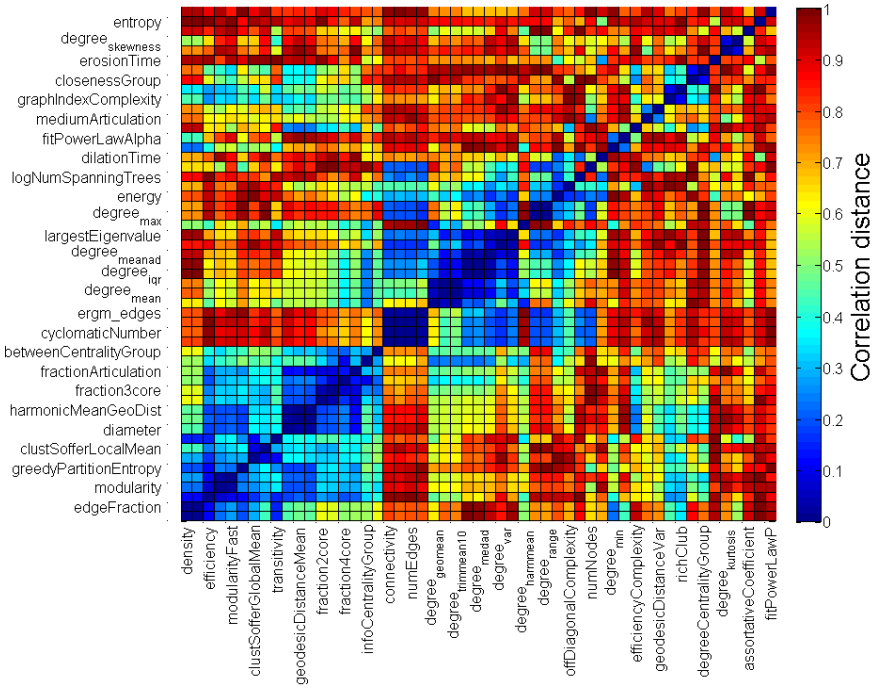
We computed pairwise correlation coefficients for each category of networks amongst a set of 281 features (we chose this as the largest set such that each feature had at least one counterpart with which a meaningful correlation could be computed over every category; the total number of such feature pairs amounted to 38,410). Absolute values of the coefficients were taken and here we report their mean and median values over each set of networks.

protein interaction networks, as noted above, and what were labelled by Onnela *et al.* [203] as 'social' networks. This suggests that the latter type are also structurally diverse (a similar observation was made in Ref. [203]); this is unsurprising, as the category label of 'social' is rather broad and includes many different types of data (see Appendix B).

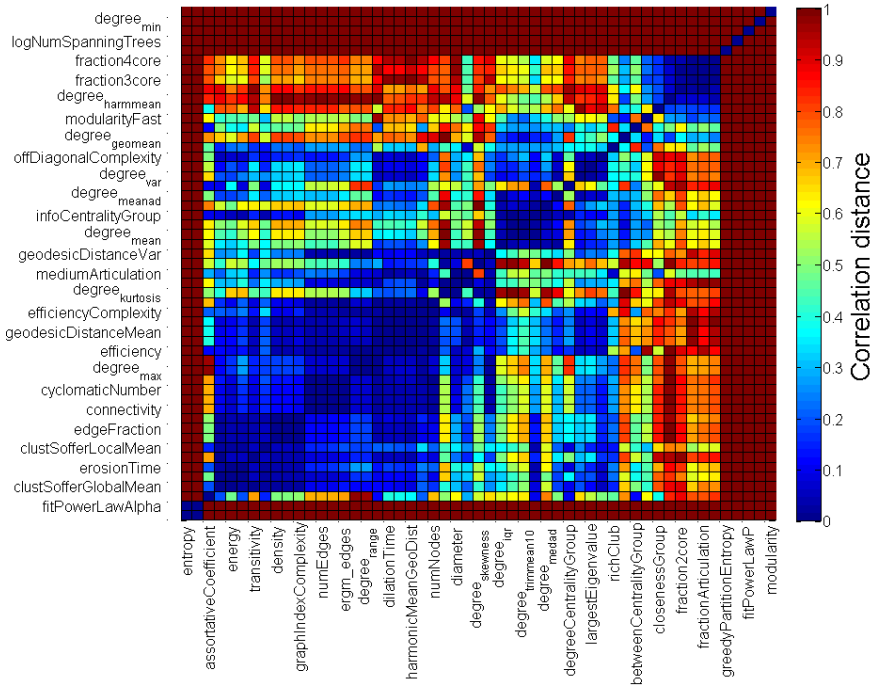
4.2.1 Granular contact networks

In order to examine in some more detail the phenomenon of feature degeneracies and how it might relate to structural constraints on certain kinds of networks, we chose to look at a specific example of networks that are spatially constrained: *granular contact networks*, which represent contacts between particles in granular materials [32]. These are spatially-embedded networks, with the nodes or particles having definite locations in two-dimensional Euclidean space. It has been shown that the topology of these networks has a significant influence on the propagation of sound through the materials in question [32].

Here we are interested in these networks as an example with definite constraints on structure, which can be compared against random graphs corresponding to null



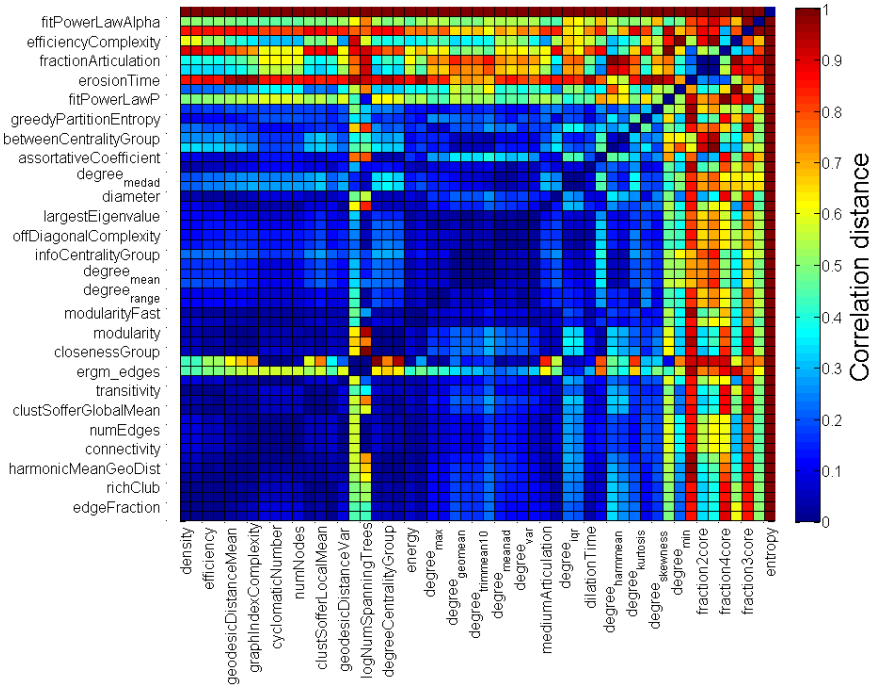
(a) 137 interaction networks



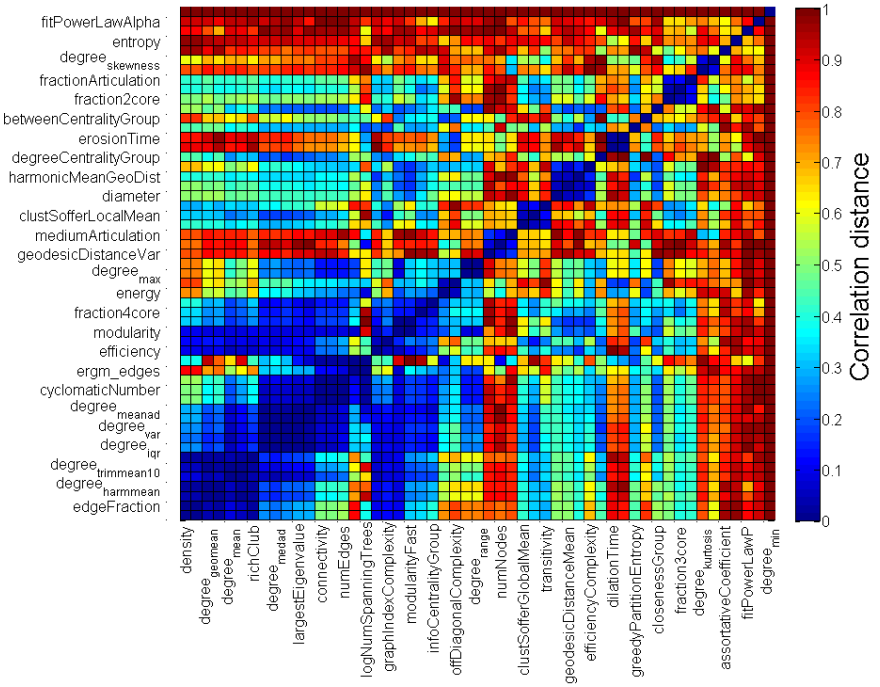
(b) 15 Facebook networks

Figure 4.1: **Features are more degenerate on restricted sets of networks.**

53×53 feature distance matrices; the correlation distance is $1 - |\rho|$. Features are alternately labeled on the x/y axes, for legibility, and in each case are ordered via



(c) 12 brain networks



(d) 25 protein interaction networks

Figure 4.1 (continued).

single-linkage clustering (see Section 1.3.3), so as to group similar features together (feature details in Appendix A). We illustrate the increased degeneracy using Facebook, brain, and protein interaction networks (see Appendix B for details).

models. For networks embedded in 2-D space, a natural null model is random geometric graphs (RGGs; Section 1.1.6.2). Granular contact networks were compared to RGGs by Bassett *et al.* [32], using a set of 19 network diagnostics, and the two were found to differ substantially. We study the same data set using our larger set of network features (which incorporates most of the earlier 19), and focus on feature-feature correlations in the real and synthetic networks. The set of 17 real granular networks (unweighted) was obtained from Bassett *et al.*[1]; alongside these we generated a set of 170 RGGs, with each real network having 10 corresponding random graphs with the same number of nodes and links.[2]

We looked at pairwise correlation coefficients between all 267 network features which were computable for the full set of 187 granular and random networks.[3] For each feature, the average absolute correlation with all other features was computed, separately for the two types of networks. We then looked at the difference in this average correlation between the real and synthetic sets of networks, as a way of depicting how much more or less a given feature tends to correlate with others on the granular networks, relative to the random ones. A histogram of these differences for all features is shown in Figure 4.2(a). It is notable that the number of features with positive differences (i.e., stronger correlations in the granular networks) is far more than the number with negative differences. This suggests that on the whole there are more feature degeneracies in the granular networks. The mean and standard

[1]This data was collected by E. T. Owens, working in the lab of K. E. Daniels.

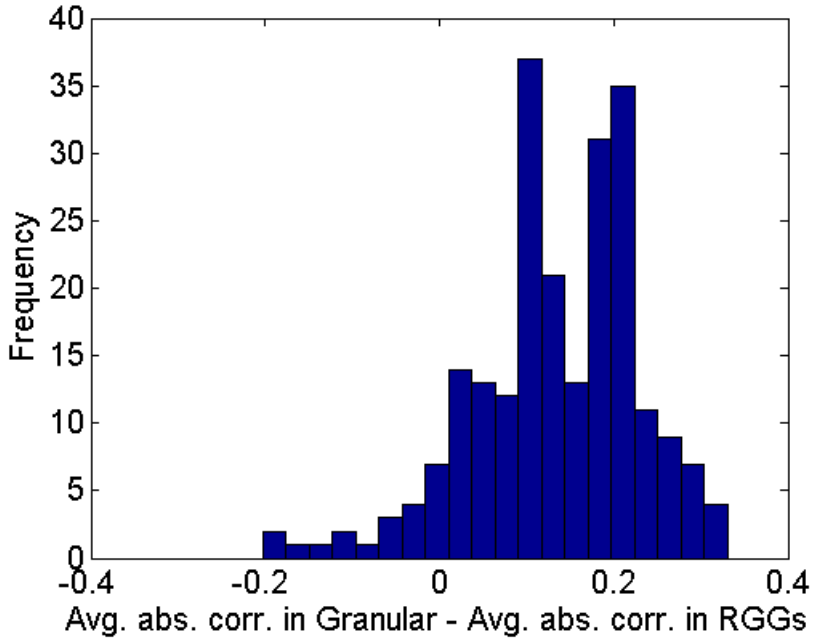
[2]We chose to use just 10 random instances per real network, because the 17 granular networks all had very similar numbers of nodes and links, and thus in effect this gave us an ensemble of 170 random graphs with nearly identical size and density. As can be seen from Figures 4.2(b)-(d), there is considerable structural variablility within this ensemble, far more than within the set of 17 real networks.

[3]This may appear to be a somewhat statistically underspecified problem, particularly for the set of real granular networks, where we have a total of $\frac{267\times266}{2}$ different pairwise correlation coefficients to compute, and a total of 17×267 data points. However, there are two ameliorating factors: firstly, as we have noted, there are many dependencies amongst the features and so the pairwise correlations are clearly not all independent parameters; secondly, we are averaging over all the correlations of a given feature with all the others, and only using these averages here (i.e., only 267 different numbers), rather than all of the pairwise correlations.

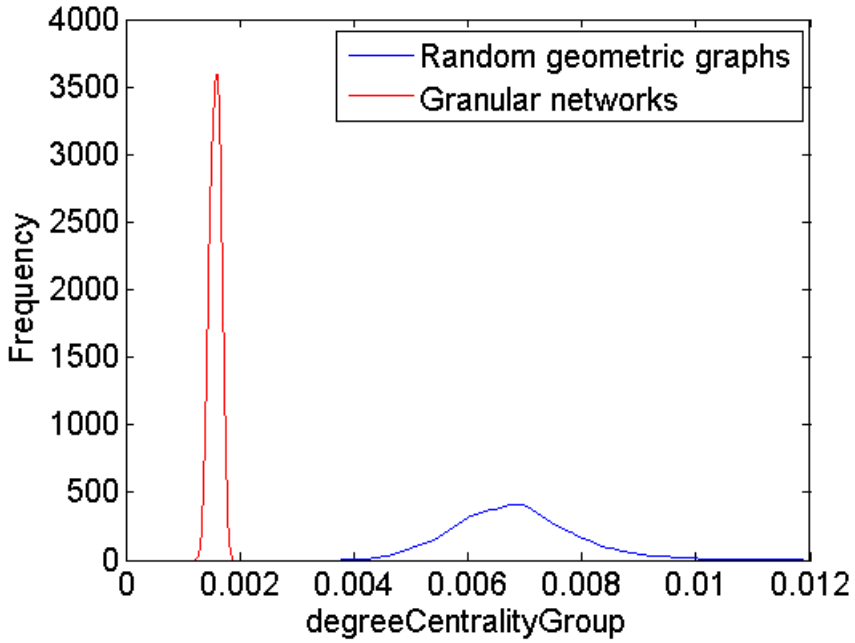
deviation of all pairwise feature correlations are 0.31 ± 0.11 for the granular networks, and 0.18 ± 0.07 for the RGGs, further indicating that the features tend to correlate more strongly on the former set.

Of more interest is to look at which specific features become more strongly correlated in the granular networks, as this might provide some insight into the nature of the structural constraints specific to those networks. In Figures 4.2(b) and 4.2(c) we depict the distributions of two features which show a particularly large increase (amounting to about 0.30 in either case) in average correlation for the granular networks: *degreeCentralityGroup* and $fiedlerValue$. Group degree centrality (see Section 1.1.4.2) can be seen as a measure of the heterogeneity in the node degrees; Figure 4.2(b) thus shows that the granular networks show more uniform degree distributions than RGGs, and also show much less variability in this measure across samples. However, $fiedlerValue$ (Section 1.1.4.7) is a measure of global connectivity and also determines the rate of convergence of a random walker on the network, which is known to be related to modularity-based (and some other) notions of community structure [156]. Thus Figure 4.2(c) indicates that the granular networks have higher global connectivity and less well-defined communities than the RGGs. This is in accord with the observations of Bassett *et al.* that the granular networks have less local connectivity and more global cohesion [32]. Additionally, the fact that both of these features are seen to have much less spread on the granular networks than the RGGs is suggestive of these network properties being under greater constraint in the granular systems.

We also used PCA (Section 1.3.3) to map the granular and RGG networks to a two-dimensional feature space. In addition to these two, we also added in a third set, of Erdős-Rényi or $G(n, m)$ random graphs, with each granular network being used to generate one $G(n, m)$ instance with a matching number of nodes and links. This provides an additional null model to compare against, in which networks are



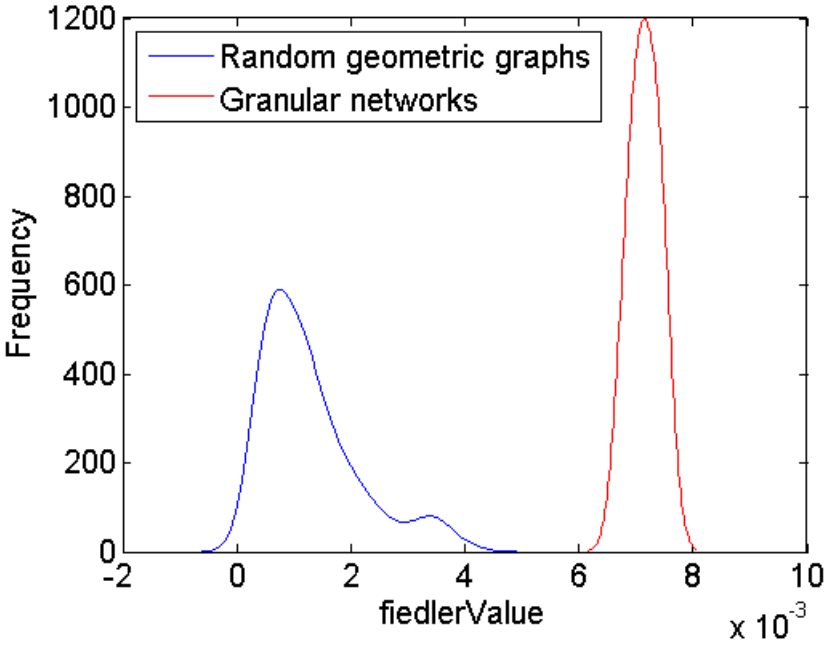
(a) Histogram of differences in average correlation



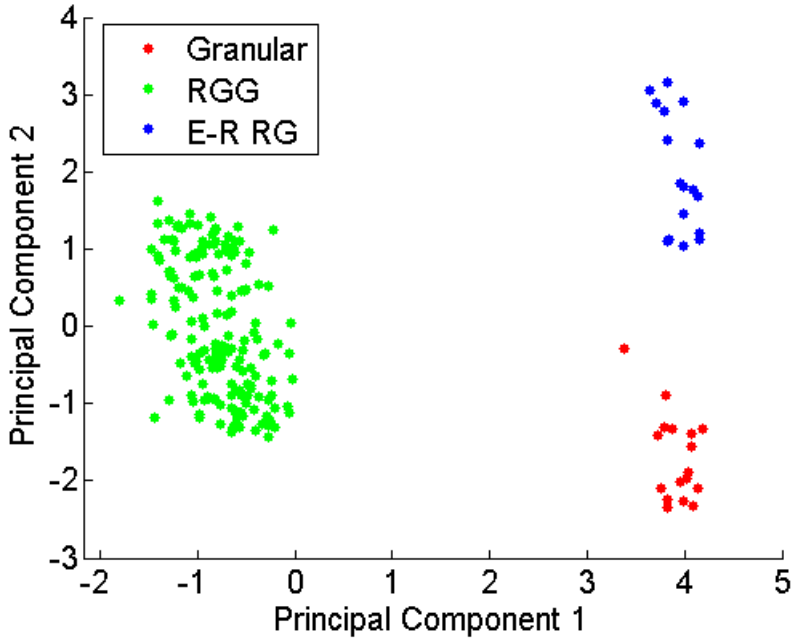
(b) Density plots for group degree centrality

Figure 4.2: **Feature correlation comparisons for granular networks and random graphs.**

(a) Distribution of the difference in the average absolute correlation of a feature with all other features, between the sets of granular and RGG networks. Histogram produced using the MATLAB `hist()` function. (b) Distribution of group degree centrality over the two sets. Density plots produced using the MATLAB `ksdensity()` function.



(c) Density plots for the Fiedler value



(d) PCA space plot of Granular, RGG, and Erdős-Rényi (E-R) networks

Figure 4.2 (continued).

(c) Distribution of Fiedler value for the two sets. (d) Granular networks and two null models (including Erdős-Rényi, which are not spatially embedded and expected to be very different structurally) mapped to a two-dimensional PCA space, representing the two principal components with the largest variation in the space of 267 network features.

not spatially embedded; we expect the Erdős-Rényi graphs to be completely different from each of the two other ensembles. The PCA space for these three categories is depicted in Figure 4.2(d), showing that the granular networks are very distinct from either of the null models. The first principal component here is very highly negatively correlated ($\rho \approx -0.9655$) with the average local clustering coefficient [see Equation (1.19)]; hence, we see as before that the granular networks have lower local clustering than the RGGs, though similar to the Erdős-Rényi networks. The second component correlates substantially with both the energy ($\rho \approx 0.8185$) and the entropy ($\rho \approx 0.6814$) of the degree distribution (Section 1.1.4.9). Thus, the granular networks appear to have lower energy and entropy than either type of random graph. This implies that their degree distributions are more restrictive, allowing for fewer different topologies. Thus, looking at our feature vectors and correlations on these granular networks has allowed us to not only recover some of the observed structural differences with RGGs that were noted in Ref. [32], but has also provided some insight (consistent with physical intuition) into which aspects of the structure of the granular networks seem to be more constrained than random graphs.

The observation that particular classes of real-world networks show degeneracies in structural features suggests the idea that the corresponding systems may be subject to particular constraints, which have the effect of lowering the number of configurations that these networks can take. In statistical physics, the concept of entropy developed by Boltzmann (which we refer to here as *thermodynamic entropy*, to distinguish it from the statistical notion of entropy in general) is used as a way of quantifying such degeneracy in the configurations of a system. We have seen that a particular type of entropy, that of the degree distribution, shows up as significant in distinguishing between granular networks and random graph models. We will now look at how the concept of entropy can be applied more generally to networks, and how it might relate to our observations.

4.3 Thermodynamic entropy of network ensembles

Many kinds of networks might be thought of as one instance from an ensemble of functionally equivalent networks, where the ensemble conserves those structural characteristics which are relevant to the network's function [43]. (More generally, one can think of a given network as a sample from a probability distribution over an ensemble of networks, where the probability of drawing a particular network corresponds to how viable it is in that functional role.) Such conserved characteristics can include things like the density of links (which leads to the Erdős-Rényi model), the degree distribution (which leads to the configuration model, as described in Section 1.1.6.3), or any number of more complex quantities. For instance, one can imagine that protein interaction networks from different species are all instances from an ensemble consisting of all functionally viable network structures in that context. Of course, in general we do not know the particular constraints operating on the formation of a given kind of network. A network can be considered as belonging to many different ensembles. If we generate a network with 100 nodes and 500 links, chosen uniformly at random from all such networks, then we have used the $G(n = 100, m = 500)$ ensemble; however we could also think of the obtained network as an instance from a configuration ensemble that preserves its specific degree sequence. In general, we would like to choose the 'least specific' (or least constrained) ensemble that can reasonably explain a given network's structural features: this is in accord with the intuitive principle of Occam's razor, of not invoking unnecessary detail in our models.

