Subgraph Covers- an Information Theoretic Approach to Motif Analysis in Networks

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr.rer.nat.)

im Fachgebiet

Informatik

vorgelegt

von M. Sc. Anatol Eugen Wegner geboren am 17.01.1985 in Meerbusch

Die Annahme der Dissertation wurde empfohlen von:

1. Professor Dr. Stefan Bornholdt (Universität Bremen)
2. Professor Dr. Jürgen Jost (MPI für Mathematik in den Naturwissenschaften)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung am 04.02.2015 mit dem Gesamtprädikat magna cum laude.

Contents

List of Tables

Abstract

A large number of complex systems can be modeled as networks of interacting units. From a mathematical point of view the topology of such systems can be represented as graphs of which the nodes represent individual elements of the system and the edges interactions or relations between them. In recent years networks have become a principal tool for analyzing complex systems in many different fields.

This thesis introduces an information theoretic approach for finding characteristic connectivity patterns of networks, also called network motifs. Network motifs are sometimes also referred to as basic building blocks of complex networks. Many real world networks contain a statistically surprising number of certain subgraph patterns called network motifs. In biological and technological networks motifs are thought to contribute to the overall function of the network by performing modular tasks such as information processing. Therefore, methods for identifying network motifs are of great scientific interest.

In the prevalent approach to motif analysis network motifs are defined to be subgraphs that occur significantly more often in a network when compared to a null model that preserves certain features of the network. How-

ever, defining appropriate null models and sampling these has proven to be challenging. This thesis introduces an alternative approach to motif analysis which looks at motifs as regularities of a network that can be exploited to obtain a more efficient representation of the network. The approach is based on finding a subgraph cover that represents the network using minimal total information. Here, a subgraph cover is a set of subgraphs such that every edge of the graph is contained in at least one subgraph in the cover while the total information of a subgraph cover is the information required to specify the connectivity patterns occurring in the cover together with their position in the graph.

The thesis also studies the connection between motif analysis and random graph models for networks. Developing random graph models that incorporate high densities of triangles and other motifs has long been a goal of network research. In recent years, two such model have been proposed [1, 2]. However, their applications have remained limited because of the lack of a method for fitting such models to networks. In this thesis, we address this problem by showing that these models can be formulated as ensembles of subgraph covers and that the total information optimal subgraph covers can be used to match networks with such models. Moreover, these models can be solved analytically for many of their properties allowing for more accurate modeling of networks in general.

Finally, the thesis also analyzes the problem of finding a total information optimal subgraph cover with respect to its computational complexity. The problem turns out to be NP-hard hence, we propose a greedy heuristic for it. Empirical results for several real world networks from different fields are

presented. In order to test the presented algorithm we also consider some synthetic networks with predetermined motif structure.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Jürgen Jost, for giving me the chance to do my thesis work in his research group. I am grateful for the opportunity he gave me to conduct research in the fascinating field of complex networks. It is needless to say that without his guidance and continuous support this thesis would not have been possible.

I also would like to thank Murat Saglam, Seraphine Wegner and Volkan Cakir for the critical reading of this thesis.

I would like to extend my gratitude to members of the Jost group for their help and fruitful discussions. I also would like to thank the administrative staff of the Max-Planck Institute for Mathematics in the Sciences for providing such a friendly and fruitful environment for doing research.

Last but not least, I thank my wife Dilara and our newborn son Mitra, my family and best friends who have been beside me every step of the way on this journey and given me the confidence, strength, and motivation to overcome obstacles along the way.

Chapter 1

Introduction

1.1 Complex networks

A wide range of systems can be represented as networks of interacting elements. Consequently, networks are studied across many disciplines. Examples include cellular networks and food webs in biology, technological networks like the internet and power grids and networks representing social relations.

In the last two decades researchers have developed a variety of tools and models to study the structure of such complex networks. As opposed to classical graph theory, that is mostly concerned with the study of small and/or highly regular graphs, network research has mostly focused on large scale statistical properties of comparatively large and complex graphs. Consequently, research has been focused of finding statistical measures that summarize key topological features of networks such as average path lengths, clustering, degree distributions, assortativity, network motifs and community structure [3].

The two main drivers of network research have been the availability of large scale network data and the observation that the topology of many real world networks significantly deviates from random graphs.

Motivated by empirical studies of networks many researchers have developed models aimed at explaining how networks come to have some commonly observed properties and their overall effect on the topology of network. On the other hand, in contrast to empirical data, most widely used random graph models that can be solved analytically produce networks that do not contain significant densities of highly connected subgraphs. Recently, several random graph models that can incorporate high densities of highly connected subgraphs have been proposed [2, 1]. However, due to the lack of a method for fitting these models, their applications have remained rather limited.

1.1.1 Network motifs

Some small connectivity pattern, called network motifs, occur in complex networks much more often than one would expect on the basis of pure chance. In social networks the presence of many triangles can be traced back to the tendency of people to associate in groups. In biological systems motifs are thought to contribute to the overal function of the network by performing modular tasks [4] and thus to be evolutionarily favored.

In the prevalent approach due to Milo et al. [5] network motifs are defined to be subgraph patterns that occur significantly more often in the network than in null model that corresponfs to a randomized version of the network which reflects certain properties of the network. The null model is in gen-

eral taken to be the ensemble of all networks that have the same degree distribution as the original network.

In this thesis instead of comparing subgraph counts in the network with a null model we look at motifs as building blocks of networks. In order to identify network motifs we use subgraph covers, which are essentially decompositions of the network into its subgraphs. Network motifs are then defined to be connectivity patterns that appear in maximally efficient decompositions of networks. In order to make the concept of efficiency mathematically precise we follow the total information approach by Gell-Mann and Lloyd [6].

1.2 Motivation and Objectives

The main objective of this thesis is to develop a method for determining characteristic connectivity patterns of networks, commonly referred to as network motifs. For this it follows an information theoretic approach that is based on using subgraph covers as representations of graphs. It provides a novel definition of network motifs in terms of total information optimal subgraph covers. The approach looks at networks motifs as regularities of the network that can be exploited to obtain a more efficient representation of the network. In order to prove the practical value of the approach several algorithms for finding network motifs are also presented.

Our motivation for developing the a new method for motif analysis is the need for a method that can be used to detect network motifs consistently even for motifs of larger size. Another aim is to establish a clear connection

between motif analysis and random graph models for networks.

1.3 Summary of the Main Results

The main result of this thesis is an information theoretic approach to motif analysis in networks that is based on finding a subgraph cover of the network that has minimal total information. By considering motifs of all sizes simultaneously and using a single universal measure, the method is able to detect even large motifs consistently.

We also show that some recently introduced random graph models that can incorporate high densities of highly connected subgraphs, can be formulated as ensembles of subgraph covers. Consequently, total information optimal subgraph covers provide a way of associating networks with specific instances of these models. This allows motif structures to be incorporated into random graph models and allows for more accurate modeling of networks in general.

In order to prove the practical value of our approach we also study the problem of finding an optimal subgraph cover from a perspective of computational complexity. The problem turns our to be a non-linear covering problem. Since, covering problems are known to be NP-hard even in the linear case we solve it heuristically using a greedy heuristic.

Finally, we present empirical results for several real world networks from different fields. These show that the methods finds very similar motifs in real world networks representing systems of the same type. We also analyzed some synthetic networks, with predetermined motif structure, in order to test

the greedy heuristic.

1.4 Thesis Structure

The remainder of the thesis is organized as follows:

Chapter 2 contains basic graph theoretical concepts and an overview of complex networks that includes definitions of commonly used network measures and random graph models.

Chapter 3 contains a brief overview of the information theoretic concepts that form the basis of our analysis.

In chapter 4 we introduce subgraph covers as representations of graphs and define the total information of subgraph covers using uniform subgraph cover ensembles. We also show that some recently introduced random graph models can be formulated as ensembles of subgraph covers and discuss the use of total information optimal subgraph covers as basis for model selection.

In chapter 5 we analyze the problem of finding a total information optimal subgraph covers with respect to its computational complexity and present several heuristics for the problem.

Chapter 6 contains empirical results for various real world and synthetic networks.

In chapter 7 we give a short summary of the thesis and discuss possible applications and generalizations of the method. We also discuss possible directions for future research.

Chapter 2

Background I: Graphs and Networks

2.1 Graph theory

In this section we give definitions of basic graph theoretical concepts that are relevant to the general development of the thesis.

Definition 2.1.1. A *graph* $G = (V, E)$ is an ordered pair of sets such that the elements of E are two element subsets of V. The elements of V are called *vertices* (or *nodes*) and the elements of E are called the *edges* of G. For an edge $\{x, y\}$ we sometimes write xy.

Definition 2.1.2. A *directed graph* (or*digraph*) $G = (V, E)$ is an ordered pair of sets such that the elements of E are ordered pairs of V. An edge (x, y) (or xy) is said to be directed from x to y.

A graph is called simple if it does not contain parallel edges or self self-

edges.Throughout this thesis we won't make an explicit distinction between directed and undirected graphs since most definitions and arguments apply to both cases. On the other hand, we will primarily use undirected graphs in examples.

Definition 2.1.3. A graph H is called a subgraph of G whenever $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph is said to be *induced* iff it contains all edges $xy \in E(G)$ such that $x, y \in V(H)$.

Definition 2.1.4. Two graphs G and H are said to be *isomorphic* ($G \simeq H$) whenever there exist a bijection $\phi : V(G) \to V(H)$ such that $(x, y) \in E(G) \Leftrightarrow (\phi(x), \phi(y)) \in E(H)$ for all $x, y \in V(G)$. Such a map ϕ is called an isomorphism. Whenever $G = H$ it is called an *automorphism*.

Definition 2.1.5. Being isomorphic ($\simeq$) is an equivalence relation and the corresponding equivalence classes are called *isomorphism classes*.

Throughout the text we generally will use the words motif or subgraph/connectivity pattern instead of isomorphism classes. An isomorphism class can be referred to by unlabeled graph or any one of its labeled members. We will in general use upper case letters for graphs and lower case letters for isomorphism classes.

Definition 2.1.6. An *m-subgraph* or *subgraph instance of m* is a subgraph that belongs to the isomorphism class m.

Definition 2.1.7. The automorphisms of a graph G form a group. This group is called the *automorphism group* of G and is denoted by $Aut(G)$.

The *type* or orbit of a vertex x is defined as the set of vertices that are images of x under the action of the automorphism group.

Definition 2.1.8. The number of edges connected to a vertex i is called its *degree* denoted by d_i. For directed graphs one can also define the in- and out- degrees of a vertex which are the number of inward and outward directed edges. Alternatively, one can also include the number of bidirectional edges attached to a vertex. The degree sequence $(d_i(G))$ of G is the sequence of the degrees of its vertices.

Definition 2.1.9. A sequence (d_i) is called *graphical* if there exists a simple graph with degree sequence (d_i).

The graphicallity of a sequence can be checked using the Erdös-Gallai theorem [7].

Definition 2.1.10. The *degree distribution* corresponding to a degree sequence (d_i) is the probability distribution that specifies the probability that a random vertex has degree k : $p(k) = \frac{n_k(d_i)}{N}$ where $n_k(d_i)$ is the number of vertices with degree k.

Definition 2.1.11. A *path* of G is an ordered tuple of distinct (except maybe the first and last) vertices $(v_0, v_1, ..., v_k)$ such that $(v_i, v_i + 1) \in E(G)$. If $v_0 = v_k$ the path is called a *cycle*.

Definition 2.1.12. The (geodesic) *distance* $d(i, j)$ between vertices i and j is the length of the shortest path connecting i and j.

Definition 2.1.13. A graph $G = (V, E)$ is said to be *connected* iff for every pair of its vertices there exists a path connecting them. In the case of directed

graphs the graph is said to be *strongly connected* if there exists a directed path between every pair of vertices in the graph and *simply connected* if the underlying undirected graph is connected.

Definition 2.1.14. A graph is said to be *biconnected* if it can not be separated into two or more disconnected components by the removal of any one of its vertices. We will assume that the graph consisting of a single edge is biconnected.

Definition 2.1.15. A *tree* is a graph that is connected and contains no cycles.

It is straightforward to show that for a tree $T = (V, E) : |E| = |V| - 1$.

Definition 2.1.16. A graph of size n which contains all possible edges is called a *complete graph*, K_n. Complete graphs are also sometimes referred to as cliques.

Definition 2.1.17. A graph is called *bipartite* if the set of vertices can be partitioned into two sets V_1 and V_2 such that: $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2$ and $E(G) \subseteq V_1 X V_2$. A maximally connected bipartite graph with $(|V_1|, |V_2|) = (n, m)$ is called *complete bipartite*, $K_{n,m}$.

2.2 Complex networks

In this section, we briefly review some classes of complex networks that have been the subject of extensive research and some commonly studied network properties. We will focus on static networks and their topological properties.

For a more extensive review of the subject we refer the reader to the following review articles and books [3, 8, 9, 10].

2.2.1 Technological networks

Technological networks in general are networks representing man-made systems which typically are designed to perform a certain function:

- Computer networks: the nodes are computers/routers and edges represent physical connections between these.
- Distribution networks: These include networks power grids, telephone lines and road networks.
- The WWW: nodes represent web pages and directed edges, hyper-links between these.
- Electronic circuits: in these networks nodes correspond to circuit components such as logic gates and flip-flops.

2.2.2 Biological networks

Various biological systems can be represented as networks. Biological networks have become the subject of intensive research in recent years. Several classes of biochemical networks have been studied extensively:

- Gene transcription networks: In these networks a directed edge from A to B indicates that A encodes a transcription factor for B [11, 5, 12].

- Metabolic networks: In these networks a directed edge from A to B indicates the existence of a metabolic reaction of which A is an educt and B a product [13].

- Protein interaction networks: These are undirected networks representing physical interactions between proteins [14].

- Neural networks: In such networks nodes represent individual neurons and directed edges, synaptic connections between these [15]. Determining the topology of neuronal networks is quite difficult in practice therefore sometimes neural networks are considered at the larger scale where nodes represent functional modules and edges connections between these [16].

- Food webs: these networks represent trophic relationships between a group of species that share the same habitat. A directed edge from A to B indicates that A preys on B [17].

2.2.3 Social and economic networks

Social sciences is one of the scientific disciplines that has a long tradition of quantitative network analysis [18, 19]. In its most general form a social network represents social interactions/relations between a set of individuals or groups of individuals. Some of the possible relations/interactions are: Friendships, acquaintances, geographical proximity, legal relations such as marriages, sexual contacts, business relations, co-ownership, communications, financial transactions and scientific collaborations.

2.2.4 Other networks

Other widely studied networks are citation networks between scientific publications [20], word adjacency and co-occurrence networks [21] and genealogical trees [22].

2.3 Network properties

There is a large number of network measures related to various structural properties of networks. Here, we will briefly review some of the most widely studied network properties and measures. Our primary focus will be on topological properties.

2.3.1 Geodesic path lengths

In many real world networks vertices seem to be connected by short paths [23, 3]. For connected graphs this can be measured by the mean geodesic distance:

$$l = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i>j} d_{ij}, \tag{2.1}$$

where d_{ij} is the length of the shortest path connecting i and j. As defined above the average path length is infinite for networks that are not connected, in such cases one might only consider the largest connected component or use one of the several other related measures such as the proposed in the literature [3].

In general networks are said to have the small world effect if l scales

logarithmically or slower with the number of vertices. The small world effect has been proven for various random graph models and has been observed in a large number of real world networks [3, 8].

2.3.2 The degree distribution

Another aspect with respect to which real world networks differ from classical random graphs is their degree distribution. In contrast to the Erdös-Renyi type random graphs that have a Poisson type degree distribution that decays exponentially for large degrees many real world networks have highly right skewed degree distributions, in other words they contain a large number of nodes with unexpectedly high connectivity. In directed networks one can differentiate between two types of degrees: the in- and out- degree. In some cases one can also define the degree corresponding to bidirectional edges. These different types of degrees are in general highly correlated.

Networks with power law degree distributions have attracted a great deal of interest [24, 3]. These networks are sometimes referred to as scale free networks. Power law degree distributions with various exponents have been observed in several real world networks including metabolic networks, the Internet, the World Wide Web and communication networks.

2.3.3 Clustering

Many real world networks have high clustering, sometimes also called transitivity, meaning that there is an increase probability that vertices v_1 and v_2 are connected if both are connected to a third vertex v_3. This phenomenon

is in general quantified the clustering coefficient:

$$C = \frac{6n_{\triangle}}{n_V}, \tag{2.2}$$

where $n_{\triangle}$ and n_V are the number of triangles and two paths in the network. One can also define a clustering coefficient for individual vertices:

$$C_i = \frac{\text{number of triangles containing vertex i}}{\text{number of two paths with central vertex i}} \tag{2.3}$$

In this approach the clustering coefficient of the whole network is defined the average of the C_i's. One can also define higher order clustering coefficients corresponding to cycles of order higher than 3.

2.3.4 Modularity and community structure

Most social networks are believed to have community structure, that is they contain groups of nodes that are more densely connected within themselves and less so to nodes in other groups. For instance, in scientific collaboration networks might represent different areas of research. In the case of the WWW communities might correspond to common subjects of web pages. In biological and technological networks communities can correspond subnetworks that perform a certain function. A lot of research effort has been focused on finding techniques that can successfully extract community structure from networks. A review of such techniques can be found in [25, 26].

