A COMPARATIVE ANALYSIS ON COMPUTATIONAL METHODS FOR FITTING AN ERGM TO BIOLOGICAL NETWORK DATA

A THESIS

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

BY

SUDIPTA SAHA

UNDER THE SUPERVISION OF

DR. MUNNI BEGUM

BALL STATE UNIVERSITY

MUNCIE, INDIANA

MAY, 2013

Acknowledgement

This thesis would not have been possible without the guidance and the help of several individuals who contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, my utmost gratitude to Dr. Munni Begum, Associate Professor of Mathematical Science, Ball State University who has been my inspiration as I hurdle all the obstacles in the completion of this research work. Her guidance helped me in all the time of research and writing of this thesis. Besides my advisor, I would like to thank my co-chair Dr. Jay Bagga, Professor of Computer Science, Ball State University for his sincerity, encouragement, patience and motivation, and Dr. Ann Blakey, Associate Professor of Biology, Ball State University for her guidance and enthusiasm.

My sincere thanks also goes to Dr. Dale Umbach, Professor, Mathematical Science for his kind help.

Last but not the least, I would like to thank my parents, for giving birth to me at the first place and supporting me spiritually throughout my life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction to the Network data

1.1 Introduction

Over the last decade, there has been a growing interest in the study of biological networks of macromolecular interactions. Identifying basic structural relationships among micro components is the main goal in the field of systems biology. In order to achieve this goal, we need an in-depth knowledge of the underlying structures or networks at the molecular level. A formal basis for handling such complex networks includes computational tools to support the modeling and simulation through methods developed in mathematical biology and bioinformatics. Since the 1960s, with some notable precursors in the preceding decades, a variety of mathematical formalisms have been proposed to describe this kind of complex networking. During the last few years, modeling efforts targeted several distinct types of networks at the molecular level, such as gene regulatory networks, metabolic networks, signal transduction networks or protein-protein interaction networks. Networks of interactions that are not restricted to a cell (intercellular communications) or take place at an altogether different level of detail (immunological networks, ecological networks) are also of immense interest.

1.2 A Brief History of Network Data Analysis

The most popular distinction of the current work on networks and their analysis from previous works of the twentieth century is the scale that measures the dimension of nodes prevalent in a network. For example, one of the popular social networks, Facebook claims to have a billion users, where each user represents a node. On the other hand, protein interaction networks typically involve many hundreds or even thousands of nodes. There are a thousand or more articles on the analysis of co-citation networks today, whereas two decades ago the analysis of co-citation networks would typically involved less than 100 articles or authors (Fienberg, 2013). The reason behind this situation is the applicability of network and analysis got diversification in different concentration in recent era.

A brief excerpt of the history of network data analysis is presented based on Fienberg (Fienberg, 2013). Initially, network studies were limited to social networks. Early network studies in sociology dealt primarily with relatively small sets of subjects whose connections allowed communication across a small set of links in a larger network setting. In their work, Stanley Milgram's group (Milgram, 1967; Travers and Milgram, 1969) presented the idea of how "small-world" phenomenon of short paths of connections linking most people in social spheres. Their studies provided the title for the play and movie "Six Degrees of Separation", which ignored the complexity of their results due to the censoring. In the mid-seventies, White (1970) and Fienberg and Lee (1975) discussed a formal Markov-chain-like model and analysis of the Milgram experimental data, including information on the uncompleted chains. Milgram's data

provided the early development of dynamic networks as his data were gathered in batches of transmission, and thus, these models can be thought of as representing early examples of generative descriptions of dynamic network evolution.

In a more recent study, such as Dodds, Muhamad, and Watts (2003) considered a global "replication" variation on the Milgram study in which more than 60,000 e-mail users attempted to reach one of 18 target persons in 13 countries by forwarding messages to acquaintances. However, only 384 out of 24,163 chains reached their targets, nonetheless the estimated median length for completions was 7. One thing notable here was that they assumed that attrition occurs at random.

Moreover, in the 1970s, several sociologists chose to study "blockmodels" and "structurally equivalent" groups of individuals again with small network datasets such as those arising in Sampson's (1968) study of the relationships among 18 novices in a monastery. During that time, Sampson's data had become a canonical example to illustrate new methods ranging from blockmodel algorithms (Breiger, Boorman, and Arabie, 1975; White, Boorman, and Breiger, 1976), to p_1 model of Holland and Leinhardt (1981) and its generalizations (Fienberg, Meyer, and Wasserman, 1985), to mixed-membership stochastic blockmodels of Airoldi et al. (2008). Other examples include Zachary's karate club network of friendships between 34 members of a karate club at a U.S. university in the 1970s (Zachary, 1977) and Lazega's study of relationships among 72 partners and associates in a law firm (Lazega and van Duijn, 1997).

Recently, evidence indicated that statistical modeling of random networks has had an impact on the empirical study of social networks. Statistical exponential family models

(Strauss and Ikeda, 1990) were a generalization of the Markov random network models introduced by Frank and Strauss (1986), which influenced the developments in spatial statistics (Besag, 1974). Complex dependencies within relational data structures were recognizable through these articles.

Statistical physics models have been used recently to detect community structure in networks (Girvan and Newman, 2002; Backstrom et al., 2006). Moreover, the probabilistic literature on random graph models from the 1990s made the link with epidemics and other evolving stochastic phenomena. Watts and Strogatz (1998) and others used the same idea in the epidemic models to capture general characteristics of the evolution of these new variations on random networks. The demand of stochastic processes as a description of dynamic network models comes from being able to exploit the extensive literature already developed, including the existence and the form of stationary distributions and other model features or properties. Chung and Lu (2006) provided a complementary treatment of these models and their probabilistic properties.

Machine learning is a relatively new approch, which emerged in several forms with the empirical studies of Faloutsos et al. (1999) and Kleinberg (2000a, 2000b, 2001), they introduced a model for which the underlying graph was a grid, the graphs generated did not have a power-law degree distribution, and each vertex has the same expected degree. The strict requirement that the underlying graph be a cycle or grid rendered the model applicable to webgraphs or biological networks.

1.3 Network Data Analysis in Biological Science

Within the fields of Biology and Medicine, potential applications of network analysis or graph theory include identifying drug targets, determining the role of proteins or genes of unknown function (Jeong et al., 2003; Samanta and Loang, 2003), designing effective containment strategies for infectious diseases (Eubank et. al., 2004), and providing early diagnosis of neurological disorders through detecting abnormal patterns of neural synchronisation in specific brain regions (Schnitzler and Gross, 2005). There are several models currently used to represent biological networks which are descriptive in nature, for example, power-law networks (sometimes called scale-free) by Barabaasi and Albert (Barabaasi and Albert, 1999). Other biological network models specify a procedure for creating networks, for example, Erdos-Renyi or an exponential random graph model (ERGM).

Broadly speaking, three classes of such bio-molecular networks attracted the most attention to date: metabolic networks of biochemical reactions; protein interaction networks consisting of the physical interactions between an organism's proteins, and the transcriptional regulatory networks which describe the regulatory interactions between different genes (Pavlopoulos et al., 2011; Mason and Verwoerd, 2007). Networks have been constructed for the transcriptional regulatory networks of *E. coli* and *S. cerevisiae* (Salgado et al., 2006a, 2006b ; Lee et al., 2002; Keseler et al., 2005) and are maintained in databases such as *RegulonDB* (Salgado et al., 2006b) and EcoCyc (Keseler et al., 2005). Such networks have usually been constructed through a combination of high-throughput genome location experiments and literature searches. Many types of gene

transcription regulatory approaches have been reported in the past. Their nature and composition involved categorization by several factors: gene expression values (Keedwell and Narayanan, 2005; Shmulevich et al., 2002); the causal relationship between genes, e.g. with Bayesian analysis or Dynamic Bayesian Networks (Zou and Conzen, 2005; Husmeier, 2003); and the time domain e.g. discrete or continuous time (Li et al., 2006; He and Zeng, 2006; Filkov et al., 2002; Qian et al., 2001). Thus, transcription regulatory network were considered over the other biological networks. This work considered the Transcription Factor- Transcription Factor (TF-TF) interaction network of *E. coli* from *RegulonDB* version *7.4*. A detailed description of TF-TF regulatory network data is given in Chapter 4. Other potential which could be subjected to analyses included the range of organisms from bacteria (genome.wisc.edu) to yeast (yeastgenome.org), to plants (maizese-quence.org) and mammals (namely humans, at genome.gov). These databases have been housed at a variety of server locations with publicly accessible data sets.

Several biological domains are accessible where graph theory techniques can be applied to knowledge extraction from data, for instance, modeling of bio-molecular networks, measurement of centrality and importance in bio-molecular networks, identifying motifs or functional modules in biological networks. Protein-protein interaction (PPI) networks, biochemical networks, transcriptional regulation networks, signal transduction or metabolic networks are the highlighted network categories in systems biology (Pavlopoulos et al., 2011).

1.4 Graphical Model Theory for Network data

The well-developed field of graph theory provides fundamental methods to study complex networks. A *network* or a *graph* is a set V of N *vertices* pairwise connected by a subset E of *edges*; Each is a pair of vertices. These edges can be oriented, weighted, signed, or not (Lesne, 2006; Fronczak, 2012). A graph may be *undirected*, meaning that there is no distinction between the orders of the two vertices associated with each edge, or its edge may be *directed* from one vertex to another. Biological networks come in a variety of forms. Nodes in biological networks represent biomolecules such as genes, proteins or metabolites, and edges connecting these nodes indicate functional, physical or chemical interactions between the corresponding biomolecules. Understanding these complex biological systems has become an important problem that has lead to intensive research in network analyses, modeling, and function and disease gene identification and prediction (Milenkovic, 2008).

Many models currently used for biological networks are descriptive, and simply specify features of a graph. For example, power-law networks (Barabaasi and Albert, 1999) are described as networks with a node degree distribution. Other biological network models specify a procedure for creating networks. Erdos-Renyi random graphs are created by considering each pair of nodes in a given node set as a potential edge. For each potential edge, a fair *n*-sided die is cast, if the die comes up above a given threshold, the edge is included. Otherwise, it is not. An exponential random graph model (ERGM) takes a different, more general approach (Saul and Filkov, 2006) as discussed below.

In particular, graphical models can be studied with respect to both global and local properties of components of the networks. The simplest versions of the local properties are: the number of vertices a network may have and the rules guiding the interactions among these vertices. These local rules are guided by probability theory in order to address the uncertainty and the lack of regularity in real networks leading to random graph models. In the literature of complex network modeling, the exponential family of random graphs is among the most widely used random graph models for social and biological networks (Begum et al., 2012; Przulj et al., 2004; Saul and Filkov, 2007; Pattison and Wasserman, 1999; Robins et al., 1999; Goodreau, 2007; Robins et al., 2007a). Although an Exponential Random Graph Model (ERGM) presents a flexible means to model complex biological models, the likelihood function for parameter estimation involves a mathematically intractable normalizing constant. Several statistical computational methods have been proposed to address this difficulty in parameter estimation in an ERGM. These are the Markov Chain Monte Carlo Maximum Likelihood Estimation (MCMCMLE) method and the Maximum Pseudo Likelihood Estimation (MPLE) method (Hunter and Handcook, 2006; Robins et al., 2007a; Snijders, 2002). The ERGM and the methods of parameter estimation in an ERGM are discusses in detail in Chapter 2.

ERGMs represent the generative process of tie formation in networks, where there are two basic types of processes: dyadic dependent and dyadic independent. A *dyad* refers to a pair of nodes and the relations between them. Dyadic dependent processes are those in which the state of one dyad depends stochastically on the state of other dyads. A classic

example is the concept that "a friend of my friend is my friend" - the presence of a friendship tie in dyads *(i, j)* and *(j, k)* increases the probability of a friendship tie in dyad *(i, k)*. Dyadic independent processes exhibit no direct dependence among dyads: An example is the related social concept that "birds of a feather flock together"- if the two nodes in a dyad have similar attributes, the probability of a friendship tie is increased. The state of the dyad depends on the attributes of the two nodes, but not on the state of other dyads (Hunter et al., 2008; Handcock et al., 2003).

The distinction between these two types of processes affects the specification, estimation and behavior of ERGMs. Models with only dyadic independent terms have a likelihood function that simplifies to a form that can be maximized using standard logistic regression methods. Intuition about how these models behave is usually straightforward, as for logistic regression models. By contrast, models for processes with dyadic dependence require computationally intensive estimation and imply complex forms of feedback and global dependence that confound both intuition and estimation (Hunter et al., 2008; Handcock et al., 2003).

1.5 Research Objectives

In this study, several random networks will be simulated by imposing the number of attributes physically and then comparing them with the observed TF-TF network. The basis for this approach lies in the biological network, itself, where the physical number of attributes might influence the overall biological process. There are several other ways of simulating a random network for instance, Erdos-Renyi method uses a simple binomial distribution with a given probability and using a fitted model to simulate similar kind of

network. However, in this study, the physical number of network attributes will be counted in the observed TF-TF network, and then impose the numbers randomly to simulate networks by keeping almost the same number of different attributes. For comparing the estimates, the ERGM was used to estimate several network attributes of interest. Then, a comparison will be made to determine how far the estimates deviated from the observed network when the same number of attributes is used in the randomly simulated model. Moreover, random network will be simulated to explore how a specific attribute reacted to increased nodes in the random model. The goal was to develop efficient computational methods for fitting ERGM to biological interaction networks through an extensive simulation study. The detailed description of the simulation study is presented in Chapter 4.

Implementation of these computational methods will be carried out using the statistical computing environment software package called R. Specifically, the ERGM, NETWORK, IGRAPH and STATNET packages of the R statistical computing environment are utilized.

