



VCU

Virginia Commonwealth University
VCU Scholars Compass

Theses and Dissertations

Graduate School

2013

COMPLEX NETWORK GROWING MODEL USING DOWNLINK MOTIFS

Ahmad Al-Musawi Jr.
Virginia Commonwealth University

Follow this and additional works at: <https://scholarscompass.vcu.edu/etd>



Part of the [Computer Sciences Commons](#)

© The Author

Downloaded from

<https://scholarscompass.vcu.edu/etd/3088>

This Thesis is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.

School of Engineering
Virginia Commonwealth University

This is to certify that the thesis prepared by Ahmad F. Al-Musawi entitled COMPLEX NETWORK GROWING MODEL USING DOWNLINK MOTIFS has been approved by his committee as satisfactory completion of the thesis requirement for the degree of Master of Science in Computer Science.

Dr. Preetam Ghosh, Computer Science, School of Engineering

Dr. Krzysztof Cios, Computer Science, School of Engineering.

Dr. Ramana M. Pidaparti, Mechanical and Nuclear Engineering, School of Engineering.

Date: _____

© Ahmad F. Al-Musawi 2013

All Right Reserved

COMPLEX NETWORK GROWING MODEL USING DOWNLINK MOTIFS

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of
Science in Computer Science at Virginia Commonwealth University

By

AHMAD F. AL-MUSAWI

B.S., Computer Science, Thi Qar University, Iraq, 2005

Director: Dr. PREETAM GHOSH
Assistant Professor of Computer Science, School of Engineering

Virginia Commonwealth University
Richmond, Virginia
May 2013

Dedication

I dedicate my thesis to my lovely wife, Noor, for all her support and encouragement, my daughters Sara and Shahad for the funny times, and to my best teachers, my parents.

I also dedicate this work to Dr.Zuhair Humadi, the head of the higher committee for education development in Iraq.

Acknowledgement

I would like to thank my advisor Dr.Preetam Ghosh for all his assistance and insights in my research work. A special thanks to my committee members, Dr. Krzysztof Cios and Dr.Ramana M. Pidaparti. I also want to thank Ahmed Abdelzaher for his precious help and comments. I am grateful to my colleagues Dr.Vijender Chaitanker, Joseph Nalluri, Bhanu Kamapantula and Murtedha Al-Maliki for their assistance, support and friendship.

Table of Contents

	Page
Dedication.....	ii
Acknowledgement.....	iii
Table of contents.....	iv
List of figures.....	vii
List of tables.....	ix
Abstract.....	xi
1 Chapter One Introduction.....	1
2 Chapter Two Literature Review.....	4
3 Chapter Three Preliminary and advance network.....	7
3.1. Identifying the problem.....	7
3.2. Network and its adjacency matrix.....	7
3.3. Node indexing and organizing for both goal and substrate network.....	8
3.4. Motifs and downlinks.....	10
3.5. Downlink List.....	12
3.6. Vertex sharing motif network (VMN).....	17

4	Chapter Four Downlink Based Network Statistics	19
4.1.	Downlink-downlink combination.....	19
4.2.	Downlink Attachment	21
4.3.	Patterns of downlink attachment	22
4.3.1.	One shared node downlink attachment	22
4.3.1.1.	Father to father	23
4.3.1.2.	Father to son.....	23
4.3.1.3.	Son to father.....	25
4.3.1.4.	Son to son.....	26
4.3.2.	Two shared nodes downlink attachment	27
4.3.2.1.	Father- son to father son	27
4.3.2.2.	Father- son to son-father	28
4.3.2.3.	Father- son to son-son	29
4.3.2.4.	Son- son to father- son	30
4.3.2.5.	Son- son to son- son.....	31
4.3.3.	Three shared nodes downlink attachment	32
4.3.3.1.	Father- son- son to son- father- son	32

4.4. Sets of applicable patterns in each downlink-downlink combination	.36
5 Chapter five Downlink Based Substrate Network Growing Model37
5.1. The probability distribution38
5.2. The selection of downlink to attach to39
5.3. Degree centrality41
5.4. The selection of downlink to be added42
5.5. Attaching the new downlink using the different patterns44
5.6. Different strategies in growing the network45
6 Chapter Six Results and Discussion48
6.1. Maximum likelihood estimation of cumulative distribution functions	48
7 Chapter Seven Growing the network using correct downlink attachment in a game based approach54
8 Chapter Eight Conclusions and further work60
8.1. Identifying the best initial substrate network structure to be grew60
8.2. Identify the most common patterns of attachment using LCS61
8.3. Grow the network using different set of downlink centralities61
References62

List of Figures

	Page
Figure (2-1) all 13 possible 13 nodes subgraphs.....	5
Figure (2-2) feed-forward loop (FFL) and Bi-fan (BF) motif structures	6
Figure (3-1) Schematic of a downlink: father node and two sons with two edges only	10
Figure (3-2) Schematic of possible types of downlink (tgg, ttg, ttt).....	11
Figure (3-3) Sample regulatory network	13
Figure (4-1) one node attachment, pattern one (P1).....	23
Figure (4-2) one node attachment, pattern two (P2).....	24
Figure (4-3) one node attachment, pattern three (P3).....	25
Figure (4-4) one node attachment, pattern three (P3).....	26
Figure (4-5) two nodes attachment, pattern one (P1).....	28
Figure (4-6) two nodes attachment, pattern two (P2).....	29
Figure (4-7) two nodes attachment, pattern three (P3).....	30
Figure (4-8) two nodes attachment, pattern three (P3).....	31
Figure (4-9) two nodes attachment, pattern four (P4).....	32
Figure (4-10) three nodes attachment, pattern one (P1).....	33
Figure (6-1) MLE difference for in degree distributions.....	52

Figure (6-2) MLE difference for motif distributions.....	52
Figure (6-3) MLE difference for out degree distributions.....	53
Figure (6-4) MLE difference for total degree distributions.....	53
Figure (7-1) Gaming difficulty dialogue box.....	55
Figure (7-2) the basic interface of the game.....	56
Figure (7-3) Selected t node with the applicable downlinks	56
Figure (7-4) the substrate network after adding several downlinks.....	57

List of Tables

	Page
Table (3-1) Nodes with outdegree ≥ 2 and corresponding number of downlinks	14
Table (3-2) List of downlinks from the network in figure (3-3).....	15
Table (4-1) downlink-downlink combination shares one node only.....	19
Table (4-2) downlink-downlink combination shares two nodes only.....	20
Table (4-3) downlink-downlink combination shares three nodes only.....	20
Table (4-4) all applicable downlink-downlink combination in one node attachment- pattern one.....	23
Table (4-5) all applicable downlink-downlink combination in one node attachment- pattern two.....	25
Table (4-6) all applicable downlink-downlink combination in one node attachment- pattern three.....	26
Table (4-7) all applicable downlink-downlink combination in one node attachment- pattern three.....	27
Table (4-8) all applicable downlink-downlink combination in two nodes attachment- pattern one.....	28

Table (4-9) all applicable downlink-downlink combination in two nodes attachment - pattern two.....	29
Table (4-10) all applicable downlink-downlink combination in two nodes attachment - pattern three.....	30
Table (4-11) all applicable downlink-downlink combination in two nodes attachment - pattern three.....	31
Table (4-12) all applicable downlink-downlink combination in two nodes attachment - pattern four.....	32
Table (4-13) all applicable downlink-downlink combination in three nodes attachment- pattern one.....	34
Table (4-14) all applicable downlink-downlink combination in each pattern.....	35
Table (4-15) applicable patterns for different downlink-downlink combinations.....	36
Table (6-1) MLE based comparison for different degree distributions.....	49
Table (6-2) MLE based motif participation distribution.....	51
Table (7-1) different modes of attachment's characters.....	59

Abstract

Understanding the underlying architecture of gene regulatory networks (GRNs) has been one of the major goals in systems biology and bioinformatics as it can provide insights in disease dynamics and drug development. Such GRNs are characterized by their scale-free degree distributions and existence of network motifs, which are small subgraphs of specific types and appear more abundantly in GRNs than in other randomized networks. In fact, such motifs are considered to be the building blocks of GRNs (and other complex networks) and they help achieve the underlying robustness demonstrated by most biological networks.

The goal of this thesis is to design biological network (specifically, GRN) growing models. As the motif distribution in networks grown using preferential attachment based algorithms do not match that of the GRNs seen in model organisms like *E. coli* and yeast, we hypothesize that such models at a single node level may not properly reproduce the observed degree and motif distributions of biological networks. Hence, we propose a new network growing algorithm wherein the central idea is to grow the network one motif (specifically, we consider one downlink motif) at a time. The accuracy of our proposed algorithm was evaluated extensively and show much better performance than existing network growing models both in terms of degree and motif distributions.

We also propose a complex network growing game that can identify important strategies behind motif interactions by exploiting human (i.e., gamer) intelligence. Our proposed gaming software can also help in educational purposes specifically designed for complex network studies.

Chapter One Introduction

Understanding the underlying architecture of gene regulatory networks (GRNs) has been one of the major goals in systems biology and bioinformatics as it can provide insights in disease dynamics and drug development ([1], [2]). As with many engineered networks like wireless networks [3, 44-46] and airline networks [4], GRNs are represented by graphs composed of a set of nodes and links connecting them together. Here the nodes signify the genes in a cell, and a set of directed links that correspond to interacting pairs of genes [5] representing the biological processes of translation and transcription [6]. Such GRNs exhibit a unique property- the phenomenon of "biological robustness" ([7], [8]) which allows genes to adapt and recover from disturbances in gene expression [9].

Gene expression robustness arise from the feed-back control nodal arrangements and other repetitive substructures [10] found in the GRN topology; in this regard GRN robustness is attributed to recently discovered specialized substructures in GRNs, termed as "network motifs" [11]. Motifs are considered to be the building blocks of many complex networks [12](including GRNs) as they appear more commonly in their topologies than would be anticipated in randomized networks [12] having the same number of nodes and links, as well as similar degree distributions although different overall topologies. Though much consideration has been focused toward unfolding their individual purposes theoretically [13] and experimentally [14], little is understood on their coupling in relation to their natural evolution.