The statistical concept of entropy, which was originally introduced by Boltzmann for the study of thermodynamic systems, has been proposed as a way of quantifying this notion of the complexity of a given network ensemble, in particular in the work of Bianconi *et al.* [22–24, 43–46]. Entropy is a measure of the amount of uncertainty

in drawing a sample from a given probability distribution; it can also be seen as the amount of information contained in the outcome of such a draw. For a discrete distribution over $\mathcal{N}$ outcomes, the standard definition of the entropy H of a distribution where the probability of the i^{th} outcome is p_i is

$$H = -\sum_{i=1}^{\mathcal{N}} p_i \log p_i. \tag{4.1}$$

In the case of network ensembles, each outcome i corresponds to one possible network structure. If the distribution over networks in the ensemble is uniform ($p_i = 1/\mathcal{N}$ for all i, also known as a *microcanonical* ensemble), then this formula reduces to

$$H = \log \mathcal{N}, \tag{4.2}$$

so the entropy is the logarithm of the number of networks. Thus, we see that larger or less constrained ensembles have higher entropy, and intuitively there is an inverse relationship between entropy and 'complexity', in the sense of Bianconi [43], i.e., the amount of information or number of structural constraints needed to specify a given network ensemble. The 'simplest' ensemble or model from amongst a set of candidates will be the one that has the maximum entropy.

To look at a couple of simple examples of computing ensemble entropies, let us consider the Erdős-Rényi model $G(n, p)$ and its related counterpart $G(n, m)$ (see Section 1.1.6.1). The $G(n, m)$ model specifies a microcanonical ensemble, as every network with n nodes and m links is equally probable under this model. Thus, for this model the entropy is given by $H = \log \mathcal{N}$, where the number of possible networks is $\mathcal{N} = \binom{n(n-1)/2}{m}$. The $G(n, p)$ model gives the corresponding *canonical* ensemble: this contains *all* networks with n nodes, but with varying probability weights; and the constraint on the density of links is not satisfied for every instance but only on average. The simplest way to evaluate the entropy in this case is to make use of the

fact that the presence or absence of each possible link is an independent event under this model; thus, the total entropy is the sum of the entropy contributions from all the links. For a given pair of nodes (i, j), they are linked with probability p and not linked with probability $1 - p$, so the entropy is $H_{ij} = -[p \log p + (1 - p) \log(1 - p)]$. The total thermodynamic entropy for this model is given by

$$H_{td}^{er} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} H_{ij} = -\frac{n(n-1)}{2}[p \log p + (1 - p) \log(1 - p)]. \tag{4.3}$$

It is worth noting that there is a direct correspondence between generative models for networks (as in Section 1.1.6) and ensembles, as both in effect specify a probability distribution over the space of all possible networks. As an example, consider the $G(n, m)$ ensemble, which assigns equal probability to all networks with n nodes and m links, and 0 probability to all other networks. This corresponds to the generative model in which one starts with n disconnected nodes and then connects m distinct node pairs at random. Thus, in principle for any proposed model (including settings for any parameters therein), we should be able to define the entropy of the corresponding ensemble, i.e., the probability distribution over networks generated by that model with those parameter settings. In practice, however, it has only been possible to evaluate this for simple models specified in terms of particular structural constraints, such as Erdős-Rényi, the configuration model, or models that constrain the degree correlations or some notion of community structure [44]. For a model specified in terms of evolutionary mechanisms rather than constraints, e.g., duplication-divergence models in biology (Section 1.1.6.7), there is no easy way to analytically evaluate the corresponding probability distribution over networks.

One way of thinking of the entropy of an ensemble is that it represents the volume it occupies in a "phase space", i.e., the space of possible network structures. As we have discussed, our feature vector representation of networks in effect maps them into

a different space; we call this a *feature space*. We can examine the distribution of a sample from a given ensemble of networks in this space and compute (an estimate of) the entropy of this distribution as well; how does this relate (if at all) to the thermodynamic entropy of the ensemble in phase space, and can it inform us about the plausibility of proposed generative mechanisms? These are the questions that motivate the following sections.

4.4 Statistical entropy in feature space

For a given network feature, we can look at how its values are distributed across a set or ensemble of networks. In theory, if we have a probability distribution over networks (such as that specified by a $G(n, p)$ model), then it specifies a corresponding probability distribution over any given feature of those networks, e.g., their mean degree. The entropy of the distribution over such a feature is a measure of disorder or uncertainty in that particular aspect of network structure over the ensemble being considered. If we look at multiple features simultaneously, then we can define the joint entropy over them. For instance, for two continuous-valued features denoted (as random variables) by X and Y, the joint entropy is

$$H(X, Y) = -\int_{\{x\}} \int_{\{y\}} P(X = x, Y = y) \log P(X = x, Y = y) dx dy. \tag{4.4}$$

However, in practice it will be difficult to know what the distribution over a particular feature is for a given ensemble. One way to estimate this is via sampling; one can generate a number of instances from a specified ensemble and compute the values of the features over these in order to examine the resulting distributions. In this case, we will use a standard binning procedure (i.e., a histogram) in order to obtain a discretised probability distribution from a sample. Suppose that the features X and Y range over $[0, 1]$; we divide this range into k equally spaced bins or intervals (in

practice, we use $k = 10$ for our results below). The discretised estimate of the joint entropy is then given by

$$\hat{H}(X,Y) = -\sum_{i=1}^{k}\sum_{j=1}^{k} P\left(\frac{i-1}{k} \leq X < \frac{i}{k}, \frac{j-1}{k} \leq Y < \frac{j}{k}\right) \log P\left(\frac{i-1}{k} \leq X < \frac{i}{k}, \frac{j-1}{k} \leq Y < \frac{j}{k}\right). \tag{4.5}$$

Thus, in this case the entropy is computed over a two-dimensional histogram, defined by a total of k^2 square-shaped bins.[4]

If we are looking at a variety of features that capture diverse aspects of network structure, then we might hope that the joint entropy over them, the disorder in the range of features, would relate to the aggregate disorder in structures in the given ensemble. Thus motivated, we sought to compute such a quantity using our library of network diagnostics. The ensuing feature vector representation of a network defines a position for it in a high-dimensional vector space. However, as we have seen in Section 3.3, there are substantial degeneracies amongst the features and it might be possible to capture much of the variance between the vectors corresponding to different types of networks in a much lower-dimensional space. Thus, rather than attempting to compute joint entropies in the full feature space (a very difficult task in any case, as for a consistent choice of binning along each dimension, the total number of bins scales exponentially with the dimensionality), we will instead do this in a low-dimensional projection. As demonstrated in Section 3.3.2, one way of obtaining such a projection is via Isomap, which we chose to use in order to be able to pick up reduced dimensions that were as far as possible uncorrelated, even in a non-linear fashion. This was

[4]Our choice of uniformly-spaced bins on either axis is the simplest and most commonly used. Other approaches have been tried as well: for instance, if the spread of the data over the range is very non-uniform, then *equal mass binning* [50, 95]—which attempts to pick bins that all contain approximately the same number of data points, rather than being of the same size—may be advisable. Here we chose to stick to the default approach, in the absence of any particular motivation for changing it.

important for being able to gain insight into genuinely orthogonal aspects of network structure that vary amongst real-world and model-generated examples. However, here we stick to a linear principal components analysis (PCA) for simplicity, as our primary goal is to obtain entropy estimates rather than to directly interpret the reduced dimensions.

Thus, our procedure is to obtain the design matrix for a set of networks corresponding to samples from particular ensembles, map these to a low-dimensional space using PCA, and then compute the joint entropies for each sample in this space, as in Equation (4.5). In practice, we use just the first 2 principal components for the examples reported here, as we find that they are sufficient to pick up a large proportion (60–70%) of the variance and they make it easy to visualise the space. As indicated, our motivation is to be able by this means to obtain an estimate of the amount of disorder in network feature space that any given ensemble permits. This is in a sense analogous to what the thermodynamic entropy discussed previously seeks to capture, but there is at least one key difference. In computing thermodynamic entropies, the nodes are regarded as distinguishable; for example, if we are considering networks with 3 nodes and 1 link $[G(n = 3, m = 1)]$, then there are three possibilities, as the link could be between nodes 1 and 2, 1 and 3, or 2 and 3. Of course all three of these represent the same topology, and they are *isomorphic*: one can be changed to another by re-labeling some nodes. In principle one could compute the thermodynamic entropy for networks with indistinguishable nodes too, but this is generically computationally intractable for all but trivially small networks, as it involves solving the *graph isomorphism* problem, i.e., finding pairs or groups of isomorphic graphs, for which no efficient general algorithm has yet been found [99].

However, in computing various network diagnostics, we ignore any node labellings that might be present, and all our network features treat nodes as indistinguishable. Hence, the 3 networks in $G(n = 3, m = 1)$ will all have exactly the same feature

vectors and are mapped to a single point in feature space. In general, ignoring node labels will reduce the size (and thus the entropy) of an ensemble, though the amount of reduction depends on the nature of the ensemble. Consequently, due to this factor alone, we cannot expect any generic relationship between thermodynamic entropy and entropy in our feature space. Additionally, mapping from a network to a feature vector itself involves loss of information, and there will be cases where even non-isomorphic graphs have the same feature values.

Thus, the entropy in feature space is measuring something quite different from thermodynamic entropy. We would like to study whether and how these two notions of network entropy might relate; we now seek to do this using example ensembles corresponding to simple network models for which the thermodynamic entropy can be obtained analytically.

4.5 Entropy comparisons

4.5.1 Erdős-Rényi networks

As a first step, we look at ensembles generated from the $G(n, p)$ model, with different settings for n and p. We generated a random sample of 100 networks from each of 3 ensembles: $G(50, 0.3)$, $G(50, 0.5)$, and $G(100, 0.5)$. We then computed the design matrix for these 300 networks and carried out PCA to map them to the space defined by the two largest principal components. The results are depicted in Figure 4.3. For each ensemble, we compute the thermodynamic entropy as per Equation (4.3) (denoted H_{td} in the figure), as well as the discretised joint entropy in the 2-D PCA space as per Equation (4.5) (denoted H_{PCA} in the figure).

As noted above, the absolute values of the entropies are not comparable, as they are defined on different spaces and are measuring different quantities. However it is notable that in this instance the relative ratios and ordering of the joint entropies (also

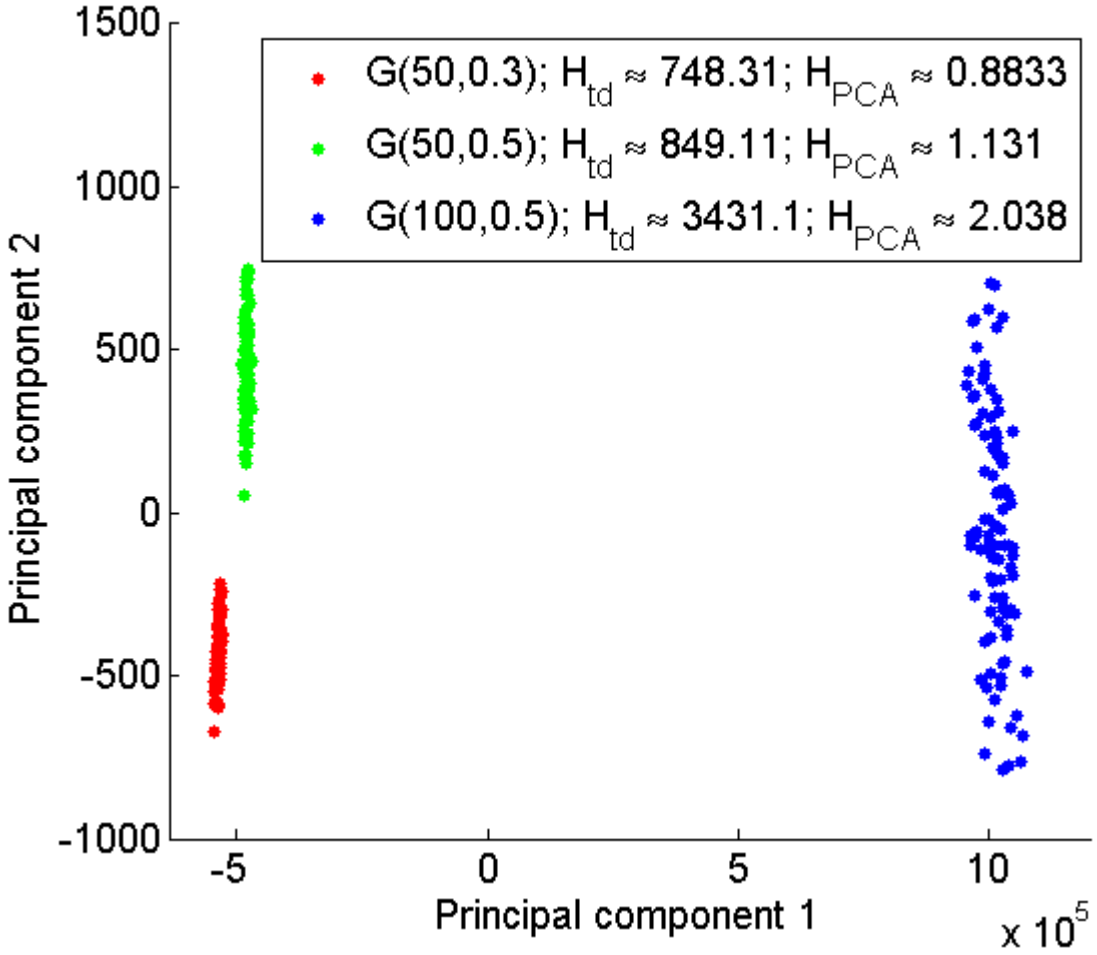


Figure 4.3: **Entropy comparisons for Erdős-Rényi ensembles.**

Samples of 100 networks per ensemble, shown in the space defined by the top two principal components, after carrying out PCA on a set of 124 features that could be successfully computed for all networks. For each ensemble, the thermodynamic entropy (H_{td}) and the discretised joint entropy (H_{PCA}, based on dividing the range of each component into 10 equi-spaced bins) in this space are shown in the legend. The first principal component is largely reflecting the number of nodes, whilst the second is substantially correlated with density, maximum degree centrality and maximum clustering coefficient (see main text discussion).

apparent from the spreads of the 3 point clouds) roughly mirror those of the thermodynamic entropies. Thus, in this case, the amount of disorder shown by the samples from the different ensembles in the low-dimensional PCA feature space provides some indication of their entropy in network phase space, even though the latter treats nodes as distinguishable whilst the former does not. This suggests that the PCA dimensions might be indicative of aspects of network structure left unconstrained within these ensembles. Here in Figure 4.3, it is evident that the first principal component is essentially just a scaled version of the network size (number of nodes), though the actual values along the component are much larger due to the presence of many other correlated features with high numerical values. The second component is significantly correlated with the density of links ($\rho \approx 0.70$), but is also capturing other aspects of local structure such as the maximum degree centrality ($\rho \approx 0.78$) and maximum local clustering coefficient ($\rho \approx 0.76$). This indicates that these are the sorts of network properties that show relatively large variation within these ensembles.

4.5.2 Modular networks

We generated modular networks in similar fashion to those described in Section 3.3.1, with 10 modules of 10 nodes each such that each module starts as a clique, and then with some probability λ, each link is rewired randomly. We choose values of λ ranging from 0.1 to 1 in steps of 0.1 and generate 100 random networks for each setting. This gives samples from 10 ensembles (one for each value of λ).

For these networks, it is possible to obtain an approximate analytical expression for the thermodynamic entropy per node H_{td}^{mod} as a function of the total number of nodes n, the mean degree $\langle k \rangle$, and λ [16]:

$$\begin{aligned} H_{td}^{\text{mod}}(\lambda) = & \frac{-1}{2} \left[\langle k \rangle \left[\epsilon_{\text{in}} \log \epsilon_{\text{in}} + (1 - \epsilon_{\text{in}}) \log(1 - \epsilon_{\text{in}}) \right] + \right. \\ & \left. (n - 1 - \langle k \rangle) \left[\epsilon_{\text{out}} \log \epsilon_{\text{out}} + (1 - \epsilon_{\text{out}}) \log(1 - \epsilon_{\text{out}}) \right] \right], \end{aligned} \tag{4.6}$$

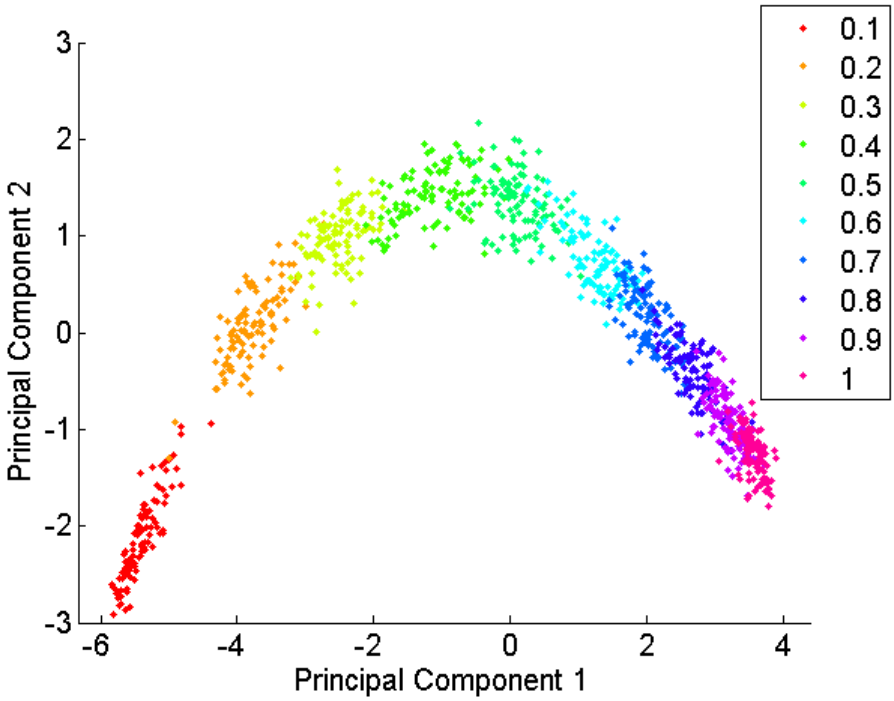
where

$$\epsilon_{\text{in}} = 1 - \lambda + \lambda \frac{\langle k \rangle}{n-1},$$
$$\epsilon_{\text{out}} = \lambda \frac{\langle k \rangle}{n-1}.$$

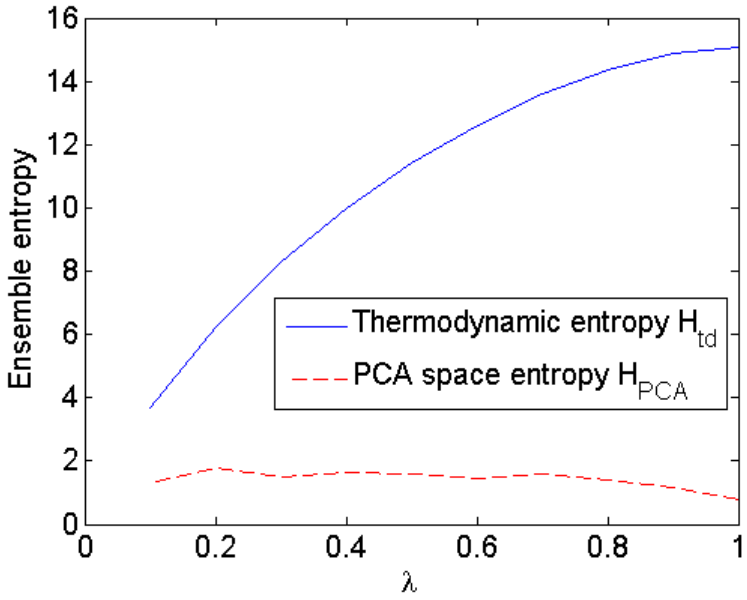
We provide our derivation for this expression in Appendix C. In our case, we have $n = 100$ and $\langle k \rangle = 9$ as constants, and we only vary λ. Because all networks we are considering for this example have the same number of nodes, we divide the ensemble entropy by the number of nodes (100) to get entropy per node; this rescaling makes it easier to plot the thermodynamic entropies on the same scale as the PCA space entropies.

As before, we map our 1000 (10×100) modular networks to a 2-D PCA feature space; this is depicted in Figure 4.4(a), with different colours denoting the different ensembles (i.e., values of λ). Joint entropy for each ensemble sample is then computed by using 10 bins on each axis, as per Equation (4.5). In Figure 4.4(b), we plot these alongside the thermodynamic entropy values computed as per Equation (4.6). It can be seen in Figure 4.4(a) that there is little change in the spread of the ensemble samples with increasing λ, which is borne out in Figure 4.4(b), where we see that whilst the thermodynamic entropy increases steadily with λ, the PCA space entropy changes little.

Thus in this case there appears to be no correspondence between the two values of entropy. It is notable that the principal components shown in Figure 4.4(a) appear to primarily capture variation between ensembles rather than within them. This suggests that entropy computed in this space will not really capture the disorder within ensembles, which may explain why no increasing trend is seen with increasing rewiring. Whilst the PCA space is still potentially useful in being able discriminate



(a) Modular networks in PCA space for varying λ



(b) Entropy comparison for modular networks

Figure 4.4: **Feature space and thermodynamic entropies for modular networks.**

(a) 10 sets of 100 modular networks each, corresponding to 10 different settings of the rewiring parameter λ, mapped to the space of the two largest principal components in feature space. (b) Plots of thermodynamic entropy values for the 10 ensembles, as well as joint feature entropy of the 100-network samples from each ensemble in the PCA space.

between networks generated with varying values of the model parameter (we attempt to pursue this further in Section 4.6 and in Chapter 5), examining additional principal components or other ways to obtain reduced dimensions or features may help to determine if and under what conditions a meaningful space for computing entropy might be obtained.