The remainder of this thesis is arranged as follows: In Chapter 2, a discussion of the Exponential Random Graph model, applications and the methods for parameter estimation are presented. Chapter 3 presents the observed biological network data. In Chapter 4, the outcomes of the simulation study are presented. And finally, Chapter 5 discusses conclusion and future directions.

Chapter 2

The Exponential Random Graph Model (ERGM)

2.1 General Introduction

A graph consists of a set of objects or individuals, called nodes (points, vertices), connected by links (edges). In the simplest notion, a graph can be considered as a way of specifying pairwise or more complicated relations among a collection of its nodes. In graph theory, a network is denoted as $G = (V, E)$, where V is the set of vertices (or nodes) of the graph, and E are two element subsets of V referred to as edges (or links or connections or other attributes) (Lesne, 2006; Fronczak, 2012). Graphical models are introduced in order to mimic the patterns of connections in real networks, in an effort to understand the implications of those patterns, or just to describe, how network structures originate, and how they evolve over time (Fronczak, 2012). There are several kinds of graphical models such as Markov graph models, Gaussian graphical models, exponential random graph models (ERGMs). ERGMs graphs are among the most widely-studied network models.

An Exponential Random Graph Model (ERGM) models the probability distribution (mass function / density function) for a given class of graphs. Given an observed graph

and a set of explanatory variables on that graph the probability distribution is estimated. The distribution provides a concise summary of the class of graphs to which the observed graph belongs, i.e. the probability distribution can be used to calculate the probability that any given graph is drawn from the same distribution as the observed graph (Wasserman and Pattison, 1996; Saul and Filkov, 2007; Robins et al., 2007a). The ERGM is particularly useful when one wants to create model networks that match the properties of observed networks as closely as possible, but without going into details of the specific process underlying network formation. Such graphical models are not only interesting in their own but also right for the light they shed on the structural properties of networks (Fronczak, 2012).

The first truly general ensemble model for networks was introduced by Solomonoff and Rapoport in 1951, who considered the collection of all undirected simple graphs with a fixed number of vertices, N, in which every pair of nodes was connected with an edge with probability p (Solomonoff and Rapoport, 1951). In the late 1950s and early 1960s, the model was fairly extensively studied by Erdos and Renyi (Erdos and Renyi, 1959; Erdos and Renyi, 1960). Ever since it is known as Bernoulli model or Erdos-Renyi model. This particular ensemble of graphs was indeed the first example of the ERGM. Holland and Leinhardt (Holland and Leinhardt, 1981) who built on statistical foundations laid by Bessag (Bessag, 1974) first introduced the ERGM formally in the early 1980s. Substantial developments were made by Frank and Strauss (Frank and Strauss, 1986; Strauss, 1986) and continued to be made by other authors throughout 1990s and 2000s

(Anderson et al., 1999; Robins et al., 1999; Wasserman and Pattison, 1996; Geyer, 1991; Snijder, 2001; Snijder et al., 2006; Robins et al., 2007a).

The ERGM represents a general and flexible methodology for modeling interactions among a number of actors in a complex network. This methodology originated and has been implemented widely in the literature of social networks. In recent years, there has been growing interest in exponential random graph models for social networks, commonly called the p^*class of models (Pattison and Wasserman, 1999; Robins et al., 1999; Goodreau, 2007; Robins et al., 2007a). These probability models for networks on a given set of actors allow generalization beyond the restrictive dyadic independence assumption of the earlier p_1 model class (Holland and Leinhardt, 1981). The exponential family of random graphs is also among the most widely used random graph models for biological networks (Begum et al., 2013; Przulj et al., 2004; Saul and Filkov, 2007). ERGMs can be used to study models of processes taking place on networks, such as epidemics spread of, diffusion of information, or opinion formation in social networks (Fronczak, 2012).

ERGMs represent the generative process of tie formation in networks with two basic types of processes namely dyadic dependent and dyadic independent. A *dyad* refers to a pair of nodes and the relations between them. Dyadic dependent processes are those in which the state of one dyad depends stochastically on the state of other dyads. Dyadic independent processes exhibit no direct dependence among dyads. In a dyadic independent case, the state of the dyad depends on the attributes of the two nodes, but not on the state of other dyads. The distinction between these two types of processes affects

the specification, estimation and behavior of ERGMs. Models with only dyadic independent terms have a likelihood function that simplifies to a form that can be maximized using standard logistic regression methods. Intuition about how these models behave is usually straightforward, as for logistic regression models. By contrast, models for processes with dyadic dependence require computationally intensive estimation and imply complex forms of feedback and global dependence that confound both intuition and estimation (Hunter et al., 2008; Handcock et al., 2003).

Although an ERGM presents a flexible means to model complex networks, the likelihood function for parameter estimation involves a mathematically intractable normalizing constant. ERGMs generalize the Markov random graph models (Frank and Strauss, 1986), and edge and dyadic independence models. Several statistical computational methods had been proposed to address this difficulty in parameter estimation in an ERGM. These are the Markov chain Monte Carlo Maximum Likelihood Estimation (MCMCMLE) method and the Maximum Pseudo Likelihood Estimation (MPLE) method (Hunter and Handcook, 2006; Robins et al., 2007a; Snijders, 2002). Two special cases of ERGM such as dyadic independence models (also known as p_1 models) and more general p^* models are discussed here in order to lay out the theoretical background of such models.

2.2 p_1 Model

A special case of an ERGM is known as p_1 model (Holland and Leinhardt, 1981). The central building block of these models is the adjacency matrix portraying the

interrelationships between the actors (or nodes) in a network. Let X denote the $v * v$ adjacency matrix with (i, j) th element defined as,

$$X_{ij} = \begin{cases} 1 & if\ i\ relates\ to\ j, \\ 0 & otherwise. \end{cases}$$

The p_1 model is derived by decomposing the adjacency matrix X into its $\binom{v}{2}$ dyads or pairs $D_{ij} = (X_{ij}, X_{ji})\ for\ i < j$. The distribution of X can be specified with the specification of the joint distribution of the D_{ij}. Under the assumption that the D_{ij} are all statistically independent, one needs to specify only the distribution of each D_{ij} in order to completely specify the distribution of X. The probability distributions of D_{ij} are specified as follows:

$$m_{ij} = P\left(D_{ij} = (1,1)\right) \quad i < j,$$

$$a_{ij} = P\left(D_{ij} = (1,0)\right) \quad i \neq j,$$

$$n_{ij} = P\left(D_{ij} = (0,0)\right) \quad i < j, and$$

$$m_{ij} + a_{ij} + a_{ji} + n_{ij} = 1, for\ all \quad i < j$$

Where, m_{ij} is the probability that the dyad *i, j* is a mutual or reciprocated pair; a_{ij} is the probability that the dyad *i,j* is an asymmetric or nonreciprocated pair; n_{ij} probability that the dyad *i, j* is a null pair.

The probability distribution of X is then expressed as follows (Holland and Leinhardt, 1981).

$$P(X = x) = \prod_{i<j} m_{ij}^{x_{ij}x_{ji}} \prod_{i \neq j} a_{ij}^{x_{ij}(1-x_{ji})} \prod_{i<j} n_{ij}^{(1-x_{ij})(1-x_{ji})} \qquad (1)$$

$$= \exp\left[\sum_{i<j} \rho_{ij} x_{ij} x_{ji} + \sum_{i \neq j} \theta_{ij} x_{ij}\right] \prod_{i<j} n_{ij} \qquad (2)$$

The parameters of this model are expressed as,

$$\rho_{ij} = \log\left(\frac{m_{ij} n_{ij}}{a_{ij} a_{ji}}\right) \quad i < j \ and$$

$$\theta_{ij} = \log\left(\frac{a_{ij}}{n_{ij}}\right) \quad i \neq j$$

Note that the parameter p_{ij} is a log-odds ratio and the parameter θ_{ij} is a log-odds. These are interpreted as

$$\exp(\rho_{ij}) = \frac{P(X_{ij} = 1 | X_{ji} = 1)}{P(X_{ij} = 0 | X_{ji} = 1)} \Big/ \frac{P(X_{ij} = 1 | X_{ji} = 0)}{P(X_{ij} = 0 | X_{ji} = 0)}$$

$$\exp(\theta_{ij}) = \frac{P(X_{ij} = 1 | X_{ji} = 0)}{P(X_{ij} = 0 | X_{ji} = 0)}$$

Here ρ_{ij} measures what Holland and Lienhardt (1981) referred to as the *force of reciprocation*. That is, if ρ_{ij} is positive and if $X_{ji} = 1$, then it is more likely to observe $X_{ij} = 1$. Also, θ_{ij} measures the probability of an asymetric dyad between the nodes i and j when it is known that $X_{ji} = 0$. In order to insure the identifiability of the model parameters, a number of restrictions are imposed on ρ_{ij} and θ_{ij}. These are

$$\rho_{ij} = \rho \ for\ all \ i < j,$$

$$\theta_{ij} = \theta + \alpha_i + \beta_i \ for\ all \ i \neq j, and$$

$$\alpha_+ = \beta_+ = 0$$

Under these assumptions the p_1 model in (1) and (2) for the adjacency matrix X becomes,

$$p_1(\mathrm{x}) = P(\mathrm{X} = \mathrm{x}) = \exp\left[\rho m + \theta x_{++} + \sum_i \alpha_i x_{i+} + \sum_j \beta_i x_{+j}\right] \times \prod_{i<j} n_{ij} \qquad (3)$$

Here the n_{ij} are functions of the parameters ρ, θ, $\{\alpha_i\}$, and $\{\beta_j\}$.

2.3 p^* Model

The p^* model is a more general model that includes p_1 model as a special case. In order to specify a p^* model, some additional notation is required.

The general log-linear form of p^* model is expressed as

$$P(\mathrm{X} = \mathrm{x}) = \frac{\exp[\theta \mathrm{z}(\mathrm{x})]}{\kappa(\theta)} \qquad (4)$$

Here θ is a vector of model parameters, $\mathrm{z}(\mathrm{x})$ is a vector of network statistics, and $\kappa(.)$ is a normalizing constant which is hard to compute for large networks. In order to ease the estimation process of the model parameters, the log-linear model form of the p^* model can be re-expressed as a logit model. A logit model is a special case of generalized linear model where log odds of a binary variable is expressed as linear combination of several explanatory variables. In particular, as per Wasserman and Pattison (1996) we define new notation: X_{ij}^{+} denotes an adjacency matrix where a tie from $i \to j$ is forced to be present. That is $X_{ij}^{+} = \{X_{kl},\ with\ X_{ij} = 1\}$. X_{ij}^{-} denotes an adjacency matrix where a tie from $i \to j$ is forced to be absent. That is $X_{ij}^{-} = \{X_{kl},\ with\ X_{ij} = 0\}$. Finally, X_{ij}^{c} denotes an adjacency matrix with complement relation for the tie from $i \to j$. That is, $X_{ij}^{c} = \{X_{kl},\ with\ (k.l) \neq (i,j)\}$. The p^* model in (4) can be turned to a logistic regression

model by considering a set of binary random variables $\{X_{ij}\}$, where $X_{ij} = 1$ implying a tie from $i\ to\ j$.

$$P\left(X_{ij} = 1|\mathrm{X}_{ij}^{c}\right) = \frac{P\left(\mathrm{X} = \mathrm{x}_{\mathrm{ij}}^{+}\right)}{P\left(\mathrm{X} = \mathrm{x}_{\mathrm{ij}}^{+}\right) + P\left(\mathrm{X} = \mathrm{x}_{\mathrm{ij}}^{-}\right)} \tag{5}$$

$$P\left(X_{ij} = 0|\mathrm{X}_{ij}^{c}\right) = \frac{P\left(\mathrm{X} = \mathrm{x}_{\mathrm{ij}}^{-}\right)}{P\left(\mathrm{X} = \mathrm{x}_{\mathrm{ij}}^{+}\right) + P\left(\mathrm{X} = \mathrm{x}_{\mathrm{ij}}^{-}\right)} \tag{6}$$

Using expression in (4) and taking the ratio in (5) and (6) one can write

$$\frac{P\left(X_{ij} = 1|\mathrm{X}_{ij}^{c}\right)}{P\left(X_{ij} = 0|\mathrm{X}_{ij}^{c}\right)} = \exp\left\{\theta\left[\mathrm{z}(\mathrm{x}_{\mathrm{ij}}^{+}) - \mathrm{z}(\mathrm{x}_{\mathrm{ij}}^{-})\right]\right\} \tag{7}$$

$$\log\left\{\frac{P\left(X_{ij} = 1|\mathrm{X}_{ij}^{c}\right)}{P\left(X_{ij} = 0|\mathrm{X}_{ij}^{c}\right)}\right\} = \omega_{ij} = \theta\left[\mathrm{z}(\mathrm{x}_{\mathrm{ij}}^{+}) - \mathrm{z}(\mathrm{x}_{\mathrm{ij}}^{-})\right] \tag{8}$$

$$\omega_{ij} = \theta\delta\left(x_{ij}\right) \tag{9}$$

Here $\delta\left(x_{ij}\right)$ is the vector of difference statistics obtained from the network statistics $\mathrm{z}(.)$ when the variable X_{ij} changes from *1* to *0*. The model in (9) is referred to as the $logit\ p^{*}$ model for single binary relation. One can work with either the log-linear form of p^{*} model given in (4) or the logit form given in (9).