In [12], the authors list the types of all 3-6 node motif structures. Among the most common motifs of the GRN of model prokaryotic bacteria such as *Escherichia coli* (herein E.

coli) and *Saccharomyces cerevisiae* (herein Yeast), are Feed-forward loops (FFLs) and Bifans (BFs), shown in Figure 2-2. In genetic networks gene A can intensify or decelerate the production of specific enzymes in gene B, if a link is projected from A to B, subsequently the terms up- and down-regulation are used to express these processes. An FFL is composed of three genes, the father gene regulates a left and a right child gene, while the left child regulates the right- a topology which allows FFLs to generate pulses, signal delays and irreversible speedups [13]. BF's constitute four genes, two of which simultaneously regulating the other two and are known to be the constituents of the dense overlapping regulons in the GRN's backbone which is responsible for conducting vital functions such as nutrient metabolism and bio-synthesis of essential branches of cellular elements [15].

It is notable to point out that all motifs are a product of the coupling between one or two and/or a mixture of one of the three two edge motif substructures presented in Figure 2; the uplink, the downlink and the three chain. For instance a BF can be viewed as two downlinks coupled by sharing both child genes and an FFL can be viewed as an uplink or a downlink sharing all three genes with a three chain. Moreover, we have conducted computational analysis to estimate the percentages of the genetic interactions covered by the mentioned structures in the GRN of *E. coli* and we observed that 54.7% of them are covered by FFLs, 82% by BF's, 99.4% by downlinks, 83.9% by uplinks and 78.3% by three chains. Considering the above observations, we hypothesize that downlinks are the most notable substructures in terms of describing the evolution GRNs due to gene coupling.

Despite the fact that the impact of motif coupling on the all-embracing function of the GRN is still a mystery, several researchers are focused in this particular area. For example, investigations of gene coupling for different motif patterns have been conducted using

mathematical modeling of transcription and translation in order to reveal substructure functionalities ([16], [17], [14]). Comprehensive experimental research have been conducted to reveal the extent at which the bacteria can endure the switching (or rewiring) of gene promoters [18]. Moreover, authors in [19] considered the occurrence of individual genes in FFL substructures of E. coli and compared these node-motif distributions with networks grown one node at a time using preferential attachment based linear and non-linear attachment kernels.

The goal of this thesis is to design biological network (specifically, GRN) growing models. As the motif distribution in networks grown using the algorithm proposed in [19] did not match that of the original E. coli GRN, we hypothesize that preferential attachment models at a single node level may not properly reproduce the observed degree and motif distributions of biological networks. Hence, we propose a new network growing algorithm wherein the central idea is to grow the network one motif (specifically, we consider one downlink motif) at a time.

Chapter Two Literature Review

A network of interconnected entities, generally termed vertices or nodes, and a set of relationship or connections among these nodes, can represent different categorical relationships in different fields such as the World Wide Web, social networks, citation networks [21], ecological networks [22], phone call networks [23], wireless sensor networks and many others. A prime example would be a social network having vertices as personal profiles, and interconnections determining profiles which are allowed to access one another (or “Friend” relationship).

In the field of biology, many biochemical networks that describe the molecular-level patterns of interaction [24] can be formed such as protein-protein interaction (PPI) networks [25], metabolic and biochemical networks [26], signal transduction network and genetic regulatory networks (GRN). Biological networks, specifically GRNs, have evolved to adapt with environmental changes [27] that was attributed to the existence of repeated patterns of small structures known as motifs [28] that are considered to be the building blocks of complex networks [12]. Basically, GRNs were scanned for different patterns of subnetworks with n -nodes.

The occurrences of n-node patterns in a GRN are compared with the occurrences of the same patterns in randomized networks of the same size and having the same degree sequences. If the pattern abundance difference between the GRN and the randomized networks is relatively high compared to other pattern abundance differences, then this pattern is labeled as a motif.

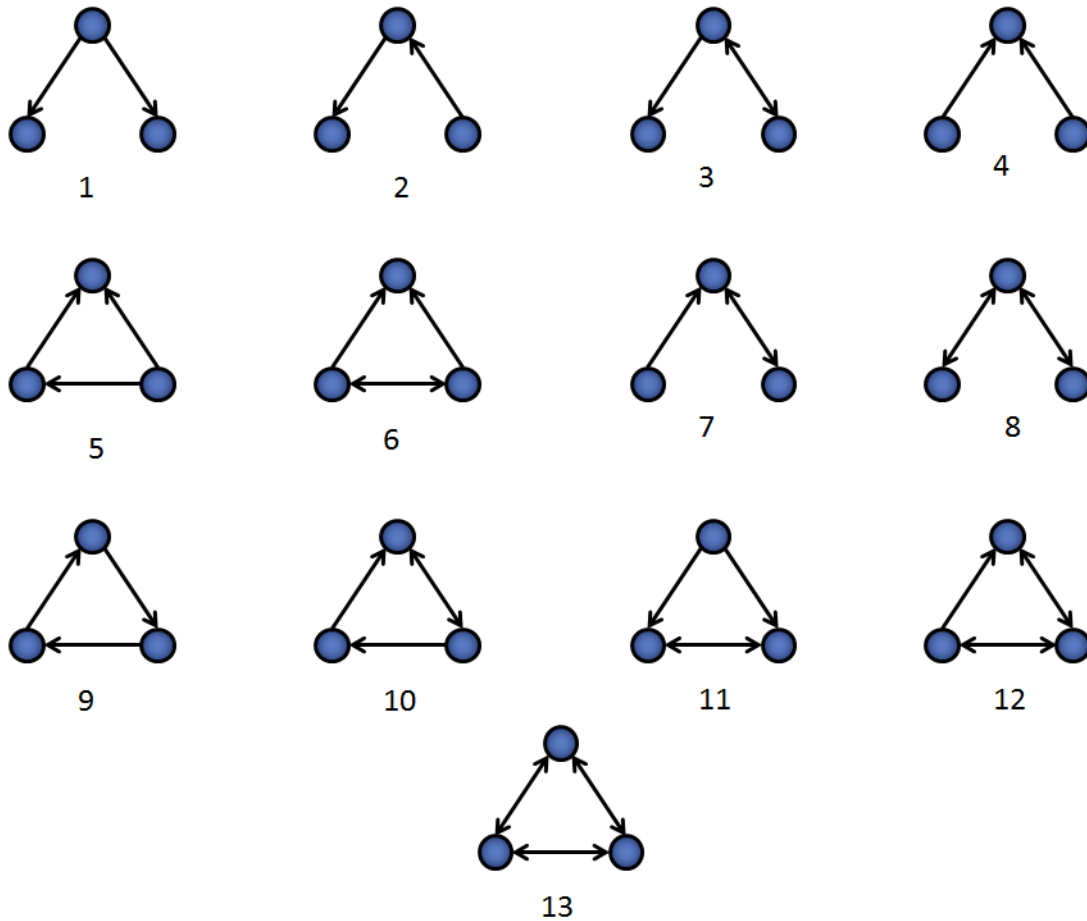


Figure (2-1) all 13 possible 3 nodes subgraphs.

Several theoretical [13] and experimental [14] studies were conducted to relate the influence of these motifs to network robustness. Figure (2-1) shows all possible patterns for three node motifs observed in *E. coli*. In [12], different kinds of motifs were counted within different types of networks in order to understand the functionality produced due to coupling of these

repeated structures. It was observed that GRNs demonstrate the Feed-Forward loop (herein FFL) and the Bi-fan (herein BF) structures as their primary 3 and 4 node motifs respectively; These motifs are depicted in Figure (2-2).

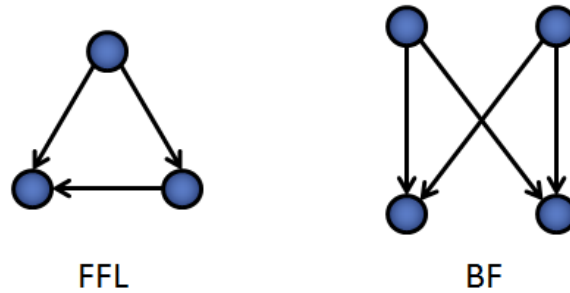


Figure (2-2) Feed-forward loop (FFL) and Bi-fan (BF) motif structures

Hence, a central goal of bioinformatics research is to design models and algorithms for robust network design that can preserve both the degree and motif distributions of biological networks. Several dynamic models known as generative networks models [24]) were designed to grow complex networks [29]. It starts with a specific set of nodes and edges and are grown adding nodes and edges to the network using defined rules of attachments. The grown networks are validated to sample biological networks specifically in terms of their degree sequences that should follow a power-law distribution, having few nodes with very high degrees compared to the average degree, and many nodes having very low degrees when compared with the average. The most popular algorithm in this area is the preferential attachment model [30] which is well known to simulate the phenomenon “The rich get richer and the poor get poorer”. However, the preferential attachment model can only create undirected networks; authors in [19] proposed a modified preferential attachment algorithm using 3 attachment kernels to design directed biological network growing models.

Chapter Three Preliminary and Advance Network Structures

3.1. Identifying the problem

The network growing algorithm designed in [19] follows the preferential attachment model where one node is added to the substrate network at a time to create directed GRNs of arbitrary size. While this algorithm showed very good correspondence with the *E. coli* GRN in terms of the degree distribution, they perform poorly on comparing the feed-forward loop motif distributions of these networks. Such motif distributions were computed by counting the number of nodes that participate in 1 feed-forward loop only; 2 feed-forward loops only and so on. This motivates the following question: do biological networks grow one node at a time? As motifs are central to their functionality and robustness and work as a single unit (or building block), we hypothesize that GRNs actually evolved (or grew in size) at a motif level. To test this hypothesis, we restrict ourselves to only downlink motifs (that cover most of the nodes and edges in the *E. coli* GRN by itself) in this thesis. Our goal is to design a preferential downlink attachment algorithm which also considers the node types (genes and transcription factor nodes) to study if that can preserve both the degree and motif distribution in the grown networks. Hence this work is just a first step towards motif based network growing models which can be perfected if one considers the other structures as well moving forward.

3.2. Networks and its adjacency matrix:

A network or graph G is defined as a pair of (V, E) where V refers to the non-empty set of vertices or nodes that participate in the network and E refers to the non-empty set of edges between nodes. An edge is presented as a combination of two nodes. There are two types of

network: directed and undirected network. The type of the edge is specified by the kind of the relationship between a pair of nodes. In this work, a biological network is used where genes and regulations represent the nodes and edges.