2.3.5 Mixing patterns and assortativity

A large class of graph measures are related to mixing patterns of vertices of different types. If vertices of the same or similar type have an increased probability of being connected the network is called assortative and disassortative in the opposite case [27]. A special case of this is mixing with respect to vertex degree for which there exist various measures [3].

2.3.6 Other properties and measures

Some authors have studied the spectral distribution of certain special matrices that can be defined on the basis of graph connectivities. The most notable of these are the adjacency matrix and the graph Laplacian. The spectra of such operators contain important information about the structure of networks and provide an almost complete set of invariants for a graph [28, 29, 30, 31]. Spectral methods have also been used to study the dynamical properties of networks as well as community structure [32, 33].

Networks also have studied extensively with respect to their dynamical properties. For instance, measures related to the resilience of a network against random and targeted node and edge removals have been studied extensively [34, 35, 36]. Networks have also been studied with respect to synchronization, navigation and spreading processes[33, 37].

2.4 Random graph models

In this section we will briefly review some widely used random graph models. We shall focus on models that are directly related to development of this thesis.

2.4.1 The Erdös-Renyi model

The Erdös-Rényi (ER) model is probably the most extensively studied random graph model [38]. In the Erdös-Rényi model edges occur independently with a fixed probability p and for undirected graphs it can be formulated as follows: Let $V = \{v_1, v_2, v_3..v_N\}$ be a set of vertices and the set $E = \{\{v_i, v_j\} : v_i, v_j \in V\}$ the corresponding set of all potential edges. Then the Erdös-Rényi model $G_{N,p}$ is the random graph where every edge occurs independently with fixed probability p. Thus the probability of any graph G is given by:

$$P(G) = p^{|E(G)|}(1-p)^{C_2^N - |E(G)|}. \tag{2.4}$$

[1]

Now, we recall some well known properties of the ER model, for a more comprehensive account we refer to[39]. The first quantity of interest is the degree distribution which is given by:

$$P(k) = C_k^{N-1} p^k (1-p)^{N-1-k} \tag{2.5}$$

For large graphs with fixed mean degree $\kappa = pN$ this approaches the Poisson

[1] $C_k^N = \frac{N!}{(N-k)!k!}$

distribution:

$$P(k) \simeq \frac{\kappa^k e^{-\kappa}}{k!} \tag{2.6}$$

The degree distribution of the ER random graph is narrowly concentrated around its mean and therefore ER model is considered to be a rather poor model for real world networks with heavy tailed degree distributions.

Since in the ER model edges occur independently, given any set s of n vertices the probability the subgraph induced on s is H is given by:

$$P(H) = p^{|E(H)|}(1-p)^{C_2^n - |E(H)|}. \tag{2.7}$$

Thus the n-node subgraph distribution of $G_{N,p}$ is $G_{n,p}$. Consequently, the probability that a motif m appears on any set of $|m|$-nodes is:

$$P(m) = \Lambda(m) p^{|E(m)|}(1-p)^{C_2^{|m|} - |E(m)|}, \tag{2.8}$$

where $\Lambda(m) = \frac{|m|!}{Aut(m)}$ is the number graphs that are in isomorphism class m. In the case of sparse graphs, that is when p is of order N^{-1}, $P(m)$ scales as $N^{-|E(m)|}$. As a result only motifs with $e(m) < |m|$ [2] have a high density. Here, by high density we mean that $\langle n(m) \rangle / N$ is nonzero as $N \to \infty$. Where $\langle n(m) \rangle$ is the expected subgraph count of m. This further implies that the clustering coefficient of the ER model scales as N^{-1}.

One can also consider the microcanonical version of the Erdös-Rényi model $G_{N,e}$ which is the uniform ensemble of all graphs with N vertices and e edges. Many results for $G_{N,p}$ can be translated to the case of $G_{N,e}$ in

[2] For connected graphs this is equivalent to being a tree.

a straightforward fashion [39].

The Erdös-Rényi model can also be formulated for directed graphs. The simplest possible model is to consider every possible directed edge with independently with equal probability p_e. However, in the sparse case the expected number of mutual edges in this model is O(1).If the graph is to have a high density of mutual edges the model can be generalized to include the addition of mutual edges with probability p_m. This model is in a certain sense the simplest version of some random graph model we will consider later.

2.4.2 The configuration model

As mentioned before the ER model has a Poison type degree distribution that is concentrated around its mean and decays exponentially for large degrees. However, most real world networks have much broader degree distributions. The configuration model was proposed in order to incorporate such broad degree distributions into network models. For a given degree sequence d_i the configuration model [40, 41] is defined as the uniform ensemble of all graphs having degree sequence d_i.

In order to generalize the configuration model to arbitrarily large graphs a degree sequence is generated by sampling the corresponding degree distribution: $p(k) = n_k(d_i)/N$. If the degree sequence obtained in this way is not graphical it is discarded and one samples the degree distribution until a graphical degree sequence is obtained.

The configuration model can be sampled in the following way: Given a degree sequence d_i each vertex is assigned a number of half edges called

'stubs' corresponding to its degree. These stubs are then connected to each other by choosing pairs of stubs at random. This process generates each possible configuration with equal probability. However one should notice that this process can generate self edges as well as parallel edges. The model can be refined to exclude such possibilities however the resulting model is much more difficult to treat analytically.

The configuration model can be generalized to directed graphs by considering both in- and out- degrees of vertices. In addition to these one can further consider the degree corresponding to mutual edges. In both cases the configuration model can be defined as the uniform ensemble of all graphs having the same degree sequence. The process for generating graphs corresponding to this model is also similar: one simply matches an incoming stub with an outgoing stub.

The configuration model can be treated in a elegant way using generating functions [41]. Using this and other techniques analytic results can be obtained for many of its properties including component sizes, path length distribution and percolation properties.

Although the configuration model can account for broad degree distributions it fails to account for the large number of triangles and other densely connected subgraphs observed in many real world networks. Estimates of subgraph densities for various degree sequence types can be found in [42].

There also exist several other models that can produce graphs with predefined degree distribution. In the model proposed by Chung and Lu [43] each vertex is assigned an expected degree k_i according to a degree distribution $p(k)$ and for each pair of vertices $\{i, j\}$ one adds an edge with probabil-

ity that is proportional to $\frac{d_i d_j}{\bar{k}}$. Although this models realizes the degree sequence only on average it is much easier to treat analytically.

Bollobas et al. [44] consider a further generalization of this type of model where nodes are labeled according to a type distribution $P(S)$ and for every pair of vertices $\{i, j\}$ an edge is added independently to the graph with probability $p(i, j) = \kappa(s_i, s_j)$ where $\kappa(\cdot, \cdot)$ is some positive real valued function.

2.4.3 Exponential random graphs

Exponential random graph models (ERGM) are a very general class of random graph models. In its most general form the models are distribution over the set off all graph on N vertices ($\mathcal{G}_\mathcal{N}$) where every graph G has probability:

$$p(G) = \frac{1}{Z} \exp(-\sum_i \beta_i \phi_i(G)), \tag{2.9}$$

where $\{\phi_i\}$ is a set of graph functions and $\{\beta_i\}$ a set of real valued free parameters. The function $\sum_i \beta_i \phi_i(G)$ is generally called the Hamiltonian. Although there is no general restrictions on the graph functions in most cases these are chosen to be subgraph counts of various motifs (edges, triangles, cliques...). $Z = \sum_{G \in \mathcal{G}_\mathcal{N}} e^{-\sum_i \beta_i \phi_i(G)}$ is the normalization factor, called the partition function.

Exponential random graphs are of special importance from an information theoretical perspective since they correspond to maximum entropy distributions under the constraint that the graph functions have certain expectation values $\langle \phi_i \rangle$ [45].

Although ERGMs offer a simple and elegant way of constructing random

graph models with certain desired properties, they are very difficult to treat analytically. The main difficulty for ERGMs seems to lie in the evaluation of the partition function. Except for some limited special cases there is no known analytical method for calculating Z. As a result, the question of how one is to perform calculations and inference with ERGMs remains mostly open.

Due to the lack of analytical results, various techniques for approximating the partition function have been proposed. However most of these procedures have very long running times for larger graphs and therefore are limited to relatively small graphs [46].

Another problem that one is faced from a modeling perspective when using ERGMs is that they tend to show some pathological behavior. For instance, when the Hamiltonian has a term favoring triangles the model tends to form large clique like regions, not observed in most real world networks [47]. Also for large portions of the parameter space ERGMs are essentially equivalent to some ER model [46].

2.4.4 Other models

Besides the models described above there exist many other types of random graph models. One such class of random graph models are generative models. These are random graph model which are formulated in terms of a random process that generates graphs. The aim of such models in general is to identify a specific mechanism that explains some commonly observed features of networks rather than being candidate models for networks. The most

prominent of such models are the small world (SW) model by Watts and Strogatz [23] and the preferential attachment (PA) model [24] by Barabási and Albert.

The small world model was proposed as model for networks that have high clustering and low average shortest path lengths. The model is based on randomly rewiring a certain fraction of the edges of a regular lattice. The lattice can in principle have any dimension, however in practice is mostly one dimensional with periodic boundary conditions i.e. a ring lattice. Watts and Strogatz showed that as a function of the rewiring probability such models show a regime where the graph has both high clustering and low average path lengths. Variants of the small world model where a certain number of random links is added on to the lattice have also been considered.

Barabási and Albert proposed a network growth model based on preferential attachment and showed that such models have power law degree distributions [24]. In this model, starting with n_0 vertices, one adds new vertices to the network in a stepwise fashion. Each newly added vertex is connected to m existing vertices with a probability that is proportional to their degree. It can be shown that graphs constructed in this way have a degree distribution that follows a power law with exponent -3 for large degrees.

2.5 Network Motifs

The concept of networks motifs was first introduced by Milo et al. in [5] where network motifs are defined to be subgraph patterns that occur more frequently in the network when compared to null model that conserves some

characteristics properties of the network.

A variety of measures have been proposed to asses the significance of motifs of which the most widely used one is the z-score:

$$z_m = \frac{n_m - \bar{n}_m}{std(n_m)}, \tag{2.10}$$

where n_m is the number of times m occurs as an induced subgraph in the network while $\bar{n}_m$ and $std(n_m)$ are the empirical mean and standard deviation of the same quantity. Motifs of which the z-score and frequency exceed thresholds z_{min} and f_{min} are classified to be network motifs. However, for most null models no analytical expressions for the mean and variance of motif counts are known therefore these are mostly determined empirically by sampling the null model.

In most applications all motifs of a certain size n are analyzed simultaneously. In this case the method involves two main steps: 1) the generation of a sample of the null model and 2) the counting of all subgraphs of size n in this sample and the original network.

The first step obviously depends on the choice of null model. In general the null model is taken to be the configuration model corresponding to the degree sequence of the network. There exist several methods for uniformly sampling the configuration model [48]. In order to avoid motifs being classified as network motifs only because they contain some smaller overrepresented motif, Milo et al. propose using a null model that in addition to the degree distribution also preserves lower order motif counts [5]. In order to generate networks corresponding to this null model a simulated annealing

(SA) algorithm is proposed. Although Milo et al. do not explicitly define a null model for this case presumably, it is the uniform ensemble of all graphs that have the same degree distribution and lower order motif counts as the network. However, there is no guarantee that the SA algorithm samples such null models uniformly. Moreover, conserving lower order motif counts is not computationally feasible for large motifs and thus the configuration model is used in most applications. Consequently, most subgraphs that contain a smaller overrepresented motif are classified as network motifs. In some cases, motifs that contain a vertex of degree 1 are excluded from the analysis in order to keep the number of network motifs manageable for large n.

The next step is to count all subgraphs of size n in the original network and randomized sample. Here one should mention that there exist several different frequency concepts used in motif analysis [49]. In their original article Milo et al. count all connected induced subgraphs effectively allowing for arbitrary edge and vertex intersections. This is also the most commonly used frequency concept and most counting algorithms are developed for this case [5, 50, 51, 52, 53]. Existing motif analysis algorithms [49] mostly differ with respect to the algorithm they use to count subgraphs.

A general problem one faces when using the method of Milo et al. is that subgraph counts are in general dependent quantities. Such dependencies can mostly be traced back to the fact that the presence of a motif implies the presence of motifs containing it as a submotif and its own submotifs. This further implies that counts of motifs that have a submotif in common are correlated. Milo et al. propose a null model that also conserves lower order motifs in order to account for the submotifs-motif dependencies. However,

this does not account for all dependencies. For instance, a certain overrepresented motif might occur almost exclusively as a submotif of one or more larger overrepresented motifs. A proper analysis of inter motif dependencies requires analytically solvable random graph models that can incorporate high densities of network motifs. In theory exponential random graph models could be used to address such questions but as mentioned before these are difficult to treat both analytically and numerically.

Other critics of the approach have argued that for certain networks the null hypothesis is ill-posed and that the presence of motifs in certain networks is rather well explained by them being embedded in physical space, their hierarchical organization and/or community structure [54, 55]. The use of the z-score as a measure of motif significance has also been criticized because the distribution of motif counts in the null model might not be narrowly concentrated around their mean.

Although methods for detecting network motifs have their shortcomings, there is a large body of evidence that suggests that network motifs play an important role in the structural and functional organization of networks. For instance, dynamical systems defined of network motifs observed in biological networks suggest that network motifs can perform modular tasks such as information processing [4]. For a more detailed review of dynamical properties of motifs we refer the reader to [4, 56]. There is also evidence that networks can be classified with respect to their motif structure [57].

Chapter 3

Background II: Information Theory and Inference

3.1 Information theory

In this chapter we introduce information theoretic concepts relevant to the development of the thesis. *A* more detailed treatment of the subjects can be found in [58], [6] and [59].

3.1.1 Entropy and Shannon information

The concept of entropy was first introduced in the context of thermodynamics by Clausius. The connection between entropy and information was later established by Shannon in his seminal paper [60]. In order to define the entropy we must first define what an ensemble is.

Definition 3.1.1. An ensemble, $E(R, p_r)$, is a set of mutually exclusive

alternative outcomes $R = \{r_1, r_2, r_3..., r_n\}$ together with a corresponding set of probabilities p_r. The probabilities p_r satisfy the conditions: $0 \leq p_r \leq 1$ and $\sum_{r \in R} p_r = 1$.

An ensemble is said to be *uniform* if all its elements have equal probability.

The uncertainty about the outcome is highest when all outcomes are equally likely similarly. Similarly, when one of the outcomes has probability 1 we have complete certainty. The entropy makes this notion mathematically precise and for an ensemble $E(R, p_r)$ is given by:

$$S(E) = -K \sum_{r \in R} p_r log p_r, \tag{3.1}$$

where K is a positive constant that determines the unit of information. When K=1 and the logarithm is base 2, the entropy is measured in bits. The entropy measures ignorance or uncertainty as a function of probabilities.

Shannon, in his seminal paper [60] also proved that $S(E) = -K \sum_{r \in R} p_r \log(p_r)$ is the unique function, up to the multiplicative constant K that satisfies the following conditions:

1. S is a continuous function of the probabilities,
2. S should be monotonically increasing function of N when all p's are 1/N,
3. $S(A \times B) = S(A) + S(B)$ whenever A and B are independent random variables.

3.1.2 Codes and Ensembles

Definition 3.1.2. Let $E(R, p_r)$ be an ensemble and $\mathcal{B}^*$ the set of all binary strings of finite length. *A* binary source code for a random variable corresponding to $E(R, p_r)$ is a mapping $C : R \to \mathcal{B}^*$. $C(r)$ is the code word for r and $l_C(r)$ its length. The expected length of C is $L(C) = \sum_r p_r l_C(r)$.

C is called *nonsingular* if every r has different code and *prefix free* if no codeword is a prefix for any other codeword.

Theorem 3.1.1. *(Kraft-McMillan inequality) For any, binary prefix code the codeword lengths $l_1, l_2, ..., l_n$ must satisfy:*

$$\sum_i 2^{-l_i} \leq 1. \tag{3.2}$$

Conversely, given a set of code lengths that satisfy this inequality there alway exists a code with corresponding code lengths.

A code is called optimal if it minimizes $L(C)$. For optimal codes the following holds:

$$S(E) \leq L(C) < S(E) + 1. \tag{3.3}$$

There are several procedures for constructing optimal codes from probabilities, the most famous being Huffmann coding. Another, widely used coding procedure is Shannon-Fano coding which assigns code lengths $L(r) = \lceil -logp_r \rceil$ [1].

The above equation establishes a correspondence between code lengths of optimal codes and probabilities. That is, given a code for a set R one

[1] $\lceil x \rceil$ is the smallest integer larger than x.

can construct a corresponding (possibly defective i.e. $\sum_{r \in R} p_r < 1$) probability distribution over it by setting $p_r = 2^{-l_r}$. The correspondence between code length functions and probabilities relates many different approaches to inductive inference [61, 62, 59, 63].