2.4 Computational Methods

There are two methods commonly used in the statistics and social/biological networks communities to estimate the maximum likelihood fit to exponential random graph models. These are the Markov chain Monte Carlo maximum likelihood estimation (MCMC MLE) and maximum pseudo-likelihood estimation (MPLE). They can also be used for network simulation. These techniques have been recently discussed by various

authors (Hunter and Handcook, 2006; Robins et al., 2007a; Snijders, 2002). To date, the most common form of estimation for random graph models has been maximum pseudo likelihood (Strauss and Ikeda, 1990). The properties of the pseudo-likelihood estimator are not well understood, the pseudo-likelihood estimates can at best be thought of as approximate, and it is not clear from existing research as to when pseudo-likelihood estimates may be acceptable. Monte Carlo Markov chain maximum (MCMC) likelihood estimation is the preferred estimation procedure. One of the advantages over maximum pseudo-likelihood estimates is that one can also obtain reliable standard errors for the estimates (Robins et al., 2007b).

2.4.1 Maximum Pseudo Likelihood Estimation (MPLE)

Before the advent of Monte Carlo methods, the only widely used methods for estimating parameters in such models were maximum pseudo likelihood estimation (Besag, 1975) and the closely related method of 'coding' (Besag, 1974), which maximum pseudo likelihood estimation superseded. These methods have been used in preference to Monte Carlo methods because they are much faster, requiring no simulations. The estimators that they produce are not maximum likelihood estimators (except in the limiting case of no dependence); hence the possibility of calculating MLEs leads to the question of whether they are so much better than maximum pseudo likelihood estimates (MPLEs) that their much greater expense is justified (Geyer and Thompson, 1992). A comparison with MLEs shows that MPLEs may seriously overestimate the dependence when it is strong. When the dependence is sufficiently weak, the MPLE behaves well and is almost

efficient (Besag, 1977), as might be expected since the MLE and the MPLE are the same when dependence is absent (Geyer and Thompson, 1992).

The pseudo likelihood function is simply the product of the probabilities of the x_{ij} with each probability conditional on the rest of the data. The method avoids the technical difficulty inherent in the maximum likelihood approach. The pseudo likelihood for model (Equation 10) is identical to the likelihood for a logistic regression model in which the (binary) response data consist of the off-diagonal elements of x_{obs} and the predictor vectors are given by the change statistics $\delta_z(\mathrm{x}_{obs})_{ij}$ of Equation (14).

$$\mathrm{logit}\left[\mathrm{P}_\theta\left(X_{ij} = 1 | X_{ij}^c = x_{ij}^c\right)\right] = \theta\,\delta_z(\mathrm{x})_{ij} \qquad (14)$$

where the logit function is defined by $logit(p) = log[p/(1-p)]$ and X_{ij}^c represents the rest of the network other than the single variable X_{ij}.

Indeed, this is exactly the likelihood that is obtained if one starts with Equation (14) and then assumes in addition that the X_{ij} are mutually independent, so that

$$\mathrm{P}_\theta\left(X_{ij} = 1 | X_{ij}^c = x_{ij}^c\right) = \mathrm{P}_\theta(X_{ij} = 1)$$

The maximum pseudo likelihood estimator (MPLE) for an ERGM, the maximizer of the pseudo likelihood, may easily be found (at least in principle) by using logistic regression as a computational device. When the ERGM in question is not a dyadic independence model, the statistical properties of pseudo likelihood estimators for social networks are not well understood (Hunter et al., 2008).

2.4.2 Monte Carlo Maximum Likelihood Estimation (MCMC MLE)

Recent developments in Monte Carlo estimation techniques for exponential random graph models have been presented and reviewed by a number of authors (Snijders, 2002; Handcock et al., 2006; Snijders et al., 2006, Wasserman and Robins, 2005). The Monte Carlo techniques proposed by Snijders (2002) and Hunter and Handcock (2006) are both based on refining approximate parameter estimates. The approximation proceeds by comparing the observed graphs against a distribution of random graphs generated by stochastic simulation using the approximate parameter values. If the parameter estimates never stabilize (converge), the model is likely to be degenerate. When convergent estimates are obtained, then simulation from the estimates will produce distributions of graphs. The number of edges can be conditioned when estimating parameters, that is, the number of edges is fixed in Monte Carlo estimation procedures (Frank and Strauss, 1986; Snijders et al., 2006). In such models there are no density parameters. Fixing the number of edges diminishes the risk of degeneracy problems and will also have minor effects on other parameter estimates (except perhaps for star parameters). Based on the experience of network scholars, that at least with smaller networks, conditioning on edges may not be necessary, and estimation procedures may successfully converge for the new specifications with density parameters included.

Maximum likelihood estimates (MLEs) in autologistic models and other exponential family models for dependent data can be calculated with Markov chain Monte Carlo methods (the Metropolis algorithm or the Gibbs sampler), which simulate ergodic Markov chains having equilibrium distributions in the model. From one realization of

such a Markov chain, a Monte Carlo approximation to the whole likelihood function can be constructed. The parameter value (if any) maximizing this function approximates the MLE. When no parameter point in the model maximizes the likelihood, the MLE in the closure of the exponential family may exist and can be calculated by a two-phase algorithm, first finding the support of the MLE by linear programming and then finding the distribution within the family conditioned on the support by maximizing the likelihood for that family (Geyer and Thompson, 1992).

Approximating an MLE

The general log-linear form of p^* model is expressed as

$$P(\mathrm{X} = \mathrm{x}) = \frac{\exp[\theta \mathrm{z}(\mathrm{x})]}{\kappa(\theta)} \tag{10}$$

Here θ is a vector of model parameters, $\mathrm{z}(\mathrm{x})$ is a vector of network statistics, and $\kappa(.)$ is a normalizing constant (Frank and Strauss 1986; Wasserman and Pattison 1996).

From the Equation (10), the loglikelihood function can be obtained such that

$$(\theta) = \theta \mathrm{z}(\mathrm{x}_{\mathrm{obs}}) - \log \kappa(\theta) \tag{11}$$

where $\mathrm{x}_{\mathrm{obs}}$ denotes the observed network. Rather than maximize () directly, instead the log-ratio of likelihood values will be considered.

$$(\theta) - (\theta_0) = (\theta - \theta_0)\, \mathrm{z}(\mathrm{x}_{\mathrm{obs}}) - \log\left[\frac{\kappa(\theta)}{\kappa(\theta_0)}\right] \tag{12}$$

where θ_0 is an arbitrarily chosen parameter vector.

The approximation of ratios of normalizing constants such as the one in expression (12) is a difficult but well-studied problem (Meng and Wong 1996; Gelman and Meng 1998). The main idea of the ratios of normalizing constants is presented by Geyer and Thompson (1992) which is described below:

$$\frac{\kappa(\theta)}{\kappa(\theta_0)} = \mathrm{E}_{\theta_0} \exp\{(\theta - \theta_0)\, z(X)\}$$

where E_{θ_0} denotes the expectation assuming that X has distribution given by $\mathrm{P}_{\theta_0,X}$. Therefore, one can exploit the law of large numbers and approximate the log-ratio by

$$(\theta) - (\theta_0) \approx (\theta - \theta_0)\, z(x_{\mathrm{obs}}) - \log\left[\frac{1}{m}\sum_{i=1}^{m} \exp\{(\theta - \theta_0)\, z(X_i)\}\right] \qquad (13)$$

where $X_1, \ldots, X_m$ is a random sample from the distribution defined by $\mathrm{P}_{\theta_0,X}$, simulated using an MCMC routine.

The stochastic estimation technique described above requires one to select a parameter value θ_0. While the approximation of Equation (13) may in theory be made arbitrarily precise by choosing the MCMC sample size m to be large enough, in practice it is extremely difficult to use this approximation technique unless the value θ_0 is chosen carefully. Initial guess for θ_0 should be "close enough" to the true maximum likelihood estimator $\hat{\theta}$ or Equation (13) will fail (Hunter et al., 2008).

2.5 ERGM for Biological Networks

Biological networks have been investigated using several network models such as the Erdos-Renyi model, the geometric random network model, and exponential random graph model (ERGM), and graphical models (Begum et al., 2013). In particular, the

Erdos-Renyi and the geometric random network models were used in the study of graphlets in *Saccharomyces cerevisiae* protein-protein interaction (PPI) networks (Przulj et al., 2004), and exponential random graph models have been employed to study biological databases such as *RegulonDB* (Saul and Filkov, 2007; RegulonDB Release 7.4, 2012). The ERGM has also been used to study large social networks (Goodreau, 2007; Robins et. al, 2007).

In this study, the ERGM for biological network data is considered rather than social network data. Each node is treated as a biological component such as gene, transcription factor, operon, protein, or metabolites. The goal of much of systems biology is to understand the functioning of biological systems which, in large part, depends on their complex underlying structure. Summarizing a biological system into a network represents the study of complex structure via the interactions among its components and the simple recurring patterns, or features, which they form. Thus, when studying the systemic nature of biological networks many modeling approaches focus on simple, but prominent, structural features, as they are easier to understand than the global networks. Once identified, these can be used as building blocks for describing the network (Saul and Filkov, 2007) under consideration.

Node degree distribution and small connected sub-graphs (graphlets), are found to capture structure in biological networks. However, methods that allow us to systematically study these and other local features are needed. Outside biology, statistical network modeling has a long history in the social and economic networks literature. (For example, the concept of network motifs, small sub-graphs that appear in a graph more

often than expected due to chance (Milo et al., 2002), were studied under the name triad census in 1970 (Holland and Leinhardt, 1970). However, biological networks are much larger than social networks and hence direct application of social network models to biological network data has not historically been possible. However, recent advances both in understanding of the behavior of these models and enhanced computational power make such application to biological networks feasible.

Many models currently used for biological networks are descriptive, and simply specify a feature of a graph. For example, power-law networks (sometimes called scale-free) are described as networks with a node degree distribution governed by a power law (Barabasi and Albert, 1999). Other biological network models specify a procedure for creating networks. Erdos-Renyi random graphs are created by considering each pair of nodes in a given node set as a potential edge. For each potential edge, a fair *n*-sided die is cast, if the die comes up above a given threshold, the edge is included. Otherwise, it is not. A more general exponential random graph model suits to explore and model complex biological interaction data.

In order to fit models by using ERGM package and R computational environment, several network attributes are considered in our model. They are edges, triangles, ostars and istars as defined below. In this study, several models with different attributes for both observed and simulated networks are considered. In graph theory, one may consider several network attributes. The formal definitions of some of the attributes are given in the next section.

2.6 Definition of Some Network Attributes

Edges

edges: This term adds one network statistic equal to the number of edges in the network. For undirected networks, edges is equal to kstar(1); for directed networks, edges is equal to both ostar(1) and istar(1) (Morris et al., 2008).

K-In-stars

istar(*k*, attrname=NULL): The k argument is a vector of distinct integers. This term adds one network statistic to the model for each element in k. The *i*th such statistic counts the number of distinct k[i]-instars in the network, where a k-instar is defined to be a node N and a set of k different nodes $\{O_1, \ldots\ldots\ldots O_k\}$ such that the ties $(O_j \rightarrow N)$ exist for $j = 1, \ldots\ldots\ldots, k$. The optional argument attrname is a character string giving the name of an attribute in the network's vertex attributes list. If this is specified then the count is over the number of k-instars where all nodes have the same value of the attribute. This term can only be used for directed networks. Note that istar(1) and ostar(1) both are equal to edge (Morris et al., 2008).

k-Outstars

ostar(k, attrname=NULL): The k argument is a vector of distinct integers. This term adds one network statistic to the model for each element in k. The *i*th such statistic counts the number of distinct k[i]-outstars in the network, where a k-outstar is defined to be a node N and a set of k different nodes $\{O_1, \ldots\ldots\ldots O_k\}$ such that the ties $(N \rightarrow O_j)$ exist for $\$ j = 1, \ldots\ldots\ldots, k$. The optional argument attrname is a character string giving the name

of an attribute in the network's vertex attribute list. If this is specified then the count is the number of k-outstars where all nodes have the same value of the attribute. This term can only be used with directed networks; Note that ostar(1) is equal to both istar(1) and edges (Morris et al., 2008).

Triangles

triangle(attrname=NULL): This term adds one statistic to the model equal to the number of triangles in the network. For an undirected network, a triangle is defined to be any set $\{(i,j),(j,k),(k,i)\ \}$ of three edges. For a directed network, a triangle is defined as any set of three edges $\{(i \to j)\ and\ (j \to k)\}$ and either $(k \to i)$ or $(k \leftarrow i)$. The former case is called a "transitive triple" and the latter is called a "cyclic triple", so in the case of a directed network, triangle equals ttriple plus ctriple — thus at most two of these three terms can be in a model. The optional argument attrname restricts the count to those triples of nodes with equal values of the vertex attribute specified by attrname (Morris et al., 2008).

Transitive triples

ttriple(attrname=NULL): This term adds one statistic to the model, equal to the number of transitive triples in the network, defined as a set of edges $\{(i \to j),(j \to k),(i \to k)\}$. Note that triangle equals ttriple+ctriple for a directed network, so at most two of the three terms can be in a model. The optional argument attrname is a character string giving the name of an attribute in the network's vertex attribute list. If this is specified then the count is over the number of transitive triples where all three nodes have the same value of the attribute. This term can only be used with directed networks (Morris et al., 2008).

Cyclic triples

ctriple (attrname=NULL): This term adds one statistic to the model, equal to the number of cyclic triples in the network, defined as a set of edges of the form $\{(i \rightarrow j), (j \rightarrow k), (k \rightarrow i)\}$. Note that for all directed networks, triangle is equal to ttriple+ctriple, so at most two of these three terms can be in a model. The optional argument attrname is a character string giving the name of an attribute in the network's vertex attribute list. If this is specified then the count is over the number of cyclic triples where all three nodes have the same value of the attribute. This term can only be used with directed networks (Morris et al., 2008).