4.5.3 Watts-Strogatz networks

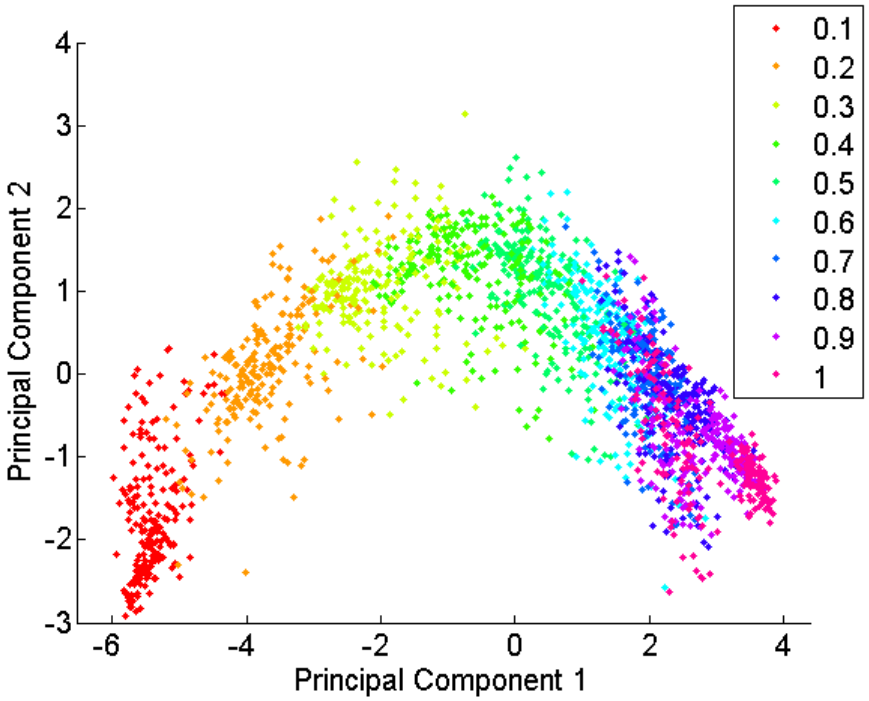
As a third example, we generated synthetic networks using the Watts-Strogatz model [273] (Section 1.1.6.4), with $n = 100$, $k = 4$, and rewiring probability p ranging from 0.1 to 1 in steps of 0.1. As for modular networks, we generated 100 instances for each setting of p. The analytical expression for thermodynamic entropy per node H_{td}^{ws} in this case is given by [16] (our derivation is in Appendix C):

$$\begin{aligned} H_{td}^{ws}(p) = &\frac{-1}{2}\left[k\left[\epsilon_{\text{near}}\log\epsilon_{\text{near}} + (1-\epsilon_{\text{near}})\log(1-\epsilon_{\text{near}})\right] + \right. \\ &\left.(n-1-k)\left[\epsilon_{\text{far}}\log\epsilon_{\text{far}} + (1-\epsilon_{\text{far}})\log(1-\epsilon_{\text{far}})\right]\right], \end{aligned} \tag{4.7}$$

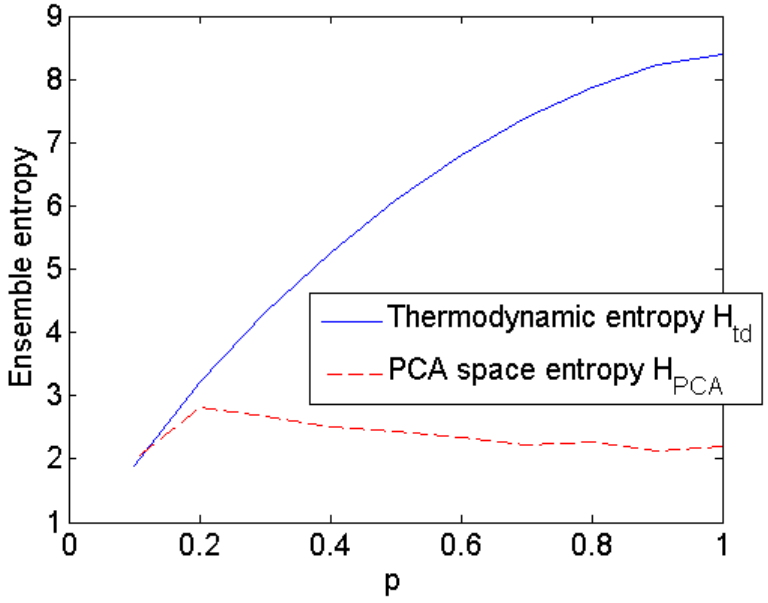
where

$$\begin{aligned} \epsilon_{\text{near}} &= 1 - p + p\frac{k}{n-1}, \\ \epsilon_{\text{far}} &= p\frac{k}{n-1}. \end{aligned}$$

In Figure 4.5(a), we show in PCA feature space the networks sampled from these 10 ensembles, corresponding to varying values of p. In Figure 4.5(b), we plot the joint entropy in feature space, along with the thermodynamic entropy [Equation (4.7)]. We see similar results to those for the modular networks, with little variation in feature space entropy across ensembles and the principal components appearing to capture predominantly inter-ensemble variation.



(a) Watts-Strogatz networks in PCA space for varying p



(b) Entropy comparison for Watts-Strogatz networks

Figure 4.5: **Feature space and thermodynamic entropies for Watts-Strogatz networks.**

(a) 10 sets of 100 Watts-Strogatz networks each, corresponding to 10 different settings of the re-wiring parameter p, mapped to the space of the two largest principal components in feature space. (b) Plots of thermodynamic entropy values for the 10 ensembles, as well as joint feature entropy of the 100-network samples from each ensemble in the PCA space.

4.6 Discussion

In this chapter, we have examined correlations between network features and observed that sets of particular types of networks show more correlations than varied or mixed sets. This suggests that the additional correlations might be indicative of structural constraints that are specific to given network types. Using the example of granular contact networks, we have demonstrated how an examination of particular network features that show increased correlations relative to a null model can give us insight into the nature of such constraints.

We have also looked at the concept of the entropy of a given network model or ensemble as a way of capturing the amount of information contained in it. We define a joint entropy over network feature space, which may be seen as a complement to the conventional entropy over phase space. We study the relation between the two quantities and find some correspondence for simple Erdős-Rényi ensembles, but not for others. A major reason for this appears to be that the principal directions of variation in feature space, for a given collection of ensemble samples, might capture variability between the ensembles rather than within them, so computing entropy in that space may not be informative about intra-ensemble disorder. Thus, the computation of feature-space entropy is clearly sensitive to the particular features or principal components selected, and a question for future work is whether it is possible to find ways of carrying out this selection such that the corresponding entropy serves as a meaningful measure of the amount of structural variability in the ensembles being studied. Another question worth studying further is what one can learn from the particular feature correlations observed for a given kind of ensemble, and the extent to which this might help to pin down both constraints and variable aspects for those ensembles.

However, we believe our representation of network ensembles in a low-dimensional PCA space provides a tractable way of estimating the range of structural variants, or

the region of structure space, captured by a given network model with given parameter settings; Figures 4.3, 4.4(a), and 4.5(a) all indicate that distinct regions of the space can be associated with networks drawn from a particular model with particular parameters. What we would like to do in practice is to compare such models to networks from the real world: to be able to fit models to data and thus obtain plausible generative mechanisms for the structures we observe. It seems natural to attempt to do this by measuring the overlap between the regions of our space occupied by model and data. In the next chapter, we show how this can be formalised using Bayesian statistical inference techniques and develop an approximate Bayesian computation (ABC) methodology for model-fitting to networks.

Chapter 5

Bayesian Model-Fitting for Networks

In this chapter we develop a methodology for matching networks to generative models via our feature space representation, making use of the framework of approximate Bayesian computation. The majority of the work in this chapter is included in a manuscript currently in preparation [17].

5.1 Background and motivation

We have shown in Section 3.5 how a simplistic evolutionary model can be used to identify network characteristics that correlate with some known functional or phenotypic property. To take this further, in many instances (such as networks representing biological or social systems), it is desirable to develop models of network evolution as a whole: models that generate synthetic networks with structure essentially similar to the real network(s) of interest. This can be useful for simulating virtual instances of a particular type of real-world network, or for gaining insight into the natural processes that generated that network structure. Typically, the fit of a model to a real network is evaluated by comparing them using only a few diagnostics. By con-

trast, our approach will allow us to compare synthetic networks to real ones more comprehensively, using any number of structural features.

In the previous chapter, we have seen how, given ensembles of networks corresponding to a particular model, it is possible to pick out directions in our vector space of features which correspond to the variations of model parameters, and also ones which correspond to variation independent of model parameters (i.e., structural characteristics not entirely constrained by the specification of the model parameters). We would like to be able to use such directions in structure space to compare models to data, as they can allow us both to discriminate between different parameter settings and to examine how well the structural variation observed in a real-world data set (i.e., its entropy, which might be defined in multiple ways as seen in the previous chapter) can be explained by a particular network model. However, in practice, for complex models with multiple parameters and intractable analytic entropy, it is not feasible to find directly features that correlate with these quantities. As we have suggested, a low-dimensional embedding that captures the bulk of the variability in a given data set (e.g., networks from a given model with varying parameter settings, as for the examples in Section 4.5)—obtained, for instance, via Principal Components Analysis (PCA)—may in practice represent the desired structural aspects to an extent. In this chapter, we develop an algorithm for model-fitting to networks in such a PCA space, using the methodology of approximate Bayesian computation, and demonstrate how this can serve as an effective means of matching models to data.

5.2 Bayesian inference for model-fitting

Bayesian inference [47, 124] is a generic and widely used framework for fitting probabilistic models to data: both for comparing distinct models and for determining appropriate parameter values for a given model. Suppose we are given model M,

inclusive of some model parameters θ, which specifies a probability distribution over data instances (in our case, networks). Given a particular observed data set x, our task is to evaluate different models and parameter settings to judge how likely they are to have generated this data. Bayes' theorem tell us that this can be expressed in the following manner:

$$P(M, \theta|x) \propto P(x|M, \theta)P(M, \theta). \tag{5.1}$$

Here $P(x|M, \theta)$ (known as the *likelihood*) is the probability of the data x under model M and parameter settings θ; and $P(M, \theta)$ is the *prior* probability of selecting this particular combination of model and parameters. The product of prior and likelihood is proportional to $P(M, \theta|x)$, the *posterior* probability of the model-parameter combination, once the data x has been observed. This posterior is the quantity that allows us to evaluate different possibilities, and in general models and parameter values that have a high posterior probability are preferable. If we wish to select a single combination, then the natural choice is the one that maximises the posterior distribution, also known as the *maximum a posteriori* estimate. However, one of the attractive features of Bayesian inference is that it does not just give us a single answer, but allows us to look at the entire distribution of posterior probabilities over possible choices, thus also providing a quantification of uncertainty in making this choice.

Another feature of this framework is the need to assign a prior probability distribution to models and parameters being considered. There are differing views about the extent to which this can or should be seen as a means of incorporating subjective human opinion [103, 104, 271]. In the absence of any specific prior knowledge, it is common to choose as uninformative a prior as possible, typically a uniform distribution over all models and the feasible ranges of all parameters therein. This is the approach we seek to adopt here. Naturally this choice too is subjective and may

not always be appropriate: for instance, the feasible range of a parameter within a mathematical model may be larger than its feasible range in the real world, due to constraints that are not included in the model. In this case taking a uniform prior over the entire range may correspond to a bias towards unrealistic values. However, in the lack of specific knowledge about such constraints on the parameters in the models we use, using a uniform distribution is the standard choice, the one which involves the fewest assumptions.

Having made a choice of prior, if we have an analytic expression for the likelihood function corresponding to a given model, then it is straightforward to get the posterior. However, for complex models defined in terms of mechanisms rather than distributions, such as the generative models for networks we are interested in here, a closed-form expression for the probability of a given data set being generated is not generally available. Thus, another means of estimating likelihoods is required; we discuss our chosen methodology next.

5.3 Approximate Bayesian computation

In general, the likelihood function might not be tractable to compute, particularly for complex models, and this makes it difficult to obtain analytic values for the posterior probability of different models and parameter settings. One way of circumventing this is to use the technique of *approximate Bayesian computation* (ABC) [173, 215, 255], which allows one to obtain an estimate of the posterior via Monte Carlo sampling of model outputs, without the need for specifying an explicit likelihood function. ABC has previously been applied for model comparison and criticism in the context of networks [218, 219, 261], and here we extend this approach to make use of the large sets of both networks and features that we have compiled.

ABC relies on the fact that whilst the likelihood may be intractable, given a model

M and a set of parameters θ, it is usually easy to generate samples from the distribution $P(\cdot|M,\theta)$. The simplest approach works by measuring the distance of these samples from the actual data, using a discrepancy function (denoted here by d) that typically combines a set of tractable summary statistics of the data. For instance, if $\mathcal{S}(x)$ denotes a vector of summary statistics of data point x, then the discrepancy between the summaries of points x (sample) and x_0 (actual data), $d(\mathcal{S}(x),\mathcal{S}(x_0))$, might be defined as the Euclidean distance between the two summary vectors. Subsequently, only those values of (M,θ) are retained for which the discrepancy lies within a certain threshold τ. These can then be used to define an *approximate likelihood* P_τ [218]:

$$P_\tau(x_0|M,\theta) = \frac{1}{\tau}\int_{\mathcal{X}} I(|d(\mathcal{S}(x),\mathcal{S}(x_0))| \leq \tau/2)P(x|M,\theta)dx. \tag{5.2}$$

Here $\mathcal{X}$ is the domain of the data points, and $I(\cdot)$ denotes the indicator function: it is 1 if its argument is true and 0 otherwise. Note that $P(x|M,\theta)$ is the unknown likelihood; however, we are able to sample from this, and thus by counting the fraction of samples retained after applying the τ threshold one can obtain a numerical estimate of the above integral. As $\tau \to 0$, $P_\tau(x_0|M,\theta)$ will converge to the true likelihood of the summaries of x_0 under the given model and parameters [218].

Ratmann et al. [218] suggested that also retaining the discrepancy or error values themselves, and examining their posterior distribution allows for model criticism (judging how well a model matches the data), in addition to model comparison (relative evaluation of different models). In this context, the threshold τ can be seen as defining a prior belief on the discrepancy between model and data (see Figure 5.1). Two important choices involved in ABC are those of the summary statistics and discrepancy function (which may be multidimensional), and the choice of the error prior, typically corresponding to a choice of τ (or multiple choices, if the discrepancy is multidimensional). In previous work on using ABC to select models for network evo-

lution, these choices have generally been based on intuition or convenience, and have included a limited range of statistics or features such as network diameter, clustering coefficient and measures of the degree distribution [218, 219]. Here we demonstrate how these selections can be carried out in a more data-driven fashion.

5.4 Data-driven parameterisation for ABC

5.4.1 Automated network summary statistics

Regarding the choice of summary statistics, the library of algorithms compiled by us allows the computation and comparison of over four hundred network features, as described previously. We would like to pick out those network statistics that show the most variability across the different kinds of networks produced by the models we are considering, so that specific models matching the data can be identified (recent work [172] provides theoretical evidence that appropriate summary statistics for use in ABC are those whose mean values are different under the models being compared, in the asymptotic limit of arbitrarily many samples from each model). As shown in Chapter 3, the full set of features can be mapped to a low-dimensional space where most of the inter-network variance is captured (Figure 3.2). Here, for simplicity, we do not use Isomap but instead use PCA (see Section 1.3.3) to carry out this dimensionality reduction. Our approach thus involves starting with a set of synthetic networks B, which we generate from the multiple different models (described in Section 5.5) being fitted (here B includes 500 networks per model). We compute the design matrix for the set of networks $B \cup D$ (where D is the data to be fitted using ABC), and then perform PCA on this matrix to pick out a small number d of dimensions.

The choice of the parameter d involves a trade-off: more dimensions will capture more variance, but in a higher-dimensional space we will need more data points to

be able to adequately sample it and obtain a substantive match between model and data. Thus the lowest dimensionality that is sufficiently discriminative is desirable; in practice we find that even $d = 2$ seems to capture 60–70% of the variance for the models and data sets examined by us. The choice of d is one of the sources of subjectivity in our methodology; in the results presented here we stick to $d = 2$, partly for ease of visualisation, but also because for these examples we find that the fraction of the variance captured by the third principal component is only about 5%, whereas doing ABC in 3 dimensions leads to considerably fewer samples being retained in the posterior compared to 2 dimensions, making the results rather less reliable. However for our examples involving comparison of different models, both for synthetic and real data, we did attempt using the first three principal components as well, and in each case the ABC fitting led to the same ranking of models as in the two-dimensional results depicted below. A more detailed study of the effects of different choices, both of the dimensionality reduction method and the number of reduced dimensions used, remains a topic for future work.

The d chosen components (each of which is a linear combination of the original feature set) become the summary statistics to be used in our ABC implementation. In principle Isomap, sparse PCA [287] or any other methods of dimensionality reduction or feature selection could also be used to obtain the specific small number of summary statistics to be considered. In some cases, the interpretability of the chosen dimensions may be aided by having them correspond to one or a small number of network diagnostics, rather than being a linear combination of all available features, as given by PCA. Thus using direct feature selection, or a method like sparse PCA, which forces the reduced dimensions to be combinations of relatively few of the original features, may be helpful in providing additional insights. However, here we stick to PCA as a means of demonstrating how our methodology works.

5.4.2 Definition of error prior

The second issue is that of choosing the error prior. In past work [218, 219], this choice had to be arbitrary as only a single network was being fit at one time, so there was no information about how much noise or variability in the match between data and model might be acceptable. Here, we propose fitting an entire ensemble of networks simultaneously (i.e., D is a set of networks rather than a single network). By examining the spread (i.e., the variance) of D after mapping to a low-dimensional feature space, we can obtain an estimate of the intrinsic variability in the type of networks we are trying to model, and use this to define a feasible region via the error prior (see Figure 5.1). Allowing the data to drive the spread of the error prior makes it relatively objective (compared to imposing an arbitrary error threshold), though a Gaussian functional form is imposed for ease of computation.

Let $G^1, G^2, ..., G^{|D|}$ denote the networks comprising the target set D, and let $\mathcal{S}_d(G^i)$ denote the vector of coordinates of G^i in the reduced d-dimensional feature space obtained via PCA. We define the error or discrepancy ϵ for a given sample network G as the vector difference from the mean of all networks in D:

$$\epsilon = \mathcal{S}_d(G) - \frac{1}{|D|}\sum_{i=1}^{|D|}\mathcal{S}_d(G^i). \tag{5.3}$$

We then define the error prior as a multivariate (d-dimensional) Gaussian (see Figure 5.1) rather than as a set of d step functions, as typical in previous work (e.g., Ratmann *et al.* [218]). We centre the Gaussian at 0, and set its covariance matrix to equal the empirically observed covariance of the d features on the set D. Thus, rather than retaining every point within a given error threshold during ABC sampling, we retain points with the probability assigned to them by the error prior (Figure 5.1).

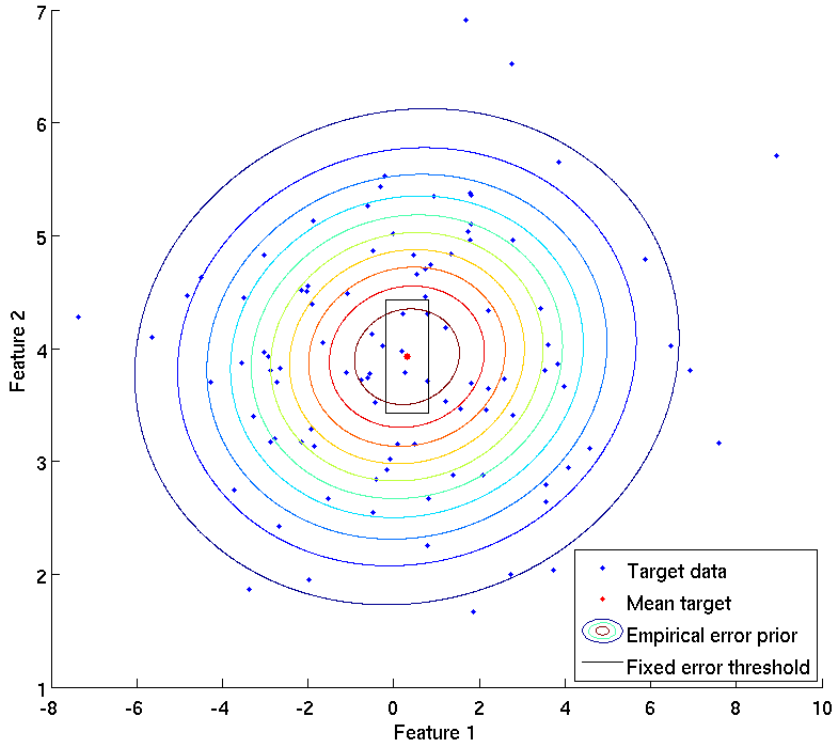


Figure 5.1: **Obtaining the empirical error prior from a given target data set.**

We define a Gaussian distribution centred at the mean (red dot) of the target data points (blue dots), and with covariance defined by the empirical covariance matrix of the data (contour lines). This distribution defines the probability of retaining samples in the posterior at any point in the feature space. In contrast, using a fixed error threshold corresponds to retaining all points inside a given area (black box) and rejecting all those outside of it.

5.4.3 Algorithm

We use the following protocol for carrying out ABC:

- For $i \in \{1, 2, 3\}$; for $j \in \{1, 2, ..., 500\}$:
  1. Draw parameters θ_{ij} from priors for model M_i
  2. Sample a network G^{ij} from $P(\cdot|M_i, \theta_{ij})$
  3. Add the element G^{ij} to set B
- Compute the design matrix of network features for $B \cup D$; perform PCA on this. Map all networks into the space $\mathcal{F}$ defined by the d largest principal components
- Define the error prior π_ϵ as a bivariate Gaussian centred at 0, with covariance matrix defined by the covariance of D (Figure 5.1)
- For $i \in \{1, 2, 3\}$; for $j \in \{1, 2, ..., 500\}$:
  1. Compute ϵ_{ij}, the discrepancy of G^{ij} from the mean of the set D in space $\mathcal{F}$, as per Equation (5.3).
  2. Retain $(M_i, \theta_{ij}, \epsilon_{ij})$ in posterior with probability $\pi_\epsilon(\epsilon_{ij})$

The last part is a probabilistic version of *rejection sampling*; this is the simplest form of sampling for ABC model-fitting, which involves generating a number of independent samples (here, the elements G^{ij}) from the model at hand, and accepting or rejecting each one based on some criterion which provides a measure of how close it is to the data being fitted. The drawback of this method is that there may be considerable wastage of samples if a large number of them are not close to the data and get rejected. A number of other sampling strategies, collectively referred to as *Monte Carlo* methods, have been used for ABC, such as Markov Chain Monte Carlo [174] and Sequential Monte Carlo [244]; these attempt to successively sample more intensively in regions closer to the data (as opposed to each sample being an independent

draw), and thus lead to fewer wasted samples, though they are also more difficult to implement and execute. Here we employ a rejection sampling protocol as a means of demonstrating the sorts of results this methodology can achieve for fitting models to networks, even with relatively few samples. Future extensions may include developing more sophisticated sampling procedures for this setting to achieve better efficiency.

The aspects of this algorithm that are novel, compared to previous work using ABC to fit models to networks, are: (1) The use of PCA on a set of several hundred network features to determine the coordinates used to define the distances (discrepancies) between networks (as opposed to using a pre-determined set of a few network summary statistics); (2) The use of a target set D, as opposed to fitting one network at a time, and the use of the covariance of D in the PCA space to define the error prior as a Gaussian distribution, rather than a set of step functions representing pre-determined thresholds (as depicted in Figure 5.1).