An adjacency matrix is used to represent the network. An adjacency matrix $A(n, n)$ is a two dimension square array with n number of columns and rows, wherein, the matrix element $A(i, j)$ would signify the edge going from the node i to the node j . In undirected network, the matrix element $A(i, j)$ would represent an edge between i and j without direction. The network adjacency matrix may contain Boolean or integer values depending on the usage of the network. Boolean adjacency matrix would be used to reflect the existence of unweighted edges between nodes. Integer adjacency matrix would be used to reflect the existence of an edge between the two related nodes with a weight specified.

$$A(n, n) = \begin{bmatrix} 1,1 & \cdots & 1,n \\ \vdots & \ddots & \vdots \\ n,1 & \cdots & n,n \end{bmatrix}$$

3.3. Node indexing and organizing for both goal and substrate network.

Here we define the goal network as the actual biological network that the algorithm will try to predict by starting from a smaller substrate network. All our goal (or reference) networks were extracted from an online scientific application called 43, which is a program that provides the transcriptional regulatory networks structure of the Escherichia Coli (herein E. coli) and Yeast organisms and options for extracting subgraphs of specific sizes.

Every gene node extracted from the GRN of E. coli has three properties that define their unique genetic representations. These properties are:

- Unique numeric identifier
- Unique string identifier
- Node type (transcription factor “T” or gene “G”).

We created a vertex structure such that it has the same characteristics as shown in the code listing below:

Define Structure *vertex*

Let *Numeric_ID* As Integer

Let *String_ID* As String

Let *Node_Type* As Boolean // True for Transcription factor and False for genes

End Structure

Note that only the goal network will use all vertex components. All other grown networks’ nodes in this study would only use the *Numeric_ID* and *Node_Type* components, since unspecified downlink nodes are added to the substrate network. When the program loads the goal network, we store each node using its unique identifiers in *Nodes^g* designating the goal network nodes array. The variable *number_of_nodes^g* will refer to the number of nodes in the goal network. Also, note that the goal network will primarily be used to assess the accuracy of our proposed algorithm by comparing the grown networks (of the same size) with the goal network.

The substrate network represents a subgraph of the goal network and its nodes *Nodes^s* initially have the same properties as goal network’s nodes have (as they are directly extracted from the goal network). However, the procedure used for storing the nodes of a substrate network is different than the one used for the goal network. Every new node has its unique numeric identifier which is the same as node’s index. In this case, we can refer to the substrate network node by either its numeric identifier or by its index. A variable is used to preserve the

number of substrate network's nodes $number_of_nodes^s$ so that with each new node added, $number_of_nodes^s$ will be incremented and the identifier of the new incoming node will be equal to the new value of $number_of_nodes^s$.

3.4. Motifs and downlinks

Downlink are motif substructures composed of three nodes, wherein a father node regulates two leaf nodes; the left and right child. Generally, a transcription factor is capable of regulating other transcription factors and genes, while genes are nodes which do not regulate others, hence do not constitute any outgoing links. Figure (3-1) shows a downlink structure.

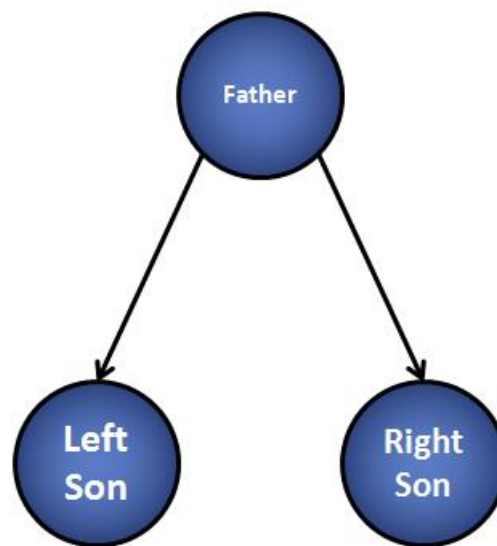


Figure (3-1) Schematic of a downlink: father node and two sons with two edges only.

Therefore the father node in a downlink is always a transcription factor and the leaf nodes can be either genes or transcription factors. Based on the gene types, the following classes of downlinks can be a result of the different gene combinations:

1. **tgg**: A downlink consisting of one transcription factor and two genes. The father node (transcription factor) regulates two genes.
2. **ttg**: A downlink consisting of two transcription factors and one gene. The father node (transcription factor) regulates a transcription factor and a gene.
3. **ttt**: A downlink that consists of three transcription factors. The father node (transcription factor) regulates two transcription factor nodes.

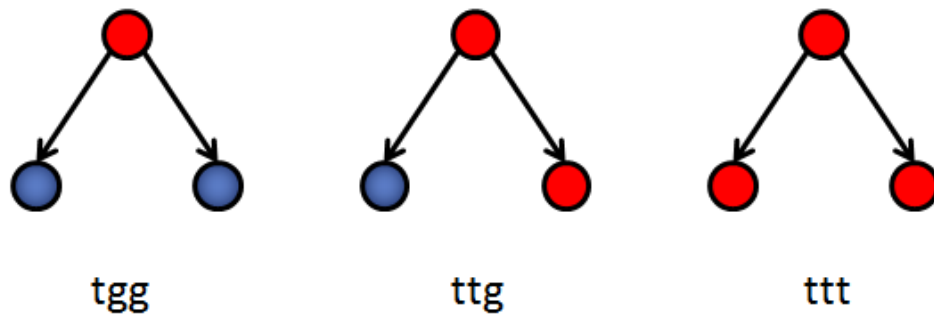


Figure (3-2) Schematic of possible types of downlink (tgg, ttg, ttt)

In each type downlink (**tgg**, **ttg** and **ttt**), the first letter refers to the father node in the GRN followed by the other two leaf nodes, such that a downlink is identified by a string identifier $\langle \text{father leaf leaf} \rangle$. The data structure for a downlink is considered as follows:

Structure downlink

Public father As vertex

Public left As vertex

Public right As vertex

End Structure

The pseudo-code below describes the procedure for identifying and returning a downlink substructure:

Type of downlink (downlink)

Count number of transcription nodes TF.

If number of TF=1 THEN

return tgg.

Else If number of TF=2 THEN

return ttg

Else

return ttt.

End

3.5. Downlink List

A downlink list represents a list of all the possible unique combinations of nodes that compose a downlink. Different networks will have different combination of downlinks based on the structure of the network. Given a regulatory network shown in figure 3-3, we can find the list of downlinks that exist in the network as following:

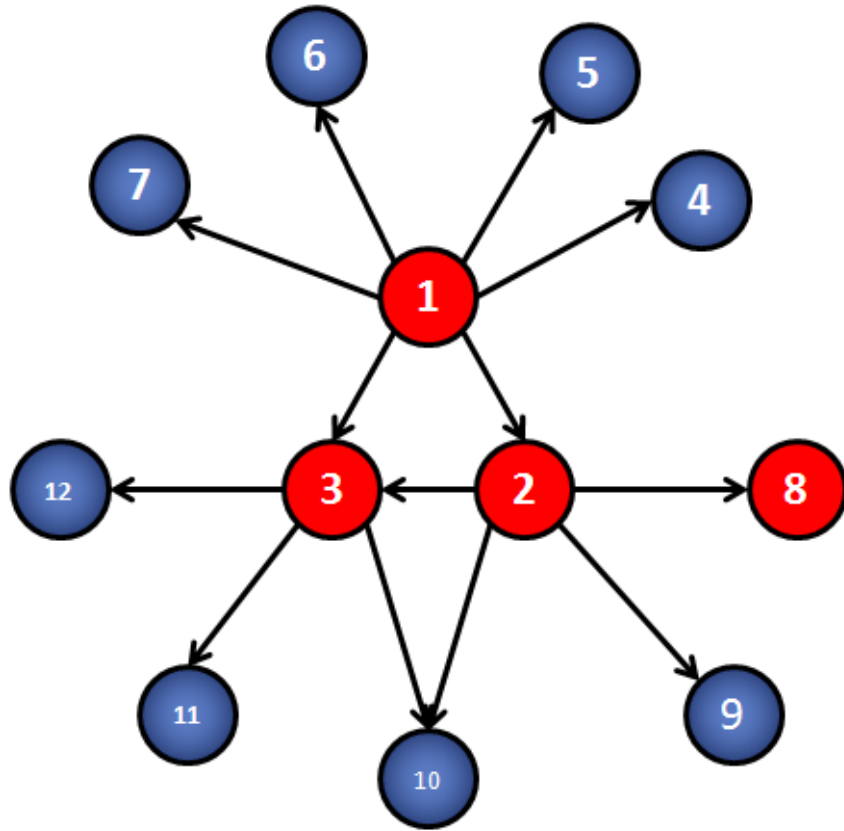


Figure (3-3) Sample regulatory network.

To calculate the number of downlinks in any network:

$$downlink(i) = \sum_{k=1}^{o-1} k \dots \dots \dots (1)$$

where $o = outdegree(downlink(i)), o \geq 2$

$$Network_{downlinks} = \sum_{i=1}^{network_{size}} downlink(i) \dots \dots \dots (2)$$

By using both equations (1) and (2) on the network in figure (3-3), only the following nodes will have non-zero outdegrees:

Node	Outdegree (≥ 2)	Number of downlinks
1	6	15
2	4	6
3	3	3

Table (3-1) Nodes with outdegree ≥ 2 and corresponding number of downlinks.