3.1.3 Information measures

Shanon information proved to be very useful quantities however in many practical instances, it might not always be easy to define an appropriate ensemble to determine Shannon information. This has lead to the formulation of several non-statistical information measures. Information measures are measures that share some key features with Shannon information, namely:

1. $I(A) \geq 0$
2. $I(A, B) = I(B, A)$
3. $I(A, B) \geq I(A)$
4. $I(A) + I(B) \geq I(A, B)$

A function that satisfies the above conditions is called an *information measure*. As a result of these properties, a number of nonnegative quantities can be associated to every information measure. The conditional information, $I(A|B) = I(A, B) - I(B)$, measures the amount of information needed to describe A given B. The mutual information, $I(A : B) = I(A) + I(B) - I(A, B)$, measures how much information A and B have in common whereas the information distance, $\delta(A, B) = 2I(A, B) - I(A) - I(B)$, measures the information that is not held in common by A and B.

Some measures only satisfy the conditions of being an information measure approximately, that is within small additive constants. One such approximate information measure is the algorithmic information content, also known as Kolmogorov complexity.

3.1.4 Algorithmic information content

One of the information measures that will be of particular interest to us is the algorithmic information content. The algorithmic information content is closely related to the theory of universal Turing machines [63].

Definition 3.1.3. The algorithmic information content (AIC) or Kolmogorov complexity of a string s with respect to a universal Turing machine U is defined as length of the shortest program that instructs U to print out s and then halt.

We will assume that U accepts binary inputs only and therefore the AIC is measured in bits. From now on we assume that U is a universal prefix Turing machine i.e. a machine for which the programs that halt form a prefix code [63]. For such machines, the AIC is an approximate information measure meaning that it satisfies the conditions of being an information measure within a small additive constant.

The crucial observation made by Kolmogorov was that the AIC is essentially computer independent. If U' is a universal Turing machine and U any other turing machine we have:

$$K_U(s) \leq K_{U'}(s) + c_{U'}, \tag{3.4}$$

where $c_{U'}$ is a constant that is independent of s. $c_{U'}$ is essentially the length of the program that instructs U to simulate U'. Although the constant might be relatively large, for sufficiently long strings one can neglect the contribution from $c_{U'}$.

Unlike Shanon information AIC is an intrinsic property of each individual string and does require the entity to be embedded in to an ensemble.

On the other hand when the set of possible strings make up an ensemble the average conditional AIC is closely approximated by the Shannon information:

$$-\sum_r p_r log p_r \leq \sum_r p_r K_U(r|E) \leq -\sum_r p_r log p_r + C_U(E), \qquad (3.5)$$

where $K_U(r|E)$ is the length of the shortest program for r given a description E and $C_U(E)$ is the length of the program that instructs U to form a optimal code for the members of the ensemble. As was demonstrated by Schack [64] for any ensemble E and any U there exists a modified universal Turing machine U' for which:

$$-\sum_r p_r log p_r \leq \sum_r p_r K_{U'}(r|E) \leq -\sum_r p_r log p_r + 1. \qquad (3.6)$$

This shows that when a description of the ensemble is given the AICs of the members of the ensemble are essentially equal to the lenghts of the corresponding Huffmann code.

As consequence of the halting problem which in turn is closely related to Gödel's incompleteness theorem, the AIC of strings is uncomputable [58]

. Therefore, in practice one is forced to work with upper bounds instead of exact values.

Given a set of strings $S = \{s_1, s_2, s_3, ..., s_n\}$, every prefix code C for S results in an upper bound for the AICs of its members that has the form:

$$K_U(r) \leq l_C(r) + c_U(C), \tag{3.7}$$

where $c_U(C)$ is the length of a description of the code. However, such upper bounds can be rather weak since the constant term can be relatively large unless the code has some short description.

The following theorem is an immediate consequence of the fact that number of programs having length less than k is bounded from above by 2^k [63]:

Theorem 3.1.2. *Let S be a set of cardinality N, then for every fixed t and positive integer k there are at most $N2^{-k}$ elements of S that for which $K_U(s|t) < logN - k$ holds.*

Alternatively, for the uniform ensemble defined on S we have:

$$P(K_U(s|t) < log(N) - k) < 2^{-k}.$$

3.1.5 Codes for integers

In principle, every integer can de encoded by the sequence that corresponds to its binary expansion giving $l(n) = \lceil log(n) \rceil$. However, this is not a prefix code. One way to resolve this would be to include a header that specifies the length of the integer to follow. Considering such headers recursively, results

in the log^* code [58]. For the log^* code we have:

$$l(n) = log^*(n) = log(n) + log(log(n)) + log(log(log(n))) \cdots , \tag{3.8}$$

where the sum is taken over all positive terms.

The log^* code is a universal code for the integers in the sense described by Rissanen [65].

3.1.6 Codes for motifs

One way of encoding motifs is to use edge lists. In such a code, one simply encodes the number of nodes and edges using a universal code for integers and given these, the list of edges using constant length codes. The resulting code length is given by:

$$l_e(m) = \log^*(|V(m)|) + \log^*(|E(m)|) + S(|V(m)|, |E(m)|), \tag{3.9}$$

where $S(|V(m)|, |E(m)|)$ is the entropy of the ensemble of all graphs with the same vertex and edge counts as m.

Such a code, however, is not one to one since for every motif m there are $\frac{|m|!}{|Aut(m)|}$ labeled graphs/edge lists. Moreover, the probability that $|Aut(m)| = 1$ is known to converge to one as $|m|$ tends to infinity [66]. The code can be made one to one, for instance, by picking the edge list that has minimum lexicographical order. However, this is less of a concern to us since we are primarily interested in code lengths and not codes themselves.

On the other hand one can also construct universal codes for motifs using

the log^* code for integers. For this, one simply needs to define a bijection between the set of integers and the set of motifs, in other words a counting procedure. Given such a mapping, the code of a motif is simply taken to be the code of its integer label. One way of counting/ordering motifs is to first order motifs according to order and then with respect to the number of edges. Motifs having the same edge and vertex counts are then ordered in the following way: for every motif consider all labelled representatives, order the edges of these (i, j) $(i < j)$ according to lexicographical order which results in a unique form $e_1 e_2 ... e_m$ $(e_1 < e_2)$ for every labelled representative. For every motif, pick the labelled representative that has smallest lexicograhical order and then order all motifs lexicographically according to these representatives. This proceedure defines a total order on all motifs and thus can be used to map every motif to a unique integer. The same ordering procedure can also be applied to more restricted motif classes such as connected motifs.

3.2 The total information framework

3.2.1 Effective Complexity

The effective complexity is a complexity measure that was proposed by Gell-Mann [67, 6]. Gell-Mann's approach is based on the idea that given a certain entity e, identifying certain regularities of e is essentially equivalent to embedding it into an ensemble E of which the members share these regularities while they differ in other aspects. The effective complexity of e with respect

to E is then defined to be algorithmic AIC of E:

$$\epsilon_E(e) = K_U(E). \tag{3.10}$$

We further require that e should be a typical member of E, in other words $-log(p_e)$ should not be much larger than the entropy of the ensemble. Otherwise E can not be considered to accurately represent regularities of e since e would be a relatively improbable member of E.

However, in general there might many ensembles into which e can be embedded as a typical member. The question of how one is to select one of such ensembles over the others, brings us to the concept of total information.

3.2.2 Total information

Total information is an approximate information measure that was introduced by Gellmann and Hartle [6, 68] . Given an entity e and an ensemble E into which it can be embeddded, the corresponding total information Σ_E is defined as the sum of the information required to describe both the regularities/ruled based features and random/probabilistic aspects of e. While the information required to describe the regularities is measured by the effective complexity, the information required to describe the random aspects of the entity is measured by the entropy of the ensemble:

$$\Sigma_E(e) = \epsilon_E(e) + S(E), \tag{3.11}$$

where $\epsilon_E(e)$ and $S(E)$ are given by equations 3.10 and 4.3 respectively.

The total information further provides a basis for comparing different models for e. The smallness of the total information is a measure of how well an ensemble describes a given entity. On the other hand, the smallness of the total information might not always be sufficient to differentiate ensembles. In such cases, the smallness of the effective complexity is used as a criterion. In other words, given ensembles with the same total information one maximizes S at the expense of ϵ. Gell-Mann and Llyod furher suggest placing a cut-off on the computational time required to generate a typical member e of the ensemble E. That is, on the time required to produce a typical member of E given a minimal description of E together with the corresponding Huffmann code for e. Here, the cut-off is taken to be larger than the time required to compute e from its minimal program but of the same order. This essentially excludes ensembles of which the membership and probabilities are difficult to compute and has the furher effect that information that is hard to compute is included in the description of E. Together with this additional constrain on computational complexity, the total information provides a framework for comparing ensembles that in many regards is independent of the observer.

3.3 Alternative approaches

Although this thesis follows the total information approach there exist many other approaches some of which are closely related to the total information. Most of these approaches are related to the total information approach and also to each other through the correspondence between probability distributions and codes. We believe that each approach provides a unique and usefull

perspective on the problem. In the following we will briefly describe some of these approaches that are closely related to the total information approach.

Many approaches to inductive inference were developed with a focus on models with continious parameters and for situations where multiple/repeated observations are made. However in this thesis, we deal with a situation that is quite different, that is we have a single discrete observation (the network) from which we try to infer a discrete representation. As we shall see later, the inference of a subgraph cover is equivalent to the inference of the latent state of certain random graph models.

3.3.1 Maximum likelihood

The simplest model selection approach is the maximum likelihood approach. In the maximum likelihood approach, given alternative models of the data one simply picks the model that maximizes the probability of the data. For uniform ensembles, maximizing the likelihood is equivalent to minimizing the entropy. However, the maximum likelihood approach is prone to overfitting.

3.3.2 Bayesian inference

Given a certain observation D, Bayesian inference assigns to every model M a posterior probability $P(M|D)$ based on a prior probability $P(M)$ over models and the likelihood $P(D|M)$ of data D given the model M:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}. \tag{3.12}$$

In the simplest form of Bayesian model selection, one picks the model that has maximum posterior probability.

3.3.3 Minimum description length (MDL)

The MDL approach introduced by Risannen [62, 69] is a compression based model selection approach. In a nutshell, it asserts that given alternative models for a given set of observations, one should select the model that results in the shortest description of the data. In its two part version the description length $L(D)$ of a data D is given by:

$$L(D) = L(D|M) + L(M),$$

where $L(M)$ is the desciption lengths of M and $L(D|M)$ is the description length of D given by M. In other words the description length of the data is the sum the code lengths of a description of the model together with an optimal encoding of the data obtained using the model.

3.3.4 Minimum message length (MML)

The minimum message length (MML) approach which was first introduced by Wallace [70] follows a messaging approach. Similar to MDL, it asserts that given alternative models for of the observation one should pick the model that minimizes the length of a message used transmit the observation to a reciever. Where the message consists of two parts the first being a code for the model M and the second part encodes the data using M: $L(D) = L(M) + L(D|M)$. Where the models are encoded using the prior of the reciever and when the

reciever is a Turing machine using minimal programs.

More detailed comparisons of various approaches can be found in [59, 61].

Chapter 4

Subgraph Covers

4.1 Subgraph covers and graph representations

Throughout the rest of this thesis we will assume that the graphs under consideration are sparse, i.e. $|E(G)| = O(N)$. Most real world networks are sparse [3]. First we give some basic definitions:

Definition 4.1.1. A subgraph cover C of a graph $G = (E, V)$ is a set of subgraphs of G such that $\bigcup_{H \in C} E(H) = E(G)$.

Subgraph covers are representations of graphs meaning that given any cover $E(G)$ can be fully recovered.

Definition 4.1.2. The motif set $(M(C))$ of a subgraph cover C is the set of isomorphism classes of the subgraphs in C.

Given a set of motifs M, an M-cover is a subgraph cover such that every element in C belongs to a class in M.

Trivial examples of a subgraph covers are cover consisting of only G itself and cover of all edges of G. The edge cover is essentially equivalent to the edge list representation of the graph.

Other examples of subgraph covers are the maximal star and clique covers of G, these are the sets of all cliques/stars that are not a sub-clique/sub-star. Clique covers are closely related to bipartite representations of graphs [72, 3]. The maximal star cover contains all edges except those connected to a vertex with degree 1 twice. For undirected graphs the maximal star cover is essentially to the adjacency list representation of the graph. Moreover, the frequencies of motifs in the maximal star cover contain the same information as the degree distribution of the graph. For directed graphs one can define the maximal inward and outward star covers. These covers cover each edge once.

As mentioned before, subgraph covers are representations of graphs in the sense that given a subgraph cover the corresponding graph can be fully recovered. One can also look at subgraphs covers as decompositions of the graph into its subgraphs. In general, every graph G has a very large number of subgraph covers. However, most of these are not very efficient representations of G. For instance, some covers contain redundant subgraphs that can be removed from the cover without affecting its ability to represent the graph. Therefore we need a measure that tells us how efficiently a subgraph cover represents the graph. Some intuitive candidates for such a measure are the total number of subgraphs in the cover or the sum of the orders of the subgraphs in the cover. Another intuitive measure is the number and complexity of motifs that occur in the cover. For instance, if we have two

covers each containing the same number of subgraphs, in general the one would consider the one of which the motif set is less complex to be a better representation. The total information provides a single measure of optimality that combines many of these intuitive measures. In this chapter in order to define the total information of subgraph covers we use uniform subgraph covers. We then proceed to define the motifs of a network to be subgraph patterns that occur in a subgraph cover that minimizes the total information.

4.2 The total information of subgraph covers

In this section we define the total information of subgraph covers by embedding subgraph covers into uniform subgraph covers. Given a motif set M and count vector $\mathbf{n}_m$ corresponding to the motif counts in the cover, the uniform subgraph cover on N vertices $E_N(M, \mathbf{n}_m)$ is the uniform ensemble of all M-covers that have motif counts $\mathbf{n}_m$. In order to calculate the entropy of $E_N(M, \mathbf{n}_m)$ we need to compute the total number of such covers. For this we first have to consider $H_N(m)$, the total number of distinct, i.e. non-automorphic, m-subgraphs on N vertices. It follows from the definition of the automorphism group that for every set of $|m|$ vertices there are $\frac{|m|!}{|Aut(m)|}$ distinct m-subgraphs. From this it follows that:

$$H_N(m) = \frac{N!}{(N - |m|)!|Aut(m)|}. \tag{4.1}$$

Since the entropy of a uniform ensemble is given by the logarithm of its size, the entropy of a set of n_m subgraph instances of m on N vertices is

given by:

$$\begin{aligned} S_N(m, n_m) =& log \binom{H_N(m)}{n_m} \\ =& log \left(\begin{array}{c} \frac{N!}{(N-|m|)!|Aut(m)|} \\ n_m \end{array} \right), \end{aligned} \tag{4.2}$$

which given m, is the information required to specify n_m instances of m on N vertices. Generalizing the above expression, the entropy of a cover C with motif set $M(C)$ and motif counts n_m is the entropy of the uniform ensemble of all covers with motif counts n_m:

$$\begin{aligned} S(C) =& log \left(\prod_{m \in M(C)} \binom{H_N(m)}{n_m(C)} \right) \\ =& \sum_{m \in M(C)} S_N(m, n_m(C)). \end{aligned} \tag{4.3}$$

When required the entropy terms can be approximated using Stirling's formula. For instance, when n_m and N are large enough and $|m| > 2$:

$$\begin{aligned} S_N(m, n_m(C)) =& n_m(C) \left(|m| log(N) - log(|Aut(m)|) - log(n_m(C)) + log(e) \right) \\ &+ O(log(N)). \end{aligned} \tag{4.4}$$

This expression indicates that covers which contain dense, symmetric and frequent motifs also have small entropy.

As in the case of the entropy, we define the effective complexity of a cover using uniform covers with the same motif counts as C:

$$\epsilon(C) = K_U(E_N(M(C), n_m(C))). \tag{4.5}$$

Consequently, the total information of a cover is:

$$\Sigma(C) = \epsilon(C) + S(C), \tag{4.6}$$

where $\epsilon(C)$ and $S(C)$ are defined as in equations 4.5 and 4.3 respectively.

According to the total information approach the cover that gives an optimal description of the graph is the one that minimizes the total information. Now, we can proceed to define the motif set of graph G: the motif set of G is the set of motifs that appear in its Σ-optimal subgraph cover, in other words $M(C_\Sigma(G))$.

In general, there might be multiple subgraph covers that minimize the total information. In such cases the smallness of the effective complexity can be used to as a further criterion. However, in some cases there might be multiple optimal covers that also have the same effective complexity. If this is the case one has to use additional criteria in order to pick one of these covers over the others. Although, in principle such covers can differ with respect their motifs sets we consider this to be a rather unlikely situation. Another more likely situation is that the optimal covers have same motif set and have almost identical motif counts. In this case one can consider them to be basically equivalent. We discuss the multiplicity of such covers in more detail in the upcoming sections in the context of random graph models.

The effective complexity of the optimal cover $\epsilon(C_\Sigma(G))$ can be interpreted as a measure of the complexity of G's subgraph structure. The effective complexity of the optimal cover is related to many other measures that are frequently used as indicators of a network's complexity such as the broadness of the degree distribution and clustering. A broad degree distribution indicates that the graph contains a large variety of star shaped subgraphs whereas high clustering can be seen as an indicator that the graph contains subgraph patterns other than trees.