5.5 Fitting network models

We chose to look at 3 models that have been proposed in the context of Protein Interaction Network (PIN) evolution [218]. These are as follows:

- *Preferential Attachment Poisson (PAP)* [218]: As described in Section 3.3.1. When generating our synthetic networks we choose m (the mean of the Poisson distribution from which the number of node attachments are drawn) uniformly at random from $[0, 30]$, following Ref. [218].

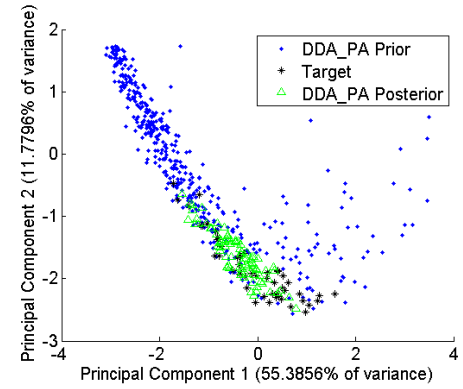
- *Duplication-Divergence-Attachment and Preferential Attachment (DDA+PA)* [219]: As described in Section 3.3.1. When generating synthetic networks for model-fitting on the toy data set (Section 5.5.1), we choose α and δ_{Att} uniformly at random from $[0, 1]$, and δ_{Div} uniformly at random from $[0, 0.3]$ (see comment below). For fitting the real PINs we allow the latter to also vary over $[0, 1]$.

- *Duplication-Divergence and Link addition and deletion and Preferential Attachment (DD+LNK+PA)* [37, 218]: In this model, one of four things can happen at each step. (1) With probability α, there is preferential attachment to a single node as in DDA+PA. The probabilities of the other three events depend on network size; they are normalised to sum to $1 - \alpha$. (2) With unnormalised weight of $\kappa_{\text{Dup}} \times N$ (where N is the number of nodes), there is a Duplication-Divergence event as above, with parameter δ_{Div}, but no parent-child attachment ($\delta_{\text{Att}} = 0$). (3) With unnormalised weight of $\kappa_{\text{LnkAdd}} \times \left[\binom{N}{2} - E\right]$ (where E is the number of links), there is link addition: a randomly chosen node is preferentially attached to another node. (4) With unnormalised weight of κ_{LnkDel} times the link addition weight, there is link deletion: a randomly chosen node is preferentially delinked from one of its interaction partners. For network generation, parameters are sampled uniformly at random from these ranges: $\alpha, \kappa_{\text{LnkDel}} \in [0, 1], \log \kappa_{\text{Dup}} \in [-1, -24], \log \kappa_{\text{LnkAdd}} \in [-2, -24], \delta_{\text{Div}} \in [0, 0.3]$ (toy data), $\delta_{\text{Div}} \in [0, 1]$ (PIN data).

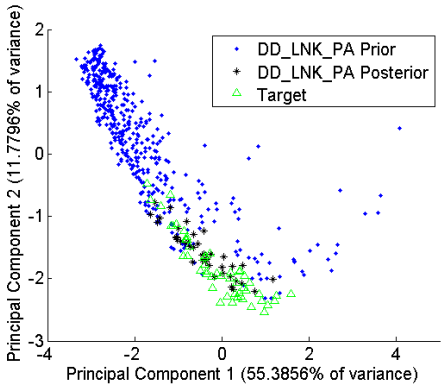
In general, all parameter ranges are taken from Ratmann et al. [218]; the only exception is δ_{Div} for the toy data, where we chose to restrict the range to $[0, 0.3]$ as a small actual value of $\delta_{\text{Div}} = 0.05$ had been used to generate the data being fitted.

5.5.1 Synthetic data

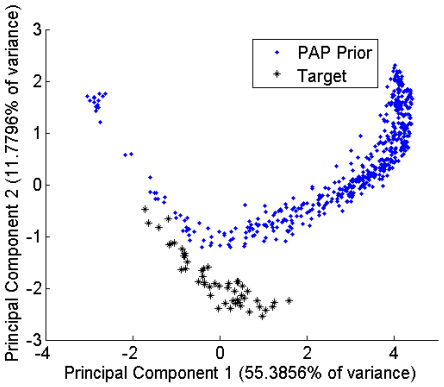
We first demonstrate our ABC algorithm on synthetic data. Specifically, we generate a target set D of 50 networks with 50 nodes each from the DDA+PA model, previously suggested as a viable mechanism for the evolution of Protein Interaction Networks [218, 219]. This set is generated with fixed values for the model parameters (specifically, $\alpha = 0.4$, $\delta_{\text{Div}} = 0.05$, $\delta_{\text{Att}} = 0.6$). We then attempt to fit to the target the three different models described above: DDA+PA, DD+LNK+PA, and PAP, with parameters for each of these drawn from uniform priors as mentioned; and we



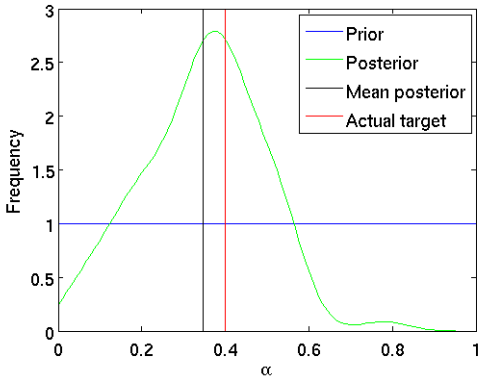
(a) PCA plot for DDA+PA



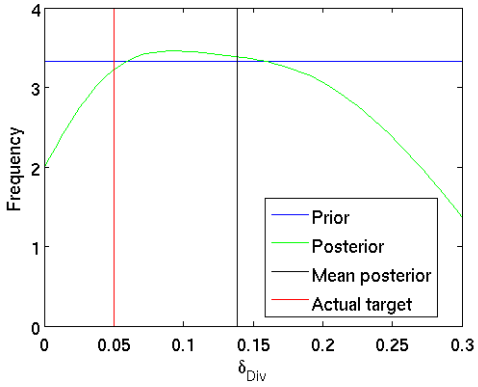
(b) PCA plot for DD+LNK+PA



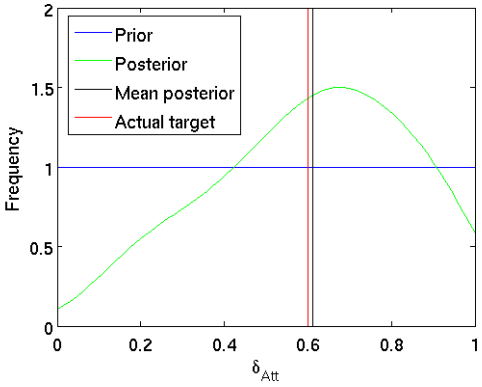
(c) PCA plot for PAP



(d) Posterior on α for DDA+PA



(e) Posterior on δ_{Div} for DDA+PA



(f) Posterior on δ_{Att} for DDA+PA

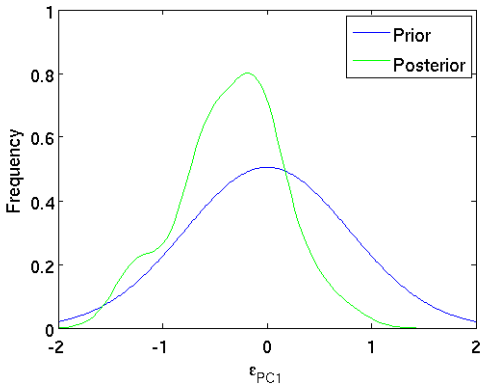
Figure 5.2: **Results of ABC model-fitting to a set of 50 synthetic networks from the DDA+PA model.**

(a)–(c): PCA space maps of the target set D (black), samples generated from priors over model parameters (blue), and retained samples (green). The first principal component has a very strong linear correlation with network *energy* ($\rho \approx 0.99$), and also with the link *density* ($\rho \approx 0.96$), whilst the second correlates maximally with $subgraphCentrality_{fit:normal}$ ($\rho \approx 0.96$; feature descriptions in Appendix A).
(d)–(f): Posterior distribution over DDA+PA model parameters. Prior uniform over $[0, 1]$ for α, δ_{Att}; uniform over $[0, 0.3]$ for δ_{Div}. Posterior mean in black, actual setting for generating D in red.

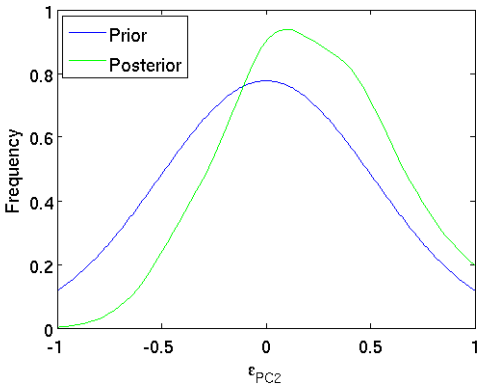
follow the simple rejection sampling protocol of Section 5.4.3, using as samples the 500 networks per model generated for set B.

As a result, for each model fitted, we obtain posteriors over the parameter set and the errors along different dimensions of the feature space. Figure 5.2 shows the results, depicting the parameter posteriors only for the DDA+PA model (the true model used to generate D). We observe that the true model matches the data substantially better than the others, in particular the very different PAP model. The first principal component depicted is found to have strong correlations with the network energy (see Section 1.1.4.9), as well as with the density of links in the network. The second component is strongly correlated to the goodness of fit (measured via log-likelihood; see Appendix A) of a normal distribution to the node subgraph centralities (see Section 1.1.4.2), respectively. This provides an indication of the specific aspects of network structure that show high variability across the models and parameter settings chosen here.

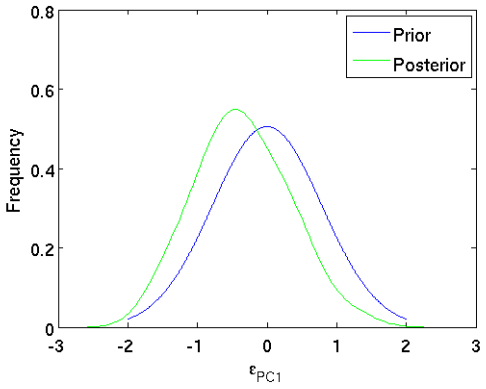
Also, the posteriors on the 3 parameters of DDA+PA shown in Figure 5.2 are peaked around the true parameter settings, showing good performance in not just detecting the correct model but also recovering model parameters. The error posteriors for DDA+PA and DD+LNK+PA are shown in Figure 5.3; they are peaked near 0 and their spreads are similar to the corresponding error priors, indicating that these models are largely able to capture both the central tendency and the spread of the real data set D. We repeated the procedure for a smaller target set D', comprising 25 networks, and we see greater uncertainty or spread in the posteriors in this case (see Figure 5.4); this is a reflection of greater spread in the error prior, due to higher variance in a smaller data set. However, the overall results obtained are similar in terms of picking out the correct model and the shapes of the parameter posteriors (Figure 5.4). As the size of the set B was the same in both cases (500×3), this suggests that for this example, the results are fairly robust to changes in the ratio of



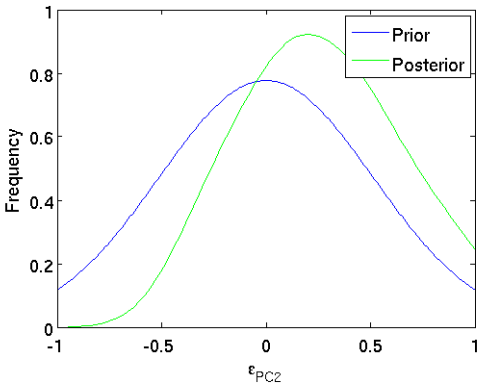
(a) Error posterior 1 for DDA+PA



(b) Error posterior 2 for DDA+PA



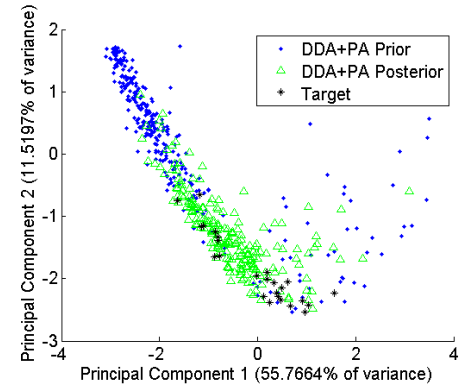
(c) Error posterior 1 for DD+LNK+PA



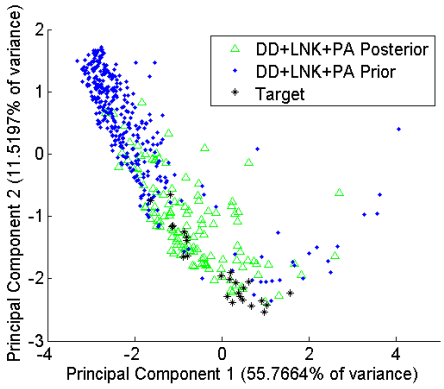
(d) Error posterior 2 for DD+LNK+PA

Figure 5.3: **Error priors and posteriors from ABC model-fitting to a set of 50 synthetic networks.**

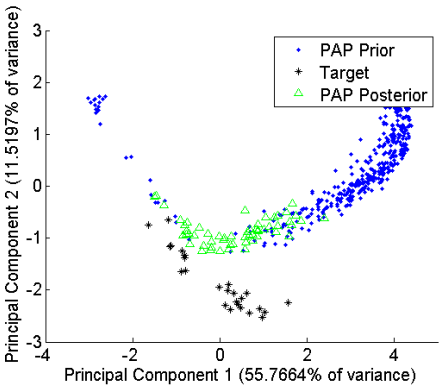
Prior and posterior distributions (plotted using MATLAB's `ksdensity()` function to perform smoothing using a Gaussian kernel, as discussed in Section 2.3) over the errors on the two features chosen via PCA, for the DDA+PA and DD+LNK+PA models (no samples were retained in the posterior for the third model, PAP).



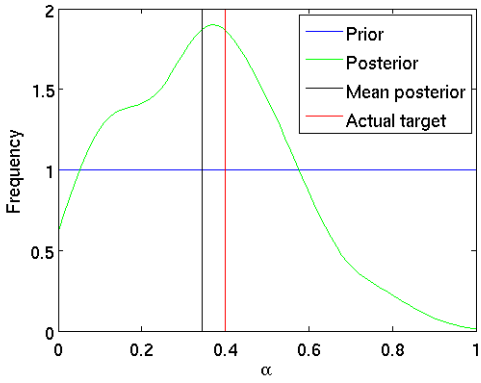
(a) PCA plot for DDA+PA



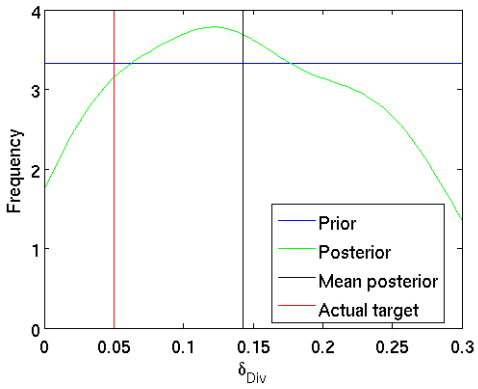
(b) PCA plot for DD+LNK+PA



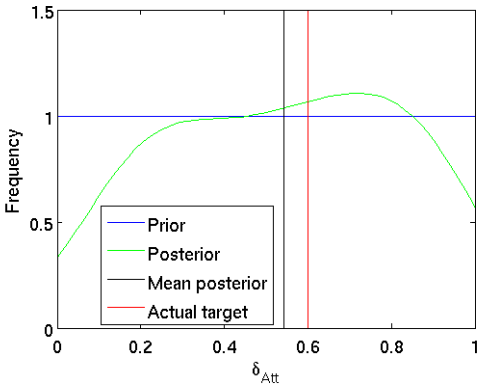
(c) PCA plot for PAP



(d) Posterior on α for DDA+PA



(e) Posterior on δ_{Div} for DDA+PA



(f) Posterior on δ_{Att} for DDA+PA

Figure 5.4: **Results of ABC model-fitting to a set of 25 synthetic networks from the DDA+PA model.**

(a)–(c): PCA space maps of the target set D (black), samples generated from priors over model parameters (blue), and retained samples (green). (d)–(f): Posterior distribution over DDA+PA model parameters. Prior uniform over $[0, 1]$ for α, δ_{Att}; uniform over $[0, 0.3]$ for δ_{Div}. Posterior mean in black, actual setting for generating D in red.

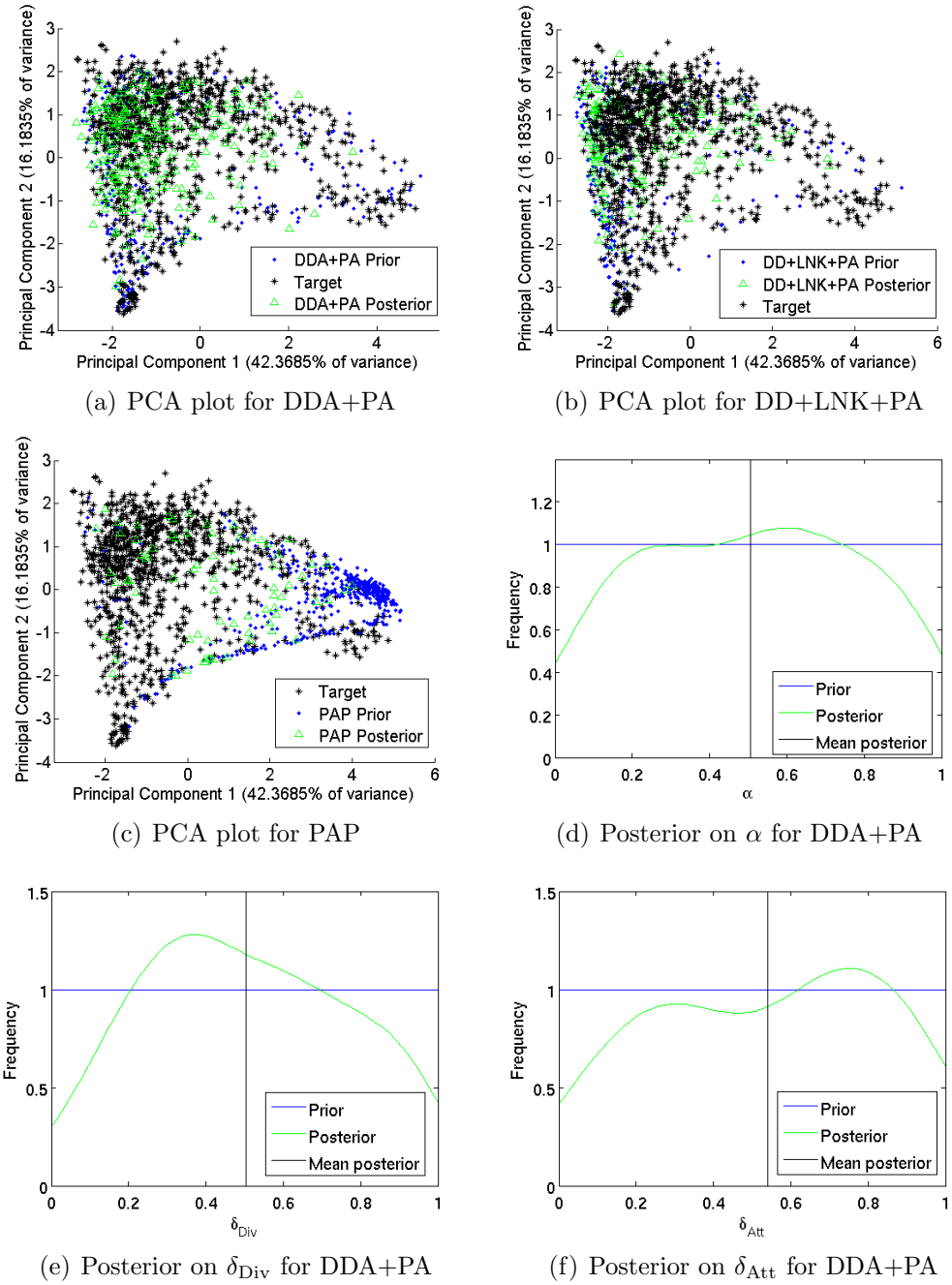
the sizes of the two sets.

5.5.2 Protein interaction networks

To test our ABC methodology on some real-world data, we took the 25 protein interaction networks (PINs) included in our set of 192 used in Chapter 3 [203] (see Appendix B). The unreliability of such data has been previously discussed in Section 1.2.2.1, and the set of 25 used here includes a mix of data from the different experimental methods mentioned there like Y2H and AP/MS. Also, the networks are for many different species, and it has already been observed that they show considerable variability in structure (Ref. [203] and Sections 3.3.2 and 4.2). Thus, it would seem unlikely that a single model could meaningfully explain this set of networks. However, we chose this as an example to study whether our approach might allow us to identify certain structural patterns even in such a seemingly incoherent data set, and to what extent those patterns could allow us to state that certain models might be more plausible than others in explaining aspects of PIN evolution in general.

As the 25 networks are all of different sizes, and some are very large, it is not feasible to try and obtain a single fit to the ensemble of the full networks. In past work on fitting models to PIN data via ABC, it has been typical to fit one network at a time [218, 219]; however we would like to be able to fit an ensemble of networks to be able to make use of an empirically determined error prior, as described in Section 5.4.2. In order to do this, we adopt the following approach: for each of our 25 real networks, we use snowball sampling [110] (see Section 1.1.4.10) to generate $\lfloor N/50 \rfloor$ samples of 50 nodes each[1], where N is the number of nodes in the respective network. This gives a total of 987 samples, which we designate as our target set D. Then, to constitute our background set B, we draw 500 samples from each of the 3 models

[1] As mentioned in Section 1.1.4.10, the choice of 50-node samples allows for fast computation of all our diagnostics. In Section 5.5.2.1, when fitting more reliable data sets for specific species, we will look at the effect of switching to 100-node samples.



(a) PCA plot for DDA+PA

(b) PCA plot for DD+LNK+PA

(c) PCA plot for PAP

(d) Posterior on α for DDA+PA

(e) Posterior on δ_{Div} for DDA+PA

(f) Posterior on δ_{Att} for DDA+PA

Figure 5.5: **Results of ABC model-fitting to a set of 25 protein interaction networks (PINs), via snowball sampling.**

(a)–(c): PCA space representations of the target set of 50-node PIN samples (black), samples generated from priors over model parameters (blue), and retained samples (green). As for Figure 5.2, the first principal component has strong correlations with network *energy* ($\rho \approx 0.99$) and *density* ($\rho \approx 0.93$). The second component is strongly correlated with statistics of the $avgNearestNeighbourDegree$ distribution, such as the mean ($\rho \approx -0.92$; feature names are described in Appendix A). (d)–(f): Posterior distribution over DDA+PA model parameters. Priors are all uniform over $[0, 1]$. The posterior mean is shown in black.

considered here, allowing the size to vary in addition to other model parameters: each time the number of nodes is drawn uniformly at random from the range $[50, 8205]$, which is the range of sizes in the actual set of 25 PINs. Subsequently, a single snowball sample of 50 nodes is taken from the generated network and added to B. Thus, B ultimately consists of 500 50-node samples for each of the three models, the samples having been obtained from full-size networks lying in the same size range as the real PINs. The sets B and D thereby defined are then used to carry out our ABC procedure as per Section 5.4.3.