Chapter 3

Biological Network Data

Biological network data arise in a variety of forms. Nodes in biological networks represent biomolecules such as genes, proteins or metabolites, and edges connecting these nodes indicate functional, physical or chemical interactions between the corresponding biomolecules. Understanding these complex biological systems has become an important problem that has led to intensive research in network data analyses, modeling, and function and disease gene identification and prediction. Transcription Regulatory Interaction Network is considered in this thesis, as these are fundamental biological interaction networks. The amount of gene expression depends on how the genes are being regulated by TFs. The regulatory network of the model organism *Escherichia Coli (E.coli)* from *RegulonDB* version 7.4[1] is considered. There are several other potential databases include the range of organisms from bacteria (genome.wisc.edu) to yeast (yeastgenome.org), to plants (maizese-quence.org) and mammals (namely humans, at genome.gov). These databases are housed at a variety of server locations with publicly accessible data sets.

[1] http://regulondb.ccg.unam.mx/

3.1 General Introduction on *RegulonDB*

A database is a complete collection of information of a certain interest. *RegulonDB* is a regulatory interaction data repository for the model organism *E. coli.* At the same time, it is also a model of the organization of the genes in transcription units, operons and simple and complex regulons. From that point of view, *RegulonDB* is a computational model of mechanisms of transcriptional regulation. Regulon research group also updates the website on a regular interval. In order to implement our method to a known biological network dataset and to obtain comparative results among MCMC MLE and MPLE, the regulatory network of *E.coli* from *RegulonDB* version 7.4 is considered. In the next section, some of the key definitions to give readers a brief overview of different biological terms is presented.

3.2 Key Definition of Biological Network Components

Some of the key definitions of biological entities belonging to the database are presented here. Although in this study we used the Transcription Factor-Transcription Factor (TF-TF) network, there are several other networks that can be explored similarly. Keeping that in mind, we report the definitions of several components below. The detailed descriptions can be found at the *RegulonDB* website.

Operon

The first component is *Operon*. An operon can be defined as the set of one or more genes and their associated regulatory elements, which are transcribed as a single unit. The classical definition is that of a group of two or more genes transcribed as a polycistronic

unit (Jacob and Monod, 1961). However, at the database they extend the definition to include the possibility of operons with only one gene. In this case, an operon is a group of one or more contiguous genes transcribed in the same direction. It is notable that given this definition, an operon must contain a promoter upstream of all genes and a terminator downstream. It is relatively common to find operons with several promoters, some of them internally located, thus transcribing a partial group of genes. In all the cases so far, one gene belongs to only one operon. The graphic display of an operon contains all the genes of its different transcription units (TUs), as well as all the regulatory elements involved in the transcription and regulation of those TUs. The genome browser shows genes and operons, accepting also monocistronic operons. In this definition, there are several biological terms. To learn more about these, a biological dictionary would be a good resources.

Transcription unit (TU)

The next term is *Transcription unit* (TU). A Transcription unit is a set of one or more genes transcribed from a single promoter. A TU may also include regulatory protein binding sites affecting this promoter and a terminator. It is notable that a complex operon with several promoters contains, several transcription units. Given the definition of an operon, at least one transcription unit must include all the genes in the operon.

Promoter

The next term is *Promoter*. A promoter is defined as the part of the DNA sequence where RNA polymerase binds and initiates transcription. Moreover, Promoter sequences are specific to different sigma factors associated to the RNA polymerase core. A promoter is

represented as a stretch of upper-case nucleotide sequence, 60 bases upstream and 20 downstream from the precise initiation of transcription or +1. In more recent studies, it has been identified that there are RNAP binding sites which do not initiate transcription. Therefore, these are not promoters since they are not functional.

Binding site

The next term is *Binding site*. The TFs binding sites are physical DNA sites recognized by transcription factors within a genome. However, binding sites for transcriptional regulators were defined as operator sites. There are several meanings of an operator site. In their wider meaning, operator sites are sites for repressors or activators. Later on, the term "activator sites" was opposed to "operator sites", where operator sites were limited to sites for the binding of repressor regulators. In bacteria, specifically for Sigma 54 promoters, the term "UAS" for upstream activator sites is also used to refer to activator site that functions remotely. A related term is that of enhancers. An enhancer has been initially defined as an activator sites, tht functions from far upstream, and which functions in either orientations in relation to the promoter.

Terminators

A *Terminator* is the region where transcription ends, and RNAP unbinds from DNA.

Gene

The formal definition of gene is a unit of heredity that is transferred from a parent to an offspring is held to determine some characteristic. In general, proteins are coded directly

by genes. For technical use, a gene is a distinct sequence of nucleotides forming part of a chromosome.

Protein

Protein can be defined as any of a class of nitrogenous organic compounds that consist of large molecules composed of one or more long chains of amino acids or such substances collectively, especially as a dietary component.

Most of the time, regulatory elements occur upstream of operons. However, there is a good number of regulatory elements (promoter and binding sites) located inside a promoter, defining a different transcription unit.

An important aspect to keep in mind in order to avoid confusion in the content of a database is the fact that the current understanding and characterization of different genes, operons and regulatory mechanisms is quite variable. For some genes, their mechanisms are very well described, whereas in other cases there is no regulation defined for a given promoter, or a promoter has been characterized upstream of a poorly characterized operon or transcription unit. Our definitions and conventions affect not only the way well-characterized systems are described, but also the way the lack of information is taken into consideration.

3.3 New Additions in Release 7.4

RegulonDB Release 7.4 is used in the analysis. It was released on March 29, 2012. New additions in this version includes consensus sequences, lengths, and symmetries corresponding to 10 TFs. Also updated were the binding sites for 4 TFs that belong to the

LysR family (ArgP, IlvY, MetR, and NhaR) and 3 response regulators that correspond to two-component systems (BaeR, CitB, and CpxR); DinJ is included in the toxin/antitoxin system, and PurR regulates genes involved in purine/pyrimidine biosynthesis. Finally, PdhR is involved in central metabolic fluxes and, more recently, has been found to be involved in the utilization of glycolate and cell division.

In addition to the above different strategies were used to identify the characteristics of the TFBSs. The regulonDB research group performed alignments of the sequences upstream of genes regulated by these proteins and compared orthologous intergenic regions, and the research group also used other databases, such as RegPrecise Novichkov et al. 2010. In addition, the binding sites of the regulator MetR were corrected based on comparisons with homologous sequences reported for Salmonella typhimurium. In all cases researchers also analyzed the available experimental evidence that corresponded to each regulatory interaction.

On the other hand, the researchs are continuing with the annotation of allosteric regulation of the RNAP by ppGpp and DksA. In this sense researchers have expanded the notes for GreB, GreA and DksA. In addition the researchers also have enriched notes for different transcriptions factors, such as: AidB, ArgP, AtoC, DcuS, DpiB, Fur, HNS, LacI, MalT, MntR, PaaX, PhoB, PutA and SoxS.

3.4 Transcription Factor-Transcription Factor (TF-TF) interaction Network of *E. coli*

In this thesis, TF-TF interaction network data is considered from *RegulonDB* version 7.4. In the original data set (represented as a table) of *E.coli* in the *RegulonDB* website there are four columns. The first column is the name of the Transcription Factor (TF), the second column is TF regulated by TF, third column is Regulatory effect of the TF on the regulated gene (+ activator, - repressor, +- dual, ? unknown) and the forth column is the evidence of supports the existence of the regulatory interaction. The first two columns are considered and it created that TF-TF interaction network. The observed TF-TF network is given in Figure 3.1. This diagram was generated with R statistical environment using Network package.

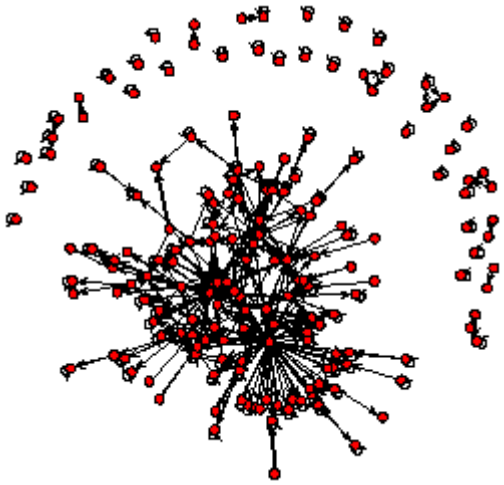

Figure 3.1: Observed TF-TF network

This is a directed network with loops. Here, each TF is considered as the vertex and the tie between two TFs as edge. An edge from a TF to another TF represents that the first

TF regulates the second TF. We explored this observed network and counted the number of several network attributes. In this observed network, there are *387* edges, *114* triangles, *20* ostar-3, *34* istar-3, 10 ostar-5, and 9 istar-5. Looking at the plot we also can say that the network data has two big clusters and several small clusters.

Chapter 4

Simulation Study

We conduct a simulation study for generating random network under varying conditions. We choose conditions by choosing different no. of nodes and network statistics. The primary reason behind conducting the simulation is to determine the cut-off points for different number of nodes for specific attributes and also to compare our simulated models with an observed model. For the comparison part, we create two networks by imposing the same number of network attributes to the models and then compare the results of estimates we get by fitting ERGM. We consider ostar-5, istar-5, ostar-6, istar-6 and triangles as our network attributes and then we mimic networks as an observed network. We physically impose these attributes into the simulated network by keeping the same number of attributes as the observed network. However, we could not able to impose exactly same numbers of attributes to our models, but we are very close to the observe network as far as the number concerns. We also do not simulate the number of edges, because if we simulate triangles, ostars and istars, edges are automatically created. We randomly assign these attributes to the simulated models. Moreover, we also simulate

networks for different number of nodes (*n=20, 50, 100*) and determine the conditions for these statistics to become insignificant. We determine the cut-off points for single attributes and also for combinations of attributes. However, due to the convergence issues, we were not able to find the cut-off points in the some cases. We define a cut-off point as the value where network attributes become significant to insignificant or vice versa. The motivation behind this that is if the biological network behaves almost the same as the random network, then if we have observed network with different number of nodes, we can say up to which point (approximately) certain statistics become insignificant for a given situation.

We also simulate networks as the Transcription Factor-Transcription Factor (TF-TF) interaction network of *E.coli.* The *RegulonDB* database (Release 7.4) contains up-to-date regulatory interaction networks of the model organism *E. coli.* The different network attributes that we consider are edges, ostar-5s, istar-5s, ostar-6s, istar-6s and triangles. We explore the original TF-TF interaction network of *E. coli* and found that there are *10* ostar-5s, *9* istar-5s, *10* ostar-6s, *8* istar-6s and *114* triangles, and also in the observed model there are *175* nodes with density *0.012*. Once we determine the number of attributes in the observed model, then we mimic this network and randomly simulate two networks. Details are presented in section 4.5. Once we get our simulated network, we consider different combinations of attributes (ostars, istars and triangles) and fit the models by ERGM. We fit the same models for the observed data by using ERGM and then compare the estimates of ERGM for both MCMC MLE and MPLE method.

At the outset of our simulation study, we consider networks with small number of nodes and then we move towards higher number of nodes. The simulation studies for different number of nodes are summarized below.

4.1 Simulation with *n=20*

For this simulation study, we start with the number of nodes *n*=20. Since TF-TF interaction network of *E. Coli* is directed in nature, we focus on the directed networks only. However, this study can be considered in a similar fashion for undirected network as well. With only *20* nodes, we consider reasonably smaller magnitude of network attributes such as edges, ostar-3s, istar-3s and triangles as our attributes of interest and then fit the models with ERGM to get the estimates and also to determine the cut-off points. We start with smaller number of attributes say 2 ostar-3, 2 istar-3 and 2 triangles. We increment each attributes one at a time and try to determine the cut-off points. We observe that if we have *75-77* or more triangles, *11-12* or more istar-3s and *13-14* ostar-3, all the attributes including edges are become insignificant in MCMC MLE method. If the network attributes are less than the numbers reported above, we might get that some of the attributes are significant at *5%* level of significance in the MCMC MLE method. To determine the cut-off points we use MCMC MLE method only. However, the estimates in MCMC MLE and MPLE method of the network attributes with *77* triangles, *12* istar-3 and *14* ostar-3 are presented in Table 4.1.

Table 4.1: Estimates of the simulated network for *n*=20

Network Attributes	MCMC MLE Estimates	MPLE Estimates	P-value (MCMC MLE)	P-value (MPLE)
Edges	-2.08935	-1.91030	<1e-04	<1e-04
Triangle	0.12671	0.14164	0.000383	0.0397
Ostar-3	0.059177	-0.01626	0.934304	0.5659
Istar-3	-0.00619	0.01533	0.431438	0.4498

Simulated network with *77* triangles, *12* istar-3 and *14* ostar-3 is presented in Figure 4.1.

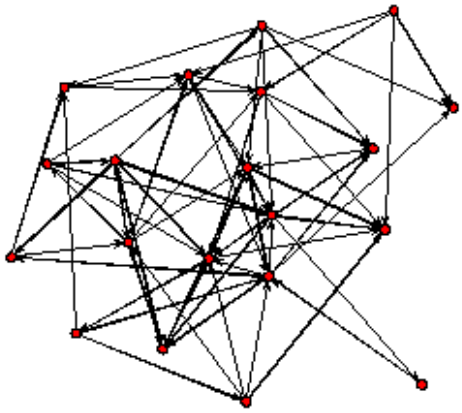

Figure 4.1: Simulated network for *n=20*

We also fit several models to determine what are the cut-off points (in MCMC MLE method) for individual network attributes along with the edges. We find that for ostar-3 the cut-off point is *7* and *17*. If the number of ostar-3s is *17*, ostar-3s are significant. Now, if we decrease the number of ostar-3s by *1*, ostar-3s become insignificant at *5%* level of significance. Again, if we increase the number of ostar-3s by *1* from *6*, ostar-3s become insignificant on an average. We define the cut-off as when a single attributes or combination of attributes become significant to insignificant and vice versa. Usually, for any specific network attributes, there are two cut-off points, one is lower cut-off point and the other one is higher cut-off point. The lower cut-off point is the number for which

any specific attribute become significant to insignificant and the higher cut-off point is the number for which any attribute become insignificant to significant. The network with *17* ostar-3s is presented in Figure 4.2.