Another quantity of interest is the amount of compression the optimal cover provides with respect to the edge cover: $\Delta_\Sigma(G) = \Sigma(C_e(G)) - \Sigma(C_\Sigma(G))$. Since, the optimal cover is also a code for G with length $\Sigma(C_\Sigma(G))$, $\Delta_\Sigma(G)$ can be seen as measure of how much G deviates from a typical Erdös-Rényi random graph (Theorem 3.1.2).

One can also associate a quantitative significance to motifs based on the compression they provide with respect to $C_\Sigma(G)$ that is similar to the z-score used by Milo et al. [5]. One such measure is the c-score c_m, which measures the effective compression provided by m with respect to $C_\Sigma(G)$:

$$c_m(G) = \frac{\Sigma(C_\Sigma(G) - m)}{\Sigma(C_\Sigma(G))} - 1, \tag{4.7}$$

where $(C_\Sigma(G) - m)$ is the cover obtained by replacing the m-subgraphs in $C_\Sigma(G)$ with the single edge subgraphs corresponding to the edges they cover in G. According to this definition c_m is always non-negative and zero for the single edge motif and motifs that are not contained in $C_\Sigma(G)$. One can further construct motif significance profiles based on the c-score that are

similar to those used by Milo et al. in [57]. As in [57] these can be used to classify/categorize networks with respect to their motif structure.

4.3 The relation to the method of Milo et al.

The amount by which G can be compressed is a measure how non-random G is. Therefore, the motifs contained in the Σ-optimal cover correspond to the motifs with respect to which the graph maximally deviates from a Erdös-Rènyi type random graph. Although the Erdös-Rènyi random graph is rarely used as a null model for the method of Milo et al., both methods essentially try to find motifs with respect to which the network differs from a random graph. In this sense, the subgraph cover approach and the method of Milo et al. can be seen as sharing a similar goal. In principle one can also use subgraph cover ensembles that also preserve the degree distribution of the network to define the total information. This would effectively allow for a more direct comparison of the two approaches but as we shall see in the following sections there are additional difficulties associated to this.

As previously discussed one of the main issues when using the method of Milo et al.is the difficulty of resolving interdependencies between motifs. The Σ-optimal cover naturally resolves such interdependence by considering motifs of all sizes simultaneously and effectively penalizing the sharing of edges between subgraphs in the cover. Despite their differences, for most networks one can expect at least some of the network motifs found by both methods to coincide especially when lower order motifs are conserved by the null model.

4.4 Random graphs with motifs

The development of random graph models that incorporate high densities of triangles and other highly connected motifs has been a long standing problem in network research. In the following, we will briefly introduce two such models: the model introduced by Bollobas et al. [2] that is a generalization of non-homogeneous random graphs [44] and the second by Karrer and Newmann [1] that is a generalization of the configuration model.

4.4.1 Sparse random graphs with clustering

In [2] the authors introduced a very general class of random graph models. These models are defined on the basis of a type space (S, μ), a set M consisting of labeled representatives of the motifs of the model and a set K of kernels associated to the elements of M. The type space (S, μ) is a discrete or continuous probability space and kernels K_m are non-negative real valued functions with domain $S^{|m|}$. Then a random graph with vertex set $V = \{1, 2, ..., N\}$ corresponding to such a set $((S, \mu), M, K)$ is defined in the following way. First, every vertex $i \in V$ is assigned a type x_i according (S, μ). Then, for every $m \in M$ and every $|m|$-tuple $(v_1, v_2, ..., v_{|m|})$ of vertices one adds a copy of m such that i^{th} vertex of m is mapped on to v_i with probability:

$$p_m = \frac{K_m(x_{v_1}, x_{v_2}, ..., x_{v_{|m|}})}{N^{|m|-1}}. \tag{4.8}$$

When $p > 1$, the subgraph is added with probability 1. The normalization factor $N^{|m|-1}$ ensures that the average number of copies of m added to the

graph is $O(N)$ i.e. that the model produces sparse graphs.

When formulated as above the model can produce parallel edges, however these are quite rare i.e. $O(1)$ we will assume that these are replaced by single edges in the graph. In their article [2] the authors showed that these models can be solved analytically for many of their properties including degree distributions, component sizes, percolation properties and subgraph densities.

When formulated as above there is a non-zero probability that certain subgraphs are added to the graph more than once. The model can be modified slightly so that every subgraph is considered for addition only once. This can be done by considering unordered $|m|$-tuples (i.e. $|m|$ subsets) of vertices and for each such subset every potential m-subgraph only once. In order to obtain a well defined expression for the probabilities the kernel K_m has to be invariant under Aut(m):

$$K_m(x_{v_1}, x_{v_2}, ..., x_{v_{|m|}}) = K_m(\sigma(x_{v_1}, x_{v_2}, ..., x_{v_{|m|}})), \forall \sigma \in Aut(m)$$

[1] For any kernel K_m one can construct version $\widetilde{K}_m$ that satisfies the condition above by taking its symmetric average:

$$\widetilde{K}_m(x_1, x_2, ..., x_{|m|}) = \frac{1}{|Aut(m)|} \sum_{\sigma \in Aut(m)} K_m(\sigma(x_1, x_2, ..., x_{|m|})).$$

Given a kernel that satisfies the above conditions the corresponding graph

[1] Bollobas et al. use this as a working hypothesis in their paper.

is obtained by adding onto each $|m|$-subset $\{v_1, v_2, ..., v_{|m|}\}$ every distinct m-subgraph m_i $(i = 1, 2, ... \frac{|m|!}{|Aut(m)|})$ with probability:

$$p_{m_i} = \frac{K_m(x_{\phi_i(1)}, x_{\phi_i(2)}, ..., x_{\phi_i(|m|)})}{N^{|m|-1}}, \tag{4.9}$$

where $\phi_i()$ is a 1-1 map from the set of vertices of m to $V = \{v_1, v_2, ...v_{|m|}\}$ such that $\phi_i(x)\phi_i(y) \in E(m_i)$ whenever $xy \in E(m)$. Since K_m is invariant under Aut(m), p_{m_i} does not depend on the choice of ϕ_i.

The models described above are actually distributions over the set of all M-covers, $\mathcal{C}_M$. The state space of these models can be written as $\prod_{m \in M} \{0, 1\}^{H_N(m)}$ where:

$$H_N(m) = \frac{N!}{(N - |m|)!|Aut(m)|},$$

is the number of distinct m-subgraphs on N vertices. The corresponding probability distribution over the space of all graphs on N vertices is obtained by projecting subgraph covers onto graphs:

$$P(G) = \sum_{C \in \mathcal{C}_M(G)} P_{S,M,K,N}(C), \tag{4.10}$$

where $\mathcal{C}_M(G)$ is the set of all M-covers of G and $P_{S,M,K,N}(C)$ is the probability of such a cover. Therefore from now on we will refer to these models subgraph cover models (SCM).

Homogeneous models

The class of models described above is very general and many type space-kernel combinations will be equivalent. In the simplest case where kernels

are constant, the type space can be left out of the formulation. In this case every possible m-subgraph is added independently to the graph with the same probability p_m that is of order $N^{1-|m|}$. We will call such models homogeneous subgraph cover models (HSCM). Uniform subgraph covers are essentially microcanonical versions of homogeneous models. For homogeneous models the probability of any M-cover C is:

$$P_{M,p_m,N}(C) = \prod_{m \in M} (1-p_m)^{H_N(m)-n_m(C)} p_m^{n_m(C)}. \tag{4.11}$$

Now, we will consider some properties of homogeneous models in order to get a better picture of uniform subgraph covers in general. For a more detailed treatment of the model we refer the reader to [2]. Similarly, for uniform subgraph covers :

$$P_{M,\mathbf{n}_m}(G) = \frac{n_{(M,n_m)}(G)}{n_{(M,n_m)}} = \frac{n_{(M,n_m)}(G)}{2^{S(M,n_m)}}, \tag{4.12}$$

where $n_{(M,n_m)}(G)$ is the number of (M,n_m)-covers of G and $n_{(M,n_m)}$ is the total number of (M,n_m) covers on N vertices.

The number of two node intersections between subgraphs: Let $\{v_1, v_2\}$ be any pair of vertices, the probability that more than one subgraph in the cover contains both these vertices is equal to the measure of the set of all subgraph covers that satisfying this condition.

$$P(I_{v_1,v_2} \geq 2) = 1 - \prod_{m \in M} (1-p_m)^{H_2(N,m)} - \sum_{m \in M} p_m H_2(N,m)(1-p_m)^{H_2(N,m)-1}, \tag{4.13}$$

where $H_2(N,m) = \binom{(N-2)}{(|m|-2)} \frac{|m|!}{Aut(m)}$ is the number of m-subgraphs that contain v_1 and v_2. $P(I_{v_1,v_2} \geq 2)$ is $O(N^{-2})$ and since there are $O(N^2)$ pairs, the expected number of two node intersections between the subgraphs in a HSCM is $O(1)$. In general the expected number of n node intersections between subgraphs in the cover $O(N^{2-n})$.

Subgraph distribution: Given a set of vertices $s = \{v_1, v_2, ..., v_n\}$ we will call the subgraph H induced on s the state of s. The only M-subgraphs of which the state contributes to state of s are those that have at least two vertices in s. In order to consider the contribution of a specific motif m to the state of s for each k-subset ($k \geq 2$) of s we have to consider the contribution of all m-subgraphs that contain this subset of vertices. For every k-subset of s ($k \leq |m|$) there are:

$$\binom{(N-n)}{(|m|-k)} \frac{|m|!}{Aut(m)}, \tag{4.14}$$

such m-subgraphs. If any of these m-subgraphs is in the cover it contributes its subgraph induced on the set vertices it shares with s to the state of s. Thus, the contributions of such m-subgraphs are determined by the distribution of induced k-submotifs of m. Since the contributions of individual m-subgraphs are independent, for any k-subset s_k of s the probability that t m-subgraphs which contain s_k have a specific m'-subgraph induced on s_k is:

$$p(t,m,m') = p_m^t (1-p_m)^{c(m,m')\binom{(N-n)}{(|m|-k)}\frac{m!}{Aut(m)}-t}, \tag{4.15}$$

where $c(m,m')$ is the number of m-subgraphs that contain the $|m'|$-subgraph under consideration as an induced subgraph. $c(m,m')$ can be expressed in

terms of the density $d(m, m')$ of m' in m as a induced subgraph:

$$c(m, m') = d(m, m')\frac{|Aut(m')||m|!}{|Aut(m)||m'|!}.$$

Moreover, this shows that the contributions corresponding to $t \geq 2$ are essentially negligible.

Thus the n-node subgraph distribution of HSCMs are essentially equivalent to a HSCM on n-nodes of which the motif set M'_n consists of all the induced submotifs of M up to size n. The subgraph distribution differs from a HSCM slightly that is, some motifs appear more than once in M'_n and that some of m'-subgraphs can be added more than once on to a specific subset. On the other hand, one can always construct a proper HSCM which induces that produces the same distribution over graphs. The distribution above shows that the probability that at a submotif $m' \in M'_n$ is contributed to a specific $|m'|$-subset of vertices is $O(N^{1-|m'|})$. This implies that the only connected n-node motifs that have non-zero density as induced subgraphs are those which are in M'_n and those which are singly connected combinations of motifs in M'_n.

4.4.2 Generalized configuration models

In the generalized configuration models introduced by Karrer and Newman [1] can be seen as a generalization of the configuration model that incorporates non trivial subgraphs. In order to formulate the model we first need the following definition:

Definition 4.4.1. Let C be a subgraph cover with motif set

$M = \{m_1, m_2, ..., m_n\}$ and for each motif m let $T_m = \{t(m)_1, t(m)_2, ..., t(m)_{k(m)}\}$ be the set of orbits of $Aut(m)$. The role $r(i)_{m,t}(C)$ of vertex i according to C is the number of m-subgraphs in C for which i is in orbit t. The sequence $\mathbf{R}(C)$=$(r(i)_{m,t}(C))$ is called the role sequence of C.

The generalized configuration model generalizes the edge configuration model by using role sequences. In their paper Karrer and Newman describe this model as a generating process similar to the stub matching method for the edge configuration model. Given a role sequence $\mathbf{r}$ corresponding to a motif set M one attaches to every vertex a motif stubs reflecting its role vector. Then a network is generated by matching stubs corresponding to the same motif m in appropriate combinations at random and connecting them to form an m-subgraph until all stubs are exhausted. This process samples all possible configurations uniformly. Although parallel edges might be formed by the process, for large N the expected number of such edges is O(1). On the other hand the process can also match stubs of the same vertex to each other resulting in a subgraph that is a vertex contraction of the original motif. If one excludes such cases from the model, every matching of the stubs corresponds to an M-cover. This allows us to formulate the generalized configuration models in terms of subgraph covers: The generalized configuration model with role sequence $\mathbf{r}$ is the uniform ensemble of all subgraph covers that have role sequence $\mathbf{r}$ and the probability of a graph G in this model is simply:

$$P_{\mathbf{r}}(G) = \frac{|\{C : \mathbf{R}(C) = \mathbf{r} \wedge Graph(C) = G\}|}{|\{C : \mathbf{R}(C) = \mathbf{r}\}|}. \tag{4.16}$$

As for the configuration model the generalized version can also be defined using a role distribution according to which vertices are assigned roles. Similar to the edge only case not every role sequence generated in this way is graphical. For instance, a sequence that is obtained by sampling the role distribution does not always contain the roles in appropriate combinations. If this is the case, the role sequence is discarded and a new role sequence is drawn from role distribution. Once a graphical role sequence is obtained the network itself is generated as described above.

The degree sequence (counting multiple edges) of graphs generated by the configuration model is fully determined by the role sequence and given by:

$$d_i = \sum_{m \in M} \sum_{t \in T_m} r(i)_{m,t} d_t, \tag{4.17}$$

where d_t is the degree of a vertex with role t in m.

Subgraph densities: As with the SRCM model the number of subgraphs that intersect on two or more nodes in the generalized configuration model is O(1). Thus, the only biconnected subgraphs with high density are the motifs of the model and their submotifs. On the other hand, densities of singly connected subgraphs are more difficult to calculated and as far as we know no general analytic formula is known. However, subgraphs that consist of one node intersections of biconnected submotifs are an exception since their density is almost conserved by the virtue of the role sequence in a similar way in which star counts are determined by the degree distribution. This suggests that one does not loose much structure if singly connected motifs are excluded from the model. In order to clarify this point we consider

several examples.

Star models vs edge models

Let us consider two models: the first one being a generalized configuration model with a motif set of made of star subgraphs and a role distribution of which the entries are independent and the second one being the edge only configuration model corresponding to the degree distribution of the first model. These two models are very similar in many regards: they are both locally tree like, have the same degree distribution and basically have no degree correlations. Therefore, in this case the edge only model seems to be superior to the star model since it is less complex.

However, this is not enough to completely exclude singly connected motifs from generalized configuration models. In order to clarify this point let us consider a second example: Let G be a graph, now compare the configuration model corresponding to the degree distribution of G and the star model corresponding to the cover obtained in the following way. Starting from the vertex with the highest degree (if there are more than one pick at random) add the corresponding star subgraph to the cover and remove the corresponding edges from the graph and repeat until no more edges are left. This can be seen as a simple heuristic for obtaining an efficient star cover. If G has degree correlations, for instance if high degree vertices have tendency to connect to other high degree vertices, we expect this to be reflected in the role sequence of this cover. Therefore, in this case it might be argued that the star model reflects the structure of G better than the edge model. On the other hand, degree correlations can also be introduced to the edge

configuration model by a simple generalization [27] resulting in models that are in general less complex than star models.

4.4.3 Subgraph Covers and Model Selection

The question we would like to address is: Given a graph/network how does one associate it with such models? In [1] the Newman and Karrer state this as an important open problem for the generalized configuration model.

In previous sections we showed that homogeneous models and generalized configuration models are essentially distributions over subgraph covers that are projected on to graphs. Therefore subgraph covers correspond to latent/unobserved states of these models and thus in the context of HSCMs our approach can be seen as inferring such latent states. Since, every cover can be associated to a unique model it also offers a method for model selection.

In the case of subgraph cover models one can further consider models with finite type spaces which can be regarded as generalizations of mixture models that include motifs. The latent state of these models consists of a vertex type configuration in addition to the subgraph cover. For these models, one would have a multitude of ensembles for each subgraph cover corresponding to the different labeling of nodes and the model selection procedure would include the extra step of finding the type assignment that minimizes the total information for each possible cover. Determining such optimal labellings is a generalization of the problem of finding communities in networks which by itself is a highly non-trivial problem. A simpler but less principled approach would be to separate the two problems. First, one could use the edge covers

to determine the vertex types and then find the motifs. Alternatively, one can use uniform models to select a subgraph cover and then using this subgraph cover find a corresponding optimal labeling. This can be further generalized to other types of correlations between the subgraphs in the Σ-optimal cover which can than be modeled using appropriate type/kernel combinations.

For the generalized configuration model the model selection problem is equivalent to determining a role sequence which in turn is essentially equivalent to selecting a subgraph cover of the graph since every subgraph cover defines a unique role sequence. The Σ-optimal cover can be seen as a viable candidate for assigning a role sequence to the network. Following previous discussions one can further consider restricting the motif set to biconnected motifs since the counts of subgraphs consisting of one node intersections of biconnected submotifs can be accounted for using the role sequence. This in general reduces subgraphs that have to be considered in the analysis significantly since the majority of connected subgraphs of sparse networks are only singly connected.