The results are depicted in Figure 5.5. We note that despite the well-known unreliability and incompleteness of PIN data [15, 119, 270] (see Section 2.5), and the extra uncertainty introduced by our subsampling,[2] our approach is able to clearly distinguish PAP as a bad model relative to the other two, whilst DDA+PA appears to be identified as the most plausible model. These observations are in agreement with Ratmann *et al.* [218,219], and also with more general notions of gene duplication and divergence as evolutionary mechanisms [113, 201, 257, 284].

Examining which specific network features are primarily represented in the principal components can provide some insight into those aspects of network structure that best discriminate between the different models and real networks being considered, something which has been difficult to discern from previous studies that have been restricted to a small number of diagnostics. The first principal component here also correlates strongly with network energy and density; however the second component has the strongest correlations with features that are summary statistics of the distribution of *average nearest neighbour degree* (denoted for node i by $\bar{k}_i^{nn}$), which is defined for any given node as the average of the degrees of all the nodes directly linked to it: $\bar{k}_i^{nn} = \frac{1}{k_i} \sum_{j:A_{ij}>0} k_j$, where k_i is the degree of node i and A_{ij} indicates

[2]Here we use just a single sampling protocol and a single sample size, and thus we cannot estimate to what extent the results obtained might be an artefact of these particular choices. In Section 5.5.2.1, where we examine some more restricted and reliable PIN data sets, we attempt to look at how our results are affected by variation in these choices.

entry (i, j) of the adjacency matrix. For instance, the second component has a strong negative correlation ($\rho \approx -0.92$) with the mean value of $\bar{k}_i^{nn}$ over all nodes. This may be related to the presence of hubs[3] in the samples from these networks: nodes with very large degrees will raise the average nearest neighbour degree of all their neighbours, so a high mean value for this quantity may reflect the presence of many such hubs. Thus the strong negative correlation with the second principal component from Figure 5.5 suggests that most of our PIN samples may have fewer or less extreme hubs than those from networks from the PAP model (preferential attachment specifically favours the emergence of some nodes with very high degree, as discussed in Section 1.1.6.3).

The posteriors on the model parameters for DDA+PA are quite flat (compared to the results with the synthetic networks), suggesting that the PIN data is insufficiently constrained to pinpoint these, and allowing for the possibility that such parameters vary across different species. We now take a closer look at one possible example of this, where we also seek to examine to what extent our results are robust to variations in the sampling procedure and sample size.

5.5.2.1 Estimating rewiring rates

As an extension of the model comparison on the 25 PINs data set, we sought to delve deeper into what model-fitting to PINs could tell us about actual evolutionary mechanisms in biology. As mentioned, collectively fitting an ensemble of PINs from different species does not seem to indicate much about model parameters for DDA+PA; hence, one question that arises is whether there might be differences in how the PINs for different organisms evolve. One parameter of particular interest in the DDA+PA model is δ_{Div}, as this determines the likelihood of the interactions of a duplicated node diverging from its parent: one way of interpreting this is as a rewiring rate

[3]As discussed in Section 1.1.2.1, hubs have been defined in multiple specific ways; here we use the term in a general sense to refer to relatively high-degree nodes.

parameter, specifying how quickly new proteins can take on novel associations and functionality. Even though DDA+PA is a simplistic model which certainly cannot capture all the intricacies of PIN evolution, both our results here as well as earlier work [218, 219] appear to suggest its viability for replicating at least some aspects of PIN structure. More generally, it is widely believed that duplication-divergence mechanisms have a prominent role to play in the evolution of gene/protein interactions (see Section 1.1.6.7), and by attempting to fit a mathematical model thereof one might hope to at least roughly quantify and compare the rates at which the PINs of different species have evolved. Thus, by estimating the value of the δ_{Div} parameter via DDA+PA model fits to PINs for different species, we might hope to obtain some indication of whether the networks of certain organisms in effect rewire faster than others.

However, as noted in Section 2.5, existing PIN data is typically rather incomplete, and this is likely to greatly affect our estimates of model parameters (and affect it to varying extents for organisms with varying degrees of network completeness). Thus, we would like to somehow try and compensate for this. Recent work by Lewis *et al.* [166] developed a methodology for estimating the percentage coverage (i.e., the percentage of actual interactions detected in our data sets) of the PINs of different species, relative to the PIN of the yeast *Saccharomyces cerevisiae* (which, as the best studied organism, is assumed to have 100% coverage). This study looked at human and fruit fly (*Drosophila melanogaster*) networks in addition to yeast and estimated coverage levels of these to be 18% and 7.5% respectively. Here, we use the PIN data sets compiled for this study for the three species (summarised in Table 5.1), and make use of the corresponding coverage estimates to attempt to put them all on a level playing field with respect to the DDA+PA model that we seek to fit. We use the following modification of our ABC procedure for fitting the model to the 3 PINs:

- For $i \in \{Yeast, Human, Fly\}$; for $j \in \{1, 2, ..., 500\}$:

Table 5.1: **PIN data sets for estimation of rewiring rates.**

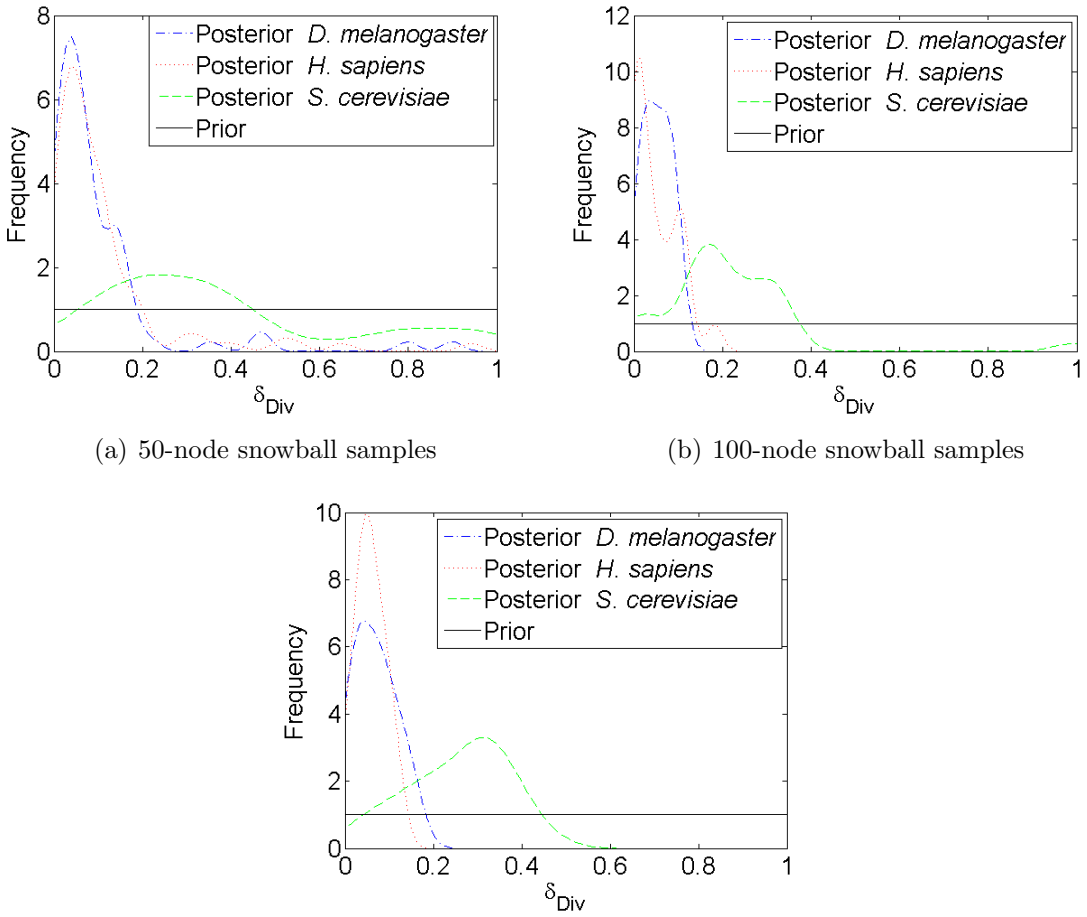
Species	Estimated gene count (S)	Nodes in PIN	Links in PIN	Estimated link coverage
S. cerevisiae	5,827	5,827	43,019	100%
D. melanogaster	14,000 [4, 14]	8,617	27,071	7.5%
H. sapiens	23,000 [131, 205]	11,601	53,738	18%

Data sets and coverage estimates from Lewis *et al.* [166]. Genome sizes for *D. melanogaster* and *H. sapiens* are our rounded estimates based on the cited references.

1. Draw parameters θ_{ij} from priors for model DDA+PA; set number of nodes N equal to S_i, an estimate of the genome size (in number of genes) of organism i
2. Sample a network G^{ij} from $P(\cdot|DDA + PA, N = S_i, \theta_{ij})$
3. If $i \in \{Human, Fly\}$: randomly remove $(100 - coverage_i)\%$ of the links from G^{ij} ($coverage_{Human} = 18$, $coverage_{Fly} = 7.5$)
4. Use snowball or forest fire sampling (see Section 1.1.4.10) to get a K-node sample G^{ij}_K from G^{ij} (we set K to 50 or 100, these being sizes for which the bulk of the network diagnostics we use can be computed within a feasible time of a few minutes)
5. Add G^{ij}_K to set B_i

This builds the background sets B_i for each of the three species; the remaining procedure is as in Section 5.4.3. The target sets D_i are built by taking $\lfloor N_i/K \rfloor$ K-node samples from the actual PIN for species i, where N_i is the number of nodes in that PIN; this is analogous to the sampling of the set of 25 PINs mentioned above. Here, however, we fit each of the three species separately and obtain posterior distributions on the model parameters for each one.

We carry out this ABC fitting procedure for the DDA+PA model for two settings of the snowball sample size, $K = 50$ and $K = 100$. We also repeat this for $K = 100$ using forest fire sampling, with the forward burning probability p set to 0.5 (which



(a) 50-node snowball samples

(b) 100-node snowball samples

(c) 100-node forest fire samples

Figure 5.6: **DDA+PA model fits suggest yeast PIN rewires faster than fruit fly and human.**

Results of our ABC procedure for fitting the DDA+PA model separately to protein interaction networks for yeast (*S. cerevisiae*), fruit fly (*D. melanogaster*), and human (*H. sapiens*) via snowball sampling and forest fire sampling. (a) Posterior distributions over the δ_{Div} parameter when using 50-node snowball samples. Priors are all uniform over $[0, 1]$. (b) Posteriors when using 100-node snowball samples. (c) Posteriors when using 100-node forest fire samples with burning probability 0.5.

is midway between the extreme choices of $p = 1$, equivalent to snowball sampling, and $p = 0$, equivalent to random node sampling; see Section 1.1.4.10). Figure 5.6 shows the posteriors obtained in each of the 3 cases for the δ_{Div} parameter, for each of the three species. Notably, the estimate for yeast is peaked around a considerably higher value than for the other two. There is some variation between the results with the 3 types of samples, for instance with the smaller (50-node samples) we see some mass in the posterior even at high values of δ_{Div}, but this vanishes for the two types of 100-node samples. This is likely to be because a smaller sample size allows for more variation in the types of samples that can be obtained from any given network. However the primary observation of a substantially higher δ_{Div} for the yeast PIN is quite consistent across the three sample types.

This observation suggests that, under the assumption of a DDA+PA-like evolutionary model, the yeast PIN rewires significantly faster than the fruit fly and human pins; in agreement with observations made in Lewis *et al.* [166] of relatively low conservation of protein-protein interactions in yeast across paralogs (pairs of proteins arising from a common ancestor via gene duplication), compared to the other two species. It has been noted earlier that smaller genomes often have faster rates of evolution [180, 209]; thus one explanation of our results might be that the small genome size of yeast implies that there is more evolutionary pressure for new proteins to rewire to enhance functional diversity.

5.6 Discussion

In this chapter, we have shown how our large library of network diagnostics allows us to develop a partly automated procedure for selecting network features or summary statistics for the purpose of model-fitting to networks using ABC. Here we use PCA as a simple means of finding statistics which are linear combinations of our full set

of network features, but an extension of the methodology may include using other approaches like feature selection to obtain a small number of summary statistics which may be more directly interpretable. We also make use of ensembles of networks to quantify uncertainty in network structure, and are able to make use of this to define a partly data-driven error prior for ABC, as opposed to arbitrary error thresholds that have typified previous work. Whilst our approach also involves arbitrariness in terms of choices such as the number of samples per model and the number of principal components, we believe it is easier to postulate fairly general guidelines for such choices, e.g., to use the minimum number of principal components that capture a certain percentage of the variance. One can then assess such a choice for multiple examples to evaluate its appropriateness. However, a choice of error threshold on a particular network feature in a particular setting is much more specific, and there is no easy way to formulate general guidelines for such choices on the basis of particular examples.

Given that earlier work has required tens of thousands of samples to fit the models of network evolution considered here, our ABC procedure provides a powerful way of doing this using relatively small amounts of data, just a few hundred samples per model. A key factor is that we appear to be able to capture aspects of network structure relevant for differentiating between models and parameter values in just 2 dimensions, as opposed to 5 or more used in previous studies [218, 219]. The partial automation of the choice of these features or summary statistics, as well as of the error priors on them, not only reduces the subjectiveness of these choices but also increases efficiency by identifying maximally informative feature combinations, thereby reducing the dimensionality of the space to be sampled. Additionally, an examination of the particular dimensions identified and what network characteristics they correspond to can help provide enhanced understanding of how the structures of model-generated and real-world networks match up, compared to previous work

that has focused on only a few structural diagnostics. One limitation of our approach is that it scales poorly with network size, due to the computational effort involved in evaluating all of our network features, many of which have runtime which grows super-polynomially with the number of nodes. Thus in our examples here we stick to small networks, and use subsampling to fit models to the bigger protein interaction networks. Further extensions of the approach, such as using feature selection to reduce the number of features to be computed and developing more efficient sampling methods, may improve the scalability and allow for larger networks to be fit directly.

We demonstrate our approach in the context of modelling the evolution of protein interaction networks. On a synthetic data set, our algorithm is able to recover model parameters fairly accurately with just a few hundred samples, even though we use simplistic rejection sampling. On some real data sets, we show how we can identify viable and bad models, and also use posteriors over model parameters to get an indication of quantities like evolutionary rates and how they compare across different organisms. Thus, alongside Section 3.5, we believe this provides another instance of how our high-throughput network analysis approach can serve to suggest feasible hypotheses about the structure and evolution of biological networks: specifically, in the examples used here, we suggest that a model like DDA+PA (a hybrid of duplication-divergence and preferential attachment processes) may partly explain the observed structural characteristics of protein interaction networks. This is in agreement with earlier work by Ratmann *et al.* [218, 219]; however, our methodology also allows for some addition speculation on the particular structural properties the model can reproduce, such as the average nearest neighbour degree discussed in Section 5.5.2. Our results also suggest that interactions in yeast (small genome size) may evolve faster than those in fruit flies and humans (larger genome sizes), which to our knowledge is a novel hypothesis. It is true that these hypotheses tend not to be very precise; but hopefully they can be useful as data-driven starting points to assist in guiding more

detailed experimental investigation of the underlying mechanisms.

Chapter 6

Conclusions

In this thesis we have sought to examine multiple issues relevant to the study of networks, with a particular focus on protein interaction networks. Here we first briefly recapitulate our key results, and then discuss in some more detail their implications and limitations, as well as directions for future research.

6.1 Key results

- We show that the proposed date/party hub dichotomy is not robust across PINs, and they do not generally have the structural attributes imputed to them. We find a lack of correspondence between the topological roles of hubs and their expression correlations, but demonstrate that link-centric (rather than node-centric) role definitions might be worth pursuing.

- We show how a simultaneous examination of a large number of networks using a large number of diagnostics can be useful for network comparison and organisation, and observe that some real-world network types appear to have highly specific structural properties.

- We demonstrate how our methodology enables identification of structural fea-

tures correlated with functional properties. For instance, many aspects of metabolic pathway networks are found to show significant correlation with evolutionary history; we also find features which strongly correlate with the solution length and runtime of the travelling salesman problem for a particular graph family, and can be thought of as relatively fast estimators of those quantities.

- We find that structural features correlate more over networks of a particular type than over diverse sets, suggesting that such increased correlations indicate particular structural constraints. We show how these can indicate distinctive aspects of structure, such as low local clustering and high global cohesion in granular contact networks.

- We develop a methodology for model-fitting to networks with partial automation of previously manual choices, which also allows for fitting of network ensembles rather than one network at a time. This accurately recovers model parameters on some synthetic data, using relatively few model samples. On real PINs, our results indicate how evolutionary rates might differ across species, in particular that yeast PINs may rewire faster than bigger genomes (fruit fly, human).

6.2 Roles in protein interaction networks

We began in Chapter 2 by looking at the roles played by hubs in protein interaction networks (PINs), specifically in light of the proposed categorisation into date and party hubs. We show, for multiple PINs, that the distribution of hub expression correlations with their interaction partners is not robustly bimodal, thus undermining a primary basis for the date-party dichotomy. Our results also indicate that the proposed date and party hubs do not have the structural roles suggested for them: date hubs in general are not as critical to global network connectivity as has been

implied, and in fact only a very small fraction of these hubs appear to have a major role in this respect. Using a community detection approach to assigning node roles, we show a general lack of correspondence between topological roles of PIN hubs and their expression correlations, thus suggesting that the basic premise of date-party type classifications was flawed.

We also adopt a link-centric (as opposed to node-centric) perspective to roles in PINs (see Section 2.6); we find a lack of correspondence between the geodesic betweenness centrality of a link (an indicator of its importance to connectivity) and the expression correlation of the proteins linked, thus mirroring the observations with hubs. However, we find substantial negative correlations between link betweenness centrality and the functional similarity of the linked proteins, implying that more 'central' links tend to be between less similar proteins. This is somewhat reminiscent of the weak/strong tie distinction in social networks, and suggests that a link-centric approach to roles in PINs might be meaningful.

Our comparison of different PIN data sets also reaffirms earlier observations about the unreliability and incompleteness of such data. This makes it difficult to obtain any conclusive results regarding these networks. However, as more and higher-quality data sets become available, one direction for future work could be to look at link-centric properties across a larger number of networks and species and see to what extent the correlations observed here hold up. It would also be of interest to explore the role of expression levels in greater detail. The notion of date and party hubs emerged out of an attempt to combine interaction and expression data to obtain some information about the dynamics of otherwise static PINs. Whilst that particular idea may have been misguided, it is certainly the case that better knowledge of what different parts of a network are 'activated' at different times or under different conditions is likely to be very helpful in understanding its organisation and functioning. As noted in Section 1.2.2.2, one caveat with attempting to use mRNA expression data for this

purpose is that there may in fact be little correlation between mRNA levels and protein levels [91]. Thus, using data on actual quantitative protein expression, which has started to become available on a substantive scale relatively recently (e.g., the Human Protein Atlas [12]), is certainly a possibility worth examining.

6.3 High-throughput analysis of networks

Our examination of date and party hubs demonstrates that looking for patterns or correlations on the basis of just one or a few network properties can be misleading; characterising hubs via multiple measures such as betweenness centrality and community-based properties led us to conclude that in fact date and party hubs do not fit the structural roles earlier imputed to them. We also noted that ideas arising from the study of networks in a different discipline like sociology could be of relevance to PINs. Motivated by these, we chose to expand the scope of the thesis towards an attempt to develop a more comprehensive methodology for investigating networks and network characteristics.

Our approach, introduced in Chapter 3, aims to examine simultaneously a large number of networks of different types, using a large number of network diagnostics from multiple disciplines. We show how this can be useful for comparing and organising both networks and diagnostics, and observe that some types of real-world networks appear to show highly specific or constrained structural properties. Given that our representation of networks as feature vectors involves substantial loss of information, and that a network itself is an abstracted representation of a complicated real-world system such as metabolic pathways, it is perhaps somewhat surprising that despite discarding so much detail, the representation still retains sufficient information that in many cases one is able to pinpoint it as a metabolic network, as opposed to any of the other diverse kinds of networks we look at.

We also look at how the use of the large set of network diagnostics enables the identification of structural features that are correlated with functional properties. In one case study, we find that many aspects of metabolic pathway networks show a significant correlation with their evolutionary history. For instance, for the data we use, the rich-club coefficient is the network property with the strongest phylogenetic signal, though it is strongly correlated with simpler features, such as link density, that also display a strong signal. Observations of this sort can help to motivate particular evolutionary mechanisms or models for these networks that might reproduce the observed signals, as discussed in Section 3.5.

In another case study (see Section 3.4), using synthetic networks generated from a preferential attachment model, we show how one can detect structural features that can serve as predictors of the solution or runtime of hard graph-theoretic problems; the particular example we use is the travelling salesman problem. We find a number of features that correlate very strongly with solution length in particular, and are relatively quick to compute (as compared to running a heuristic algorithm to solve the full problem). Specific examples of such features include the maximum node betweenness centrality and the average cyclic coefficient, which show respectively strong positive and negative correlations with the TSP solution length, across a set of 3 different solvers we examine. It would be difficult to highlight such specific features without the sort of large-scale study employed here. We also make explicit the process of testing many different hypotheses (i.e., features) and assessing their statistical significance having taken this into account. This is as opposed to many traditional studies where a single hypothesis may be tested and published as being statistically significant, without having accounted for the fact that many other hypotheses may have been tested too but not found significant and thus never published, leading to a false sense of significance (this is also known as 'fishing') [51, 132].

There is of course a lot of scope for extending and improving upon the results

obtained via our high-throughput methodology. When looking at structure-function correlations, we make particular choices of data sets, and we have also chosen a particular set of network diagnostics to examine. It is also the case that not all these diagnostics were feasibly computable for all the networks, and in most case studies only a subset of features could be used. We include some features computed via subsampling the network, but as discussed in Section 3.2 we do this only on an experimental basis using a single sampling method and sample size, and these features do not appear in any of our results. Given more time it would be desirable to properly examine the relationship between features on subsamples versus the full network, to assess how robust various features are to such subsampling and to what extent it might be possible to draw conclusions about a full network on the basis of samples obtained in a particular manner.

Regarding the other choices involved in our methodology, in our case studies we have attempted to vary some of them to examine how they affect our results, but again this could be done more thoroughly given more time, to obtain a better notion of the generality of our observations. The set of diagnostics (listed in Appendix A) can certainly be built upon further, and in particular greater coverage of diagnostics applicable to weighted and directed networks could be aimed for. There are also a variety of other sorts of case studies that could be attempted with our methodology: one example would be change detection in time-evolving networks, i.e., obtaining snapshots of the network at multiple points in time, mapping them all to our feature space and then studying the rate and pattern of evolution of the different features. It may be that correlations can be found between changes in certain network features and some external events; for instance, studies of changes in the co-voting network of U.S. Senators have found a strong correspondence between modularity and a measure of political polarisation [203, 274].