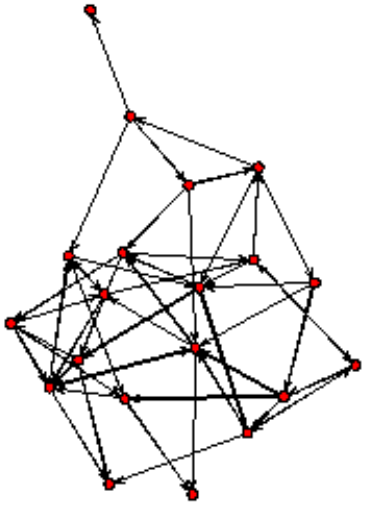

Figure 4.2: Simulated network with *17* ostar-3s

The cut-off points (in MCMC MLE method) of istar-3 are *6* and *17*. If we add one more istar-3 in the network model after *5*, istar-3s become insignificant and also increase the number of istar-3 after *16*, the istar-3 become significant. The network with *7* istar-3 is presented in Figure 4.3.

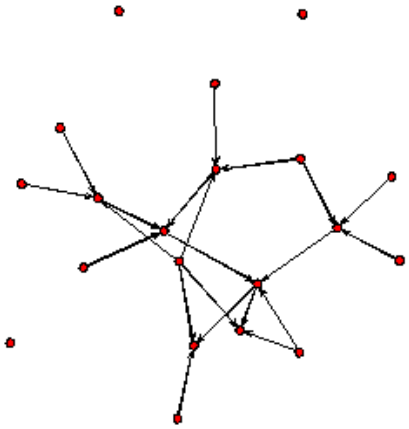

Figure 4.3: Simulated network with *7* istar-3

The higher cut-off point for triangles is in between *76-80* in MCMC MLE method. After this range both edges and triangles become insignificant. Here, we couldn't find the lower cut-off point for triangles in MCMC MLE method due to the convergence issue. The network with *79* triangles shows in Figure 4.4.

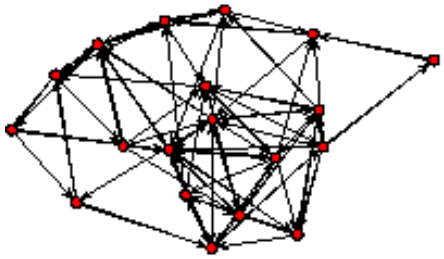

Figure 4.4: Simulated network with *79* triangles

For edges it also shows a similar pattern. It becomes significant for small number and also become significant for bigger numbers.

4.2 Simulation with *n=50*

In this section we simulate random directed networks for 50 nodes. Here we simulate networks with different number of istar-3s, ostar-3s and triangles and try to determine the cut-off points. As before, we start with a small number of attributes and gradually increase the number. We use MCMC MLE method to determine the cut-off points. For the full model including edges, ostar-3s, istar-3s and triangles, our lower cut-off points for combination of the attributes are *30-31* for istar-3, *30-32* for ostar-3 and *70* for

triangles in MCMC MLE method. The attributes are become insignificant after this combination. We couldn't find the higher cut-off point for the combination of attributes because of the convergence issue. The simulated network with *31* istar-3, *32* ostar-3 and *70* triangles is presented in Figure 4.5.

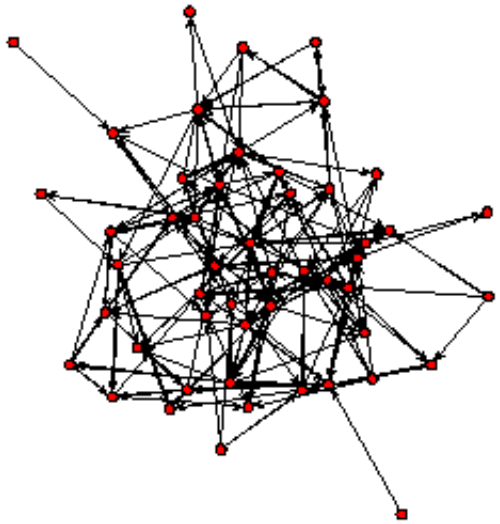

Figure 4.5: Simulated network with *31* istar-3, *32* ostar-3 & *70* triangles

We considered the MCMC MLE method to find the cut-off points. For ostar-3s, the higher cut-off point is *35*. After this number ostar-3s becomes significant. We couldn't find the lower cut-off point for ostar-3s, because of the convergence issue. Again just for istar-3, the higher cut-off point is *35*. After this point istar-3s become significant. The lower cut-off point for istar-3s is *3*. If we increase the number of istar-3 after *2*, the istar-3 becomes insignificant. On the other hand, for just triangles, we couldn't find the cut-off points.

4.3 Simulation with *n=100*

For *n=100* we simulate multiple directed networks with various number of istar-3s, ostar-3s and triangles. We consider each attribute with edge say edges and triangles, edges and ostar-3s and so on. We also consider combination of all attributes i.e. edges, triangles, ostar-3s and istar-3s together. Here we also consider the MCMC MLE method to determine the cut-off points. However, for *n=100* we couldn't find the cut-off point for the combination of attributes. The reason behind that is, as we are proceeding to higher order networks, the number of possible combinations also increased proportionally. Therefore, determining cut-off points becomes tedious. However, we noticed a combination for which all network attributes became insignificant except edges. For *26* istar-3s, *23* ostar-3s and *32* triangles, all the network attributes except edges become insignificant. Again, for *23* istar-3s, *24* ostar-3s and *27* triangles, we have all the network attributes are significant except ostar-3 at *5%* level of significance. The network model with *26* istar-3, *23* ostar-3 and *32* triangles looks like Figure 4.6.

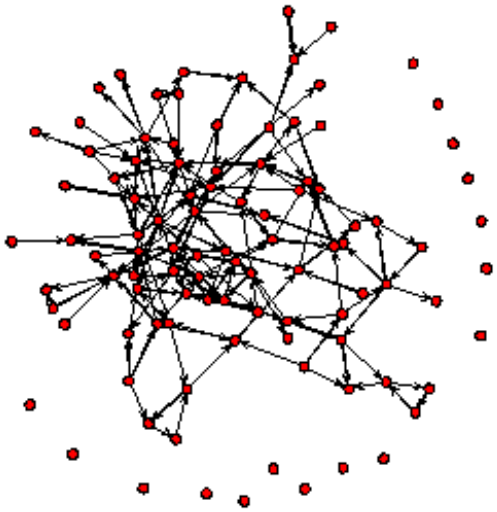

Figure 4.6: Simulated network with *26* istar-3, *23* ostar-3 and *32* triangles

For ostar-3 the lower cut-off point is *4* and the higher cut-off point is *64*. Again for istar-3 the cut-off points are *5* and *64*. For triangles, we couldn't reach into a cut-off point due to the convergence issue.

We summarize our simulation studies for different numbers of nodes in the following Table 4.2.

Table 4.2: Summary of simulation studies for different numbers of nodes

	For *n*=20			**For *n*=50**			**For *n*=100**		
	Triangles	Ostar-3	Istar-3	Triangles	Ostar-3	Istar-3	Triangles	Ostar-3	Istar-3
Lower cut-offs	-	7	6	-	-	3	-	4	5
% of *n* (apps)		35%	30%			6%		4%	5%
Higher cut-offs	76-80	17	17	-	35	-	-	64	64
% of *n* (apps)	390%	85%	85%		70%			64%	64%

From the above table, we can say that, the cut-off points for ostar-3s and istar-3s are quite similar, although we couldn't find any conclusive answer for *n=50*. For ostar-3 and istar-3, we can say that, the cut-off points spread out with the increase in the nodes i.e. if we move forward towards higher orders, the lower cut-off points become smaller and higher cut-off points become smaller as well with respect to percentage of the nodes. For *n=20*, the total spread of insignificant region is close to *(85-35) = 50%* and which is approximately *60%* for *n=100*. As we couldn't find a conclusive result for triangles, we cannot make any conclusion in this regard. However, we can say that, for triangles cut-off points should be bigger than the number of nodes i.e. *n*. In summary, we can say that,

for network data if we increase the order of the nodes, the spread of the insignificant region gradually gets bigger for any specific attributes. To determine the exact percentage of cut-off points, we have to do similar study for more nodes, then and we can generalize the idea.

4.4 Simulation with *n=175*

The *E. coli* TF-TF regulatory network taken from the *regulonDB* network has a total of 175 nodes. In order to mimic this network we simulate directed networks with *175* nodes. We start with ostar-3s, istar-3s and triangles. In our observed TF-TF network, we count *114* triangles, *20* ostar-3 and *34* istar-3. Thus we simulate a directed network having same numbers of network attributes. However, the closer we can get to the observed network is a network with *122* triangles, *50* istar-3s and *51* ostar-3s. The reason behind that is, if we randomly impose *114* triangles, a certain number of istar-3s and ostar-3s already created and which is greater than our observed numbers. The directed network with *122* triangles, *50* istar-3 and *51* ostar-3 is presented in Figure 4.7.

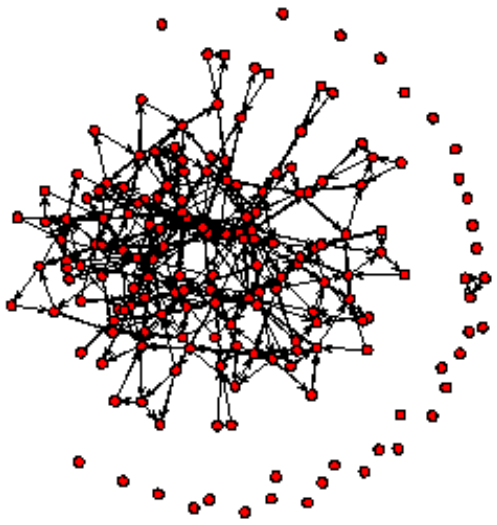

Figure 4.7: *50* istar-3, *51* ostar-3 and *122* triangles

As before, we increase the order of ostars and istars sequentially. For ostar-5s, istar-5s and triangles, we reach close to our observed model. Ostar-6s, istar-6s and triangles also give a very good approximation to our observed model. Therefore, we simulate two directed (as TF-TF is directed) networks, one with ostar-5s, istar-5s, and triangles (network-1) and the other with network same number of ostar-6s, istar-6s, and triangles (network-2). The triangles differ by just *1* for both networks. The detailed comparison based on the number of attributes is presented in section *4.7*. In the next section, we try to fit different modes by using ERGM for both the networks we simulate here and also compare the estimates of attributes with the observed TF-TF network attribute estimates.

4.5 Observed Vs Simulated

There are several biological domains where graph theory techniques are applied for knowledge extraction from data. In this section, we compare our observed TF-TF network of *E. coli* with our simulated model for both MCMC MLE and MPLE methods. Since ERGM package cannot handle self loop while model fitting we exclude loops from the network. If we have self loops in the network data, the model doesn't converge. If this self pool problem can be addressed, it would be a good extension of the ERGM model.

Here we consider TF-TF interaction model because it has a relatively smaller number of nodes which is comparatively easier to mimic. However, this procedure can be applied in similar fashion to simulate networks of more complex organisms. In our observed TF-TF model, we have *175* nodes, *114* triangles, *10* ostar-5s, *9* istar-5s, *10* ostar-6s and *8* istar-5s. An R-script is written to count the number of attributes in the model. Then we randomly simulate two different network models to compare the estimates of these

network attributes with the observed network. In both cases, we have very close estimates of network attributes from the simulated models compared to the actual model. In this section, we start with the observed network first and then we move on to our simulated networks. In Figure 4.8 and 4.9 we represent the observed TF-TF network with and without loops.

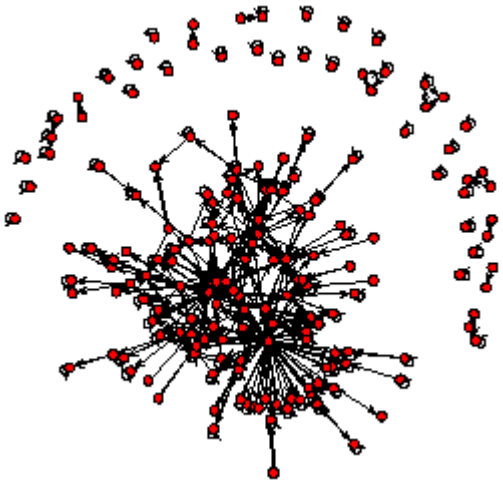

Figure 4.8: Observed TF-TF network with looping

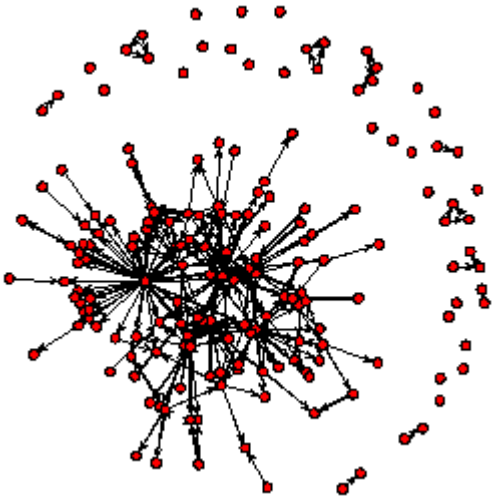

Figure 4.9: Observed TF-TF network without looping

We consider five models based on the combinations of the attributes. The combinations of the attributes and the corresponding estimates in both MCMC MLE and MPLE method are reported in the Table 4.3 and 4.4 respectively.