In principle the generalized configuration models can also be used to define the total information of subgraph covers. In such an approach the total information of a subgraph cover C would be defined by the entropy and effective complexity of the ensemble of all subgraph covers with the same role sequence as C. However, this requires the enumeration of all such covers and as far as we know even in the case of the edge configuration model only approximate expression are known [73, 41]. Although, one could use such approximations, these expressions tend to be rather complicated which might pose additional difficulties when devising algorithms for the problem.

These ensembles further have high effective complexity since their description includes the full role sequence which in general requires O(N) bits to describe.

Here, we should note that this is a somewhat indirect way of doing model selection. The classical model selection problem would be to infer a model, that is a set (M, n_m), using the probability distribution over graphs. In other words one would select the model (M, n_m) that minimizes:

$$\begin{aligned}\Sigma_G(M, n_m)(G) =& - log(P_{M,\mathbf{n}_m}(G)) + \epsilon(M, n_m) \\ =& S(M, n_m) - log(n_{M,\mathbf{n}_m}(G)) + \epsilon(M, n_m), \end{aligned} \tag{4.18}$$

where $n_{M,\mathbf{n}_m}(G)$ is the number of $(M, \mathbf{n}_m)$ covers of G. The second line follows from equation 4.12. Thus the total information of the model selection problem and the subgraph cover selection problem differ with respect to the term $-log(n_{M,\mathbf{n}_m}(G))$. Since, $n_{M,\mathbf{n}_m}(G)$ is a function of G exhaustive enumeration of the covers of G seems to be the only way of determining its value which in general will require exponential time. Thus using the probability distribution over graphs as a basis of model selection seems to be rather unpractical.

On the other hand the $-log(n_{M,\mathbf{n}_m}(G))$ in many cases is expected to be much smaller than $S(M, n_m)$ and thus the models corresponding to the optimal subgraph cover can be considered to be rather good approximation to the model selected using the distribution over graphs. In order to illustrate this point let us consider the following example: Let G be a large graph that has a triangle edge cover with edge count e and triangle count t (O(e),O(t)=N),

such that the cover contains all triangles in G and that no two triangles in G have an edge in common. In this case this is also the unique Σ-optimal edge-triangle cover. Moreover, in this case the number of $(e+3k, t-k)$ covers of G is C_k^t. As a result we have:

$$\Sigma_G(e+3k, t-k) = S(e+3k, t-k) - \log(C_k^t) + \epsilon. \tag{4.19}$$

This in turn implies $\Sigma_G(e,t) - \Sigma_G(e+3k, t-k) = 3k\log(e/N) + k\log(4/3) - \log(k!) + O(1/N)$. Since the factorial grows faster than the exponential this show that for such G the edge triangle model corresponding to the optimal cover differs only slightly from the model that is optimal with respect to distribution over graphs. One can obtain similar bounds for the case where the triangles in graph intersect on only O(1) edges, which corresponds to the generic configuration of a random cover. In other words if the network is generated by a uniform edge-triangle cover both methods will find very similar covers ($O(\Delta(n_m)) = 1$) with high probability. Similar arguments also apply to more general HSCMs of which the motifs are biconnected since for these one expects most subgraphs instances to come from the underlying cover. If the motifs are dense enough one can even show that the model found by both methods coincide exactly for large enough N, for instance for the edge-K_4 model.

However, in the case of singly connected motifs the situation is rather different since subgraph instances of these are easily created by one node intersections of other motifs. For instance, consider a cover that contains two 3-stars that share the same central vertex forming a 6 star. This 6-star

can then be covered in $C_3^6 = 20$ different/equivalent ways using two 3-stars each resulting in a different cover with the same motif counts. Thus, if M contains stars and other singly connected motifs the term $log(n_{M,\mathbf{n}_m}(G))$ might not be negligible.

The above discussion also applies to the multiplicity of Σ-optimal covers. That is, if the optimal cover(s) contain large numbers of singly connected subgraphs the number of optimal covers can also be rather large and therefore these might also differ significantly with respect to the subgraphs they contain. On the other hand, if the optimal covers contains only biconnected subgraphs the multiplicity is expected to be comparatively low which also implies that in this case optimal covers can be expected to contain almost the same subgraphs.

Chapter 5

The Optimal Subgraph Cover Problem

In this chapter we present some algorithms for finding optimal covers in order to prove the practical value of our approach. We first examine computational complexity of the problem. For this, we briefly review some related classical problems. The problem of finding optimal covers turns out to be NP-hard therefore we propose a greedy algorithm.

5.1 Related Problems and Algorithms

In this section we briefly review some classical problems that are directly related to Σ-optimal subgraph cover problem. A general introduction to the subject of NP-completeness can be found in [74] and [75]. We also examine some widely used algorithms for these problems some of which we use as subroutines later.

5.1.1 The set cover problem

In its most general form, set covering problems can be formulated as follows. Let $U = \{s_1, s_2, ..., s_n\}$ be a set, $\mathcal{S} = \{S_1, S_2, ...S_M\}$ a collection of subsets of U such that $\bigcup_{s \in \mathcal{S}} S = U$ and $F : 2^{\mathcal{S}} \rightarrow \mathbb{R}^+$ a cost function. Then the corresponding set covering problem is to find some $C \in 2^{\mathcal{S}}$ that minimizes $F(C)$ under the constraint $\bigcup_{S \in C} S = U$.

In the linear version of the problem $F(C) = \sum_{S \in C} c(S)$, where $c(S)$ is the cost of S. When all the subsets have cost 1, the problem reduces to finding a cover of minimum cardinality. This problem is one of Karp's 21 NP-complete problems. In another widely studied version of the problem is $c(S) = |S|$, which is equivalent to finding a cover with a minimal number of intersections.

A commonly used heuristic for the linear set cover problem is the greedy algorithm [76]. In the greedy algorithm a cover is constructed stepwise by picking subsets based on their cost per uncovered element. The greedy algorithm has a worst case approximation ratio of $H(n) = \sum_{i=1}^{n} \frac{1}{i} \leq lnn + 1$, where $n = |U|$.

The set cover problem has also been extensively studied with respect to its approximability. Feige proved that unless NP has quasi-polynomial algorithms set cover can not be approximated within a factor of (1-o(1))ln(n) [77] and Alon et al. proved that similar results under the assumption $P \neq NP$ [78]. These show that for the linear problem the greedy algorithm is almost optimal in polynomial time.

The Σ-optimal subgraph cover problem is non linear set cover problem with, $U = E(G)$, $\mathcal{S} = \{$E(S): S is a subgraph of G$\}$ and cost function $\Sigma(C)$.

5.1.2 The graph isomorphism and automorphism problems

The graph isomorphism problem is the problem of determining whether two graphs are isomorphic. The graph isomorphism problem together with integer factorization is one of only two problems for which the computational complexity remains unknown. In other words, the problem is neither known to be NP-complete nor is there a known polynomial time algorithm for it.

Another problem that is closely related to the graph isomorphism problem is the problem of computing the automorphism group of a given graph. The graph automorphism problem is at least as difficult as the graph isomorphism problem since the later can be reduced to the former.

Although no polynomial time algorithm is known for the graph automorphism problem, fortunately there exist several algorithms [79, 80, 81, 82] that can efficiently compute automorphisms of graphs some of which are available as software packages. In our implementation we use NAUTY developed by B.D. McKay [79].

5.1.3 Generating all motifs of size n

The number of motifs of size n is bounded from below by $\frac{2^{\frac{n(n-1)}{2}}}{n!}$ in the undirected and by $\frac{2^{n(n-1)}}{n!}$ in the directed case. In other words, the number of motifs of size n grows faster than exponential with n. For instance in the undirected case there are 11716571 connected motifs on only 10 vertices. Thus, even if one had an efficient way of finding isomorphism classes together with their automorphism groups, computing and storing all possible motifs

together with their respective automorphism groups becomes prohibitive for large n.

The directg and geng routines included in the NAUTY [79] can be used to generate motifs with various properties.

5.1.4 The subgraph isomorphism problem

The problem of finding whether a graph G has a subgraph that is isomorphic to some other graph H is called the subgraph isomorphism problem. The subgraph isomorphism problem can be reduced to the maximum clique problem which is NP-complete [83].

There exist several exact algorithms that can compute subgraph isomorphisms rather efficiently. Some widely used algorithms are [82] and [84].

5.1.5 The maximum independent set problem

Given a graph $G = (E, V)$, an independent vertex set is a subset of vertices of which no two elements are adjacent. An independent set is called maximal if it is not a subset of any other independent set and is called maximum if it has maximum cardinality among such sets. Finding a maximum independent set is NP-hard [83].

The maximum independent set problem is one of the classical NP-complete problems and there exist several approximation algorithms. One of the simplest and most widely used heuristics is the minimum degree greedy heuristic which is based on the stepwise construction of an independent set where at each step one adds the vertex with smallest degree to the set and then removes

all vertices connected to it from the graph together with the corresponding edges until no vertices are left. There are several known approximation ratios for the greedy algorithm: $(d_{max}+2)/3$ in terms of the maximum degree and $(2\bar{d}+3)/5$ in terms of the average degree $\bar{d}$ [85]. Another widely used approximation algorithm based on excluding cliques from the graph is due to Boppana and Halldorsson [86].

Although the maximum independent vertex set problem might not seem to be directly connected to the Σ-optimal cover problem it appears as a subroutine of our approximation algorithm.

5.2 The set of candidate motifs

In general, it might be desirable to use the most general set of potential motifs when finding optimal subgraph covers. However, as discussed above the number of motifs grows faster than exponential with size and finding subgraph instances of motifs is also computationally expensive. Therefore, in practical applications, one is forced to restrict the set of motifs of which the instances are to be included in the analysis. We will call these candidate motifs. In general, several factors have to be taken into consideration when determining the set of candidate motifs:

- Computation time: this is determined by the size of the network, the computational resources (including time) available and the algorithms used to perform the analysis.

- Goal of the analysis: although our primary goal is discovering network

motifs, the method can also be used to obtain efficient decompositions of the network into special classes of motifs.

- Prior knowledge of the structure of the network: if one has any prior knowledge about the local structure of the network, one might be able to make an educated guess about the general form of subgraph that occur frequently in the network. Similarly, one might know that certain types of motifs simply do not occur in the network and thus exclude these from the candidate set.

Some potential candidate motif sets include:

- Connected motifs up to size n: Such a set of candidate motifs is most suitable when the primary goal of the analysis is discovering motifs.

- Biconnected motifs up to size n: The goal of the analysis is to determine a role sequence to be used in the generalized configuration model.

- Special classes (complete graphs, complete bipartite graphs, cycles and other highly symmetric motifs) of motifs of potentially unrestricted size: here the goal might be to obtain an efficient decomposition of the network into such special classes of motifs or data compression.

- Motifs with known dynamical properties: if a certain class of motifs is known to have certain dynamical properties that are thought to contribute to the function of the network, one can consider such motifs and generalizations of them in the set of candidate motifs. Decomposing the network into such motifs might further facilitate the analysis of dynamical processed defined on the network [87].

- Network specific motifs: Certain networks have a natural underlying subgraph cover because of the way they are constructed. For instance, collaboration networks are constructed by connecting all nodes (scientists, board members, actors) taking part in a certain collaboration (scientific publications, executive boards, movies) and therefore have a natural underlying clique cover. Chemical reaction networks also have an underlying subgraph cover of which the members correspond to chemical reactions and thus including the motifs corresponding to various reaction types into the candidate set might be good strategy. Similarly for electronic circuits, motifs corresponding to various known subcomponents might be good candidates. If one wants to find higher order motifs, the candidate set can further expanded to include various intersection patterns of such network specific motifs.

- Symmetric motifs: one can also restrict motifs with respect to the size of their automorphism group. This significantly reduces the number of motifs for large n since most large motifs have trivial automorphism groups [66].

In general classes of special motifs can also be expanded by including motifs that differ only slightly from these motifs i.e. motifs that differ from these motifs only by a few edges. One can also consider motif sets that are combinations of the sets described above. For instance, including stars into the set of candidate motifs will in general result in covers that better reflect the degree distribution of network. Disconnected motifs can be excluded from the analysis since it can be shown that the cover that independently

contains the connected components of such subgraphs always has lower total information.

5.3 Finding subgraphs

Finding a Σ-optimal cover by definition involves finding subgraph instances of various motifs. For this, one can follow one of these two approaches: the motif centric approach or the network centric approach.

The motif centric approach can be summarized as finding subgraph instances for each motif separately [53]. For this one simply runs a subgraph isomorphism algorithm [53, 84, 82] for each individual motif. This is the approach we will follow in our implementation.

In the network centric approach one finds all connected n-node subgraphs of G using a single algorithm. In general such algorithms first find all connected n-node subsets of G and then sort the subgraphs occurring on these sets using an isomorphism algorithm. Most motif analysis algorithms use network centric approaches [5, 50, 51, 52] to enumerate subgraphs. However, these algorithms in general focus on finding induced subgraphs. Since we also consider subgraphs that are not induced, these need algorithms would have to be modified accordingly.

In applications, both approaches have their advantages and dis-advantages. For instance, the network centric approach might find a large number of subgraphs that in the end not included in the analysis if the set of candidate motifs does not include all connected motifs. On the other hand, in the motif centric approach, one might spend a lot of time on motifs that do not appear

in the network at all.

5.4 Practical definitions of the effective complexity

Another issue that has to be addressed in practice is that the algorithmic information content is not computable and is computer dependent. As a result in practical applications one has to work with approximations in the form of upper bounds which can be obtained using efficient codes:

$$\epsilon(M, n_m) \simeq l_C(M, n, m) + c_U(C),$$

where $c_U(C)$ is the length of the program that describes the code C. Thus, in our case the computer dependence can be reduced to the constant $c_U(C)$. Although this constant determines the numerical value of the practical effective complexity, it is the same for all (M, n_m) and therefore the optimization problem is essentially the same for all choices of U. Consequently, we will omit such constant terms from now on.

Another important simplification we make is to assume that motifs are independent which results in an effective complexity term that is additive in motifs. We will use the log^* code for integers. Thus we have:

$$\Sigma(C) = \sum_{m \in M(C)} (S_N(m, n_m) + \epsilon(m) + log^* n_m) + log^* N, \qquad (5.1)$$

where $\epsilon(m)$ is the practical effective complexity given by the length of the

code used for motifs. Thus in the case of edge list code:

$$\epsilon(m) = \log^* |m| + \log^* e(m) + S(|m|, e(m)),$$

where $S(|m|, e(m))$ is the entropy of the ensemble of all graphs with the same node and edge counts as m. Another alternative is the log^* code for motifs described in Sec. 3.1.6. These codes are in a certain sense universal codes for motifs thus in the context of unrestricted motif sets they seem to be a natural choice. However, for instance when the set of candidate motifs is restricted to special classes of motifs (cliques, stars, cycles etc...) more suitable codes can be found since these have obvious better/shorter encodings than their edge list. A universally applicable strategy for obtaining such codes is to label motifs with integers starting from 1 and then to set:

$$\epsilon(m) \simeq log^* n(m),$$

where $n(m)$ is the integer label of m.

Although, the codes we presented can be considered as reasonably efficient, it might be argued that the choice of code used to approximate effective complexity is subjective. However, all alternative approaches to inductive inference involve similar subjective choices in practice [62, 61, 59]. The main goal of this thesis is not to advocate a specific approach to inductive inference but rather to show that subgraph covers can be used as a basis for motif analysis. The reason why we chose the total information approach is that it accounts for the fact that the parameters of the ensembles are graphs/motifs in an intuitive way.

5.4.1 Alternative approaches and interpretations

From a Bayesian viewpoint, the effective complexity term can be associated to a prior distribution using the correspondence between codes and probability distributions. From viewpoint of Bayesian statistics, the subjectivity involved in choice of code is less of a problem. In case where the set candidate motifs is infinite, the log^* code corresponds to a universal prior [65]. However, it might be argued that other priors are more suitable, for instance when one has prior knowledge about the structure of the network. In general, any prior knowledge of the properties of the network limits the set of motifs that can occur in the network. For instance, the size of the network and/or its maximal degree will in general limit the set of motifs that can occur in the network. Similarly, if one for instance knows that the network is a gene regulatory network and that certain motifs are likely to correspond to functional subunits, assigning such motifs higher prior probability might be justified. Thus, one can also look at the determination of the set of candidate motifs from a Bayesian point of view, since excluding some motifs from the candidate set is essentially equivalent to giving them zero prior. From this point of view we are faced with an interesting situation where the prior distribution is not only determined by prior beliefs/knowledge but also by our expectation for the time required to do the analysis, which in turn is determined by computational resources at our disposal and the algorithms we choose.