6.4 Feature degeneracies and network entropies

We also find substantial correlations amongst the different network features we examine. In Chapter 4, we examine how network features correlate with each other over different sets of networks and find that networks of a particular type (e.g., brain connectivity or Facebook) show substantially greater feature-feature correlations than are observed for diverse sets of networks. This suggests that feature correlations or degeneracies may be indicative of particular structural constraints. For a set of granular (spatially-embedded) networks (see Section 4.2.1), we show that such correlations can provide an indication of which aspects of structure show distinctive behaviour, relative to a null model such as random geometric graphs. For instance, we pick out the group degree centrality and the Fiedler value as two features that show a substantial increase in correlations with other features on the granular networks, compared to the random ones; and it turns out that these two features are much more constrained in their distribution on the granular networks, in a manner that is in accord with earlier observations that these networks have relatively low local connectivity and relatively high global cohesion [32].

The observation of feature correlations corresponding to structural constraints also suggested a relationship to the notion of network entropy, a way of quantifying the amount of spread or uncertainty within a given network ensemble. We define a notion of statistical entropy in a low-dimensional feature space (obtained via PCA), and examine whether we can relate this in any way to the thermodynamic entropy of certain chosen ensembles, for which the latter quantity is analytically obtainable (see Sections 4.4 and 4.5). We observe a rough correspondence in the two types of entropy for Erdős-Rényi ensembles, but for the other two kinds we study, Watts-Strogatz networks and what we call 'modular networks', we find that the statistical entropy in 2-dimensional PCA space shows very little variation with model parameters that influence the thermodynamic entropy. One reason for this appears to be that the first

two principal components are largely picking up variation between ensembles, rather than within ensembles, and thus the space defined by them is not really capturing the amount of spread within an ensemble.

Thus, whilst our results show that mapping networks to a low-dimensional PCA space can be useful for picking out directions that most effectively distinguish between ensembles with differing parameter settings, the meaning of statistical entropy in such a space and its relation (if any) to thermodynamic entropy is unclear. Our study of these concepts of network entropy was quite preliminary due to time constraints, but it appears worthwhile to explore further to what extent and under what conditions the feature space entropy can be a meaningful measure of ensemble uncertainty. Looking at lower principal components (beyond the second), or using alternative methods like Isomap to carry out dimensionality reduction, may provide ways of exploring this question.

6.5 Bayesian model-fitting for networks

The observation that we can use PCA on our feature space to obtain directions that correspond to variations in model parameters suggested that the PCA space might be useful for comparing models to actual data, to see how well a given model with a given set of parameters reproduces the structure observed in some real network(s). In Chapter 5, we make use of this idea to develop an approximate Bayesian computation (ABC) methodology for fitting generative models to networks. Our approach allows for partial automation of choices that have been made manually in previous use of ABC for model-fitting to networks, in particular the choice of which network summary statistics to use and the size of the error prior. It also allows one to fit ensembles of networks rather than one network at a time as has been done previously. We show that our approach accurately recovers model parameters on examples of synthetic data,

using a relatively small number of samples (despite our use of simple but inefficient rejection sampling). On real PINs (see Section 5.5.2), our results provide an indication of how evolutionary rates across species might differ, and in particular suggest that under the assumption of a duplication-divergence mechanism, divergence rates in yeast may be faster than for bigger genomes like fruit fly and human. This result appears to be in agreement with the observation of lower conservation of interactions across paralogous proteins in yeast, compared to the other two species [166].

The ABC approach presented here also offers plenty of scope for extension and improvement. For the real PINs, we use subsampling in order to fit a model to the data, due to the difficulty of computing some diagnostics on large networks and also in order to have an ensemble of networks to fit, rather than a single one. Whilst our results broadly seem to be consistent across 3 different choices of sampling procedure, it is desirable to probe this further to investigate how robust the results are to different ways of obtaining network samples, and also whether it is possible to fit the full networks for comparison. The ABC procedure itself involves making choices of how many PCA dimensions to consider, how many samples to generate from the model(s) being fit, the number of networks in the real ensemble, and the use of a Gaussian-shaped error prior. Whilst we attempt to provide some justification for our particular choices and also allow for some variation, a more thorough study of the effects of different possible choices remains to be done. We also use the simplest possible model sampling protocol, i.e., rejection sampling, which involves throwing away a lot of data, and future extensions may include developing a more intelligent sampling algorithm, such as Markov chain Monte Carlo (MCMC).

6.6 Summary

We have attempted to adopt and develop some new approaches to the study of networks, with a focus on protein interaction networks. Our case studies provide examples of how these can generate insights in the context of specific scientific questions, and can assist in guiding and motivating further investigation. In particular, for protein interaction networks, we have presented results that throw some light on several different aspects, such as roles and modularity, their interplay (or lack thereof) with gene expression dynamics, and mechanisms and rates of evolution. More generally, our high-throughput methodology provides a way of leveraging both large quantities of data and a large number of ways of characterising that data, in order to illuminate relationships amongst both networks and ways of thinking about networks, as well as relationships between structural properties of networks and aspects of their functionality. Whilst the methodology involves abstracting away much detail and thus leaves work to be done in relating any insights obtained back to the real-world system(s) under consideration, we believe it can serve as a tool to complement more detailed human efforts and experimentation, and help to focus them in fruitful directions.

Appendix A

List of Network Features

Here we list all of the diagnostics and summary statistics[1] that were utilised in the high-throughput methodology presented in this thesis. For each diagnostic, the short name given is that generally used to refer to it in the main thesis. We also specify whether each diagnostic returns a vector or function over nodes (F) or a set of communities (C), and, where necessary, provide a reference for or description of the diagnostic. For summary statistics, short names use a subscript to denote the summary (e.g., the maximum of the degree distribution is $degree_{\max}$); shorthand summary names used in such subscripts (where applicable) are given in parentheses.

Table A.1: List of network diagnostics.

Short name	Full name	Notes	Reference
Connectivity			
degree	Degree distribution	F	
avgNearestNeighbourDegree	Average of degrees of adjacent nodes	F	
assortativeCoefficient	Assortative coefficient		Degree assortativity [191]
density	Density		Number of edges present as fraction of all possible edges
fractionArticulation	Fraction of articulation nodes		Fraction of nodes whose removal results in a disconnected graph
erosionTime	Erosion time		Number of steps for dilation,

[1]These lists were prepared using a template obtained from Gabriel Villar.

Table A.1: List of network diagnostics.

Short name	Full name	Notes	Reference
			starting from node of maximum degree, to cover whole network, normalised to network size [65]
dilationTime	Dilation time		Number of steps for erosion, starting from nodes of degree 1, to cover whole network, normalised to network size [65]
fraction2core	Fraction of vertices comprising 2-core		Fraction of nodes that form 2-core
fraction3core	Fraction of vertices comprising 3-core		Fraction of nodes that form 3-core
fraction4core	Fraction of vertices comprising 4-core		Fraction of nodes that form 4-core
richClub	Rich-club index		[64]
richClubNormalised	Normalised rich-club index		[64]
Centrality			
degreeCentrality	Degree centrality	F	[272]
degreeCentralityGroup	Group degree centrality		[272]
betweenCentrality	Betweenness centrality	F	[272]
betweenCentralityGroup	Group betweenness centrality		[272]
closeness	Closeness	F	[272]
closenessGroup	Group closeness		[272]
evectorCentrality	Eigenvector centrality	F	[195]
subgraphCentrality	Subgraph centrality	F	[65]
subgraphCentralisation	Subgraph centralisation		[66, 79]
bipartivity	Estrada's measure of bipartivity		[79]
infoCentrality	Information centrality	F	[272]
infoCentralityGroup	Group information centrality		[272]
vulnerability	Vulnerability	F	[65, 109]
Community			
modularity	Spectrally optimised modularity		[193, 194]
modularityFast	Louvain optimised modularity		[48]
greedyPartitionEntropy	Entropy of Louvain partition		[48]
spectralf	Newman's spectral community detection	C	[193]
greedyComm	Louvain community detection	C	[48]
pottsModel	Potts model community detection	C	[220]
infomap	Infomap community detection	C	[226]
Clustering			
transitivity	Transitivity		
clusteringCoeff	Clustering coefficient	F	[65]
clustSofferGlobalMean	Global mean Soffer clustering coefficient		[246]

Table A.1: List of network diagnostics.

Short name	Full name	Notes	Reference
$clustSofferLocalMean$	Local mean Soffer clustering coefficient		[246]
Distance			
$diameter$	Graph diameter		
$radius$	Graph radius		
$szegedIndex$	Szeged index		[145]
$cyclicCoefficient$	Cyclic coefficient	F	[148]
$geodesicDistanceMean$	Mean geodesic distance		
$geodesicDistanceVar$	Variance of geodesic distance		
$harmonicMeanGeoDist$	Harmonic mean geodesic distance		
Complexity			
$cyclomaticNumber$	Cyclomatic number		[149]
$edgeFraction$	Edge fraction		Number of edges as fraction of the maximum possible
$connectivity$	Connectivity		[149]
$logNumSpanningTrees$	log(number of spanning trees)		[149]
$graphIndexComplexity$	Graph index complexity		[149]
$mediumArticulation$	Medium articulation		[149]
$efficiency$	Efficiency		[160]
$efficiencyComplexity$	Efficiency complexity		[149]
$offDiagonalComplexity$	Off-diagonal complexity		[63]
$chromaticNumber$	Chromatic number		Minimum colours for graph colouring
$tspl$	TSP length from cross-entropy algorithm		[3, 68, 232]
$tspl_{ga}$	TSP length from genetic algorithm		[2, 128]
$tspl_{sa}$	TSP length from simulated annealing		[1, 57]
Spectral			
$largestEigenvalue$	Largest eigenvalue		
$spectralScalingDeviations$	Deviations from 'perfect spectral scaling'	F	[78]
$algebraicConnectivity$	Algebraic connectivity		Second smallest eigenvalue of graph Laplacian
$algebraicConnectivityVector$	Algebraic connectivity vector	F	Eigenvector corresponding to algebraic connectivity
$fiedlerValue$	Fiedler value		Smallest non-zero eigenvalue of graph Laplacian
Statistical physics			
$energy$	Energy		[65]
$entropy$	Entropy		[65]
Motif			

Table A.1: List of network diagnostics.

Short name	Full name	Notes	Reference
$fraction3motifs$	Fraction of 3-motifs		[186]
$fraction4motifs$	Fraction of 4-motifs		[186]
Size			
$numNodes$	Number of nodes		
$numEdges$	Number of edges		
$totStrength$	Sum of all link weights		
Model			
$ergm\_edges$	Exponential random graph model for edges		Log-likelihood of model fit to the edge count [206]
$fitPowerLawAlpha$	Fitted power law exponent for degrees		[62]
$fitPowerLawP$	p-value of power law fit to degrees		[62]

Table A.2: List of distribution summary statistics.

Central tendency	Dispersion	Shape	Model fit log-likelihoods[1]
Mean	Minimum (min)	Kurtosis	Normal
Geometric mean (geomean)	Maximum (max)	Skewness	Log-normal
Harmonic mean (harmmean)	Variance (var)		Exponential
Mean excluding 10% outliers (trimmean10)	Range		Extreme value
RMS of positive values (posrms)	Inter-quartile range (iqr)		Gamma
RMS of negative values (negrms)	Mean absolute deviation (meanad)		Weibull (wbl)
	Median absolute deviation (medad)		

[1] These features return the log-likelihood of fitting the corresponding model distribution to the actual data. In shorthand feature names used in the main thesis, these are denoted by a subscript of the form $fit: distribution$, e.g., $betweenCentrality_{\text{fit}:\ \text{wbl}}$.

Table A.3: List of community structure summary statistics.

Partition summaries[1]	**Distributions over nodes**
Number of communities (numComm)	Node role counts[2]
Partition entropy (entropy)	Role entropy[3]
Number of inter-community links (icl)	Within-module degree variability (wmdPCAwt)[4]
	Participation coefficient variability (pcPCAwt)[4]

[1] For multi-resolution community detection using the Potts method [220], we compute the values of each of these (and additionally for numComm and entropy, finite-difference approximations of their first and second derivatives), at 10 different evenly-spaced settings of the resolution parameter [203]. For numComm and entropy, we also add a feature recording the area under the curve for these quantities over all 10 resolutions. In shorthand feature names used in the thesis, we add to the end of the subscript $\_resN$ to denote the value at resolution N; $\_df\_resN$ and $\_d2f\_resN$ for the first and second derivatives respectively; and $\_auc$ for area under the curve. For example, $pottsModel_{\text{entropy\_res1}}$ or $pottsModel_{\text{numComm\_auc}}$.

[2] We assign one of the 7 Guimerà-Amaral roles [114] to each node, then count the fraction of nodes in each role and report these as 7 separate features with subscript $roleN$ corresponding to role N; for example, $infomap_{\text{role1}}$.

[3] This is the entropy of the distribution of the nodes into the 7 Guimerá-Amaral roles [114]; we report it as a feature with subscript $roleEntropy$.

[4] To compute these, we carry out PCA on the set of nodes in the two-dimensional space defined by the within-module degree and the participation coefficient [114], and return the weights of the two measures in the first principal component obtained.

Appendix B

Set of 192 Real-World Networks

The set of 192 networks used for several of the case studies in this thesis was obtained from Dan Fenn, who compiled it for use in his own D.Phil. thesis [87] as well an associated manuscript [203]. The details of these networks presented here have also largely been collated from these two sources.

Table B.1: List of 192 real-world networks.

Index	Name	Category	Weighted	Nodes	Links	Reference
1	Human brain cortex: participant A1	Brain	Y	994	13,520	[118]
2	Human brain cortex: participant A2	Brain	Y	987	14,865	[118]
3	Human brain cortex: participant B	Brain	Y	980	14,222	[118]
4	Human brain cortex: participant D	Brain	Y	996	14,851	[118]
5	Human brain cortex: participant E	Brain	Y	992	14,372	[118]
6	Human brain cortex: participant C	Brain	Y	996	14,933	[118]
7	Cat brain: cortical	Brain	Y	52	515	[235]
8	Cat brain: cortical/thalmic	Brain	Y	95	1,170	[235]
9	Macaque brain: cortical	Brain	N	47	313	[85]
10	Macaque brain: visual/sensory cortex	Brain	N	71	438	[85]
11	Macaque brain: visual cortex 1	Brain	N	30	190	[279]
12	Macaque brain: visual cortex 2	Brain	N	32	194	[279]
13	Co-authorship: astrophysics	Collaboration	Y	14,845	119,652	[190]
14	Co-authorship: comp. geometry	Collaboration	Y	3,621	9,461	[7, 69]
15	Co-authorship: condensed matter	Collaboration	Y	13,861	44,619	[190]
16	Co-authorship: Erdős	Collaboration	N	6,927	11,850	[8]
17	Co-authorship: high energy theory	Collaboration	Y	5,835	13,815	[190]
18	Co-authorship: network science	Collaboration	Y	379	914	[193]
19	Hollywood film music	Collaboration	Y	39	219	[84]
20	Jazz collaboration	Collaboration	N	198	2,742	[108]
21	Facebook: Caltech	Facebook	N	762	16,651	[263]
22	Facebook: Cornell	Facebook	N	18,621	790,753	[263]
23	Facebook: Dartmouth	Facebook	N	7,677	304,065	[263]
24	Facebook: Georgetown	Facebook	N	9,388	425,619	[263]

Table B.1: List of 192 real-world networks.

Index	Name	Category	Weighted	Nodes	Links	Reference
25	Facebook: Harvard	Facebook	N	15,086	824,595	[263]
26	Facebook: Indiana	Facebook	N	29,732	1,305,757	[263]
27	Facebook: MIT	Facebook	N	6,402	251,230	[263]
28	Facebook: NYU	Facebook	Y	21,623	715,673	[263]
29	Facebook: Oklahoma	Facebook	N	17,420	892,524	[263]
30	Facebook: Texas80	Facebook	N	31,538	1,219,639	[263]
31	Facebook: Trinity	Facebook	N	2,613	111,996	[263]
32	Facebook: UCSD	Facebook	N	14,936	443,215	[263]
33	Facebook: UNC	Facebook	N	18,158	766,796	[263]
34	Facebook: USF	Facebook	N	13,367	321,209	[263]
35	Facebook: Wesleyan	Facebook	N	3,591	138,034	[263]
36	NYSE: 1980-1999	Financial	Y	477	113,526	[202]
37	NYSE: 1980-1983	Financial	Y	477	113,526	[202]
38	NYSE: 1984-1987	Financial	Y	477	113,526	[202]
39	NYSE: 1988-1991	Financial	Y	477	113,526	[202]
40	NYSE: 1992-1995	Financial	Y	477	113,526	[202]
41	NYSE: 1996-1999	Financial	Y	477	113,526	[202]
42	Phanerochaete velutina control1-2	Fungal	Y	117	136	[203]
43	Phanerochaete velutina control1-5	Fungal	Y	526	588	[203]
44	Phanerochaete velutina control1-8	Fungal	Y	721	821	[203]
45	Phanerochaete velutina control1-11	Fungal	Y	823	954	[203]
46	Phanerochaete velutina control7-2	Fungal	Y	232	240	[203]
47	Phanerochaete velutina control7-5	Fungal	Y	816	874	[203]
48	Phanerochaete velutina control7-8	Fungal	Y	1,113	1,303	[203]
49	Phanerochaete velutina control7-11	Fungal	Y	1,205	1,469	[203]
50	Phanerochaete velutina control4-2	Fungal	Y	461	490	[203]
51	Phanerochaete velutina control4-5	Fungal	Y	1,380	1,476	[203]
52	Phanerochaete velutina control4-8	Fungal	Y	1,869	2,061	[203]
53	Phanerochaete velutina control4-11	Fungal	Y	2,190	2,431	[203]
54	Online Dictionary of Computing	Language	Y	13,356	91,471	[35]
55	Online Dictionary Of Information Science	Language	Y	2,898	16,376	[9, 69]
56	Reuters 9/11 news	Language	Y	13,308	148,035	[138]
57	Roget's thesaurus	Language	N	994	3,640	[10, 69]
58	Word adjacency: English	Language	N	7,377	44,205	[185]
59	Word adjacency: French	Language	N	8,308	23,832	[185]
60	Word adjacency: Japanese	Language	N	2,698	7,995	[185]
61	Word adjacency: Spanish	Language	N	11,558	43,050	[185]
62	Metabolic: CE	Metabolic	N	453	2,025	[137]
63	Metabolic: CL	Metabolic	N	382	1,646	[137]
64	Metabolic: CQ	Metabolic	N	187	663	[137]
65	Metabolic: CT	Metabolic	N	211	772	[137]
66	Metabolic: DR	Metabolic	N	800	3,789	[137]
67	Metabolic: HI	Metabolic	N	505	2,325	[137]
68	Metabolic: NM	Metabolic	N	369	1,708	[137]
69	Metabolic: OS	Metabolic	N	285	1,168	[137]
70	Metabolic: PA	Metabolic	N	720	3,429	[137]
71	Metabolic: PG	Metabolic	N	412	1,772	[137]
72	Metabolic: PH	Metabolic	N	318	1,394	[137]
73	Metabolic: PN	Metabolic	N	405	1,829	[137]
74	Metabolic: SC	Metabolic	N	552	2,595	[137]
75	Metabolic: ST	Metabolic	N	391	1,756	[137]
76	Metabolic: TP	Metabolic	N	194	788	[137]
77	Bill cosponsorship: U.S. House 96	Political: cosponsorship	Y	438	95,529	[92, 93]

Table B.1: List of 192 real-world networks.

Index	Name	Category	Weighted	Nodes	Links	Reference
78	Bill cosponsorship: U.S. House 97	Political: cosponsorship	Y	435	94,374	[92, 93]
79	Bill cosponsorship: U.S. House 98	Political: cosponsorship	Y	437	95,256	[92, 93]
80	Bill cosponsorship: U.S. House 99	Political: cosponsorship	Y	437	94,999	[92, 93]
81	Bill cosponsorship: U.S. House 100	Political: cosponsorship	Y	439	96,125	[92, 93]
82	Bill cosponsorship: U.S. House 101	Political: cosponsorship	Y	437	95,263	[92, 93]
83	Bill cosponsorship: U.S. House 102	Political: cosponsorship	Y	437	95,051	[92, 93]
84	Bill cosponsorship: U.S. House 103	Political: cosponsorship	Y	437	95,028	[92, 93]
85	Bill cosponsorship: U.S. House 104	Political: cosponsorship	Y	439	95,925	[92, 93]
86	Bill cosponsorship: U.S. House 105	Political: cosponsorship	Y	442	97,373	[92, 93]
87	Bill cosponsorship: U.S. House 106	Political: cosponsorship	Y	436	94,820	[92, 93]
88	Bill cosponsorship: U.S. House 107	Political: cosponsorship	Y	442	97,233	[92, 93]
89	Bill cosponsorship: U.S. House 108	Political: cosponsorship	Y	439	96,104	[92, 93]
90	Bill cosponsorship: U.S. Senate 96	Political: cosponsorship	Y	101	5,050	[92, 93]
91	Bill cosponsorship: U.S. Senate 97	Political: cosponsorship	Y	101	5,050	[92, 93]
92	Bill cosponsorship: U.S. Senate 98	Political: cosponsorship	Y	101	5,050	[92, 93]
93	Bill cosponsorship: U.S. Senate 99	Political: cosponsorship	Y	101	5,049	[92, 93]
94	Bill cosponsorship: U.S. Senate 100	Political: cosponsorship	Y	101	5,050	[92, 93]
95	Bill cosponsorship: U.S. Senate 101	Political: cosponsorship	Y	100	4,950	[92, 93]
96	Bill cosponsorship: U.S. Senate 102	Political: cosponsorship	Y	102	5,142	[92, 93]
97	Bill cosponsorship: U.S. Senate 103	Political: cosponsorship	Y	101	5,050	[92, 93]
98	Bill cosponsorship: U.S. Senate 104	Political: cosponsorship	Y	102	5,151	[92, 93]
99	Bill cosponsorship: U.S. Senate 105	Political: cosponsorship	Y	100	4,950	[92, 93]
100	Bill cosponsorship: U.S. Senate 106	Political: cosponsorship	Y	102	5,151	[92, 93]
101	Bill cosponsorship: U.S. Senate 107	Political: cosponsorship	Y	101	5,049	[92, 93]
102	Bill cosponsorship: U.S. Senate 108	Political: cosponsorship	Y	100	4,950	[92, 93]
103	Committees: U.S. House 101, comms.	Political: committee	N	159	3,610	[211, 212]
104	Committees: U.S. House 102, comms.	Political: committee	N	163	4,093	[211, 212]
105	Committees: U.S. House 103, comms.	Political: committee	N	141	2,983	[211, 212]
106	Committees: U.S. House 104, comms.	Political: committee	N	106	1,839	[211, 212]
107	Committees: U.S. House 105, comms.	Political: committee	N	108	1,997	[211, 212]
108	Committees: U.S. House 106, comms.	Political: committee	N	107	2,031	[211, 212]
109	Committees: U.S. House 107, comms.	Political: committee	N	113	2,429	[211, 212]
110	Committees: U.S. House 108, comms.	Political: committee	N	118	2,905	[211, 212]
111	Committees: U.S. House 101, Reps.	Political: committee	N	434	18,714	[211, 212]
112	Committees: U.S. House 102, Reps.	Political: committee	N	436	20,134	[211, 212]
113	Committees: U.S. House 103, Reps.	Political: committee	N	437	18,212	[211, 212]
114	Committees: U.S. House 104, Reps.	Political: committee	N	432	17,130	[211, 212]
115	Committees: U.S. House 105, Reps.	Political: committee	N	435	18,297	[211, 212]
116	Committees: U.S. House 106, Reps.	Political: committee	N	435	18,832	[211, 212]
117	Committees: U.S. House 107, Reps.	Political: committee	N	434	19,824	[211, 212]
118	Committees: U.S. House 108, Reps.	Political: committee	N	437	21,214	[211, 212]
119	Roll call: U.S. House 101	Political: voting	Y	440	96,505	[179, 210, 274]
120	Roll call: U.S. House 102	Political: voting	Y	441	96,811	[179, 210, 274]
121	Roll call: U.S. House 103	Political: voting	Y	441	96,348	[179, 210, 274]
122	Roll call: U.S. House 104	Political: voting	Y	445	98,720	[179, 210, 274]
123	Roll call: U.S. House 105	Political: voting	Y	443	97,841	[179, 210, 274]
124	Roll call: U.S. House 106	Political: voting	Y	440	96,557	[179, 210, 274]
125	Roll call: U.S. House 107	Political: voting	Y	443	97,816	[179, 210, 274]
126	Roll call: U.S. House 108	Political: voting	Y	440	96,561	[179, 210, 274]
127	Roll call: U.S. Senate 101	Political: voting	Y	100	4,950	[179, 210, 274]
128	Roll call: U.S. Senate 102	Political: voting	Y	102	5,148	[179, 210, 274]
129	Roll call: U.S. Senate 103	Political: voting	Y	102	5,080	[179, 210, 274]
130	Roll call: U.S. Senate 104	Political: voting	Y	103	5,247	[179, 210, 274]

Table B.1: List of 192 real-world networks.