Table 4.3: MCMC MLE Estimates of the different models for observed network

Models	Network Attributes	Corresponding MCMC MLE Estimates
1	edges+ostar-5+istar-5+triangle	-5.3500647,0.0003852,0.0022043,0.9089341
2	edges+ostar-5+triangle	-5.3356386,0.0003851,0.9355000
3	edges+ostar-5+istar-5	-4.9965056,0.0003874,0.0044687
4	edges+ostar-6+istar-6+triangle	-5.33e+00,7.797e-05,1.034e-01,9.220e-01
5	edges+ostar-6+triangle	-5.33e+00,7.796e-05,9.385e-01

Table 4.4: MPLE Estimates of the different models for observed network

Models	Network Attributes	Corresponding MPLE Estimates
1	edges+ostar-5+istar-5+triangle	-5.35, 1.564e-05, 2.204e-01, 9.089e-01
2	edges+ostar-5+triangle	-5.336, 1.551e-05, 9.355e-01
3	edges+ostar-5+istar-5	-4.997e+00, 1.783e-05, 4.469e-03
4	edges+ostar-6+istar-6+triangle	-5.337e+00, 1.676e-06, 1.034e-03, 9.220e-01
5	edges+ostar-6+triangle	-5.33, 1.667e-06, 9.385e-01

From Table 4.3, we observe that for MCMC MLE whichever combinations of network attributes we consider i.e. edges, istar-5s or triangles, the estimates of that network attributes remain approximately same. The numbers on the table represent the corresponding estimates of the network attributes separated by commas. For example, for edges, we consider edges in all the five models in Table 4.3, and for all cases, we get approximately same estimate, which is *-5.35*. Again we consider ostar-5 in models 1 through 3, and in each case we get approximately same estimates which is *0.0003852*. This is also true for the other attributes (i.e. ostar-6, istar-6, triangles) in the Table 4.3 (in MCMC MLE method).

Now in the Table 4.4, we report the estimates that we get for different models in MPLE method. Here, we also notice the same feature of the network statistics; the estimates are approximately same irrespective of the models or combinations. The reason behind that is, the physical numbers of a specific network attributes remain same irrespective whatever models or combinations we consider i.e. the number of triangles in a specific network model remains same irrespective of the combinations of attributes. Keeping that in mind, we summarize the estimates of different attributes that we get from under different model into a table (for both MCMC MLE and MPLE methods). The summary

of the estimates of attributes of the observed network under different model is presented in Table 4.5.

Table 4.5: Summary table of estimates of the observed network

Network Attributes	MCMC MLE Estimates	MPLE Estimates
Edges	-5.3500647	-5.35
Triangle	0.9355000	9.355e-01
Ostar-5	0.0003851	1.564e-05
Istar-5	0.0022043	2.204e-01
Ostar-6	7.797e-05	1.676e-06
Istar-6	1.034e-01	1.034e-03

To compare the estimates of network attributes between the observed and simulated network, we randomly simulate two networks by imposing the same number of attributes as TF-TF, one with ostar-5s, istar-5s, and triangles (network-1) and the other with network same number of ostar-6s, istar-6s, and triangles (network-2). The first one has almost same number of network attributes as the observed TF-TF network. It has the same number of ostar-5s and istar-5s and *115* triangles which differs by only *1* triangle compared to our observed network, as in our observed network we have *114* triangles. Then we fit several models with ERGM on our first network (network-1) (Figure 4.10) like our observed network reported in Table 4.5. For instance, we consider a model with edges, ostar-5, istar-5 and triangles; then we consider another model with just edges and ostar-5 and so on. Like the observed network, we notice a similar feature of the network attributes that the estimates do not change no matter whatever model or whichever combinations we are considering. As long as it is the same network, the attributes under that network give the same estimates. Keeping this observed property in mind, we

summarize the estimates of network attributes that we get from network-1, under different models, in Table 4.6.

Table 4.6: Summary table of estimates of the simulated network-1

Network Attributes	MLE Estimates	MLPE Estimates
Edges	-5.73286	-5.675632
Triangle	2.90743	2.905757
Ostar-5	-0.01720	-0.016141
Istar-5	-0.08434	-0.083797

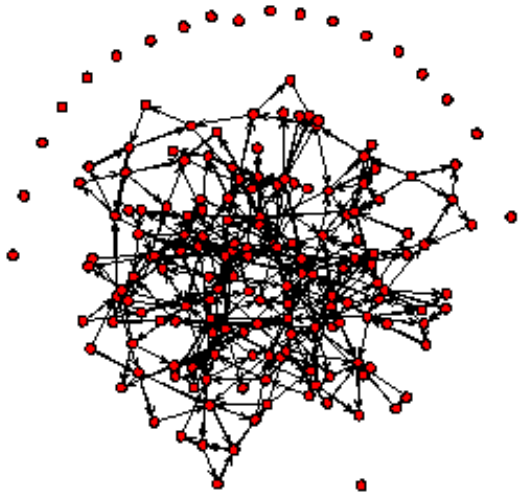

Figure 4.10: Simulated network-1

In our second simulated network (Figure 4.11), we randomly impose almost the same numbers of ostar-6 and istar-6 and triangles. In network-2, the numbers of istar-6, ostar-6 are exactly similar. However, the number of triangle is *115* which differs by only *1* triangle compared to our observed network, as in our observed network we have *114* triangles. Then we fit several models considering all four attributes together and individually. In all case, we get approximately same estimates for a specific attributes like before no matter whatever model or whichever combinations we are considering. As

this property is true for any attribute, we can represent the estimates in a single table. For reporting we consider both MCMC MLE and MPLE method. The summary of the estimates of attributes that we get from network-2 under different models, is presented in Table 4.7.

Table 4.7: Summary table of estimates of the simulated network-2

Network Attributes	MCMC MLE Estimates	MPLE Estimates
Edges	-5.539e+00	-5.4661274
Triangle	2.570e+00	2.5659793
Ostar-6	-1.342e-01	-0.1343137
Istar-6	-8.702e-04	-0.0006957

The simulated network with ostars, istars and triangles is presented in Figure 4.11.

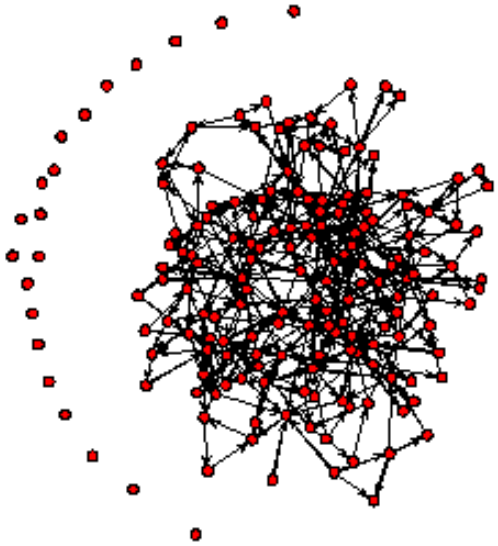

Figure 4.11: Simulated network-2

Finally, for comparison purposes, we create two summary tables of estimates of the observed and simulated models, one for MCMC MLE and the other MPLE, to compare between the estimates we have from the observed model and our simulated models. Although we simulate two models considering different attributes, we present them under

a single table. The reason behind that is we simulate these models based on the number of attributes that we have in the observed TF-TF model. We can make two different tables but we present it in a single table so that we can summarize the big picture in a single table. The summary tables for MCMC MLE and MPLE is presented in Table 4.8 and 4.9 respectively.

Table 4.8: Summary table of estimates OBSERVED Vs SIMULATED in MCMC MLE method

Network Attributes	Estimates from observed networks	Estimates from simulated networks
Edges	-5.3500647	-5.73286
Triangle	0.9355000	2.90743
Ostar-5	0.0003851	-0.01720
Istar-5	0.0022043	-0.08434
Ostar-6	7.797e-05	-1.342e-01
Istar-6	1.034e-01	-8.702e-04

Table 4.9: Summary table of estimates OBSERVED Vs SIMULATED in MPLE method

Network Attributes	Estimates from observed networks	Estimates from simulated networks
Edges	-5.35	-5.675632
Triangle	9.355e-01	2.905757
Ostar-5	1.564e-05	-0.016141
Istar-5	2.204e-01	-0.083797
Ostar-6	1.676e-06	-0.1343137
Istar-6	1.034e-03	-0.0006957

From the above tables, we conclude that except triangles the rest of the estimates of network attributes are very close (for both MCMC MLE and MPLE method). Therefore, from the biological point of view, if we can figure out the network and the number of certain network attributes in a given model, then it will behave almost same as the

random model for most of the cases. Although, to generalize the case we need more experiment and also need exploration among higher order of species.

From this experiment, we can say that if we want to simulate a biological data, then a good way would be to explore the observed data and count the number of statistics that we are interested and then physically impose the number of statistic and then compare. There are several other ways to simulate network models using several packages on R. The simplest way is to take the density of the observed model and simulate it using binomial distribution. Also, once a model is fitted by using ERGM package, then you can simulate one from the observed fitted model. ERGM takes the estimates of the network attributes and simulates a similar type of model. However, in such a case the physical number of attributes differs by substantially. Again, we can also simulate networks by using Erdos-Renyi model. However, for all the cases the physical number of attributes significantly differs. Therefore, from biological point of view, the total number might have a significant influence over the whole process.

4.6 Comparison over other Simulation Methods

In this section, we simulate several networks by the existing simulation scheme. We simulated a network by using Erdos-Renyi modeling scheme where we consider *175* nodes to create similarity with our observed TF-TF network and then consider the density of the TF-TF model. As already mentioned, Erdos-Renyi simulates network by considering just the density. After simulating the network models we fit several model by ERGM to estimate the attributes of interest so that we can compare the estimate with the observed model. As before, the estimates are approximately similar irrespective of

combinations or models. The summary of the estimates that we get under different models are provided in Table 4.10 (for both MCMC MLE and MPLE).

Table 4.10: Summary table of estimates from Erdos-Renyi model

Network Attributes	MLE Estimates	MPLE Estimates
Edges	-4.438846	-4.42969
Triangle	-0.058951	-0.06431
Ostar-5	-0.007336	-0.00546
Istar-5	-0.120974	-0.15166
Ostar-6	-0.01302	-0.01366
Istar-6	-0.75202	-0.84257

The simulate network using Erdos-Renyi modeling scheme is represented in Figure 4.12.

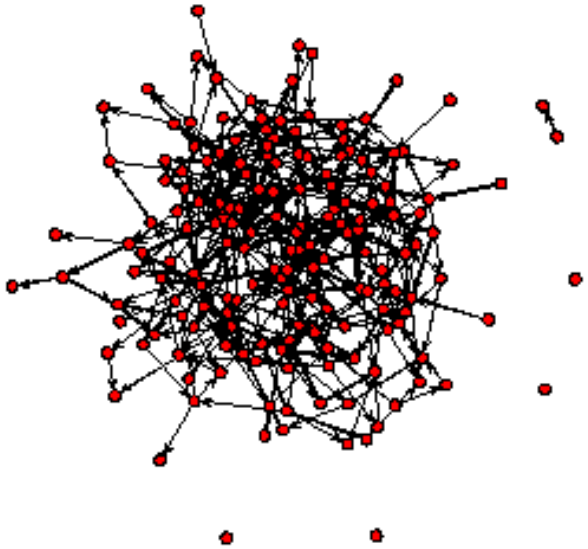

Figure 4.12: Simulated from Erdos-Renyi model

Similarly, we also simulate a model by using simple binomial density. We consider the density of our observed TF-TF network and also keep *175* nodes to keep resembles with our observed network. After simulating the network, we fit several model by ERGM to estimate the attributes of interest so that we can compare the estimate with the observed model. As before, the estimates are approximately similar irrespective of the

combinations or models. The summary of the estimates that we get under different models are presented in Table 4.11 (for both MCMC MLE and MPLE).

Table 4.11: Summary table of estimates from Binomial simulated model

Network Attributes	MCMC MLE Estimates	MPLE Estimates
Edges	-4.33561	-4.30465
Triangle	-0.06395	-0.08794
Ostar-5	-0.01032	-0.04404
Istar-5	-0.07059	-0.07491
Ostar-6	-0.10063	-0.25368
Istar-6	-0.31184	-0.29370

The simulated network using binomial density is represented in Figure 4.13.

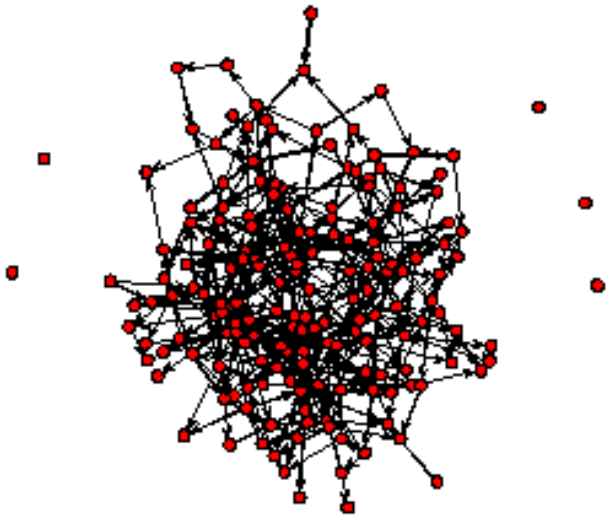

Figure 4.13: Simulated network using binomial probability

We also simulate two models by using ERGM package on R. There is a built in simulation comment in ERGM package on R to simulate networks. To do that, we first fit our observed TF-TF network model for edges, istar's ostar's and triangles and simulate two models using the fitted models. In our first model we consider edge, ostar-5s, istar-5s and triangles and in the other model we consider edge, ostar-6s, istar-6s and triangles.