Another, more practical, way of looking at the effective complexity is as a safeguard against overfitting. From this point of view, the effective com-

plexity of a motif corresponds to the minimal entropy gain it has to provide in order to be included in the optimal cover. This, however, is an oversimplification, since in general the entropy gain of a motif also depends on other motifs. This, in turn, can be seen as setting a frequency threshold for the motif. Thus from a more practical point of view the problem can be formulated as a problem of entropy minimization under frequency constraints. In order to clarify the connection between the effective complexity of motifs and frequency threshold, we consider a simple example. Let m be a motif and G be a graph with $|E| = O(N)$, now we would like to find lowest minimum number n_{min} of disjoint copies of m such that:

$$\Sigma(E, 0) - \Sigma(E - n_{min}e_m, n_{min}) > 0.$$

For small n we have:

$$\begin{aligned}\Sigma(E,0)-\Sigma(E-ne_m,n) &= n(e_m-|m|)logN + log(n!) \\ &+ n(log|Aut(m)| + log(e)(e_m-1) - e_m(log(e_G)+1)) \\ &- \epsilon(m) - log^*(n) + O(1/N),\end{aligned}$$

where $e_G = E/N$ and e_m are the number of edges of G and m, respectively. This shows that if $e_m > |m|$, n_{min} is $O(1)$ and converges to 1 for large N. If $e_m = |m|$, n_{min} is $O(1)$ and larger than 1 even for large N, in general. The case where m is a tree i.e. $e_m = |m| - 1$ is more involved. However, the thresholds can be shown to be always of $O(N)$. Note that, the thresholds are in close correspondence with the expected number of copies of motifs Erdös-

Rènyi random graph and thus can be seen as reflecting natural expectations about frequencies of motifs. However, in this example we considered only covers that consisted of the motif and the single edge. In a more general setting one would also have to take into account the covers that contain the submotifs of m.

From the point of view of overfitting, one is not constrained to use effective complexity terms that correspond to code lengths. Thus if overfitting is less of a concern, one can even set the effective complexity term to zero which in our case is equivalent to the maximum likelihood approach. Another alternative is to consider cost functions of the form:

$$S(C) + \alpha\epsilon,$$

where $0 \leq \alpha \leq 1$ and compare motifs obtained for various values of α. In situations where overfitting is less of a concerns and/or if one wants to find a maximal number of potentially relevant motifs the using reduced effective complexity terms might be justified. Using reduced effective complexity terms might be especially useful for small networks ($N < 200$). Since, by definition, such networks can only contain a small/limited number of sparsely intersecting copies of each motif, the entropy gain motifs can provide is also limited. Consequently, for small networks the thresholds set by the effective complexity might be too stringent and impede the discovery of motifs.

The algorithms we shall present in the next section apply to all choices of effective complexity type terms provided that they are additive in the motifs. The algorithms can also be modified in a straightforward manner to

incorporate frequency thresholds.

5.5 The greedy heuristic

Unlike in the linear cover problem where each subset has an individual cost, in the Σ-optimal cover problem each subgraph can not be assigned an individual efficiency because Σ() is nonlinear. Therefore, making use of the fact that Σ is additive in the motifs, we base our greedy algorithm on the efficiency of motifs instead. Given a partial cover C, the efficiency of a set S_m of m-subgraphs is defined as:

$$\sigma(S_m, C) = \frac{\Sigma(S_m)}{|E(S_m) - E(C)|}, \tag{5.2}$$

where $E(C)$ and $E(S_m)$ are the set of edges covered by C and S_m respectively and $\Sigma(S_m)$ is the total information corresponding to S_m. More precisely:

$$\Sigma(S_m) = S(m, |S_m|) + \epsilon(m) + log^*(|S_m|).$$

Following this definition, an optimal instance set of m is defined as a set of m-subgraphs that minimizes σ. At each step, the algorithm determines the efficiency of all motifs in the candidate motif set by finding an optimal instance set for m in the set of candidate motifs. In the next step, the algorithm checks for each motif whether including its optimal instance set into the cover decreases the overall total information of the cover. Here, the total information of partial covers is calculated by adding to them the single edge subgraphs corresponding to uncovered edges. At this point we should

mention that motifs can not be selected based only on their efficiency because adding the optimal instance set of a motif to the cover in general decreases the efficiency of other motifs which, sometimes might lead to an increase of the overall total information. In the next step the algorithm picks the motif which is most efficient among the motifs of which the optimal instance set does not increase the total information. Once this motif is determined its optimal instance set is added to the cover. Then the set of covered edges is updated and the process is repeated until all edges of the graph are covered. To ensure that the algorithm terminates, we require the single edge motif to always be included in the set of candidate motifs.

Algorithm 1 GreedyOptimalCover (G(E,V),MS)

$CoveredEdges = \emptyset, Cover = \emptyset, Motifs = \emptyset$

while $|CoveredEdges| < |E|$ **do**

$C, m = FINDMOTIF(G, MS, CoveredEdges)$

$CoveredEdges \leftarrow CoveredEdges \cup_{i \in C} e(i)$

$Cover \leftarrow Cover \cup C$

$Motifs \leftarrow Motifs \cup \{m\}$

end while

return Cover, Motifs

function FINDMOTIF(G,MS,CoveredEdges)

for $m \in MS$ **do**

C(m)=OptimalInstanceSet(m,CoveredEdges,G(E,V))

end for

M=$argmin_{m \in MS}\{\sigma(C(m), CoveredEdges) | \Sigma(Cover \cup C(m)) \leq \Sigma(Cover)\}$

return C(M),M

end function

Here, OptimalInstanceSet is a function that finds an optimal instance set given a motif and a set of covered edges and MS is the set of candidate motifs.

Given a motif m and a set of covered edges, finding an optimal instance set is a nontrivial optimization problem on its own. For instance, if subgraphs in the cover are not allowed to share edges, finding an optimal instance set is equivalent to finding a set of m-subgraphs of maximum cardinality such

that no two of the subgraphs in the set have an edge in common. This problem is equivalent to the maximum independent vertex set problem which is NP-complete [74, 83]. Therefore in most practical situations, some type of heuristic has to be used. In the following section, we present two such heuristics. Depending on the heuristic, finding an optimal instance set might require some or all motif subgraphs to be computed. In our implementation, we will follow a motif centric approach that uses subgraph isomorphism algorithms [82, 84].

5.5.1 Maximum independent set heuristic

Let us first consider a very simple greedy heuristic for constructing an optimal instance set. Starting from the empty set, at each step we pick the m-subgraph that contains the maximal number of uncovered edges. Note that, this is the m-subgraph that gives us the maximal reduction in $\sigma(S)$. This is then repeated until there are no more m-subgraphs that decrease $\sigma(S)$. The maximum independent set heuristic improves this by also maximizing the number efficient motifs that are added to the set.

As the name suggests, the maximum independent set heuristic is based on finding maximum independent vertex sets of various intersection graphs of the subgraph instances of m. The intersection graph of m-subgraphs containing n covered edges is defined as follows: the vertices of this graph are all the m-subgraphs of G that contain n covered edges and there is an edge between two subgraphs whenever they have at least one uncovered edge in common. Finding a maximum independent set is known to be NP-complete

therefore in most instances a heuristic has to be used.

When constructing an optimal instance set, we start with $n = 0$, since these are the m-subgraphs that are most efficient in covering edges. Then a maximum independent vertex set of the corresponding intersection graph is found and the subgraphs in this set are added to the optimal instance set one at a time provided they decrease σ. When all the m-subgraphs with intersection number n are exhausted, the set of covered edges is updated and the procedure is repeated for $n + 1$. The algorithm terminates when there are no m-subgraphs left that decrease σ.

Algorithm 2 Maximum-IS heuristic for OptimalInstanceSet(G,m,CoveredEdges)

```
mSet = SubGraphInstances(G, m), OIS(m) = ∅
for n:=0 until e(m)-1 do
    IG = IntersectionGraph(mSet, CoveredEdges ∪ Edges(OIS(m)), n)
    MIS=MaximumIndependentSet(IG)            ▷ Maximum-IS heuristic
    while MIS ≠ ∅ do
        s=random pick from MIS
        if σ(m, OIS(m)) ≤ σ(m, OIS(m) ∪ {s}) then
            OIS(m) ← OIS(m) ∪ {s}
            MIS ← MIS − {s}
        else
            end while, end for
        end if
    end while
end for
return OIS(m)
function IntersectionGraph(mSet,CoveredEdges,n)
    V = {m ∈ mSet : |Edges(m) ∩ CoveredEdges| = n}
    E = {{m, m'} : m, m' ∈ mSet and Edges(m) ∩ Edges(m') − CoveredEdges ≠ ∅}
    return G(E,V)
end function
```

In the code above MaximumIndependentSet is a heuristic for finding the

maximum independent sets [85, 86] and SubGraphInstances(G,m) [84, 82] is a function that finds and returns all instances of m in G.

Maximal independent set heuristic

Constructing an intersection graphs at each step of the maximum independent set heuristic in general demands a lot of computational resources since some subgraphs might occur in quite large numbers. Moreover, such subgraphs also tend to intersect quite heavily and thus their intersection graphs can occupy a lot of memory. To overcome this, we introduce a lighter/faster version of the above algorithm which uses maximal independent sets instead of maximum independent sets. Maximal independent vertex sets are independent sets that are not subsets of any other independent set. Finding a maximal independent set is much easier that finding a maximum independent set. One can easily obtain a maximal independent set of m-subgraphs by stepwise picking an instance of m, removing the edges of this subgraph from the graph and then repeating the procedure until the graph contains no more copies of m. The maximal independent set of subgraphs is then used as a candidate for the optimal instance set. Another advantage of this heuristic is that it allows for subgraphs to be detected on the fly and does not require all m-subgraph of the network to be computed.

Using maximal independent sets instead of maximum independent sets introduces more variability in terms of the cover obtained by the greedy heuristic. The maximal independent set heuristic always produces covers of which the subgraphs do not share edges. However, the heuristic can be easily be modified to allow such intersections between subgraphs. This algorithm

Algorithm 3 Maximal-IS heuristic for OptimalInstanceSet(G,m,CoveredEdges)

```
OIS(m) = ∅
remove CoveredEdges from G
while Subgraph(G, m) ≠ ∅ do
    s = Subgraph(G, m)                    ▷ Find an m-subgraph of G
    OIS(m) ← OIS(m) ∪ {s}
    remove edges in s from G
end while
return OIS(m)
```

is equivalent to the simple greedy heuristic mentioned in the first paragraph and corresponds to approximating the maximum independent sets by maximal ones in the first algorithm. However in experiments, including such intersecting subgraphs did not result in covers with significantly lower total information.

An important feature of the maximal independent set heuristic is that it combines the determination of optimal instance sets and detection of subgraphs. Consequently, one does need to compute all subgraph instances of m in advance which significantly reduces its running time and memory requirements. This makes it much more suitable for larger networks and motifs when computational resources are limited.

5.5.2 Discussion

Due to its probabilistic nature, the greedy heuristic might find different covers for the same networks on different runs. When using the greedy heuristic, this variability essentially comes from the heuristic used to obtain optimal instance sets. More specifically, in the case of the maximum independent set

heuristic, the source of this variability is the heuristic used to approximate maximum independent sets. In general one expects that heuristics which are able to find better solutions (that is larger independent sets) to also have less variability. The maximal independent set heuristic can be seen as the crudest way to approximate maximum independent sets and as a result one also expects it to have the largest variability. In general the variability of the optimal instance sets obtained by the different heuristics also strongly depends on the network. Depending on whether the greedy algorithm is able to produce a stable solution or not, one can opt for more sophisticated algorithms to approximate maximum independent sets. However, as exemplified by the maximum and maximal independent set heuristics, this in general might involve significant trade-offs in terms of computational complexity. On the other hand, one can also devise heuristics that do not rely on independent sets for finding optimal instance sets.

For the networks we considered, we observed that the results of greedy heuristics are quite stable over runs even when the maximal independent set heuristic is used. Although for some networks the motif sets obtained on different runs differ, these are mostly restricted to motifs that only occur a few times in the cover or are one node intersections of smaller motifs. For instance, one cover might contain triangles and the other subgraphs that are made of two triangles connected at one node.

As is the case with any heuristic, the success of the greedy heuristic depends on the structure of the network and in certain situations the greedy heuristic might get stuck in a local minima. For instance, if a motif contains a sub-motif that is more dense and symmetric compared to the entire

motif, the greedy algorithm will choose the sub-motif over the motif even if the inclusion of the larger motif might result in a cover with smaller total information since the submotif covers edges more efficiently. In principle the greedy heuristic could be modified to avoid at least some of its local minima by picking motifs not only based on their efficiency but also the overall gain in total information. On the other hand, one can also apply other widely used approximation approaches such as simulated annealing or genetic algorithms to the problem [88, 89, 90]. Although we don't expect there to be one algorithm that outperforms all the others for every network, given a network one could solve the problem using every algorithm that has acceptable running time and pick the solution with minimal total information. Therefore we consider the development of alternative algorithms to be an important topic for further research.

Chapter 6

Empirical Results

In this chapter we present empirical results obtained for several real world networks using the greedy heuristic [71]. We also consider some synthetic networks that are realizations of uniform subgraph covers in order to test whether the heuristic can recover the motif structure from the graph corresponding to these. Due to restricted computational resources, the size of the subgraphs used in the analysis is limited to 5 in the directed and to 6 in the undirected case. We also consider biconnected subgraph covers for some of the networks. All presented results were obtained using the maximal independent set heuristic for finding optimal instance sets and practical effective complexities corresponding to the edge list encoding. In the tables N and E stand for the number of vertices and edges respectively. Σ_e stands for the total information of the corresponding edge cover and Σ for the total information of the obtained subgraph cover, both quantities are rounded to the closest integer and are given in bits.

For some of the networks we also present empirical results concerning

maximum likelihood/minimum entropy covers. These can be regarded as giving the maximal number of potentially relevant motifs for these networks.

Due to its probabilistic nature, the greedy heuristic might find different covers for the same networks on different runs. For each network the cover with smallest total information obtained over 10 runs is given. For the power grid network we also include the range of the motif counts obtained over 10 runs in parenthesis.

6.1 Real world networks

6.1.1 The power grid of the western United States

Network	**N**	**E**	Σ_e	Σ		
Power Grid	4941	6594	81084	77109		
						
4109	141	112	44	31	17	47
(4109-4129)	(138-145)	(111-122)	(44-45)	(30-31)	(15-17)	(45-47)
						
11	68	2	15	42	2	
(10-11)	(67-68)	(2-2)	(15-16)	(41-43)	(2-2)	

Table 6.1: The motifs of the network representing the Western States Power Grid of the United States found using connected subgraphs up to size 6. The motif counts correspond to the cover with lowest total information obtained over 10 runs. The range of the motif counts obtained are also shown in parenthesis.

Table 6.1 shows the motifs contained in the optimal cover of the network representing the Western State Power Grid of the United States [23]. All motifs except the motif consisting of two triangles connected by a single vertex are biconnected. Therefore when the candidate set is restricted to be biconnected motifs the optimal cover contains these triangles individually. The table further shows the covers obtained on different runs all contained

the same motifs and also did not differ significantly with respect to their motif counts.

6.1.2 Gene regulatory networks

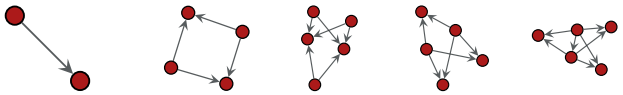
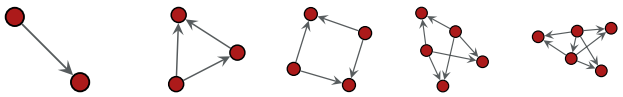
Network	**N**	**E**	Σ_e	Σ
S.Cerevisiae	688	1079	11024	9811
				
547	23	4	60	8
E.Coli	423	519	5124	4810
				
323	9	14	13	5

Table 6.2: The motifs of the transcription networks of E.coli and S.cerevisiae obtained using all biconnected motifs up to size 5.

Network	**N**	**E**	Σ_e	Σ			
S.Cerevisiae	688	1079	11024	9309			
							
59	26	16	23	94	5	61	8
E.Coli	423	519	5124	4637			
							
130	12	51	13	4	5		

Table 6.3: The motifs of the transcription networks of E.coli and S.cerevisiae obtained using all connected motifs up to size 5.

Tables 6.2 and 6.3 show the motifs found for the transcription networks of E.Coli [91] and S.Cerevisiae [5]. These show that including singly connected motifs in the candidate motif set has almost no effect on the biconnected motifs and mostly results in star shaped motifs and/or motifs that consist of one vertex intersections of biconnected motifs. The two networks share 3 out of 4 motifs in the case of biconnected motifs.

For these networks the covers observed over the different runs did in some instances differ with respect to their motifs. For instance, in the case of biconnected motifs, some covers of the S.Cerevisiae network did not contains the motif consisting of two inward 3-stars (4/10) and the motif consisting of 3 feed-forward loops sharing an edge (3/10). Similarly, 3 out of the 10 covers of the E.Coli network contained 3 copies of the 4-node motif consisting of two feed forward loops sharing an edge in the biconnected case. In the case of

general motifs in addition to similar variations some covers of the S.Cerevisiae network also contained inward 2-stars (2/10) and 3-stars (1/10).

6.1.3 Electronic circuits

Network	**N**	**E**	Σ_e	Σ
s208	122	189	1460	1454
				
165	8			
s420	252	399	3491	3404
				
220	7	4	13	11
s838	512	819	7995	7652
				
456	15	8	25	23

Table 6.4: The motifs of electronic circuits (digital fractional multipliers) obtained using all connected motifs up to size 5.