Therefore, number of downlinks in the given network is $15 + 6 + 3 = 24$. The elements of the downlink list of the sample network shown in figure (3-3) are given in the table below:

Father	Left son	Right son
1	2	3
1	2	4
1	2	5
1	2	6
1	2	7
1	3	4
1	3	5
1	3	6
1	3	7
1	4	5

1	4	6
1	4	7
1	5	6
1	5	7
1	6	7
2	8	3
2	8	9
2	8	10
2	9	3
2	9	10
2	10	3
3	10	11
3	10	12
3	11	12

Table (3-2) List of downlinks from the network in figure (3-3)

Note that the number of unique downlinks increases exponentially when there are more connections available. The algorithm below shows how to generate the downlink list:

Let n be the number of nodes in the network

Let $nodes(n)$ be the set of n nodes participating in the current network;
 $network_adjacency_matrix(n \times n)$ be a two dimension array storing the network structure; and
 n_dl be the number of downlinks

for $i := 1$ to n

for $j := 1$ to n

if $network_adjacency_matrix(i,j) = 1$ and $i \neq j$ **then**

for $k := 1$ to n

 father = $nodes(i)$

 found = **false**

if $j < k$ **then**

 left_son = $nodes(j)$

 right_son = $nodes(k)$

else

 left_son = $nodes(k)$

 right_son = $nodes(j)$

end if

for $l=1$ to n_dl

 //checking the existence of the downlink in the downlink list.

if $downlink_list(l).father = father$ and

$downlink_list(l).left = left_son$ and

$downlink_list(l).right = right_son$ **then**

 found = **true**

exit for

```

    end if
    if not found then
        n_dl ++
        downlink_list(n_dl).father = father
        downlink_list(n_dl).left = left_son
        downlink_list(n_dl).right = right_son
    end if
end if
end for
end if
end for
end for
end for

```

3.6. Vertex sharing Motif Network (VMN)

Vertex sharing motif networks VMN^{s-nw} uses graph transformation on the original GRN to create an undirected weighted network representing the network of connected downlinks, wherein every downlink in the GRN is represented here as a node. A VMN^{s-nw} can be represented as a square matrix of size $(N_{downlink}^{s-nw}, N_{downlink}^{s-nw})$ where $N_{downlink}^{s-nw}$ represents the number of downlinks. An element in the $VMN^{s-nw}(i, j)$ would represent the number of shared nodes between the downlink in index i and the downlink in index j , which is either 0,1,2 or 3 (as two downlinks can at most share 3 vertices). This matrix represents an undirected weighted network since both downlinks of i and j share the same number of nodes. As mentioned before the number of downlinks increase exponentially with more links in the GRN, hence the number of nodes and edges in the VMN increase exponentially as the network keeps growing.

$$VMN^{s_nw} \begin{bmatrix} 1,1 & \dots & 1, N_{downlinks}^{s_nw} \\ \vdots & \ddots & \vdots \\ N_{downlinks}^{s_nw}, 1 & \dots & N_{downlinks}^{s_nw}, N_{downlinks}^{s_nw} \end{bmatrix}$$

Chapter four Downlink Based Network Statistics

4.1. Downlink-downlink combinations.

As we have seen so far that many downlinks share node(s) with one another, hence we can utilize this property as a relationship between each pair of downlinks. In a network, we can find the list of available downlink-downlink combinations in terms of sharing nodes through the downlinks list mentioned above. First, we must define the different classes of downlink-downlink combination:

- 1- Downlink-downlink combinations that share one node.
- 2- Downlink-downlink combinations that share two nodes.
- 3- Downlink-downlink combinations that share three nodes.

All the possible downlink-downlink combinations of one, two and three shared nodes are listed as shows in tables (4-1), (4-2) and (4-3) considering the different types of downlinks comprising 'T' or 'G' nodes. The combinations that are not supported in each case are shown as deleted in the corresponding entries. These combinations were created based on the fact that a gene node cannot have any outgoing edges.

$N_{tgg-tgg}^1$	$N_{ttg-tgg}^1$	$N_{ttt-tgg}^1$
$N_{tgg-ttg}^1$	$N_{ttg-ttg}^1$	$N_{ttt-ttg}^1$
$N_{tgg-ttt}^1$	$N_{ttg-ttt}^1$	$N_{ttt-ttt}^1$

Table (4-1) Downlink-downlink combinations sharing one node only

$N_{tgg-tgg}^2$	$N_{ttg-tgg}^2$	$N_{\cancel{ttt-tgg}}^2$
$N_{tgg-ttg}^2$	$N_{ttg-ttg}^2$	$N_{\cancel{ttt-ttg}}^2$
$N_{\cancel{tgg-ttt}}^2$	$N_{ttg-ttt}^2$	$N_{\cancel{ttt-ttt}}^2$

Table (4-2) Downlink-downlink combinations sharing two nodes only

$N_{\cancel{tgg-tgg}}^3$	$N_{\cancel{ttg-tgg}}^3$	$N_{\cancel{ttt-tgg}}^3$
$N_{\cancel{tgg-ttg}}^3$	$N_{ttg-ttg}^3$	$N_{\cancel{ttt-ttg}}^3$
$N_{\cancel{tgg-ttt}}^3$	$N_{\cancel{ttg-ttt}}^3$	$N_{\cancel{ttt-ttt}}^3$

Table (4-3) shows downlink-downlink combination shares three nodes only

In order to calculate the different combinations we use the downlink list to compare every downlink with all other downlinks. The comparison algorithm is showed below:

Set all the network's counts to zero

For i := 1 to number of downlink in the network

Get the type of downlink i and increase that type counts (either N_{tgg}^{nw} , N_{ttg}^{nw} or N_{ttt}^{nw})

For j = i + 1 to number of downlink

x = get the number of shared nodes between downlink i and downlink j.

Get the string combination of the downlink i and downlink j.

Increase the same combination counts of the same number of shared nodes.

End For

End For

We next present the pseudo-code of the algorithm for getting the number of shared nodes between any two downlinks:

Get the number of shared nodes (downlink 1, downlink2) As Integer

Let a(3) be array of downlink 1 nodes

Let b(3) be array of downlink 2 nodes

For i = 1 To 3

For j = 1 To 3

If a(i) = b(j) Then k += 1

End For

End For

Return k

End

And to get the string combination of two downlinks we use the following strategy:

Get string of downlink-downlink combination (downlink 1, downlink 1)

Return type of downlink (downlink 1) + type of downlink (downlink 2)

End

By using the above algorithms we could find the downlink combination counts for any given network.

4.2. Downlink attachment:

The substrate network is grown by adding one downlink at a time. The process of adding downlinks has to meet a specific condition; that is, the new downlink must share node(s) with the network. The reason behind this condition is that the substrate network should be a singly connected component, because adding a downlink that does not share nodes with the substrate network would result in creating disjoint components which is not common in GRN topologies.

Since downlinks can either share one, two or three nodes, there are only three possible lists of “sharing nodes” with the downlink in the substrate network:

- List of all substrate downlinks that share one node with the new downlink
- List of all substrate downlinks that share two nodes with the new downlink
- List of all substrate downlinks that share three nodes with the new downlink

Once the number of shared nodes ($N_{shared_nodes}^S$) is specified, we can target the appropriate list from the substrate network and choose a downlink to attach to. The process of attachment would simply mean *matching* the shared node(s) between the selected downlink of the growing substrate and the new downlink and add the other unmatched node(s) and edge(s) of the new downlink to the growing network. Note that downlink attachment does not necessarily add new nodes to the substrate network as some downlinks share all there three nodes, However, such attachment process would certainly add two edges to the growing substrate network.

4.3. Patterns of downlink attachment:

The diversity in downlink types will create specific patterns of attachment. Basically, there are three categories of attachment depending on the number of shared nodes: category one, two or three. The downlinks comparison below depends on the fact that both downlinks share a specified number of nodes regardless of the use of numeric identifiers. We used the numeric identifiers for both downlinks to simply refer and identify the nodes before the attachment process. Once a downlink is added to the substrate network, all its numeric identifiers will be specified based on the sequence of nodes of the substrate network.

4.3.1. One shared node downlink attachment

There are four possible categories in which two downlinks share one node:

4.3.1.1. Father to father

If we have two downlinks, $d1=\{1,2,3\}$ and $d2 = \{4, 5, 6\}$ where $d1$ and $d2$ share the same father:

$$d1.father = d2.father$$

or (node 1 = node 4), then the attachment process will add nodes 5 and 6 to the substrate network as shows in figure (4-1).

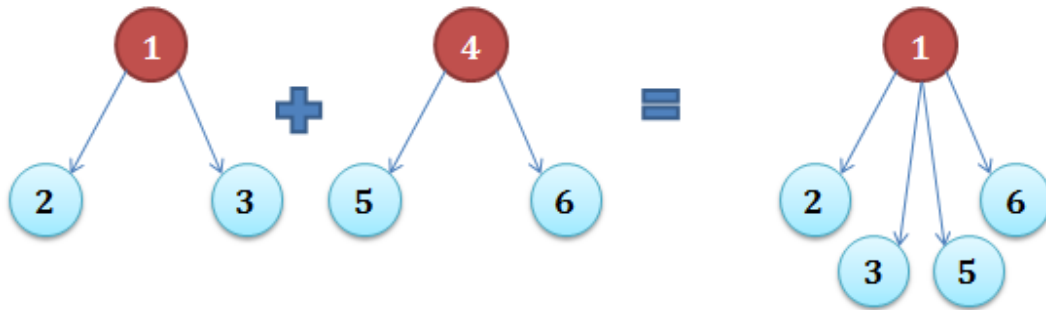


Figure (4-1) One node attachment, pattern one (P1).

The applicable set of downlink-downlink combinations are shown in table (4-4)

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	A	ttg - tgg	A	ttt - tgg	A
tgg - ttg	A	ttg - ttg	A	ttt - ttg	A
tgg - ttt	A	ttg - ttt	A	ttt - ttt	A

Table (4-4) All applicable downlink-downlink combination in one node attachment- pattern one.

4.3.1.2. Father to son :

If we have two downlinks, $d1=\{1,2,3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share one node where :

$$(d1.father = (d2.left \text{ or } d2.right))$$

or (node 1 = either node 5 or node 6), then the attachment process will add either (nodes 4 and 6) or (nodes 4 and 5) to the substrate network as shown in figure (4-2).