Index	Name	Category	Weighted	Nodes	Links	Reference
131	Roll call: U.S. Senate 105	Political: voting	Y	100	4,950	[179, 210, 274]
132	Roll call: U.S. Senate 106	Political: voting	Y	102	5,148	[179, 210, 274]
133	Roll call: U.S. Senate 107	Political: voting	Y	102	5,148	[179, 210, 274]
134	Roll call: U.S. Senate 108	Political: voting	Y	100	4,950	[179, 210, 274]
135	U.K. House of Commons voting: 1992-1997	Political: voting	Y	668	220,761	[11]
136	U.K. House of Commons voting: 1997-2001	Political: voting	Y	671	223,092	[11]
137	U.K. House of Commons voting: 2001-2005	Political: voting	Y	657	215,246	[11]
138	U.N. resolutions 59	Political: voting	Y	191	18,140	[269]
139	U.N. resolutions 60	Political: voting	Y	191	18,110	[269]
140	U.N. resolutions 61	Political: voting	Y	192	18,331	[269]
141	U.N. resolutions 62	Political: voting	Y	192	18,331	[269]
142	Biogrid: *A. thaliana*	Protein interaction	N	406	625	[249]
143	Biogrid: *C. elegans*	Protein interaction	N	3,353	6,449	[249]
144	Biogrid: *D. melanogaster*	Protein interaction	N	7,174	24,897	[249]
145	Biogrid: *H. sapiens*	Protein interaction	N	8,205	25,699	[249]
146	Biogrid: *M. musculus*	Protein interaction	N	710	1,003	[249]
147	Biogrid: *R. norvegicus*	Protein interaction	N	121	135	[249]
148	Biogrid: *S. cerevisiae*	Protein interaction	N	1,753	4,811	[249]
149	Biogrid: *S. pombe*	Protein interaction	N	1,477	11,404	[249]
150	DIP: *H. pylori*	Protein interaction	N	686	1,351	[6]
151	DIP: *H. sapiens*	Protein interaction	N	639	982	[6]
152	DIP: *M. musculus*	Protein interaction	N	50	55	[6]
153	DIP: *C. elegans*	Protein interaction	N	2,386	3,825	[6]
154	Human: CCSB	Protein interaction	N	1,307	2,483	[229]
155	Human: OPHID	Protein interaction	N	5,464	23,238	[55, 56]
156	Protein: serine protease inhibitor (1EAW)	Protein interaction	N	53	123	[185]
157	Protein: immunoglobulin (1A4J)	Protein interaction	N	95	213	[185]
158	Protein: oxidoreductase (1AOR)	Protein interaction	N	97	212	[185]
159	STRING: *C. elegans*	Protein interaction	N	1,762	95,227	[135]
160	STRING: *S. cerevisiae*	Protein interaction	N	534	57,672	[135]
161	Yeast: Oxford Statistics	Protein interaction	N	2,224	6,609	[59]
162	Yeast: DIP	Protein interaction	N	4,906	17,218	[6]
163	Yeast: DIPC	Protein interaction	N	2,587	6,094	[6]
164	Yeast: FHC	Protein interaction	N	2,233	5,750	[41]
165	Yeast: FYI	Protein interaction	N	778	1,798	[121]
166	Yeast: PCA	Protein interaction	N	889	2,407	[254]
167	Corporate directors in Scotland (1904-1905)	Social	Y	131	676	[69, 238]
168	Corporate ownership (EVA)	Social	N	4,475	4,652	[199]
169	Dolphins	Social	N	62	159	[168]
170	Family planning in Korea	Social	N	33	68	[224]
171	Unionization in a hi-tech firm	Social	N	33	91	[154]
172	Communication within a sawmill on strike	Social	N	36	62	[182]
173	Leadership course	Social	N	32	80	[185]
174	Les Miserables	Social	Y	77	254	[151]
175	Marvel comics	Social	Y	6,449	168,211	[20]
176	Mexican political elite	Social	N	35	117	[105]
177	Pretty-good-privacy algorithm users	Social	N	10,680	24,316	[49]
178	Prisoners	Social	N	67	142	[185]
179	Bernard and Killworth fraternity: observed	Social	Y	58	967	[38, 39, 225]
180	Bernard and Killworth fraternity: recalled	Social	Y	58	1,653	[38, 39, 225]
181	Bernard and Killworth HAM radio: observed	Social	Y	41	153	[40, 146, 147]
182	Bernard and Killworth HAM radio: recalled	Social	Y	44	442	[40, 146, 147]
183	Bernard and Killworth office: observed	Social	Y	40	238	[40, 146, 147]

Table B.1: List of 192 real-world networks.

Index	Name	Category	Weighted	Nodes	Links	Reference
184	Bernard and Killworth office: recalled	Social	Y	40	779	[40, 146, 147]
185	Bernard and Killworth technical: observed	Social	Y	34	175	[40, 146, 147]
186	Bernard and Killworth technical: recalled	Social	Y	34	561	[40, 146, 147]
187	Kapferer tailor shop: instrumental (t1)	Social	N	35	76	[140]
188	Kapferer tailor shop: instrumental (t2)	Social	N	34	93	[140]
189	Kapferer tailor shop: associational (t1)	Social	N	39	158	[140]
190	Kapferer tailor shop: associational (t2)	Social	N	39	223	[140]
191	University Rovira i Virgili (Tarragona) e-mail	Social	N	1,133	5,451	[115]
192	Zachary karate club	Social	N	34	78	[283]

Appendix C

Approximate Analytic Expressions for Thermodynamic Network Entropy

C.1 Modular networks

These networks are generated such there are a total of n nodes, divided into N equally-sized modules, with each node initially being connected to all others in its module, i.e, the average degree $\langle k \rangle = \frac{n}{N} - 1$. Subsequently, each link is rewired with probability λ; i.e., it is disconnected from one of its endpoints and joined up to a node chosen uniformly at random. Consider first a pair of nodes in different modules. They can only be linked by a rewired link. Each of the two nodes has $\langle k \rangle$ links to start with. For each such link, there is a probability λ of rewiring; if it rewires, there is a probability $\frac{1}{2}$ that it remains linked to the node being considered; and if that happens, the other end can join up to any one of the remaining $n - 1$ nodes in the network, so the probability of joining to the other node being considered is $\frac{1}{n-1}$. Thus, for each such link, the probability that it ends up linking the pair under consideration

is $\lambda\frac{1}{2(n-1)}$. Since there are a total of $2\langle k\rangle$ such links, $\langle k\rangle$ for each of the two nodes, the total probability of the considered pair being linked, which we denote by ϵ_{out}, is

$$\epsilon_{\text{out}} = \lambda\frac{\langle k\rangle}{n-1}. \tag{C.1}$$

Here we are ignoring the possibility that multiple links might rewire to join the same pair of nodes, which is negligible if n is sufficiently larger than $\langle k\rangle$.

Now consider a pair of nodes within the same module. There is a probability $1-\lambda$ that the initial link between them will remain in place. Additionally, there is a probability $\lambda\frac{\langle k\rangle}{n-1}$ that they will be connected by a rewired link. Thus the total probability of such a pair being linked, which we denote by ϵ_{in}, is

$$\epsilon_{\text{in}} = 1-\lambda+\lambda\frac{\langle k\rangle}{n-1}. \tag{C.2}$$

There are a total of $\frac{n\langle k\rangle}{2}$ within-module node pairs, and $\frac{n(n-1-\langle k\rangle)}{2}$ between-module node pairs. Given the assumption that n is sufficiently larger than $\langle k\rangle$, we can sum independently over the entropy contributions from each of these pairs. This gives us a total entropy per node (denoted for these networks by H_{td}^{mod}) of

$$H_{td}^{\text{mod}}(n,\langle k\rangle,\lambda) = \frac{-1}{2}\left[\langle k\rangle\left[\epsilon_{\text{in}}\log\epsilon_{\text{in}} + (1-\epsilon_{\text{in}})\log(1-\epsilon_{\text{in}})\right] + (n-1-\langle k\rangle)\left[\epsilon_{\text{out}}\log\epsilon_{\text{out}} + (1-\epsilon_{\text{out}})\log(1-\epsilon_{\text{out}})\right]\right]. \tag{C.3}$$

C.2 Watts-Strogatz networks

A Watts-Strogatz small-world network [273] is obtained by starting with a circular lattice where each node is connected to its k nearest neighbours, and subsequently rewiring links independently with probability p, in a fashion similar to the modular networks just described. The derivation here proceeds in a fashion exactly analogous to the above. Again we can divide node pairs into two types: those that are initially connected and those that are not. For the latter, the probability of being linked after

rewiring (which we denote by ϵ_{far}) is

$$\epsilon_{\text{far}} = p\frac{k}{n-1}. \tag{C.4}$$

For those node pairs close enough together in the lattice to be initially linked, the final probability of being linked (denoted ϵ_{near}) is

$$\epsilon_{\text{near}} = 1 - p + p\frac{k}{n-1}. \tag{C.5}$$

There are a total of $\frac{nk}{2}$ initially linked pairs, and $\frac{n(n-1-k)}{2}$ other pairs. Summing independently over the entropy contributions of the two types of pairs, we get a total entropy per node for these networks (denoted by H_{td}^{ws}) of

$$H_{td}^{ws}(n,k,p) = \frac{-1}{2}\left[k\left[\epsilon_{\text{near}}\log\epsilon_{\text{near}} + (1-\epsilon_{\text{near}})\log(1-\epsilon_{\text{near}})\right] + (n-1-k)\left[\epsilon_{\text{far}}\log\epsilon_{\text{far}} + (1-\epsilon_{\text{far}})\log(1-\epsilon_{\text{far}})\right]\right]. \tag{C.6}$$

Bibliography

[1] Code by Aravind Seshadri. http://www.mathworks.com/matlabcentral/fileexchange/9612-traveling-salesman-problem-tsp-using-simulated-annealing (accessed May 2011).

[2] Code by Joseph Kirk. http://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm (accessed May 2011).

[3] Cross-entropy toolbox. http://www.maths.uq.edu.au/CEToolBox/ (accessed May 2011).

[4] Drosophila Virtual Library. http://www.ceolas.org/fly/intro.html (accessed October 2011).

[5] Matlab code for computing Hartigan's DIP statistic implemented by Ferenc Mechler. Shared by Nicholas Price. http://www.nicprice.net/diptest/ (accessed February 2010).

[6] See http://dip.doe-mbi.ucla.edu/dip/main.cgi.

[7] See http://vlado.fmf.uni-lj.si/pub/networks/data/collab/geom.htm.

[8] See http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm.

[9] See http://vlado.fmf.uni-lj.si/pub/networks/data/dic/odlis/odlis.htm.

[10] See http://vlado.fmf.uni-lj.si/pub/networks/data/dic/roget/roget.htm.

[11] See http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/firth/software/tapir/.

[12] See http://www.proteinatlas.org/.

[13] B. Adamcsek, G. Palla, I. J. Farkas, I. Derenyi, and T. Vicsek. CFinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023 (2006).

[14] M. D. Adams, S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, *et al.* The genome sequence of *Drosophila melanogaster*. *Science*, 287(5461):2185–2195 (2000).

[15] S. Agarwal, C. M. Deane, M. A. Porter, and N. S. Jones. Revisiting date and party hubs: Novel approaches to role assignment in protein interaction networks. *PLoS Computational Biology*, 6(6):e1000817 (2010).

[16] S. Agarwal, S. Johnson, and N. Jones. Multiple notions of network entropy. In preparation.

[17] S. Agarwal, G. Villar, and N. S. Jones. High-throughput analysis of networks. In preparation.

[18] S. Agarwal, G. Villar, and N. S. Jones. High throughput network analysis (extended abstract). In *Machine Learning in Systems Biology (MLSB), Proceedings of the Fourth International Workshop*. Edinburgh, Scotland (2010). http://www.physics.ox.ac.uk/systems/agarwal/mlsb10.pdf.

[19] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764 (2010).

[20] R. Alberich, J. Miro-Julia, and F. Rossello. Marvel universe looks almost like a real social network (2002). arXiv:cond-mat/0202174.

[21] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, *et al. Molecular Biology of the Cell (fourth edition)*. Garland Science (2002).

[22] K. Anand and G. Bianconi. Entropy measures for networks: Toward an information theory of complex topologies. *Physical Review E*, 80(4):045102 (2009).

[23] K. Anand and G. Bianconi. Gibbs entropy of network ensembles by cavity methods. *Physical Review E*, 82(1):011116 (2010).

[24] K. Anand, G. Bianconi, and S. Severini. Shannon and von Neumann entropy of random networks with heterogeneous expected degree. *Physical Review E*, 83(3):036109 (2011).

[25] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, 25(1):25–29 (2000).

[26] G. D. Bader and C. W. Hogue. Analyzing yeast protein-protein interaction data obtained from different sources. *Nature Biotechnology*, 20(10):991–997 (2002).

[27] J. S. Bader, A. Chaudhuri, J. M. Rothberg, and J. Chant. Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnology*, 22(1):78–85 (2004).

[28] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512 (1999).

[29] A.-L. Barabási and Z. N. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113 (2004).

[30] D. P. Bartel. MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–297 (2004).

[31] D. P. Bartel. MicroRNAs: Target recognition and regulatory functions. *Cell*, 136(2):215–233 (2009).

[32] D. S. Bassett, E. T. Owens, K. E. Daniels, and M. A. Porter. The influence of topology on signal propagation in granular force networks (2011). `arXiv: 1110.1858`.

[33] N. N. Batada, T. Reguly, A. Breitkreutz, L. Boucher, B.-J. Breitkreutz, *et al.* Still stratus not altocumulus: Further evidence against the date/party hub distinction. *PLoS Biology*, 5(6):e154 (2007).

[34] N. N. Batada, T. Reguly, A. Breitkreutz, L. Boucher, B.-J. J. Breitkreutz, *et al.* Stratus not altocumulus: A new view of the yeast protein interaction network. *PLoS Biology*, 4(10):e317 (2006).

[35] V. Batagelj, A. Mrvar, and M. Zaveršnik. Network analysis of dictionaries. In T. Erjavec and J. Gros, editors, *Jezikovne tehnologije / Language Technologies*, pages 135–142. Ljubljana (2002).

[36] A. Békéssy, P. Békéssy, and J. Komlós. Asymptotic enumeration of regular matrices. *Studia Scientiarum Matematicarum Hungarica*, 7:343 (1972).

[37] P. Beltrao and L. Serrano. Specificity and evolvability in eukaryotic protein interaction networks. *PLoS Computational Biology*, 3(2):e25 (2007).

[38] H. Bernard, P. D. Killworth, and L. Sailer. Informant accuracy in social network data IV: a comparison of clique-level structure in behavioral and cognitive network data. *Social Networks*, 2(3):191–218 (1979–1980).

[39] H. Bernard, P. D. Killworth, and L. Sailer. Informant accuracy in social-network data V. An experimental attempt to predict actual communication from recall data. *Social Science Research*, 11(1):30–66 (1982).

[40] H. R. Bernard and P. D. Killworth. Informant accuracy in social network data II. *Human Communication Research*, 4(1):3–18 (1977).

[41] N. Bertin, N. Simonis, D. Dupuy, M. E. Cusick, J.-D. J. Han, *et al.* Confirmation of organized modularity in the yeast interactome. *PLoS Biology*, 5(6):e153 (2007).

[42] G. Bianconi. Degree distribution of complex networks from statistical mechanics principles (2006). `arXiv:cond-mat/0606365`.

[43] G. Bianconi. The entropy of randomized network ensembles. *Europhysics Letters*, 81(2):28005 (2008).

[44] G. Bianconi. Entropy of network ensembles. *Physical Review E*, 79(3):036114 (2009).

[45] G. Bianconi, A. C. C. Coolen, and C. J. Perez Vicente. Entropies of complex networks with hierarchically constrained topologies. *Physical Review E*, 78(1):016114 (2008).

[46] G. Bianconi, P. Pin, and M. Marsili. Assessing the relevance of node features for network structure. *Proceedings of the National Academy of Sciences of the United States of America*, 106(28):11433–11438 (2009).

[47] C. Bishop. *Pattern Recognition and Machine Learning*. Information science and statistics. Springer (2006).

[48] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, 2008(10):P10008 (2008).

[49] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas. Models of social networks based on social distance attachment. *Physical Review E*, 70:056122 (2004).

[50] B. V. Bonnlander and A. S. Weigend. Selecting input variables using mutual information and nonparametric density estimation. In *1994 International Symposium on Artificial Neural Networks*, pages 42–50. Tainan, Taiwan (1994).

[51] A.-L. Boulesteix. Over-optimism in bioinformatics research. *Bioinformatics*, 26(3):437–439 (2010).

[52] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, *et al.* On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188 (2008).

[53] P. Braun, M. Tasan, M. Dreze, M. Barrios-Rodiles, I. Lemmens, *et al.* An experimentally derived confidence score for binary protein-protein interactions. *Nature Methods*, 6(1):91–97 (2009).

[54] L. Breiman. *Classification and regression trees*. The Wadsworth and Brooks-Cole statistics-probability series. Chapman & Hall (1984).

[55] K. Brown and I. Jurisica. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biology*, 8(5):R95 (2007).

[56] K. R. Brown and I. Jurisica. Online predicted human interaction database. *Bioinformatics*, 21(9):2076–2082 (2005).

[57] R. E. Burkard and F. Rendl. A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, 17(2):169–174 (1984).

[58] J. Chen and B. Yuan. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics*, 22(18):2283–2290 (2006).

[59] P.-Y. Chen, C. M. Deane, and G. Reinert. Predicting and validating protein interactions using network structure. *PLoS Computational Biology*, 4(7):e1000118 (2008).

[60] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society (1997).

[61] F. D. Ciccarelli, T. Doerks, C. von Mering, C. J. Creevey, B. Snel, *et al.* Toward automatic reconstruction of a highly resolved tree of life. *Science*, 311(5765):1283–1287 (2006).

[62] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703 (2009).

[63] J. Claussen. Offdiagonal complexity: A computationally quick complexity measure for graphs and networks. *Physica A: Statistical Mechanics and its Applications*, 375(1):365–373 (2007).

[64] V. Colizza, A. Flammini, M. A. Serrano, and A. Vespignani. Detecting rich-club ordering in complex networks. *Nature Physics*, 2(2):110–115 (2006).

[65] Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242 (2007).

[66] D. Cvetković, P. Rowlinson, and S. Simić. *Eigenspaces of Graphs*. Cambridge University Press, Cambridge, UK (1997).

[67] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics*, 2005(09):P09008 (2005).

[68] P.-T. de Boer, D. Kroese, S. Mannor, and R. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134:19–67 (2005).

[69] W. de Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, Cambridge, UK (2004).

[70] D. de Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5):292–306 (1976).

[71] C. M. Deane, Ł. Salwiński, I. Xenarios, and D. Eisenberg. Protein interactions: two methods for assessment of the reliability of high throughput observations. *Molecular & Cellular Proteomics*, 1(5):349–356 (2002).

[72] I. Derényi, G. Palla, and T. Vicsek. Clique percolation in random networks. *Physical Review Letters*, 94:160202 (2005).

[73] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer (1999).

[74] R. Dunn, F. Dudbridge, and C. M. Sanderson. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC Bioinformatics*, 6(1):39 (2005).

[75] V. Emilsson, G. Thorleifsson, B. Zhang, A. S. Leonardson, F. Zink, *et al.* Genetics of gene expression and its effect on disease. *Nature*, 452:423–428 (2008).

[76] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae*, 6:290–297 (1959).

[77] E. Estrada. Spectral scaling and good expansion properties in complex networks. *Europhysics Letters*, 73(4):649 (2006).

[78] E. Estrada. Topological structural classes of complex networks. *Physical Review E*, 75(1):016103 (2007).

[79] E. Estrada and J. A. Rodríguez-Velázquez. Spectral measures of bipartivity in complex networks. *Physical Review E*, 72(4):046105 (2005).

[80] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Physical Review E*, 71:056103 (2005).

[81] L. Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum imperialis Petropolitanae*, 8:128–140 (1736).

[82] T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):016105 (2009).

[83] T. S. Evans and R. Lambiotte. Line graphs of weighted networks for overlapping communities. *The European Physical Journal B*, 77(2):265–272 (2010).

[84] R. R. Faulkner. *Music on Demand. Composers and Careers in the Hollywood Film Industry.* Transaction Books, New Brunswick, NJ, USA (1983).

[85] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47 (1991).

[86] J. Felsenstein. Phylogenies and the comparative method. *The American Naturalist*, 125(1):1–15 (1985).

[87] D. Fenn. *Network Communities and the Foreign Exchange Market.* Ph.D. thesis, University of Oxford (2010).