Table 6.4 shows the results for three networks representing electronic circuits that are digital fractional multipliers [5]. In s208 we only find the 3-cycle motif-as we shall see in Sec.6.3 this is mainly due to s208 being relatively small. In the other two networks the algorithm not only finds the same motifs but the motif counts also scale almost exactly with network size. For

these networks the algorithm found covers with the same motifs and motif frequencies on all runs.

6.1.4 Metabolic networks

Table 6.5 shows the motifs found in metabolic networks [13] of several species from different domains of life: AA= Aquifex aeolicus(bacteria), AB= Actinobacillus actinomycetemcomitans (bacteria), EC= Escherichia coli (bacteria), CE= Caenorhabditis elegans (eukaryote), AG= Archaeoglobus fulgidus (archea), AP= Aeropyrum pernix(archea). The table only shows motifs that occur at least 4 times in any one of the covers. For each network at most 2 motifs are not shown in the table. Again, we not only find approximately the same motifs in these networks but the counts of common motifs also scale approximately with network size.

Network	**N**	**E**	Σ_e	Σ				
AA	1057	2527	25844	21255				
								
423	16	6	16	130	147	97	0	0
AB	993	2368	24012	19882				
								
408	22	4	23	128	131	82	0	0
EC	2275	5763	64842	52590				
								
935	117	5	40	264	345	202	5	0
CE	1173	2864	29634	24380				
								
478	13	3	31	137	178	100	0	0
AG	1268	3011	31616	25960				
								
509	23	6	26	140	168	120	0	4
AP	490	1163	10610	8856				
								
195	11	0	12	55	67	46	0	0

Table 6.5: The motifs found in metabolic networks of various species using biconnected motifs up to size 5. The table only shows motifs that occur at least 4 times in any one of the covers.

6.1.5 Autonomous systems networks

Motifs (id)															
	 (0)	 (1)	 (2)	 (3)	 (4)	 (5)	 (6)	 (7)							
	 (8)	 (9)	 (10)	 (11)	 (12)	 (13)	 (14)							
Network	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}
AS-1	710	31	0	55	7	495	227	34	7	41	0	0	17	5	9
AS-2	695	18	0	73	9	505	226	35	10	40	0	0	13	9	8
AS-3	791	18	0	85	14	507	243	39	10	44	10	0	15	7	9
AS-4	1016	28	0	124	5	650	356	55	17	48	13	21	24	7	11
AS-5	980	31	0	127	10	615	368	59	17	54	12	30	38	9	18
AS-6	1227	30	0	183	14	826	569	83	23	83	13	23	35	12	13
AS-7	3162	27	0	127	8	0	299	45	13	71	0	27	34	11	14
AS-8	913	38	144	189	18	981	682	87	30	98	14	30	47	18	14

Network	**N**	**E**	Σ_e	Σ
AS-1	3015	5156	57877	52769
AS-2	3042	5232	58755	53557
AS-3	3213	5624	63458	57784
AS-4	4235	7674	89266	80224
AS-5	3962	7931	90351	80411
AS-6	5599	10728	128250	113240
AS-7	3570	7033	79225	71227
AS-8	6474	12572	152686	134176

Table 6.6: Motifs found in networks representing the internet at the level of autonomous systems using all connected motifs up to size 5.

Table 6.6 shows the network motifs found in networks representing the internet at the level of autonomous systems [92]. As in the case of metabolic networks whenever a certain motif occurs in the optimal cover of more than one of these networks its counts also scale approximately with network size.

The analysis of various networks shows that networks representing similar systems also have the similar motif structure. This can be regarded as further evidence that motifs play an important role in the structural organization of complex networks. We also observe that motif counts scale approximately with the vertex and edge counts of the networks in the same type. This

also shows that the method can be used to categorize networks in a similar fashion to [57]. The results also indicate that subgraph covers can be used to obtain representations that are up to 20% shorter compared to their edge list encoding.

In principle the method can be further evaluated by comparing the networks with the generalized configuration models corresponding to the Σ-optimal subgraph covers. However, the models corresponding to the obtained covers are in general quite complex and therefore, analyzing them would require developing computer algebra systems and/or sampling algorithms for these models. As a result, such comparisons are beyond the scope of this thesis.

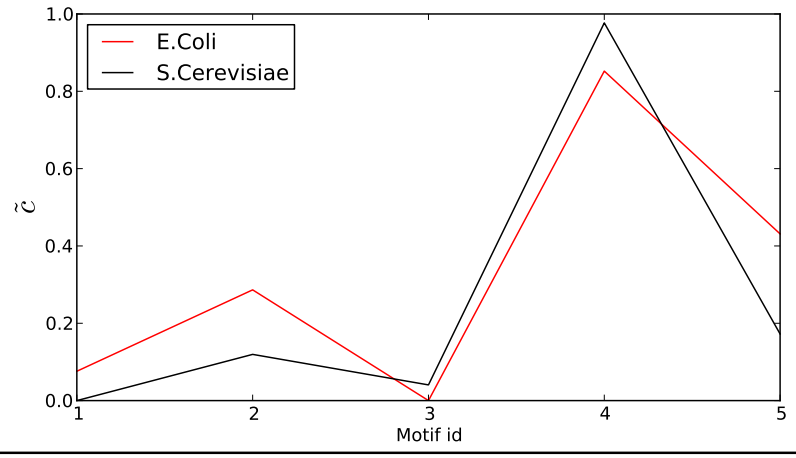
6.1.6 Motif significance profiles

The method can also be used to construct motif significance profiles based on the c-score. These are analogues of the significance profiles based on the z-score used by Milo et al. in [57] and can be used to classify networks according to their motif structure. We use motif significance profiles that are given by the normalized c-score:

$$\tilde{c}_m = \frac{c_m}{\sqrt{\sum_{m' \in M(C_\Sigma)} c_{m'}^2}}. \tag{6.1}$$

The tables presented below show the motif significance profiles of various networks corresponding to the covers presented in previous sections. The tables contain only the regions of the significance profiles for which the c-score is non-zero. Compared to the full significance profile these regions are

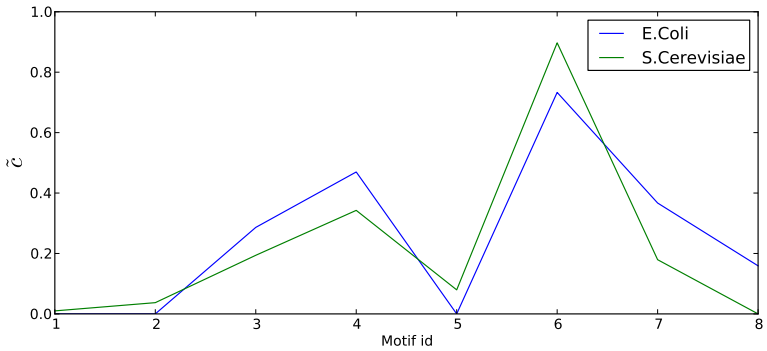
comparatively small since in the directed case there are 9578 connected and 7585 biconnected motifs up to size 5. Similarly, in the undirected case there are 30 connected motifs up to size 5. Since the triad significance and subgraph ratio profiles used in [57] have only 6 and 4 degrees of freedom [93, 57], respectively, the motif significance profiles based on the c-score provide a much finer grained classification compared to [57].



Motifs (id)

(1) (2) (3) (4) (5)

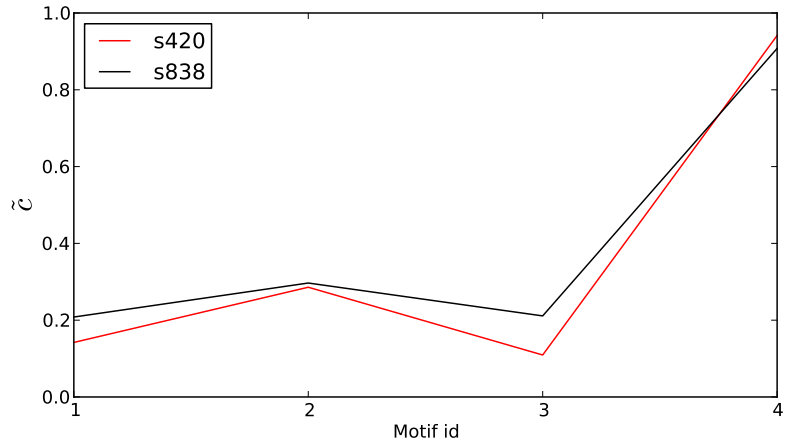
Table 6.7: The motif significance profiles corresponding to the covers given in Table 6.2.



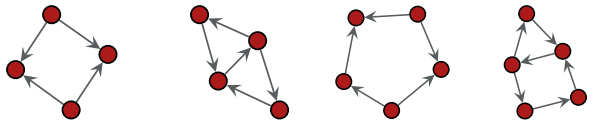
Motifs (id)							
							
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

Table 6.8: The motif significance profiles corresponding to the covers given in Table 6.3.

Tables 6.7 and 6.8 show the significance profiles corresponding to the optimal covers obtained using singly connected and biconnected subgraphs up to size 5, respectively. In both cases the significance profiles of both networks are in broad agreement.



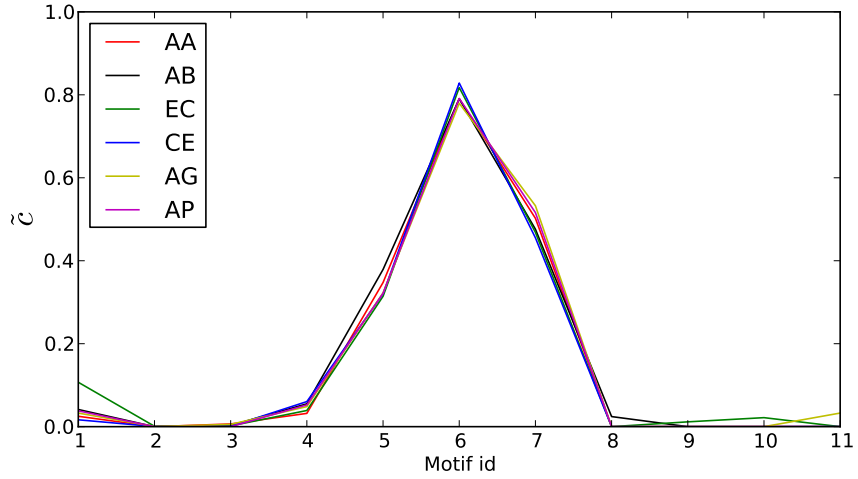
Motifs (id)



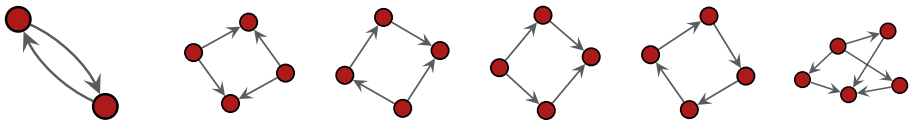
(1) (2) (3) (4)

Table 6.9: The motif significance profiles corresponding to the covers obtained for the electronics circuits s420 and s838 given in Table 6.4.

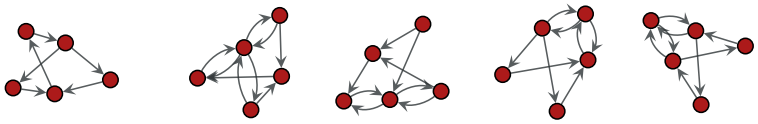
In Table 6.9 the significance profiles of electronic circuits s420 and s838 are shown. The significance profiles are so coincide almost exactly.



Motifs (id)



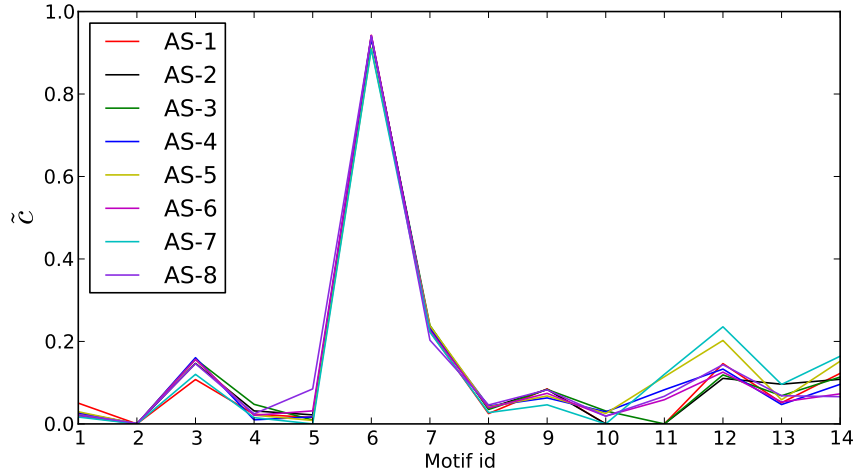
(1) (2) (3) (4) (5) (6)



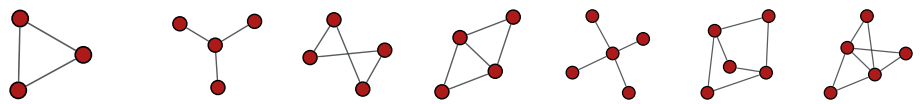
(7) (8) (9) (10) (11)

Table 6.10: The motif significance profiles corresponding to the covers given in Table 6.5. The profiles include the motifs not shown in Table 6.5.

Table 6.10 shows the significance profiles of the metabolic networks corresponding to various species. Again, we find that the significance profiles of these networks match extremely well.



Motifs (id)



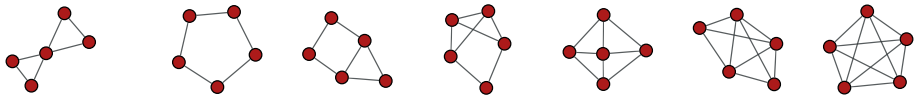


Table 6.11: The motif significance profiles corresponding to the covers given in Table 6.6.

Table 6.11 shows the significance profiles of the metabolic networks corresponding to various autonomous systems networks. Although there are slight differences between the profiles for denser motifs, the overall agreement of the profiles is quite well.

The significance profiles of various network types given in the tables above show that networks of the same type also have very similar significance profiles. This demonstrates that motif significance profiles based on the c-score

provide an effective, fine grained measure for classifying/ categorizing networks.

6.1.7 A comparison with the method of Milo. et al

In commonly analyzed networks we find 3 and 4 node motifs that are almost identical to those found by Milo et al. in [5]. Here, we only compare 3 and 4 node motifs for relatively small networks since for larger motifs and networks conserving lower order motifs is not computationally feasible.

In principle for larger motifs one can use the configuration model as a null model instead but in this case the method of Milo et al. will identify most subgraphs that contain a smaller overrepresented motif as network motifs.

In the transcription networks we find all the motifs found by Milo et al. though in the S.Cerevisiae network the feed forward loop (FFL) only appears as a submotif of the larger motif that consists of 3 FFLs sharing an edge. Similarly, for the electronic circuit s420 and s838 networks we find the same 3 and 4 node motifs though the 3-and 4-cycles appear only as submotifs. Moreover, the optimal subgraph covers show that in these networks 3- and 4-cycles occur almost exclusively as subgraphs of larger motifs (3-cycles: s420-19/20, s822-39/40; 4-cycles: s420-11/11, s838-23/23). For s208 we only find the 3-cycle motif in the cover that minimizes the total information. On the other hand, the maximum likelihood cover (See Sec.6.3 Table 6.15) of s208 contains the same motifs we found in the other two electronic circuit. Moreover, the maximum likelihood cover shows that in s208 almost all copies 3 and 4 cycles occur as subgraphs of some larger motifs.

6.2 Synthetic Networks

In this section we consider some synthetic networks corresponding to uniform subgraphs in order to test the performance of the greedy heuristic in recovering the underlying covers and motifs.

Network	**N**	**E**	Σ_e	Σ	
Network 1	1500	3115	34069	20566(20566)	
					
150(150)	80(80)	125(125)	75(75)	125(125)	75(75)
Network 2	512	819	7795	7646(7652)	
					
452(456)	16(15)	8(8)	25(25)	23(23)	
Network 3	750	2065	17594	16296(16089)	
					
273(150)	124(125)	119(125)	45(45)	47(45)	78(100)
Network 4	1500	2065	21746	18413(18277)	
					
192(150)	123(125)	120(125)	44(45)	47(45)	95(100)

Table 6.12: The motifs obtained for several networks corresponding to uniform subgraph cover ensembles. The quantities corresponding to these ensembles are given in parenthesis.

As seen in Table 6.12, the algorithm is able to recover the motif sets of the underlying cover for all random networks. For Network 1 the algorithm recovers the underlying cover exactly. For Network 2, which replicates the motifs found in the electronic circuit s838 (Table 6.4), the cover found by the algorithm differs from the underlying cover only with respect to one subgraph. On the other hand, for Networks 3 and 4 the motif counts differ significantly from the counts of the uniform subgraph covers used to generate the networks, especially with respect to the 5-star counts. As discussed previously this is caused by the fact that these networks contain a large number of 5-stars of which not all are explicitly contained in the underlying cover. As a result for 5-stars the determination of an optimal instance set becomes more difficult. This effect is more pronounced in Network 3 because Network 4 has a higher edge density which results in more 5-star subgraphs.