Then we simulate two networks to estimate the attributes of our interest. We observe the similar pattern in the estimates that as long as we consider the same network, estimates of a certain attributes is always similar. Therefore, we get approximately same estimate for whatever combination we consider. That is why we try to present the estimates under a single table. The estimates of both MCMC MLE and MPLE method for this simulating scheme is presented in Table 4.12.

Table 4.12: Summary tables of estimates from fitted ERGM models

Network Attributes	MLE Estimates	MPLE Estimates
Edges	-5.3318479	-5.332e+00
Triangle	0.7194116	7.194e-01
Ostar-5	0.0001207	6.297e-06
Istar-5	0.0016440	1.644e-03
Ostar-6	1.484e-05	5.333e-07
Istar-6	-5.887e-02	-5.887e-02

Although some of the estimates we get under this fitted simulation scheme are very close, the physical numbers of statistics differ substantially. In this scheme, as the simulation scheme takes the fitted estimates into account, the physical number of different attributes should be close to the observed model. In particular, in biological simulation this is very important since the exact numbers of network statistics might have a significant influence on the overall process. In the next section, we physical counted these numbers of network attributes for different simulation models.

The simulated graph that we get from this scheme is presented in Figure 4.14.

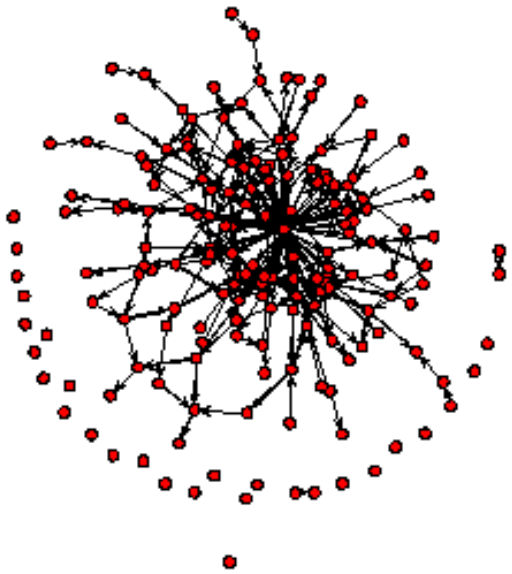

Figure 4.14: Simulated network from fitted ERGM model

4.7 Comparison of Network Attributes

In the previous sections, we simulated several networks by using our process and several other schemes. In Table 4.13 we represented the number of different attributes under different simulation schemes that we considered in sections 4.5 and 4.6.

Table 4.13: Summary table of estimates OBSERVED Vs SIMULATED

Network Attributes	Observed TF-TF network	Our Simulated network	Simulation using density	Erdos-Renyi simulation	ERGM fitted simulation
Edges	263	327	377	375	247
Triangle	114	115	12	9	82
Ostar-5	10	10	17	6	7
Istar-5	9	9	12	8	2
Ostar-6	10	10	6	2	1
Istar-6	8	8	3	1	3

From the table, we can say that the network attributes are different for all the simulation schemes. In our process as we are physically imposing the attributes, it is really close to the observer model. The only difference in the attributes is for the triangles which differ by just *1*. From the table above, we can say that, in terms of number, the ERGM simulated network is giving us close result. However, the numbers of triangles substantially differ from the original observed model. For the simple binomial simulation, we simulate network by considering the density from the observed model. However, the edges under this scheme do not even come close and the other attributes also significantly differ. We find similar characteristic for Erdos-Renyi modeling scheme. The reason behind this could be that both the binomial and Erdos-Renyi consider the density only while simulation. Thus, the number of attributes along with the edges are very close. Also in the ERGM simulation scheme, the other attributes such as istar-5s or ostar-5s are also not very close. In our random simulation, we emphasize on the number of attributes because a biological process is a very complicated process. A single edge might have significant influence over the entire process. Therefore, for biological simulation, we should always keep in mind the physical number of attributes that we are interested in.

Chapter 5

Further Directions and Conclusion

5.1 Further Directions

There are several ways to extend this study. In this study, an extensive simulation was conducted to explore the role and significance of network attributes in ERGM under several types of set-ups. While performing this simulation study, several issues arose that can be addressed in future. The first issue involved employing Bayesian method to see how the estimates differ from the MCMC MLE and MPLE methods. Moreover, ERGM cannot handle self loops. Therefore, inclusion of self loops may improve the estimates of ERGM. Specially, in biological networks the inclusion may have significant impacts on the estimates associated with overall biological processes. The other issue involved observations that some of the models did not converge using ERGM package. Exploration of the problem in convergence could potentially be accomplished using similar simulation techniques.

5.2 Conclusions

The number of commonly used network attributes such as istars, ostars and triangles in the TF-TF regulatory network of *E. coli* was determined. These networks attributes statistically serve as the significant local structures for the *E. coli* regulatory network. An observed regulatory network of the model organism *E. coli* was explored in terms of finding statistically local structure in this study. Simulation of two network models and comparison of the estimates of the observed and simulated models were made. In the first simulated model simulated with istar-5s, ostar-5s and triangles in the model, and in the second simulated model simulated with ostar-6s, istar-6s and triangles. In both cases, the estimates we obtained are very similar with the observed TF-TF network just except triangles. Networks were also simulated in other ways using existing methods were compared using these estimates as well. At the end, our models provide close results and same number of network attributes, which is very important in the biological data. In fact, in biology it is very important to keep the numbers the same because the numbers might have influence over the entire biological process. Therefore, it can concluded that at least for *E. coil*, the network can reproduced by taking the counts for different attributes, and the simulated network will behave as the observed network. To generalize this across species, the same techniques need to be applied across the species and in more complex organisms. If the technique works properly with higher order organisms, then the technique can be applied to generalize the ideas across the species.

Simulation of different networks with different number of nodes and network attributes were performed. The cut-off points were determined for a number of attributes at which

point specific attributes become significant to insignificant, or vice versa. It was observed that for smaller numbers of network attributes, the estimates usually become significant. If the number of attributes were increased in a given model, the attributes become insignificant, yet with very large numbers of attributes become significant again. Therefore, for a higher order of number of node, it becomes a difficult to determine with a number as the number of combinations get large. However, for those cases a number can be determined where the attributes become insignificant after a certain combination. Thus the attributes were considered separately just with edges and together in a combined model.

It was also observed that the models in ERGM do not always converge. This would be a desirable upgrade for the computational method and would allow for addressing the convergence issue. For the several models considered, convergence failure occurred while estimating parameters for any of the methods. For example, for our observed network, the model with edges, istar-4, ostar-4 and triangles did not converge. Also, due to the convergence issue, cut-off points could not be determined for several network attributes. In addition, if the network has self loops then the model did not converge. Therefore, while the ERGM shows promise, these issues remain in need of further analysis.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed Membership Stochastic Blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Anderson, C.J., Wasserman, S., and Crouch, B. (1999). A p* primer: logit models for social networks. *Social Networks*, 21:37–66.

Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X. (2006). Group Formation in Large Social Networks: Membership, Growth, and Evolution. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York: ACM Press, 44–54.

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

Begum, M., Bagga, J., and Blakey, C.A. (2012). Graphical Modeling for High Dimensional Data, *J Mod. Appl. Stat. Meth*, 11.

Besag, J.E. (1974). Spatial interaction and the statistical analysis of lattice systems. *J R Stat Soc Ser B Stat Methodol*, 36:192–225.

Besag, J. (1975). Statistical Analysis of non-lattice data, *Statistician*, 24: 179-195.

Besag, J. (1974). Spatial Interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36: 192-236.

Besag, J. (1977). Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrika*, 64:616-618.

Breiger, R., Boorman, S., and Arabie, P. (1975). An Algorithm for Clustering Relational DataWith Applications to Social Network Analysis and Comparison with Multidimensional Scaling. *Journal of Mathematical Psychology*, 12:328–383.

Chung, F., and Lu, L. (2006). *Complex Graphs and Networks*. Providence, RI: American Mathematical Society.

Dodds, P. S., Muhamad, R., and Watts,D. J. (2003). An Experimental Study of Search in Global Social Networks. *Science*, 301: 827–829.

Erdos, P., and Renyi, A. (1959). On random graphs. *Publicationes Mathematicae*, 6:290–297.

Erdos, P., and Renyi, A. (1960). On the evolution of random graphs. Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 5:17–61.

Eubank, S., Guclu, H., Anil Kumar, V.S., et al. (2004). Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180–184.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On Power-Law Relationships of the Internet Topology. *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '99)*, New York: ACM Press, 251–261.

Fienberg, S. E., and Lee, S. K. (1975). On Small World Statistics. *Psychometrika*, 40: 219–228.

Fienberg, S. E., Meyer, M. M., and Wasserman, S. S. (1985). Statistical Analysis of Multiple Sociometric Relations. *Journal of the American Statistical Association*, 80:51–67.

Fienberg, S.E. (2012). A Brief History of Statistical Models for Network Analysis and Open Challenges. *Journal of Computational and Graphical Statistics*, 21: 825-839.

Filkov, V. et al. (2002). Analysis techniques for microarray time-series data. *J Comput Boil*, 9:317-331.

Fronczak, A. (2012). Exponential Random Graph Models. Available from: http://arxiv.org/abs/1210.7828.

Frank, O., and Strauss, D. (1986). Markov graphs. *J Am Stat Assoc*, 81:832–842.

Gelman, A., Meng, X.L. (1998). Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling. *Statistical Science*. 13:163–185.

Geyer, C.J. (1991). Markov Chain Monte Carlo Maximum Likelihood. Computing Science and Statistics, *Proceedings of the 23rd Symposium on the Interface*, 156-163.

Geyer,C.J. and Thompson,E,A. (1992) Constrained monte carlo maximum likelihood for dependent data. J. *R. Stat. Soc. Ser. B*, 54:657–699.

Girvan,M., and Newman, M. E. J. (2002). Community Structure in Social and Biological Networks. *Proceedings of the National Academy of Sciences*, 99: 7821–7826.

Goodreau, S.M. (2007). Advances in Exponential Random Graph (*p**) Models Applied to a Large Social Network, *Social Networks,* 26: 231-248.

Handcook, M.S., Hunter, D.R., Butts, C.T., Goodreau, S.M., and Morris, M. (2003). statnet: Software tools for the Statistical Modeling of Network Data. Avaialable from: http://statnetproject.org.

He, F., and Zeng, A. P. (2006). In search of functional association from time-series microarray data based on the change trend and level of gene expression. *BMC Bioinformatics*, 7:69-84.

Holland, P.W. and Leinhardt, S. (1970). A method for detecting structure in sociometric data. *American Journal of Sociology*, 70: 492–513.

Holland, P.W., and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *J Amer Stat Assoc*, 76:33–50.

Hunter, D.R., Handcook, M.S., Butts, C.T., Goodreau, S.M. and Morris, M. (2008). ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks. *J Stat Softw*, 24(3): nihpa54860.

Hunter, D.R., Handcock, M.S. (2006). Inference in CurvedExponential Family Models for Networks. *Journal of Computational and Graphical Statistics*. 15:565-583.

Husmeier, D. (2003). Sensitivity and Specificity of Inferring Genetic Regulatory Interactions from Microarray Experiments with Dynamic Bayesian Networks. *Bioinformatics*, 19:2271-2282.

Jacob, F., and Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3: 318–356.

Jeong, H., Oltvai, Z., and Barabasi, A. (2003). Prediction of protein essentiality based on genomic data. *ComPlexUs*, 1:19–28.

Keedwell, E., and Narayanan, A. (2005). Discovering Gene Networks with a Neural-Genetic Hybrid. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2:231-242.

Keseler, et al. (2005). EcoCyc: a comprehensive database resource for Escherichia coli. *Nucleic Acids Research*, 33:334-337.

Kleinberg, J. M. (2000a). Navigation in a Small World—It Is Easier to Find Short Chains Between Points in Some Networks Than Others. *Nature*, 406, 845.

Kleinberg, J. M. (2000b). The Small-World Phenomenon: An Algorithmic Perspective. *Proceedings of the 32nd ACM Symposium on Theory of Computing*, New York: ACM Press, 163–170.

Kleinberg, J. M. (2001). Small-World Phenomena and the Dynamics of Information. *Advances in Neural Information Processing Systems (NIPS)*, 14:431–438, Cambridge, MA: MIT Press.

Lazega, E., and van Duijn, M. (1997). Position in Formal Structure, Personal Characteristics and Choices of Advisors in a Law Firm: A Logistic Regression Model for Dyadic Network Data. *Social Networks*, 19:375–397.

Lee, T. et al. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, 298:799-804.

Lesne, A. (2006). Complex Networks: from Graph Theory to Biology. *Letters in Mathematical Physics*, 78:235–262.

Li, X. et al. (2006). Wand QK: Discovery of time-delayed gene regulatory networks based on temporal gene expression profiling. *BMC Bioinformatics*, 7:26-46.