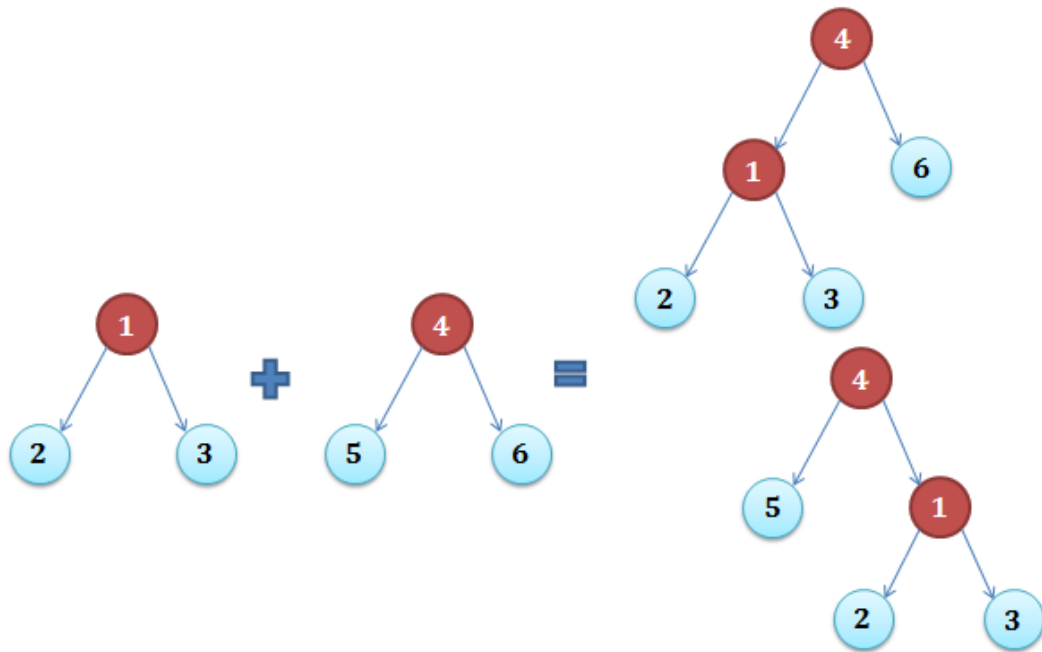


Figure (4-2) One node attachment, pattern two (P2).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	NA	ttg - tgg	NA	ttt - tgg	NA

tgg - ttg	A	ttg - ttg	A	ttt - ttg	A
tgg - ttt	A	ttg - ttt	A	ttt - ttt	A

Table (4-5) All applicable downlink-downlink combination in one node attachment- pattern two.

4.3.1.3. Son to Father

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ where $d1$ and $d2$ share one node where:

$$d1.left = d2.father \text{ or } d1.right = d2.father$$

Or (either node 2 or 3 = node 4), then the attachment process will add nodes 5 and 6 to the substrate network as shown in figure (4-3).

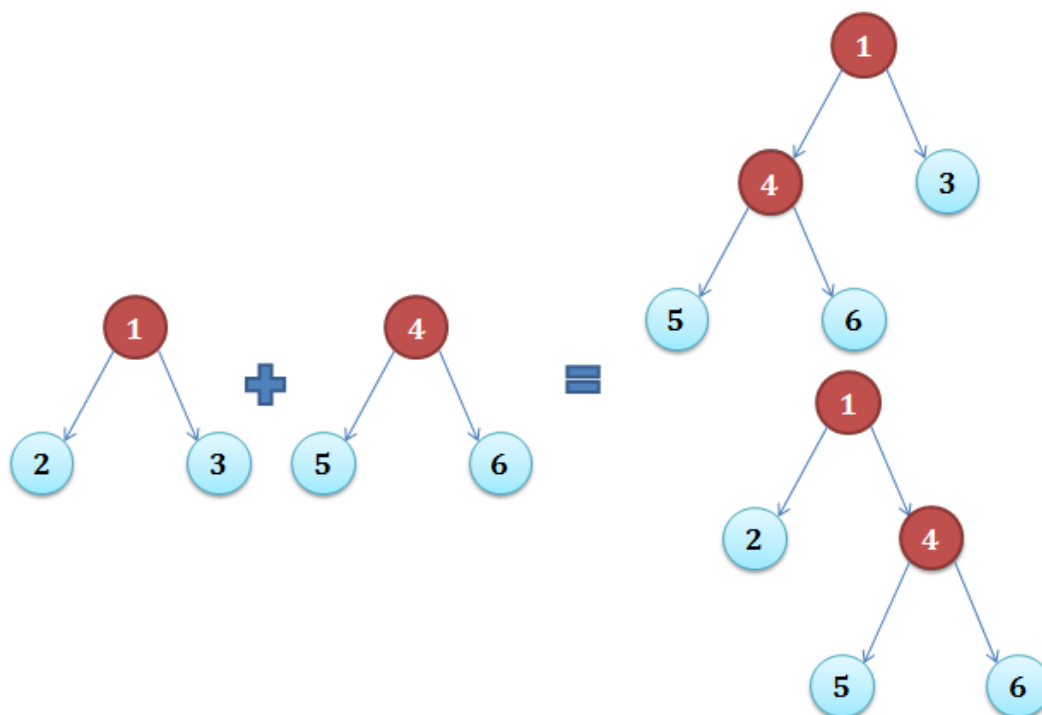


Figure (4-3) One node attachment, pattern three (P3).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	NA	ttg - tgg	A	ttt - tgg	A
tgg - ttg	NA	ttg - ttg	A	ttt - ttg	A
tgg - ttt	NA	ttg - ttt	A	ttt - ttt	A

Table (4-6) All applicable downlink-downlink combination in one node attachment- pattern

three.

4.3.1.4. Son to son.

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share one node where:

$$(d1.left = (d2.left \text{ or } d2.right)) \text{ or } (d1.right = (d2.left \text{ or } d2.right))$$

Or (either node 2 or 3 = either node 5 or 6), then the attachment process will add nodes 5 and 6 to the substrate network as shown in figure (4-4).

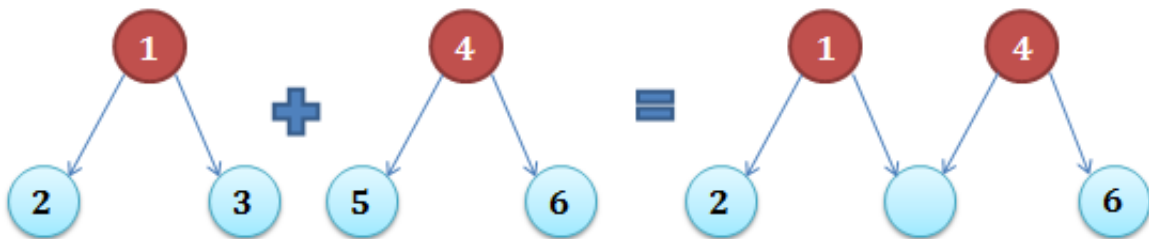


Figure (4-4) One node attachment, pattern three (P3).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	A	ttg - tgg	A	ttt - tgg	NA
tgg - ttg	A	ttg - ttg	A	ttt - ttg	A
tgg - ttt	NA	ttg - ttt	A	ttt - ttt	A

Table (4-7) All applicable downlink-downlink combination in one node attachment- pattern

three.

4.3.2. Two shared nodes downlink attachment

There are five possible categories in which two downlinks share two nodes:

4.3.2.1. Father – son to father – son:

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share two nodes where:

$$d1.father = d2.father \text{ and } (d1.left = (d2.left \text{ or } d2.right) \text{ or } d1.right = (d2.left \text{ or } d2.right))$$

Or (node 1 = node 4 and (either node 2 or 3 = either node 5 or 6)), then the attachment process will add only one node to the substrate network as shown in figure (4-5).

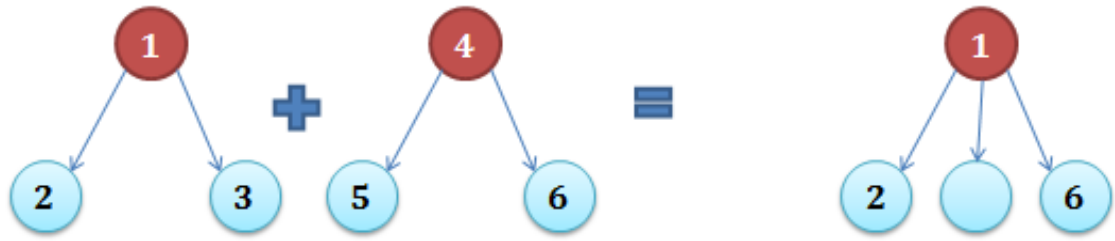


Figure (4-5) Two nodes attachment, pattern one (P1).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	A	ttg - tgg	A	ttt - tgg	NA
tgg - ttg	A	ttg - ttg	A	ttt - ttg	A
tgg - ttt	NA	ttg - ttt	A	ttt - ttt	A

Table (4-8) All applicable downlink-downlink combination in two nodes attachment- pattern one.

4.3.2.2. Father – son to son – father.

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ provided $d1$ and $d2$ share two nodes:

$$d1.father = (d2.left \text{ or } d2.right) \text{ and } d2.father = (d1.left \text{ or } d1.right)$$

Or (node 1 = (either node 5 or 6) and node 4 = (either node 2 or 3)), then the attachment process will add one node to the substrate network as shown in figure (4-6).

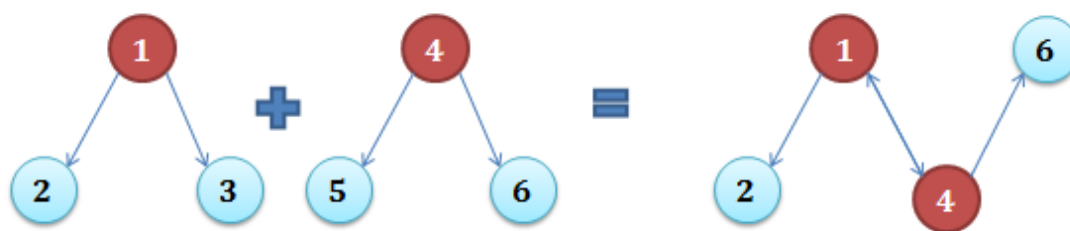


Figure (4-6) Two nodes attachment, pattern two (P2).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	NA	ttg - tgg	NA	ttt - tgg	NA
tgg - ttg	A	ttg - ttg	A	ttt - ttg	NA
tgg - ttt	NA	ttg - ttt	NA	ttt - ttt	A

Table (4-9) All applicable downlink-downlink combination in two nodes attachment - pattern two.

4.3.2.3. Father – son to son – son.

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share two nodes:

$$d1.father = (d2.left \text{ or } d2.right) \text{ and } (d1.left = (d2.left \text{ or } d2.right) \text{ or } d1.right = (d2.left \text{ or } d2.right))$$

Or (node 1 = (either node 5 or 6) and node 4 = (either node 2 or 3)), then the attachment process will add nodes 5 and 6 to the substrate network as shows in figure (4-7).