6.3 Maximum likelihood covers

In this section we present results regarding maximum likelihood covers i.e covers that minimize the entropy. These also show how the motifs found vary with respect to the effective complexity term. For each network we give the best cover obtained over 5 runs. Since maximum likelihood covers tend to contain motifs that occur only a few times in the cover, the covers obtained on different runs sometimes also differ with respect to such motifs. For the gene regulatory networks we observed 2 such motifs over 5 runs. While the covers for the power grid network only differed with respect their motif counts, for the electronic circuits the motifs and their frequencies were

the same on all runs.

6.3.1 Power Grid

Network	**N**	**E**	$S_e(G)$	$S(C)$			
Power Grid	4941	6594	81067	76488			
							
4050	207	94	38	21	9	5	66
							
4	1	2	2	1	2	11	4
							
3	41	16	3	1	1	1	1
							
5	1	2	3	1	1	1	2

Table 6.13: The motifs counts of the maximum likelihood cover of the Western States Power Grid of the United States found using all biconnected subgraphs up to size 6.

The above table shows that compared to the Σ-optimal cover the maximum likelihood cover contains extra motifs that mostly appear only a few times in the cover.

6.3.2 Gene regulatory networks

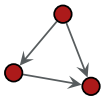
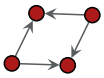
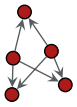
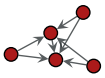
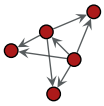
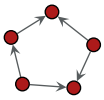
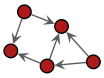
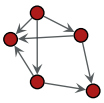
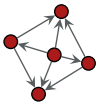
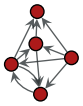
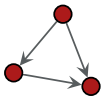
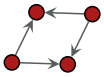
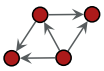
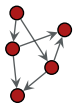
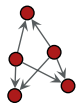
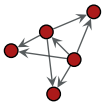
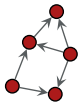
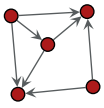
Network	**N**	**E**	S_e	$S(C)$	
S.Cerevisiae	688	1079	11010	9630	
					
518	2	19	4	62	1
					
5	2	1	1	1	1
E.Coli	423	519	5110	4709	
					
321	8	9	2	1	12
					
5	2	1			

Table 6.14: The motifs contained in the maximum likelihood cover of transcription networks of E.coli and S.cerevisiae obtained using all biconnected motifs up to size 5.

As with the power grid network the effect of setting the effective complexity term to zero results in additional motifs that occur at most twice in the cover. The extra motifs appearing in the maximum likelihood covers are different for the two networks except for the feed forward loop.

6.3.3 Electronic circuits

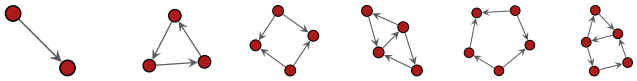
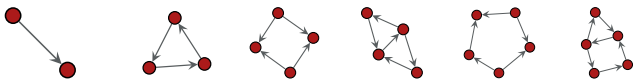
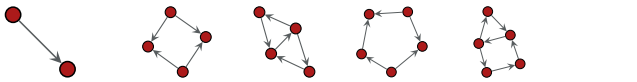
Network	**N**	**E**	S_e	$S(C)$	
s208	122	189	1392	1448	
					
104	1	3	2	6	5
s420	252	399	3478	3320	
					
222	1	7	4	12	11
s838	512	819	7995	7652	
					
456	15	8	25	23	

Table 6.15: Motifs appearing in the maximum likelihood covers of electronic circuits obtained using all connected motifs up to size 5.

Table 6.15 shows that the maximum likelihood covers for the s420 and s838 networks are almost identical to their Σ-optimal covers. On the other hand, for s208 the maximum likelihood cover contains the same motifs as s420 and s838. Moreover, the motif counts also scale almost exactly with respect to network size. This further supports using maximum likelihood or reduced effective complexity terms for small networks.

Chapter 7

Conclusion

7.1 Summary of the main contributions

We proposed an information theoretical approach to motif analysis in networks that is based on using subgraph covers as formal representations of graphs and total information of subgraphs covers as a measure of optimality. By considering motifs of different sizes simultaneously with respect to a single measure the method can detect even large motifs consistently.

An important feature of the presented method is that it provides an explicit and efficient decomposition of the network into motif subgraphs. This allows motifs to be studied in the context of the whole network rather than in isolation.

We also examined the relation between subgraph covers and several random graph models that can incorporate motifs. We showed that total information optimal subgraph covers can used to match networks with specific instances of these models. This effectively allows for more realistic network

models in general. These, models also can be used to study the relation between motifs and structural and dynamical properties of networks.

In order to prove the practical value of our approach we also studied the total information optimal subgraph cover problem from a perspective of computational complexity and proposed a greedy heuristic for the problem. The heuristic is able to recover the motif structure of synthetic networks and also produces consistent results for real world networks.

Empirical results for several real world networks were also presented. These show that networks with similar function not only have similar motif structures but also that motif counts scale approximately with the number of vertices. Consequently, the method provides a fine grained measure, in the form of motifs significance profiles, for classifying networks.

7.2 Directions for future research

7.2.1 The structure of optimal subgraph covers

In this thesis we mostly concentrated on finding optimal covers and their motif sets and did not study the structure of the optimal covers we obtained. Further insights might be gained by examining the structure of optimal covers in more detail. Properties that can be studied include preferred attachment patterns between motifs and the overall distribution of the motifs in the network. If for instance, a certain set of motifs appears on the same set of nodes and/or in specific combinations this can be seen as indicating a functional relation between these motifs. On the other hand, if the instances

of a certain motif are concentrated on a specific set of vertices, this set might correspond to a region of the network that is responsible of performing a specific function.

7.2.2 Generalization to colored networks

Subgraph covers and the total information can be generalized to graphs with colored/labeled vertices and edges in a straightforward manner. Such labels might correspond to different types of vertices in the network or membership in a network community or module. Including such additional information into the analysis might further facilitate the detection of motifs. Moreover, one would expect that the motifs a vertex participates in to be correlated with its type. For instance, if the vertex types in a network are related to functional roles or when the communities of the network differ with respect to their internal structure. The greedy algorithm can also be modified in a straightforward manner to the case of colored graphs.

On the other hand, if the types of vertices are not known a priori the Σ-optimal cover also be used as a starting point for inferring functional roles of vertices and/or network communities. For community detection, the inhomogeneous model of Bollobas et al. with discrete types could be used since these effectively generalize the widely used mixture models to subgraph covers.

7.2.3 Random graph models

Another important direction for future research would be to study the random graph models corresponding to the optimal covers. Such comparisons

would allow for further assessment of the method. Moreover, these models allow the relation between the motif structure of the network and more general/global properties to be studied. This, can further provide insights into the question why certain motifs appear in certain types of networks. Moreover, various dynamical systems that can be defined on the network can be used to study the interplay between the motif structure and dynamical properties of the network.

As is the case with many random graph models, the generalized configuration models and SCMs can only be solved analytically as the number of vertices goes to infinity. Thus, especially in the context of small networks, efficient algorithms for sampling such models are required. On the other hand, the models can also become quite complex especially as the size of motifs increases and therefore, implementing a computer algebra system for such models might be required in order to do calculations.

7.2.4 Heuristics

We consider the development of further heuristics an important topic for future research. While the greedy algorithm can be further modified/improved to avoid some local minima, other widely used approaches such as genetic algorithms and simulated annealing can also be applied to the problem. In the view of the problem's high computational complexity developing parallel algorithms is also of interest.

Bibliography

[1] B. Karrer and M.E.J Newman. Random graphs containing arbitrary distributions of subgraphs. *Physical Review E*, 82(6):066118, 2010.

[2] B. Bollobás, S. Janson, and O. Riordan. Sparse random graphs with clustering. *Random Structures & Algorithms*, 38(3):269–323, 2011.

[3] M.E.J. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[4] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[5] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824, 2002.

[6] Murray Gell-Mann and Seth Lloyd. Information measures, effective complexity, and total information. *Complexity*, 2(1):44–52, 1996.

[7] P Erdos-T Gallai. Graphs with prescribed degree of vertices (hungarian), mat. *Lapok*, 11:264–274, 1960.

[8] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[9] Mark Newman, Albert-László Barabási, and Duncan J Watts. *The structure and dynamics of networks*. Princeton University Press, 2006.

[10] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

[11] Tong Ihn Lee, Nicola J Rinaldi, François Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.

[12] Eric Davidson and Michael Levin. Gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(14):4935–4935, 2005.

[13] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[14] Soon-Hyung Yook, Zoltán N Oltvai, and Albert-László Barabási. Functional and topological characterization of protein interaction networks. *Proteomics*, 4(4):928–942, 2004.

[15] Martin Chalfie, JOHN E Sulston, JOHN G White, Eileen Southgate, J Nicol Thomson, and Sydney Brenner. The neural circuit for touch

sensitivity in caenorhabditis elegans. *The Journal of neuroscience*, 5(4):956–964, 1985.

[16] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[17] Stuart L Pimm. *Food webs*. Springer, 1982.

[18] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[19] David Knoke and Song Yang. *Social network analysis*, volume 154. Sage, 2008.

[20] S Lehmann, B Lautrup, and AD Jackson. Citation networks in high energy physics. *Physical Review E*, 68(2):026113, 2003.

[21] Loet Leydesdorff and Liwen Vaughan. Co-occurrence matrices and their applications in information science: Extending aca to the web environment. *Journal of the American Society for Information Science and Technology*, 57(12):1616–1628, 2006.

[22] Bernard Derrida, Susanna C Manrubia, and Damián H Zanette. Statistical properties of genealogical trees. *Physical Review Letters*, 82(9):1987, 1999.

[23] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[24] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[25] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.

[26] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.

[27] Mark EJ Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[28] Anirban Banerjee and Jürgen Jost. Spectral characterization of network structures and dynamics. In *Dynamics on and of Complex Networks*, pages 117–132. Springer, 2009.

[29] Anirban Banerjee and Jürgen Jost. Graph spectra as a systematic tool in computational biology. *Discrete Applied Mathematics*, 157(10):2425–2431, 2009.

[30] Sergey N Dorogovtsev, Alexander V Goltsev, José FF Mendes, and Alexander N Samukhin. Spectra of complex networks. *Physical Review E*, 68(4):046109, 2003.

[31] Fan Chung, Linyuan Lu, and Van Vu. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003.

[32] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[33] Fatihcan M Atay, Türker Bıyıkoğlu, and Jürgen Jost. Network synchronization: Spectral versus statistical properties. *Physica D: Nonlinear Phenomena*, 224(1):35–41, 2006.

[34] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000.

[35] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical review letters*, 85(21):4626, 2000.

[36] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[37] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1):1–101, 2011.

[38] Paul Erdös and A Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

[39] Béla Bollobás. *Random graphs*. Springer, 1998.

[40] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

[41] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[42] Shalev Itzkovitz, Ron Milo, Nadav Kashtan, Guy Ziv, and Uri Alon. Subgraphs in random networks. *Physical review E*, 68(2):026127, 2003.

[43] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[44] Béla Bollobás, Svante Janson, and Oliver Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1):3–122, 2007.

[45] E. Olbrich, T. Kahle, N. Bertschinger, N. Ay, and J. Jost. Quantifying structure in networks. *European Physical Journal B*, 77:239–247, 2010.

[46] Sourav Chatterjee, Persi Diaconis, et al. Estimating and understanding exponential random graph models. *The Annals of Statistics*, 41(5):2428–2461, 2013.

[47] Juyong Park and MEJ Newman. Solution for the properties of a clustered network. *Physical Review E*, 72(2):026136, 2005.

[48] R Milo, N Kashtan, S Itzkovitz, MEJ Newman, and U Alon. On the uniform generation of random graphs with prescribed degree sequences. *arXiv preprint cond-mat/0312028*, 2003.

[49] Pedro Ribeiro, Fernando Silva, and Marcus Kaiser. Strategies for network motifs discovery. In *e-Science, 2009. e-Science'09. Fifth IEEE International Conference on*, pages 80–87. IEEE, 2009.

[50] Sebastian Wernicke and Florian Rasche. Fanmod: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006.

[51] Falk Schreiber and Henning Schwöbbermeyer. Mavisto: a tool for the exploration of network motifs. *Bioinformatics*, 21(17):3572–3574, 2005.

[52] Zahra RM Kashani, Hayedeh Ahrabian, Elahe Elahi, Abbas Nowzari-Dalini, Elnaz S Ansari, Sahar Asadi, Shahin Mohammadi, Falk Schreiber, and Ali Masoudi-Nejad. Kavosh: a new algorithm for finding network motifs. *BMC bioinformatics*, 10(1):318, 2009.

[53] Joshua A Grochow and Manolis Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking. In *Research in Computational Molecular Biology*, pages 92–106. Springer, 2007.

[54] Yael Artzy-Randrup, Sarel J Fleishman, Nir Ben-Tal, and Lewi Stone. Comment on" network motifs: simple building blocks of complex networks" and" superfamilies of evolved and designed networks". *science*, 305(5687):1107–1107, 2004.

[55] Jörg Reichardt, Roberto Alamino, and David Saad. The interplay between microscopic and mesoscopic structures in complex networks. *PloS one*, 6(8):e21282, 2011.

[56] Uri Alon. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2006.

[57] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538, 2004.

[58] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[59] Brendan D McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.

[60] Claude E Shannon and Warren Weaver. The mathematical theory of communication (urbana, il. *University of Illinois Press*, 19(7):1, 1949.

[61] Christopher S Wallace. *Statistical and inductive inference by minimum message length*. Springer, 2005.

[62] Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *Information Theory, IEEE Transactions on*, 44(6):2743–2760, 1998.

[63] Ming Li and Paul MB Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2009.

[64] Rüdiger Schack. Algorithmic information and simplicity in statistical physics. *International Journal of Theoretical Physics*, 36(1):209–226, 1997.

[65] Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, pages 416–431, 1983.

[66] Frank Harary and Edgar M Palmer. Graphical enumeration. Technical report, DTIC Document, 1973.

[67] Murray Gell-Mann. What is complexity? remarks on simplicity and complexity by the nobel prize-winning author of the quark and the jaguar. *Complexity*, 1(1):16–19, 1995.

[68] Murray Gell-Mann and James B Hartle. Strong decoherence. *arXiv preprint gr-qc/9509054*, 1995.

[69] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[70] Christopher S Wallace and David M Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.

[71] Anatol E. Wegner. Subgraph covers: An information-theoretic approach to motif analysis in networks. *Phys. Rev. X*, 4:041026, Nov 2014.

[72] Jean-Loup Guillaume and Matthieu Latapy. Bipartite structure of all complex networks. *Information processing letters*, 90(5):215–221, 2004.

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

[74] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[75] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[76] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[77] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[78] Noga Alon, Dana Moshkovitz, and Shmuel Safra. Algorithmic construction of sets for k-restrictions. *ACM Transactions on Algorithms (TALG)*, 2(2):153–177, 2006.

[79] Brendan D McKay. The nauty page. *Computer Science Department, Australian National University, 2004 http://cs. anu. edu. au/bdm/nauty*, 2004.

[80] Adolfo Piperno. Search space contraction in canonical labeling of graphs. *arXiv preprint arXiv:0804.4881*, 2008.

[81] Derek Gordon Corneil and Calvin C Gotlieb. An efficient algorithm for graph isomorphism. *Journal of the ACM (JACM)*, 17(1):51–64, 1970.

[82] Julian R Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

[83] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[84] Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub) graph isomorphism algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10):1367–1372, 2004.

[85] Magnús M Halldórsson and Jaikumar Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18(1):145–163, 1997.

[86] Ravi Boppana and Magnús M Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2):180–196, 1992.

[87] Anne-Ly Do, Johannes Höfener, and Thilo Gross. Engineering mesoscale structures with distinct dynamical implications. *New Journal of Physics*, 14(11):115022, 2012.

[88] Lawrence Davis. Genetic algorithms and simulated annealing. 1987.

[89] Peter JM Van Laarhoven and Emile HL Aarts. *Simulated annealing*. Springer, 1987.

[90] John E Beasley and Paul C Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.

[91] Shai S Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64–68, 2002.

[92] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[93] Anatol E Wegner. Motif conservation laws for the configuration model. *arXiv preprint arXiv:1408.6303*, 2014.

Bibliographische Daten

Subgraph Covers-An information theoretic approach to motif analysis in networks

(Subgraph Überdeckungen-Ein informationstheoretischer Ansatz für Motivanalyse in Netzwerken)

Wegner, Anatol Eugen

Universität Leipzig, Dissertation, 2014

129 Seiten, 15 Abbildungen, 93 Referenzen

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den February 11, 2014

. . . . . .

(Anatol Eugen Wegner)

Daten zum Autor

Name:	Anatol Eugen Wegner
Geburtsdatum:	17/01/1985 in Meerbusch
09/2004 - 06/2008	Bachelor Studium in Physik
	Middle East Technical University -Ankara/Türkei
09/2004 - 06/2008	Bachelor Studium in Mathematik
	Middle East Technical University - Ankara/Türkei
08/2008 - 01/2010	Master Studium in Physik
	University of Illinois at Urbana-Champaign- IL/USA
seit 04/2010	Doktorand am Max Planck Institut für Mathematik
	in den Naturwissenschaften- Leipzig