[88] V. Filkov, Z. M. Saul, S. Roy, R. M. D'Souza, and P. T. Devanbu. Modeling and verifying a broad array of network properties. *Europhysics Letters*, 86(2):28003 (2009).

[89] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer (2006).

[90] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174 (2010).

[91] E. J. Foss, D. Radulovic, S. A. Shaffer, D. R. Goodlett, L. Kruglyak, *et al.* Genetic variation shapes protein networks mainly through non-transcriptional mechanisms. *PLoS Biology*, 9(9):e1001144 (2011).

[92] J. H. Fowler. Connecting the Congress: A study of cosponsorship networks. *Political Analysis*, 14(4):456–487 (2006).

[93] J. H. Fowler. Legislative cosponsorship networks in the US House and Senate. *Social Networks*, 28(4):454–465 (2006).

[94] O. Frank and D. Strauss. Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842 (1986).

[95] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33:1134–1140 (1986).

[96] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41 (1977).

[97] M. Fromont-Racine, A. E. Mayes, A. Brunet-Simon, J.-C. Rain, A. Colley, *et al.* Genome-wide protein interaction screens reveal functional networks involving sm-like proteins. *Yeast*, 1(2):95–110 (2000).

[98] M. Fromont-Racine, J.-C. Rain, and P. Legrain. Toward a functional analysis of the yeast genome through exhaustive two-hybrid screens. *Nature Genetics.*, 16(3):277–282 (1997).

[99] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A series of books in the mathematical sciences. Freeman (2003).

[100] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, *et al.* Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11(12):4241–4257 (2000).

[101] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, *et al.* Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636 (2006).

[102] A.-C. C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, *et al.* Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147 (2002).

[103] A. Gelman. Induction and deduction in Bayesian data analysis. In D. Mayo, A. Spanos, and K. Staley, editors, *Rationality, Markets and Morals, special topic issue "Statistical Science and Philosophy of Science: Where Do (Should) They Meet In 2011 and Beyond?"* (2011).

[104] A. Gelman and C. R. Shalizi. Philosophy and the practice of Bayesian statistics (2011). `arXiv:1006.3868`.

[105] J. Gil-Mendieta and S. Schmidt. The political network in mexico. *Social Networks*, 18(4):355–381 (1996).

[106] E. N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30(4):1141–1144 (1959).

[107] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826 (2002).

[108] P. Gleiser and L. Danon. Community structure in jazz. *Advances in Complex Systems*, 6(4):565–573 (2003).

[109] V. Gol'dshtein, G. Koganov, and G. Surdutovich. Vulnerability and hierarchy of complex networks (2004). `arXiv:cond-mat/0409298`.

[110] L. A. Goodman. Snowball sampling. *Annals of Mathematical Statistics*, 32:148–170 (1961).

[111] S. M. Goodreau. Advances in exponential random graph (p*) models applied to a large social network. *Social Networks*, 29(2):231–248 (2007).

[112] M. S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380 (1973).

[113] X. Gu, Z. Zhang, and W. Huang. Rapid evolution of expression and regulatory divergences after yeast gene duplication. *Proceedings of the National Academy of Sciences of the United States of America*, 102(3):707–712 (2005).

[114] R. Guimerà and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900 (2005).

[115] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68:065103 (2003).

[116] A. Gursoy, O. Keskin, and R. Nussinov. Topological properties of protein interaction networks from a structural perspective. *Biochemical Society Transactions*, 36(Pt 6):1398–1403 (2008).

[117] G. Gutin and A. P. Punnen, editors. *The Traveling Salesman Problem and Its Variations*. Springer (2002).

[118] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, *et al.* Mapping the structural core of human cerebral cortex. *PLoS Biology*, 6(7):e159 (2008).

[119] L. Hakes, J. W. Pinney, D. L. Robertson, and S. C. Lovell. Protein-protein interaction networks and biology—what's the connection? *Nature Biotechnology*, 26(1):69–72 (2008).

[120] L. Hakes, D. L. Robertson, S. G. Oliver, and S. C. Lovell. Protein interactions from complexes: A structural perspective. *Computational and Functional Genomics*, 2007:49356 (2007).

[121] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, *et al.* Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995):88–93 (2004).

[122] J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84 (1985).

[123] P. M. Hartigan. Algorithm AS 217: Computation of the dip statistic to test for unimodality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 34(3):320–325 (1985).

[124] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer (2009).

[125] S. R. Hegde, P. Manimaran, and S. C. Mande. Dynamic changes in protein functional linkage networks revealed by integration with gene expression data. *PLoS Computational Biology*, 4(11):e1000237 (2008).

[126] H. Hitotumatu and K. Noshita. A technique for implementing backtrack algorithms and its application. *Information Processing Letters*, 8(4):174–175 (1979).

[127] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, *et al.* Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415(6868):180–183 (2002).

[128] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press (1975).

[129] E. Hubbell, W.-M. Liu, and R. Mei. Robust estimators for expression analysis. *Bioinformatics*, 18(12):1585–1592 (2002).

[130] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218 (1985).

[131] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945 (2004).

[132] J. P. A. Ioannidis. Why most published research findings are false. *PLoS Medicine*, 2(8):e124 (2005).

[133] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, *et al.* A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8):4569–4574 (2001).

[134] P. Jaccard. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:241–272 (1901).

[135] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, *et al.* STRING 8—a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research.*, 37:D412–416 (2009).

[136] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42 (2001).

[137] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654 (2000).

[138] J. C. Johnson and L. Krempel. Network visualization: The "Bush team" in Reuters news ticker 9/11–11/15/01. *Journal of Social Structure*, 5 (2004).

[139] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15 (1989).

[140] B. Kapferer. *Strategy and transaction in an African factory.* Manchester University Press, Manchester, UK (1972).

[141] G. Kar, A. Gursoy, and O. Keskin. Human cancer protein-protein interaction network: A structural perspective. *PLoS Computational Biology*, 5(12):e1000601 (2009).

[142] P. Kemmeren, N. L. van Berkum, J. Vilo, T. Bijma, R. Donders, *et al.* Protein interaction verification and functional annotation by integrated analysis of genome-scale data. *Molecular Cell*, 9:1133–1143 (2002).

[143] F. Képès, editor. *Biological Networks.* World Scientific (2007).

[144] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–307 (1970).

[145] P. V. Khadikar, N. V. Deshpande, P. P. Kale, A. Dobrynin, I. Gutman, *et al.* The Szeged index and an analogy with the Wiener index. *Journal of Chemical Information and Computer Sciences*, 35(3):547–550 (1995).

[146] P. D. Killworth and H. Bernard. Informant accuracy in social network data. *Human Organization*, 35(3):269–286 (1976).

[147] P. D. Killworth and H. Bernard. Informant accuracy in social network data III: A comparison of triadic structure in behavioral and cognitive data. *Social Networks*, 2(1):19–46 (1979–1980).

[148] H. J. Kim and J. M. Kim. Cyclic topology in complex networks. *Physical Review E*, 72(3):036109 (2005).

[149] J. Kim and T. Wilhelm. What is a complex graph? *Physica A*, 387(11):2637–2652 (2008).

[150] P. M. Kim, A. Sboner, Y. Xia, and M. Gerstein. The role of disorder in interaction networks: a structural analysis. *Molecular Systems Biology*, 4:179 (2008).

[151] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, USA (1993).

[152] D. E. Knuth. Dancing links. *Millenial Perspectives in Computer Science*, pages 187–214 (2000).

[153] K. Komurov and M. White. Revealing static and dynamic modular architecture of the eukaryotic protein interaction network. *Molecular Systems Biology*, 3:110 (2007).

[154] D. Krackhardt. The ties that torture: Simmelian tie analysis in organizations. *Research in the Sociology of Organizations*, 16:183 (1999).

[155] O. Kuchaiev, A. Stevanovic, W. Hayes, and N. Przulj. Graphcruch 2: Software tool for network modeling, alignment and clustering. *BMC Bioinformatics*, 12(1):24 (2011).

[156] R. Lambiotte, J. C. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks (2009). `arXiv:0812.1770`.

[157] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80:056117 (2009).

[158] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015 (2009).

[159] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110 (2008).

[160] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701 (2001).

[161] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816 (2005).

[162] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. Springer (2007).

[163] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 631–636. ACM, New York, NY, USA (2006).

[164] I. Letunic and P. Bork. Interactive tree of life (iTOL): an online tool for phylogenetic tree display and annotation. *Bioinformatics*, 23(1):127–128 (2007).

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

[166] A. C. F. Lewis, N. S. Jones, M. A. Porter, and C. M. Deane. What evidence is there for the homology of protein interactions? (2011). Submitted.

[167] W. K. Lim, K. Wang, C. Lefebvre, and A. Califano. Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. *Bioinformatics*, 23(13):i282–i288 (2007).

[168] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, *et al.* The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405 (2003).

[169] M. Macholán. The mouse skull as a source of morphometric data for phylogeny inference. *Zoologischer Anzeiger*, 247(4):315–327 (2008).

[170] D. R. Maddison and K.-S. Schulz, editors. The tree of life web project. `http://tolweb.org` (2007).

[171] I. Maraziotis, K. Dimitrakopoulou, and A. Bezerianos. An *in silico* method for detecting overlapping functional modules from composite biological networks. *BMC Systems Biology*, 2(1):93 (2008).

[172] Marin J.-M., N. Pillai, C. Robert, and J. Rousseau. Relevant statistics for Bayesian model choice (2011). `arXiv:1110.4700`.

[173] Marin J.-M., P. Pudlo, C. P. Robert, and R. Ryder. Approximate Bayesian computational methods (2011). `arXiv:1101.0955`.

[174] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavar. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–15328 (2003).

[175] E. P. Martins. Estimating the rate of phenotypic evolution from comparative data. *The American Naturalist*, 144(2):193–209 (1994).

[176] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913 (2002).

[177] A. Mazurie, D. Bonchev, B. Schwikowski, and G. A. Buck. Phylogenetic distances are encoded in networks of interacting pathways. *Bioinformatics*, 24(22):2579–2585 (2008).

[178] A. Mazurie, D. Bonchev, B. Schwikowski, and G. A. Buck. Evolution of metabolic network organization. *BMC Systems Biology*, 4(1):59 (2010).

[179] N. M. McCarty, K. T. Poole, and H. Rosenthal. *Polarized America: The Dance of Ideology and Unequal Riches*. MIT Press, Cambridge, MA, USA (2007).

[180] J. P. McCutcheon and N. A. Moran. Extreme genome reduction in symbiotic bacteria. *Nature Reviews Microbiology*, 10(1):13–26 (2012).

[181] H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, *et al.* MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, 30(1):31–34 (2002).

[182] J. H. Michael and J. G. Massey. Modeling the communication network in a sawmill. *Forest Products Journal*, 47:25–30 (1997).

[183] T. Milenkovic, J. Lai, and N. Przulj. Graphcrunch: A tool for large network analyses. *BMC Bioinformatics*, 9(1):70 (2008).

[184] S. Milgram. The small world problem. *Psychology Today*, 1(1):60–67 (1967).

[185] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, *et al.* Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542 (2004).

[186] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, *et al.* Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827 (2002).

[187] P. V. Missiuro, K. Liu, L. Zou, B. C. Ross, G. Zhao, *et al.* Information flow analysis of interactome networks. *PLoS Computational Biology*, 5(4):e1000350 (2009).

[188] A. Mithani, G. M. Preston, and J. Hein. A Bayesian approach to the evolution of metabolic networks on a phylogeny. *PLoS Computational Biology*, 6(8):e1000868 (2010).

[189] F. Mosteller and J. W. Tukey. Data analysis, including statistics. In *Handbook of Social Psychology*. Addison-Wesley, Reading, MA, USA (1968).

[190] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409 (2001).

[191] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701 (2002).

[192] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256 (2003).

[193] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104 (2006).

[194] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582 (2006).

[195] M. E. J. Newman. Mathematics of networks. In L. E. Blume and S. N. Durlauf, editors, *The New Palgrave Encyclopedia of Economics*. Palgrave Macmillan, Basingstoke, 2 edition (2008).

[196] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press (2010).

[197] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113 (2004).

[198] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press (2006).

[199] K. Norlen, G. Lucas, M. Gebbie, and J. Chuang. EVA: Extraction, visualization and analysis of the telecommunications and media ownership network. In *Proceedings of International Telecommunications Society 14th Biennial Conference (ITS2002)*. Seoul, Korea (2002).

[200] T. Obayashi, S. Hayashi, M. Shibaoka, M. Saeki, H. Ohta, *et al.* COXPRESdb: a database of coexpressed gene networks in mammals. *Nucleic Acids Research*, 36(Database issue):D77–D82 (2008).

[201] S. Ohno. *Evolution by Gene Duplication*. Springer-Verlag (1970).

[202] J.-P. Onnela, A. Chakraborti, K. Kaski, J. Kertész, and A. Kanto. Dynamics of market correlations: Taxonomy and portfolio analysis. *Physical Review E*, 68:056110 (2003).

[203] J.-P. Onnela, D. J. Fenn, S. Reid, M. A. Porter, P. J. Mucha, *et al.* Taxonomies of networks (2011). `arxiv:1006.5731`.

[204] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, *et al.* Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences of the United States of America*, 104(18):7332–7336 (2007).

[205] M. Ortiz, N. Guex, E. Patin, O. Martin, I. Xenarios, *et al.* Evolutionary trajectories of primate genes involved in HIV pathogenesis. *Molecular Biology and Evolution*, 26(12):2865–2875 (2009).

[206] P. Pattison and S. Wasserman. Logit models and logistic regressions for social networks: II. Multivariate relations. *British Journal of Mathematical and Statistical Psychology*, pages 169–193 (1999).

[207] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572 (1901).

[208] M. Penrose. *Random Geometric Graphs*. Oxford University Press (2003).

[209] J. Pons and A. P. Vogler. Complex pattern of coalescence and fast evolution of a mitochondrial rRNA pseudogene in a recent radiation of tiger beetles. *Molecular Biology and Evolution*, 22(4):991–1000 (2005).

[210] K. T. Poole and H. Rosenthal. *Congress: A Political-Economic History of Roll Call Voting*. Oxford University Press, Oxford, UK (1997).

[211] M. A. Porter, P. J. Mucha, M. Newman, and A. Friend. Community structure in the United States House of Representatives. *Physica A*, 386(1):414–438 (2007).

[212] M. A. Porter, P. J. Mucha, M. E. J. Newman, and C. M. Warmbrand. A network analysis of committees in the U.S. House of Representatives. *Proceedings of the*

National Academy of Sciences of the United States of America, 102(20):7057–7062 (2005).

[213] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, 1164–1166 (2009).

[214] W. Press. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press (2007).

[215] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798 (1999).

[216] A. Rapoport. Contributions to the theory of random and biased nets. *Bulletin of Mathematical Biophysics*, 19:257–277 (1957).

[217] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press (2008).

[218] O. Ratmann, C. Andrieu, C. Wiuf, and S. Richardson. Model criticism based on likelihood-free inference, with an application to protein network evolution. *Proceedings of the National Academy of Sciences of the United States of America*, 106(26):10576–10581 (2009).

[219] O. Ratmann, O. Jørgensen, T. Hinkley, M. Stumpf, S. Richardson, *et al.* Using likelihood-free inference to compare evolutionary dynamics of the protein networks of *H. pylori* and *P. falciparum*. *PLoS Computational Biology*, 3(11):e230 (2007).

[220] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110 (2006).

[221] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453 (1995).

[222] A. W. Rives and T. Galitski. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(3):1128–1133 (2003).

[223] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p*) models for social networks. *Social Networks*, 29(2):173–191 (2007).

[224] E. M. Rogers and D. L. Kincaid. *Communication Networks. Toward a New Paradigm for Research*. The Free Press, New York, NY, USA (1981).

[225] A. K. Romney and S. C. Weller. Predicting informant accuracy from patterns of recall among individuals. *Social Networks*, 6(1):59–77 (1984).

[226] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences of the United States of America*, 104(18):7327–7331 (2007).

[227] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326 (2000).

[228] S. Roy and V. Filkov. Strong associations between microbe phenotypes and their network architecture. *Physical Review E*, 80(4):040902 (2009).

[229] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, *et al.* Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437:1173–78 (2005).

[230] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190 (1999).

[231] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112 (1997).

[232] R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag (2004).

[233] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31:581–603 (1966).

[234] R. Saeed and C. Deane. An assessment of the uses of homologous interactions. *Bioinformatics*, 24(5):689–695 (2008).

[235] J. W. Scannell, G. A. P. C. Burns, C. C. Hilgetag, M. A. O'Neil, and M. P. Young. The connectional organization of the cortico-thalamic system of the cat. *Cereb. Cortex*, 9(3):277–299 (1999).

[236] E. E. E. Schadt, C. Molony, E. Chudin, K. Hao, X. Yang, *et al.* Mapping the genetic architecture of gene expression in human liver. *PLoS Biology*, 6(5):e107 (2008).

[237] A. S. Schwartz, J. Yu, K. R. Gardenour, R. L. Finley Jr, and T. Ideker. Cost-effective strategies for completing the interactome. *Nature Methods*, 6(1):55–61 (2009).

[238] J. Scott and M. Hughes. *The anatomy of Scottish capital: Scottish companies and Scottish capital, 1900–1979*. Croom Helm, London, UK (1980).

[239] S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287 (1983).

[240] I. Sendiña-Nadal, Y. Ofran, J. A. Almendral, J. M. Buldú, I. Leyva, *et al.* Unveiling protein functions through the dynamics of the interaction network. *PLoS ONE*, 6(3):e17679 (2011).

[241] M. A. Serrano and M. Boguna. Weighted configuration model. *American Institute of Physics Conference Proceedings*, 776(1):101–107 (2005).

[242] R. Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34 (1973).

[243] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42(3-4):425–440 (1955).

[244] S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1760–1765 (2007).

[245] K. Smith-Miles, J. van Hemert, and X. Lim. Understanding TSP difficulty by learning from evolved instances. In C. Blum and R. Battiti, editors, *Learning and Intelligent Optimization*, volume 6073 of *Lecture Notes in Computer Science*, pages 266–280. Springer Berlin / Heidelberg (2010).

[246] S. N. Soffer and A. Vázquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(5):057101 (2005).

[247] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, *et al.* Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297 (1998).

[248] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):12123–12128 (2003).

[249] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, *et al.* BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, 34(suppl 1):D535–D539 (2006).

[250] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37 (1989).

[251] M. P. H. Stumpf and M. A. Porter. Critical truths about power laws. *Science*, 335(6069):665–666 (2012).

[252] M. P. H. Stumpf, C. Wiuf, and R. M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4221–4224 (2005).

[253] A. I. Su, T. Wiltshire, S. Batalov, H. Lapp, K. A. Ching, *et al.* A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):6062–6067 (2004).

[254] K. Tarassov, V. Messier, C. R. Landry, S. Radinovic, M. M. Serna Molina, *et al.* An *in vivo* map of the yeast protein interactome. *Science*, 320(5882):1465–1470 (2008).

[255] S. Tavare, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518 (1997).

[256] I. W. Taylor, R. Linding, D. Warde-Farley, Y. Liu, C. Pesquita, *et al.* Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nature Biotechnology*, 27:199–204 (2009).

[257] J. Taylor and J. Raes. Duplication and divergence: The evolution of new genes and old ideas. *Annual Review of Genetics*, 38:615–643 (2004).

[258] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323 (2000).

[259] The UniProt Consortium. The universal protein resource (UniProt). *Nucleic Acids Research*, 36(Database issue):D190–D195 (2008).

[260] R. Toivonen, L. Kovanen, M. Kivelä, J.-P. Onnela, J. Saramäki, *et al.* A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks*, 31(4):240–254 (2009).

[261] T. Toni and M. P. H. Stumpf. Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, 26(1):104–110 (2010).

[262] A. L. Traud, C. Frost, P. J. Mucha, and M. A. Porter. Visualization of communities in networks. *Chaos*, 19(4):041104 (2009).

[263] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM Review*, 53(3):526–543 (2011).

[264] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443 (1969).

[265] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, *et al.* A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403(6770):623–627 (2000).

[266] R. R. Vallabhajosyula, D. Chakravarti, S. Lutfeali, A. Ray, and A. Raval. Identifying hubs in protein interaction networks. *PLoS ONE*, 4(4):e5344 (2009).

[267] K. Venkatesan, J.-F. Rual, A. Vazquez, U. Stelzl, I. Lemmens, *et al.* An empirical framework for binary interactome mapping. *Nature Methods*, 6(1):83–90 (2009).

[268] G. Villar, S. Agarwal, and N. S. Jones. High throughput network analysis (extended abstract). In *Proceedings of the Workshop on Analysis of Complex NEtworks (ACNE), ECML PKDD*. Barcelona, Spain (2010).

[269] E. Voeten. Clashes in the Assembly. *International Organization*, 54(2):185–215 (2000).

[270] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, *et al.* Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403 (2002).

[271] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer texts in statistics. Springer (2004).

[272] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press (1994).

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

[274] A. S. Waugh, L. Pei, J. H. Fowler, P. J. Mucha, and M. A. Porter. Party polarization in Congress: A network science approach (2011). arXiv:0907.3509.

[275] M. R. Wilkins and S. K. Kummerfeld. Sticking together? Falling apart? Exploring the dynamics of the interactome. *Trends in Biochemical Sciences*, 33(5):195–200 (2008).

[276] Z. Wu, R. A. Irizarry, R. Gentleman, F. M. Murillo, and F. Spencer. A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99(468):909–917 (2004).

[277] T. Yamada and P. Bork. Evolution of biomolecular networks: lessons from metabolic and protein interactions. *Nature Reviews Molecular Cell Biology*, 10(11):791–803 (2009).

[278] S.-H. H. Yook, Z. N. Oltvai, and A.-L. L. Barabási. Functional and topological characterization of protein interaction networks. *Proteomics*, 4(4):928–942 (2004).

[279] M. P. Young. The organization of neural systems in the primate cerebral cortex. *Proceedings of the Royal Society B: Biological Sciences*, 252(1333):13–18 (1993).

[280] H. Yu, P. Braun, M. A. Yildirim, I. Lemmens, K. Venkatesan, *et al.* High-quality binary protein interaction map of the yeast interactome network. *Science*, 322:104–110 (2008).

[281] H. Yu, P. M. Kim, E. Sprecher, V. Trifonov, and M. Gerstein. The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLoS Computational Biology*, 3(4):e59 (2007).

[282] G. U. Yule. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. *Philosophical Transactions of the Royal Society of London, Series B*, 213:21–87 (1925).

[283] W. A. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473 (1977).

[284] J. Zhang. Evolution by gene duplication: an update. *Trends in Ecology & Evolution*, 18(6):292–298 (2003).

[285] S. Zhou and R. Mondragon. The rich-club phenomenon in the Internet topology. *Communications Letters, IEEE*, 8(3):180–182 (2004).

[286] E. Zotenko, J. Mestre, D. P. O'Leary, and T. M. Przytycka. Why do hubs in the yeast protein interaction network tend to be essential: Reexamining the connection between the network topology and essentiality. *PLoS Computational Biology*, 4(8):e1000140 (2008).

[287] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286 (2006).