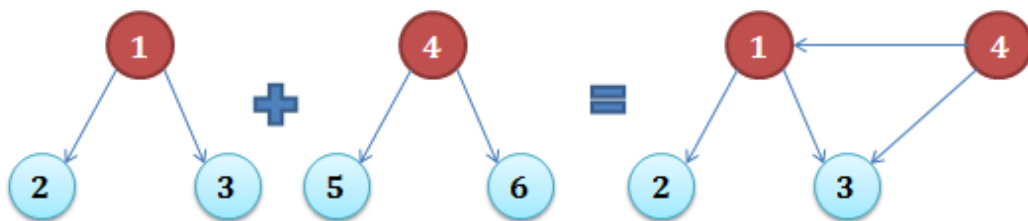


Figure (4-7) Two nodes attachment, pattern three (P3).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	NA	ttg - tgg	NA	ttt - tgg	NA
tgg - ttg	A	ttg - ttg	A	ttt - ttg	NA
tgg - ttt	NA	ttg - ttt	NA	ttt - ttt	A

Table (4-10) All applicable downlink-downlink combination in two nodes attachment - pattern

three.

4.3.2.4. Son – son to father – son.

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share two nodes where:

$$d2.father = (d1.left \text{ or } d1.right) \text{ and } (d1.left = (d2.left \text{ or } d2.right) \text{ or } d1.right = (d2.left \text{ or } d2.right))$$

Or (node 4 = (either node 2 or 3) and (either node 5 or 6 = either node 2 or 3)), then the attachment process will add only one node to the substrate network as shown in figure (4-8).

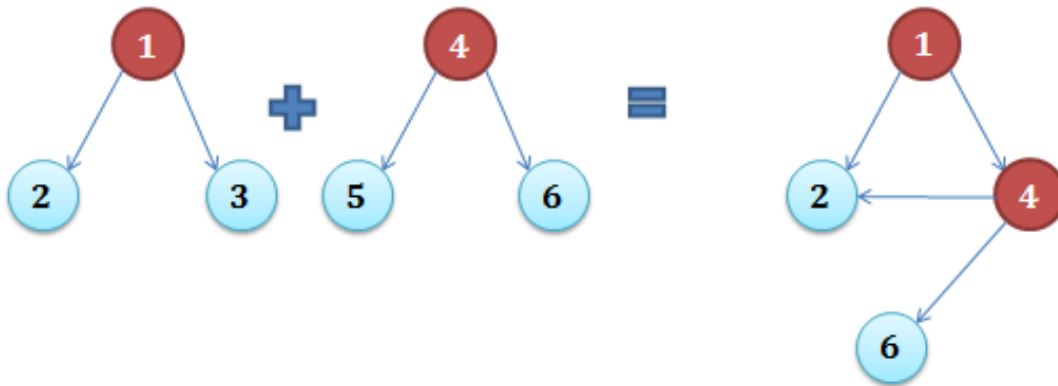


Figure (4-8) Two nodes attachment, pattern three (P3).

Type	A: applicable	Type	A: applicable	Type	A: applicable
	NA: not applicable		NA: not applicable		NA: not applicable
tgg - tgg	NA	ttg - tgg	A	ttt - tgg	NA
tgg - ttg	NA	ttg - ttg	A	ttt - ttg	A
tgg - ttt	NA	ttg - ttt	NA	ttt - ttt	A

Table (4-11) All applicable downlink-downlink combination in two nodes attachment - pattern three.

Even though the node attachment is different in (son-son to father-son) to that in (father-son to son-son), it results in same pattern. Therefore, we will refer to the (son-son to father-son) as pattern three.

4.3.2.5. Son – son to son – son.

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share two nodes where:

$$d1.left = (d2.left \text{ or } d2.right) \text{ and } d1.right = (d2.left \text{ or } d2.right)$$

Or (node 2 = (either node 5 or 6) and node 3 = (either node 5 or 6)), then the attachment process will add only one node to the substrate network as shown in figure (4-9).

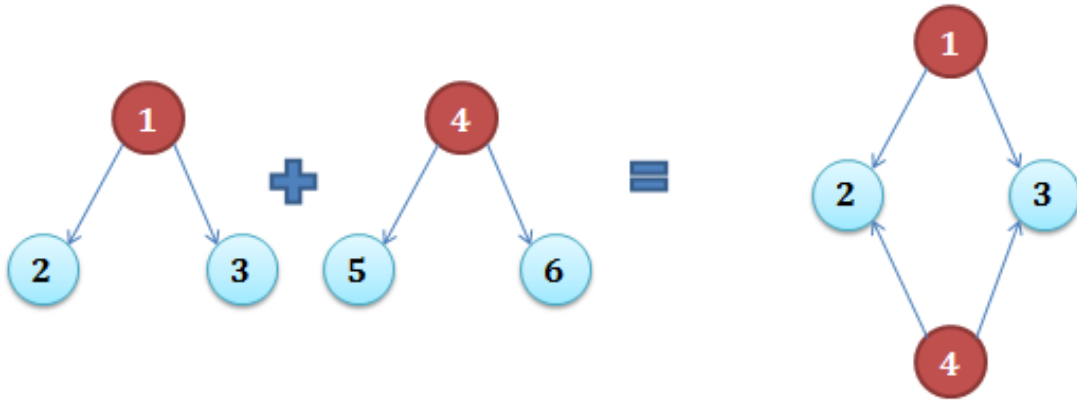


Figure (4-9) Two nodes attachment, pattern 4 (P4).

Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	A	ttg - tgg	NA	ttt - tgg	NA
tgg - ttg	NA	ttg - ttg	A	ttt - ttg	NA
tgg - ttt	NA	ttg - ttt	NA	ttt - ttt	A

Table (4-12) All applicable downlink-downlink combination in two nodes attachment - pattern

four.

4.3.3. Three shared nodes downlink attachment

4.3.3.1. Father – son – son to son – father – son.

If we have two downlinks, $d1 = \{1, 2, 3\}$ and $d2 = \{4, 5, 6\}$ and $d1$ and $d2$ share three nodes where:

$d1.father = (d2.left \text{ or } d2.right)$ and $d2.father = (d1.left \text{ or } d1.right)$ and $(d1.left \text{ or } d1.right) = (d2.left \text{ or } d2.right)$

Or (either (nodes (1-3-2) attached to (5-4-6) or (6-4-5)) or (nodes (1-2-3) attached to (5-4-6) or (6-4-5)), then the attachment process will add no nodes to the substrate network as shown in figure (4-10)

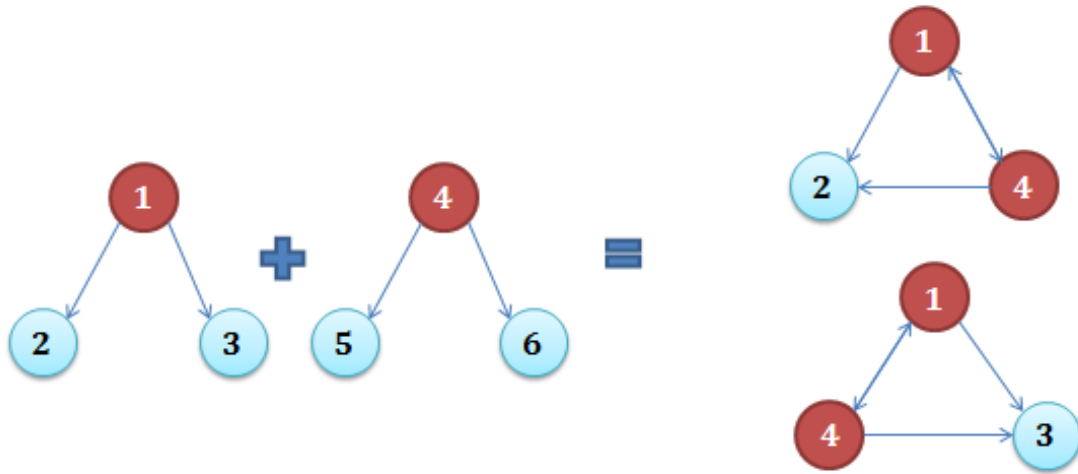


Figure (4-10) three nodes attachment, pattern one (P1).

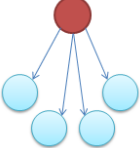
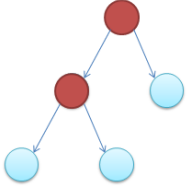
Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable	Type	A: applicable NA: not applicable
tgg - tgg	NA	ttg - tgg	NA	ttt - tgg	NA
tgg - ttg	NA	ttg - ttg	A	ttt - ttg	NA
tgg - ttt	NA	ttg - ttt	NA	ttt - ttt	A

Table (4-13) All applicable downlink-downlink combination in three nodes attachment- pattern one.

So far we considered all the possible combinations of downlink attachments. However, we saw there are few recurrent patterns appearing in the one and two node attachment cases:

- 1- Repeated patterns of attachment in one node attachment: As we can see in the one node attachment, there are only three patterns. Note that each of the second and the third attachment cases (father to son and son to father) result in same structure of downlink attachment. Therefore we consider both cases as one.
- 2- Repeated patterns in the two nodes attachment cases: Same as in one node attachment, we can find only four patterns of downlinks attachment. Cases: three and four yield the same structures.

The table below lists all the non-repeated patterns and sets of applicable downlink-downlink combination in each pattern.

Category	Pattern	Pattern graph	Applicable Downlink-downlink combinations		
One node attachment	P1		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt
	P2		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt

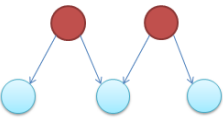
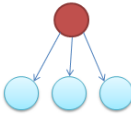
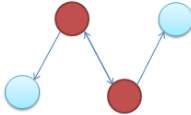
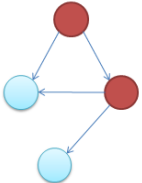
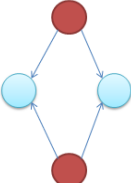
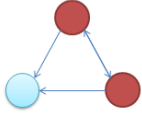
	P3		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt
Two nodes attachment	P1		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt
	P2		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt
	P3		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt
	P4		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt
Three nodes attachment	P1		tgg - tgg	ttg - tgg	ttt - tgg
			tgg - ttg	ttg - ttg	ttt - ttg
			tgg - ttt	ttg - ttt	ttt - ttt

Table (4-14) All applicable downlink-downlink combinations in each pattern.