Mason, O., and Verwoerd, M. (2007). Graph theory and networks in Biology. *IET Syst. Biol.*, 1:89–119.

Meng, X.L., and Wong, W.H. (1996). Simulating Ratios of Normalizing Constants via a Simple Identity: A Theoretical Exploration. *Statistica Sinica*, 6:831–860.

Milo, R., Shen-Orr, S., Itzkovitzl, S. et al. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298: 824–827.

Milgram, S. (1967). The Small World Problem. *Psychology Today*, 1:60–67.

Milenkovic, T. (2008). *Graph-theoretical approaches for studying biological networks*. Department of Computer Science. University of California, Irvine. Available from: http://www.doc.ic.ac.uk/~natasha/course/docs/SURVEY_FINAL.pdf

Morris, M., Handcock, M.S. and Hunter, D.R. (2008). Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects, *Journal of Statistical Software*, 24: 1-24.

Pattison, P.E., and Wasserman, S. (1999). Logit models and logistic regressions for social networks: II. Multivariate relations. *British Journal of Mathematical and Statistical Psychology*, 52, 169–194.

Pavlopoulos, G.A., Secrier, M., Moschopoulos, C.N., et al. (2011). Using graph theory to analyze biological networks. *BioData Min*, 4: 10.

Przulj, N., Corneil, D.G. and Jurisica, I. (2004). Modeling interactome: scale-free or geometric? *Bioinformatics*, 20:3508–3515.

Qian, J. et al. (2001). Beyond synexpression relationships: local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. *J Mol Biol*, 314:1053-1066.

Robins, G.L., Pattison, P.E., and Wasserman, S., (1999). Logit models and logistic regressions for social networks: III. Valued relations. *Psychometrika* 64, 371–394.

Robins, G.L., Pattison, P.E., Kalish, Y. and Lusher, D. (2007a) An introduction to exponential random graph (*p**) models for social networks. *Social Networks*, 29: 173–191.

Robins, G., Snijders, T., Wang, P., Handcock, M., Pattison, P. (2007b). Recent developments in exponential random graph (p*) modelsfor social networks. *Soc Networks*. 29:192-215.

Salgado, H. et al. (2006a). The comprehensive updated regulatory network of Escherichia coli K-12. *BMC Bioinformatics*, 7:5.

Salgado, H. et al. (2006b). RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, 34: 394-397.

Sampson, F. S. (1968). *A Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships*, Ph.D. thesis, Cornell University.

Samanta, M., and Liang, S. (2003). Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc. Nat. Acad. Sci.,* 100:12579–12583.

Saul, Z.M., and Filkov, V. (2007). Exploring biological network structure using exponential random graph models. *Bioinformatics*. 23:2604-2611.

Schnitzler, A., and Gross, J. (2005). Normal and pathological oscillatory communication in the brain. *Nat. Rev. Neurosci*., 6:285–295.

Shmulevich, et al. (2002). From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks. *Proceedings of the IEEE*, 90:1778-1790.

Snijders, T.A.B. (2001). The statistical evaluation of social network dynamics. *Sociological Methodology*, 31, 361–395.

Snijders, T.A.B. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure*, 3, 2:1-40.

Snijders, T.A.B., Pattison, P.E., Robins, G.L., and Handock, M. (2006). New specifications for exponential random graph models. *Sociological Methodology*, 36:99-153.

Solomonoff, R., and Rapoport, A. (1951). Connectivity of random nets. *B Math Biophys*, 13:107–117.

Strauss, D. (1986). On a general class of models for interaction. *SIAM Rev*, 28:513–527.

Strauss, D., and Ikeda, M. (1990). Pseudolikelihood estimation for social networks. Alexandria, VA, ETATS-UNIS: American Statistical Association.

Travers, J., and Milgram, S. (1969). An Experimental Study of the Small World Problem. *Sociometry*, 32:425–443.

Wasserman, S., and Pattison, P.E. (1996). Logit Models and Logistic Regressions for Social Networks: I. An Introduction to Markov Graphs and *p**. *Psychometrika*, 61:401-425.

Wasserman, S., Robins, G.L. (2005). An introduction to random graphs, dependence graphs, and *p**. In: Carrington, P., Scott, J., and Wasserman, S. (Eds.), *Models and Methods in Social Network Analysis*. Cambridge University Press, New York, 148–161.

Watts, D. J., and Strogatz, S. H. (1998). Collective Dynamics of 'Small-World' Networks. *Nature*, 393:440–442.

White, H. C. (1970). Search Parameters for the Small World Problem. *Social Forces*, 49:259–264.

White, H. C., Boorman, S. A., and Breiger,R.L. (1976). Social Structure From Multiple Networks. I. Blockmodels of Roles and Positions. *The American Journal of Sociology*, 81:730–780.

Zachary, W. W. (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33:452–473.

Zou, M., and Conzen, S. D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21:71- 79.

Appendices

```
################################################

# counting ostar-3 in a given square martix                    #

################################################



countostarr<-function(M,r)

{

  countvec<-0

  for (i in 1:nrow(M))

  {

    ifelse (sum(M[i,])>=r, countvec<-countvec+1,countvec<-countvec+0)

  }
```

```
  countvec
}

################################################
# counting istar-3 in a given square martix                    #
################################################

countistarr<-function(M,r)
{
  countvec<-0
  for (i in 1:nrow(M))
  {
    ifelse (sum(M[,i])>=r, countvec<-countvec+1,countvec<-countvec+0)
  }
  countvec
}
```

```
#####################################################
# counting ctriple in a given square martix         #
#####################################################
contctriple<-function(M){
  count<-0
  for (i in 1:ncol(M)){
    for (j in 1:ncol(M)){
      if (M[i,j]==1) {
        for (k in 1:ncol(M)){
          if (M[j,k]==1 & M[k,i]==1){count<-count+1}
        }
      }
    }
  }
  count/3
```

```
}

##################################################
# counting ttriple in a given square martix      #
##################################################
conttriple<-function(M){
  count<-0
  for (i in 1:ncol(M)){
    for (j in 1:ncol(M)){
      if (M[i,j]==1) {
        for (k in 1:ncol(M)){
          if (M[j,k]==1 & M[i,k]==1){count<-count+1}
        }
```

```
    }
   }
  }
  count
}
```

```
#####################################################################
# counting triangle in a given square martix                        #
#####################################################################
tri<-function(M){
  count<-0
  count1<-0
  for (i in 1:ncol(M)){
    for (j in 1:ncol(M)){
```

```
      if (M[i,j]==1) {
        for (k in 1:ncol(M)){
          if (M[j,k]==1 & M[k,i]==1){count<-count+1}
          if (M[j,k]==1 & M[i,k]==1){count1<-count1+1}
        }
      }
    }
  }
  count/3+count1
}
```

```
#################################################################
#Assigning random number of ostar-r in a n*n adjence matrix     #
#################################################################
ass.ostr.null<-function(r,n,m){
```

```
  Adj.Mat<-matrix(0,n,n)
  for(i in sample(1:n,m)){
    done<-FALSE
    while(done==FALSE){
      sam<-sample(1:n,r)
      ifelse(any(sam==i),done<-FALSE, done<-TRUE)
    }
    Adj.Mat[i,sam]<-1
  }
  Adj.Mat
}
```

```
#########################################################################
#Assigning random number of ostar-r in a specific n*n adjence matrix     #
#########################################################################
assig.ran.ostarr<-function(r,M,m){
  for(i in sample(1:nrow(M),m)){
    done<-FALSE
    while(done==FALSE){
      sam<-sample(1:nrow(M),r)
      ifelse(any(sam==i),done<-FALSE, done<-TRUE)
    }
    M[i,sam]<-1
  }
  M
}
```

```
#################################################################

#Assigning random number of ctriple in a n*n adjence matrix#

#################################################################

library(combinat)

ass.ctripl<-function(n,m){

  Adj.Mat<-matrix(0,n,n)

  posibl<-t(combn(n,3))

  s<-posibl[sample(nrow(posibl),m),]

  for(i in 1:nrow(s)){

    sam<-sample(3,3)

    Adj.Mat[s[i,sam[1]],s[i,sam[2]]]<-1

    Adj.Mat[s[i,sam[2]],s[i,sam[3]]]<-1

    Adj.Mat[s[i,sam[3]],s[i,sam[1]]]<-1

  }

  Adj.Mat

}
```

```
#################################################################
#Assigning random number of ttriple in a n*n adjence matrix      #
#################################################################
library(combinat)
ass.ttrip<-function(n,m){
  Adj. Mat<-matrix(0,n,n)
  posibl<-t(combn(n,3))
  s<-posibl[sample(nrow(posibl),m),]
  for(i in 1:nrow(s)){
    sam<-sample(3,3)
    Adj.Mat[s[i,sam[1]],s[i,sam[2]]]<-1
    Adj.Mat[s[i,sam[2]],s[i,sam[3]]]<-1
    Adj.Mat[s[i,sam[1]],s[i,sam[3]]]<-1
  }
  Adj.Mat
}
```

```
#################################################################
#Assigning random number of triangle in a n*n adjence matrix     #
#################################################################
library(combinat)
ass.try.null<-function(n,m){
  Adj.Mat<-matrix(0,n,n)
  posibl<-t(combn(n,3))
  s<-posibl[sample(nrow(posibl),m),]
  for(i in 1:nrow(s)){
    sam<-sample(3,3)
    Adj.Mat[s[i,sam[1]],s[i,sam[2]]]<-1
    Adj.Mat[s[i,sam[2]],s[i,sam[3]]]<-1
    sam1<-sample(c(1,3),2)
    Adj.Mat[s[i,sam[sam1[1]]],s[i,sam[sam1[2]]]]<-1
  }
```

```
  Adj.Mat
}
#########################################################################
#Assigning random number of triangle in a specific n*n adjence matrix                  #
#########################################################################
library(combinat)
ass.try<-function(M,m){
  posibl<-t(combn(nrow(M),3))
  s<-posibl[sample(nrow(posibl),m),]
  for(i in 1:nrow(s)){
    sam<-sample(3,3)
    M[s[i,sam[1]],s[i,sam[2]]]<-1
    M[s[i,sam[2]],s[i,sam[3]]]<-1
    sam1<-sample(c(1,3),2)
    M[s[i,sam[sam1[1]]],s[i,sam[sam1[2]]]]<-1
  }
```

```
  M
}
#########################################################################
#counting commom triangle with combination so that we can figure the no. has          #
#   1>2>3 and 3>2>1 both                                                               #
#########################################################################
library(combinat)
tri.comn<-function(M){
  count<-0
  posibl<-t(combn(nrow(M),3))
  for (i in 1:nrow(posibl)){
    if (M[posibl[i,1],posibl[i,2]]==1 & M[posibl[i,2],posibl[i,3]]==1 &
M[posibl[i,3],posibl[i,1]]==1) {
      if (M[posibl[i,1],posibl[i,3]]==1 & M[posibl[i,3],posibl[i,2]]==1 &
M[posibl[i,2],posibl[i,1]]==1){count<-count+1}
    }
  }
```

```
  count
}
############################################################
# assigning istar-r in a square martix of n*n dimension    #
############################################################
as.istr.null<-function(r,n,m){
  Adj.Mat<-matrix(0,n,n)
  for(i in sample(1:n,m)){
    done<-FALSE
    while(done==FALSE){
      sam<-sample(1:n,r)
      ifelse(any(sam==i),done<-FALSE, done<-TRUE)
    }
    Adj.Mat[sam,i]<-1
  }
  Adj.Mat
```

```
}

#########################################################################

# assigning istar-r in a specific square martix of n*n dimension                #

#########################################################################

as.ran.istr<-function(r,M,m){

  for(i in sample(1:nrow(M),m)){

    done<-FALSE

    while(done==FALSE){

      sam<-sample(1:nrow(M),r)

      ifelse(any(sam==i),done<-FALSE, done<-TRUE)

    }

    M[sam,i]<-1

  }

  M

}
```

```
#################################################################

#Assigning the statistics altogether n*n adjence matrix                    #

#################################################################


altogether<-function(r,n,os,is,tr){        # here n is the dimesion, r is the ostar-r or istar-r
  Adj.Mat<-matrix(0,n,n)                  # os is the # of ostar-r, is is the # of istar-r and tr is
3 of triangle
  while (countistarr(Adj.Mat,r)!=is & countostarr(Adj.Mat,r)!=os & tri(Adj.Mat)!=tr){
    for(i in sample(1:n,os)){
    done<-FALSE
    while(done==FALSE){
      sam<-sample(1:n,r)
      ifelse(any(sam==i),done<-FALSE, done<-TRUE)  # this do loop is to create ostar-r
    }
    Adj.Mat[i,sam]<-1
```

```
}
Adj.Mat
for(j in sample(1:n,is)){
  done<-FALSE
  while(done==FALSE){
    sam<-sample(1:n,r)
    ifelse(any(sam==j),done<-FALSE, done<-TRUE)    # this do loop is to create istar-r
  }
  Adj.Mat[sam,j]<-1
}
Adj.Mat
posibl<-t(combn(n,3))
s<-posibl[sample(nrow(posibl),tr),]
for(k in 1:nrow(s)){
  sam<-sample(3,3)                          # this do loop is to create triangles
  Adj.Mat[s[k,sam[1]],s[k,sam[2]]]<-1
```

```
    Adj.Mat[s[k,sam[2]],s[k,sam[3]]]<-1
    sam1<-sample(c(1,3),2)
    Adj.Mat[s[k,sam[sam1[1]]],s[k,sam[sam1[2]]]]<-1
  }
  Adj.Mat
}
Adj.Mat
}

aa<-altogether(2,20,1,1,2)
tri(aa)
```