4.4. Sets of applicable patterns in each downlink-downlink combination:

It is important to specify the patterns of attachment that can be applied in two downlink attachments. Different downlink-downlink attachments can result in different patterns. Here, we determine all possible and acceptable patterns of attachment for one, two and three shared nodes that specified to the type of the two downlinks. Table (4-15) shows the applicable patterns in one, two and three node attachments.

	dl-dl combinati on	Applicable pattern	dl-dl combinati on	Applicable pattern	dl-dl combinat ion	Applicable pattern
One node attachment	tgg - tgg	{P1,P3}	ttg – tgg	{P1,P2,P3}	ttt – tgg	{P1,P2}
	tgg - ttg	{P1,P2,P3}	ttg – ttg	{P1,P2,P3}	ttt – ttg	{P1,P2,P3}
	tgg - ttt	{P1,P2}	ttg – ttt	{P1,P2,P3}	ttt – ttt	{P1,P2,P3}
Two nodes attachment	tgg - tgg	{P1,P4}	ttg – tgg	{P1,P3}	ttt – tgg	NA
	tgg - ttg	{P1,P2,P3}	ttg – ttg	{P1,P2,P3,P4}	ttt – ttg	{P1,P3}
	tgg - ttt	NA	ttg – ttt	{P1}	ttt – ttt	{P1,P2,P3,P4}
Three nodes attachment	tgg - tgg	NA	ttg – tgg	NA	ttt – tgg	NA
	tgg - ttg	NA	ttg – ttg	{P1}	ttt – ttg	NA
	tgg - ttt	NA	ttg – ttt	NA	ttt – ttt	{P1}

Table (4-15) Applicable patterns for different downlink-downlink combinations.

Chapter Five Downlink Based Substrate Network Growing Model

A downlink based network growing model is an algorithm that uses downlinks as a basic structure for analyzing and growing a small sized substrate network into a large scale complex network. Several models exist to grow a network of small scale to large scale networks by adding single nodes or edges at a time.

In our downlink based algorithm of growing large scale complex networks, we analyze and measure specific downlink based network statistics in model GRNs and incorporate them into the algorithm as set of rules used to grow the network.

The network growing model uses the addition of downlink as set of three nodes and two edges with very specific types and patterns of attachment to the substrate network as discussed before. That is, every time a new downlink comes, its components (nodes and edges) would be added to the network's structure. We have avoided creating more than two separated (disconnected) components in the same network. In this way, every time we add a new downlink to the network, we increase the network size in terms of nodes and edges.

Generally, our motivation is that we want to predict the rules of growing a biological network such that it would have the same characteristics as another target GRN. Assume we have a goal network called $G = (V, E)$ which has set of vertices V and set of edges E . We next extract a substrate network $S = (v, e)$ which is a subnetwork of the goal network G . v is a subset of V while e is a subset of E ; note that e is set of edges over v only. The intention is to find the methodology in which we grow the substrate network that starts with small set of nodes and edges $S (v, e)$ to be a network that has the same characteristics and properties as the goal network $G (V, E)$. The algorithm will have two basic input networks:

- A goal network G.
- A substrate network S.

And it outputs the grown substrate network Grown_S.

However, the growing concepts are the same in term of adding nodes to the substrate network, but the methodology and structure of why and how we are adding these nodes are different here. As we have found earlier that specific network structures would have a very specific number of patterns of attachment and downlink-downlink combinations, we can grow the substrate network by using these statistics by using the probability distributions of either the goal network or the current substrate itself.

Growing the substrate network requires specifying what kind of downlink should be added to the substrate and where exactly and why did we choose these positions and structures.

5.1. The probability distributions

In this study where emulating the distribution of the nodes and edges of the goal network represents the target goal, a process of emulation is implemented so that it tries to derive the grown substrate network having the same distribution as the goal network. To do so, a very basic function $D \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ is used to determine a random value within several ranges. The ranges may have several meanings such as the distribution of number of shared nodes or number of attachment patterns existing in a specific downlink-downlink combination. The utility behind repeatedly using this function is that we use the same probability distribution as provided by the goal network. Large values are more expected to dominate the returning value than the smaller values

depending on the generated random value of r . The probability distribution $D\left(\begin{smallmatrix} x \\ y \\ z \end{smallmatrix}\right)$ is defined as

follows:

$$D\left(\begin{smallmatrix} x \\ y \\ z \end{smallmatrix}\right) = \begin{cases} 1, & \text{if } 0 \leq r \leq \frac{x}{x+y+z} \\ 2, & \text{if } \frac{x}{x+y+z} < r \leq \frac{x+y}{x+y+z} \\ 3, & \text{if } \frac{x+y}{x+y+z} < r \leq 1 \end{cases}$$

The algorithm below explains how to find the probability distribution for three different values:

Probability_distribution(x, y, z)

Let total = x + y + z

let r be a value between (1,0)

If $r \geq 0$ And $r \leq \frac{x}{\text{total}}$ Then

Return 1

Else If $r > \frac{x}{\text{total}}$ and $r \leq \frac{x+y}{\text{total}}$ Then

Return 2

Else

Return 3

End If

End algorithm

Same algorithm is used to work over different number of parameters.

5.2. The selection of downlinks to attach to.

A downlink is selected from the substrate network by using a probability distribution over the specified network's statistics. The process of selecting the downlink is shown below:

First, we have to find the type of downlink to be chosen from the substrate network based on the probability distribution over the number of $N_{tgg}^{s_nw}$, $N_{ttg}^{s_nw}$ and $N_{ttt}^{s_nw}$ as follows:

Get a random variable r between 0-1.

$$\text{Let } N_{total}^{s_nw} = N_{tgg}^{s_nw} + N_{ttg}^{s_nw} + N_{ttt}^{s_nw}$$

Let *selected_type* be a string of the selected type.

If r lies in the range of $(0, \frac{N_{tgg}^{s_nw}}{N_{total}^{s_nw}})$ then *selected_type* = tgg.

If r lies in the range of $(\frac{N_{tgg}^{s_nw}}{N_{total}^{s_nw}}, \frac{N_{tgg}^{s_nw} + N_{ttg}^{s_nw}}{N_{total}^{s_nw}})$ then *selected_type* = ttg.

If r lies in the range of $(\frac{N_{tgg}^{s_nw} + N_{ttg}^{s_nw}}{N_{total}^{s_nw}}, 1)$ then *selected_type* = ttt.

Second, we should store all downlinks that have the same type as *selected_type* does.

Let $i_{selected_type}$ be number of existing downlinks of the *selected_type*, $i_{selected_type} = 0$

Define Probable_Downlink() as an array of downlink that holds the downlinks of the selected type.

For $i=1$ to $N_{downlink}^{s_nw}$

If type of downlink(downlink i) = *selected_type* then

$$i_{selected_type}^{++};$$

$$\text{Probable_Downlink}(i_{selected_type}) = \text{downlink_List}(i)$$

End if

Next i

Now we have all the potential downlinks to be attached to, one downlink should be selected amongst them based on its degree centrality of the corresponding VMN representation of the substrate network.

5.3. Degree Centrality

Degree of a node can be interpreted as how influential is the node in terms of its connection within the network. The node that has many edges (relationships) than other nodes means that this node is central and it has a significant role in specifying the functionality of the network. In a directed network, the node with high indegree means it is essential to other nodes and the node that has high outdegree means it has high influence into other connected nodes.

A degree centrality metric is used to specify the substrate downlinks t having the highest potential to be selected for attachment with the new downlink. Let's define the $DL_DC(i_{selected_type})$ to be the array that is going to hold the degree centrality for each downlink. In order to calculate the degree centrality for each single downlink, a VMN matrix is used. A degree centrality basically is defined in the following equation:

$$C_i = \frac{degree_i}{\sum_j^n degree_j}$$

The degree of any given downlink is calculated as follows:

$$degree_i = \sum_{j, i \neq j}^n VMN_{(i,j)}$$

The downlink degree centrality is stored in an array called $DL_DC()$. After we have calculated the degree centrality for each downlink in the substrate network, a downlink should be selected based on the probability distribution of the degree centralities of the different downlinks. The probability distribution of downlinks degree centrality is calculated by the following algorithm:

Let $Temp_DL_DC(i_{selected_type})$ be the array that contains the probability of each downlink to be chosen.

Let $temp_total = 0$

For i=1 to $i_{selected_type}$

$$temp_total = temp_total + DL_DC(i)$$

Next i

$$temp_DL_DC(1) = \frac{DL_DC(1)}{temp_total}$$

For i = 2 To $i_{selected_type}$

$$Temp_DL_DC(1) = \frac{DL_DC(1)}{temp_total}$$

$$Temp_DL_DC(i) = Temp_DL_DC(1) + \frac{DL_DC(i)}{temp_total}$$

Next

Let r be a random value between (0,1).

Let temp =1

For i = 1 To $i_{selected_type}$

If $r > Temp_DL_DC(i)$ Then

temp = i

Else

Exit For

End If

Next

Return substrate downlink (temp)

5.4. The selection of downlinks to be added.

The selection of the new downlink would follow a certain methodology so that an appropriate network is generated with the potential of high similarity to the goal network. We use the goal

network's statistics to find out the probable type of downlink and pattern of attachment. In general, the strategy of selecting and adding the downlink is as follows:

- 1- Depending on the strategy of growth of the substrate network, we would generate the number of shared nodes $N_{shared_nodes}^S$ between the selected substrate downlink and the new downlink.
- 2- Based on the type of the selected substrate downlink (*selected_type*) and the number of shared nodes ($N_{shared_nodes}^S$), the type of the new downlink would be specified by using the probability distribution over all the combination of downlinks whose first downlink type is the same as the selected type. We used table (4-15) to implement this strategy.
- 3- Now we have the downlink-downlink combination set and we can choose the pattern of attachment using the probability distribution over all the patterns of that combination.

In order to specify the number of nodes ($N_{shared_nodes}^S$) we used the following algorithm:

$$\text{Let } total = N_1^{S_nw} + N_2^{S_nw} + N_3^{S_nw}$$

Let r be a random value between (0,1)

If r lies in the range of $(0, \frac{N_1^{S_nw}}{total})$ then $N_{shared_nodes}^S = 1$

If r lies in the range of $(\frac{N_1^{S_nw}}{total}, \frac{N_1^{S_nw} + N_2^{S_nw}}{total})$ then $N_{shared_nodes}^S = 2$

If r lies in the range of $(\frac{N_1^{S_nw} + N_2^{S_nw}}{total}, 1)$ then $N_{shared_nodes}^S = 3$

The next algorithm shows how to specify the type of the downlink and pattern of attachment:

$N_{shared_nodes}^S$ = get the number of shared nodes.

// Now we specified what table to use (one, two or three nodes attachment table.)

Substrate_dl = Get the type of downlink (selected substrate downlink)

// Now we get the first section of the downlink-downlink combination. We would use this value to specify the type of the second downlink.

New_dl = select the type of the new downlink using the probability distribution over the applicable downlink-downlink combinations where *substrate_dl* represents the first part.

//Now we have specified the downlink-downlink combination and number of shared nodes, we should target what pattern to use under that combination:

Pattern = get the pattern using the probability distribution over the applicable patterns of the selected downlink-downlink combination.

Attach ($N_{shared_nodes}^S$, *pattern*, *substrate_dl*, *new_dl*)

5.5. Attaching the new downlink using the different patterns:

A downlink should be attached to the substrate network using very specific structure of matching and node addition as described earlier. Each pattern has a unique representation inside the network and needs very discrete requirements to be applied. The node and edge addition was done normally for each individual pattern case in such a way that we only add the new node(s) and edges as contained in the new downlink to the selected substrate downlink. The downlink addition is implemented in two steps: First, the new node(s)' numeric identifier(s) is(are) added to the substrate nodes list while preserving their indices. Second, we add new edges among the preserved nodes as specified for each pattern.

In one node attachment patterns, two new nodes and edges will be added to the substrate network. In two node attachment patterns, one new node and two edges will be added to the substrate network. And in three node attachment pattern no new node is added to the substrate network while two new edges are added.

Apparently, a gene node will not be allowed to have an out degree as other transcription factor nodes. Moreover, when the system is required to choose one of two nodes of the same type to be matched, we should check for the following:

- If we have to add an incoming edge to either two gene nodes or transcription factor nodes, we use the probability distribution between the incoming degrees of each node and then we generate a random value between (0,1) and we pick the node that falls in the range of incoming degree distribution $(\frac{node^1_{incoming\ degree}}{node^1_{incoming\ degree} + node^2_{incoming\ degree}}, 1)$ to add the edge to it.
- Similarly, if we have to add an outgoing edge to two transcription factor nodes, we will use the probability distribution between the outgoing degrees of each node and then we generate a random value between (0,1) and we pick the node that falls in the range of outgoing degree distribution $(\frac{node^1_{outgoing\ degree}}{node^1_{outgoing\ degree} + node^2_{outgoing\ degree}}, 1)$ to add the edge to it.

5.6. Different strategies in growing the network:

In this study, four different growing strategies were used so that later we can check the performance of the algorithm in terms of measuring the similarity between the grown networks

and the goal network. Generally, we need to specify what network statistics should be used in order to grow the substrate network? Should it be the same as the goal network's statistics or the substrate network's statistics? Also, we need to specify the number of shared nodes between the selected substrate downlink and the new coming downlink in every single attachment; should it be chosen based on the probability distribution over the specified network N_1^{S-NW} , N_2^{S-NW} and N_3^{S-NW} or by the number of patterns in each of the three attachment modes(3,4,1)?

In fact, four different strategies are implemented in this thesis:

- 1- Using goal network's statistics with unbiased distribution of probability over N_1^g , N_2^g and N_3^g .
- 2- Using goal network's statistics with biased distribution of probability over (3, 4, 1).
- 3- Using substrate network's statistics with unbiased distribution of probability over N_1^s , N_2^s and N_3^s .
- 4- Using substrate network's statistics with biased distribution of probability over (3, 4, 1).

Finding the number of shared nodes would be using one of the following methods:

- 1- **Unbiased distribution:** by using the N_1^{S-NW} , N_2^{S-NW} and N_3^{S-NW} which represent the number of one shared node, two shared nodes and three shared nodes respectively of the specified network. The number of shared nodes between the selected substrate downlink and the new downlink ($N_{shared_nodes}^s$) would be calculated by using the probability distribution over N_1^{S-NW} , N_2^{S-NW} and N_3^{S-NW} . The algorithm below shows how to generate the number of shared nodes:

$$\text{Let } total = N_1^{S_NW} + N_2^{S_NW} + N_3^{S_NW}$$

Let r be a random value between (0,1)

$$\text{If } r \text{ lies in the range of } (0, \frac{N_1^{S_NW}}{total}) \text{ then } N_{shared_nodes}^S = 1$$

$$\text{If } r \text{ lies in the range of } (\frac{N_1^{S_NW}}{total}, \frac{N_1^{S_NW} + N_2^{S_NW}}{total}) \text{ then } N_{shared_nodes}^S = 2$$

$$\text{If } r \text{ lies in the range of } (\frac{N_1^{S_NW} + N_2^{S_NW}}{total}, 1) \text{ then } N_{shared_nodes}^S = 3$$

2- **Biased distribution:** By using the probability distribution of the number of patterns within each category of attachments. In one node attachment case, there are three different possible patterns. In two nodes attachment case, there are four different patterns and in the three nodes attachment case there is only one pattern. The algorithm below shows how to generate the number of shared nodes:

$$\text{Let } total = 8$$

// 3 in one node attachment, 4 in two nodes attachment and 1 in three nodes attachment

Let r be a random value between (0,1)

$$\text{If } r \text{ lies in the range of } (0, \frac{3}{total}) \text{ then } N_{shared_nodes}^S = 1$$

$$\text{If } r \text{ lies in the range of } (\frac{3}{total}, \frac{3+4}{total}) \text{ then } N_{shared_nodes}^S = 2$$

$$\text{If } r \text{ lies in the range of } (\frac{3+4}{total}, 1) \text{ then } N_{shared_nodes}^S = 3$$

Chapter Six Results and Discussion

Here we report the performance of our proposed network growing algorithm by comparing it with sample GRNs extracted out of *E. coli* and also the modified preferential attachment algorithm presented in [19]. We first discuss the maximum likelihood estimation strategy used for comparing the degree and motif distributions.

6.1. Maximum likelihood estimation of cumulative distribution functions

Several features of interest in biology when subjected to repeated measurement show a cumulative probability distribution that follows power-law type mathematical relationship [41]. For that reason, the in-, out-, or total-degree distributions of a network may support a power-law type tail depending on the scheme of the attachment kernel used to build it. However if there are no *a priori* theoretical considerations to predict whether experimental data should best fit to a particular distribution, then curve-fitting methodologies are commonly used to justify empirical relationships among features in these data. It is known, for example, that using a least squares based optimization algorithm does not accurately determine whether the data are power-law distributed [42].

Addressing this problem, [42] presented a maximum likelihood estimation based approach that determines whether data are power-law distributed or not. To carry out our analyses on the motif participation and degree distributions extracted from the experimental and synthetic networks described above, we employed MATLAB implementations of the maximum likelihood estimation method of [41].

The modified preferential attachment algorithm from [19] was implemented here with all 3 attachment kernels: linear, sigmoidal and power-law (details on the kernels can be found in [19]).

MLE for the different degree distributions															
Kernel type	Sample Substrates														
	100n-1			100n-2			100n-3			100n-4			100n-5		
	In	Out	Total	In	Out	Total	In	Out	Total	In	Out	Total	In	Out	Total
1. Linear	0.91	0.94	0.81	0.25	0.55	0.18	0.86	0.74	0.63	1.18	0.87	0.75	0.8	1.92	0.21
2. Power-law	1.09	1.08	0.99	0.23	0.57	0.16	0.8	0.71	0.73	1.09	0.99	0.46	0.88	1.91	0.19
3. Sigmoidal	0.92	0.98	0.97	0.42	0.63	0.15	1.01	0.66	0.82	1.25	0.65	1.09	0.62	1.91	0.3
4. p = goal	0.08	0.96	0.13	0.38	0.21	0.07	0.62	0.12	0.1	0.22	0.44	0.07	1.89	1.9	0.35
5. p = goal (biased)	0.12	0.79	0.12	0.35	0.57	0.05	0.3	0.9	0.1	0.36	0.96	0.2	1.9	1.9	0.3
6. p = substrate	0.16	1.4	0.61	0.37	0.69	0.02	0.38	0.9	0.36	0.48	0.94	0.37	1.9	1.9	0.41
7. p = substrate (biased)	0.24	1.38	0.43	0.37	0.53	0.04	0.38	0.9	0.34	0.30	0.91	0.41	1.89	1.89	0.35

Table (6-1) MLE based comparison for different degree distributions

Originally, the MLE gives a best fit scale free distribution for the degree distribution at hand. For our 5 target networks of size 100 nodes extracted from the largest connected component (LCC) of the E. coli GRN, each network has its own alpha (that designate the exponent of the power law distribution). Moreover, we extracted subgraphs from our target networks of sizes equal to 10, 20, 30, and 40% of their original target networks, 5 networks per percentage, which gives us

a total of 20 subgraphs per sample target network (herein initial formations). For each initial formation, the networks are grown to 100 nodes using the algorithm discussed in the thesis with the seven kernels listed in the table and the differences between their alpha values and the alpha values of the target networks are stored (herein alpha').

Table 6-1 gives the average of alpha' across these 20 cases. Similarly, Table 6-2 shows the averaged alpha' for the feed-forward loop motif distributions for the 20 cases. It can be observed that the biased goal network kernel works better than the unbiased version, whereas the biased substrate network kernel works better than the unbiased version. Also, the goal network based kernels work better than the substrate network based kernels which is to be expected; the substrates being of smaller size will not generate the best values for computing the probability distributions for the attachment algorithm. Regardless, all 4 versions of our proposed algorithm work better than the modified preferential attachment based algorithm proposed in [19] improving both the degree and motif distributions significantly.

MLE for the motif participation distribution					
Kernel type	Sample Substrate				
	100n-1	100n-2	100n-3	100n-4	100n-5
1. Linear	1.17	1.19	0.79	1.32	0.33
2. Power-law	1.17	1.19	0.79	1.32	0.33
3. Sigmoidal	1.17	1.19	0.79	1.32	0.33

4. p = goal	0.67	0.43	0.34	0.75	0.63
5. p = goal (biased)	0.47	0.41	0.34	0.7	0.63
6. p = substrate	0.75	1.2	0.34	0.69	0.62
7. p = substrate (biased)	0.77	1.23	0.34	0.91	0.62

Table(6-2) MLE based motif participation distribution

Figures 6-1 to 6-4 plot the same statistics as shown in the above two tables for the total, in- and out- degree distributions and the motif distribution.

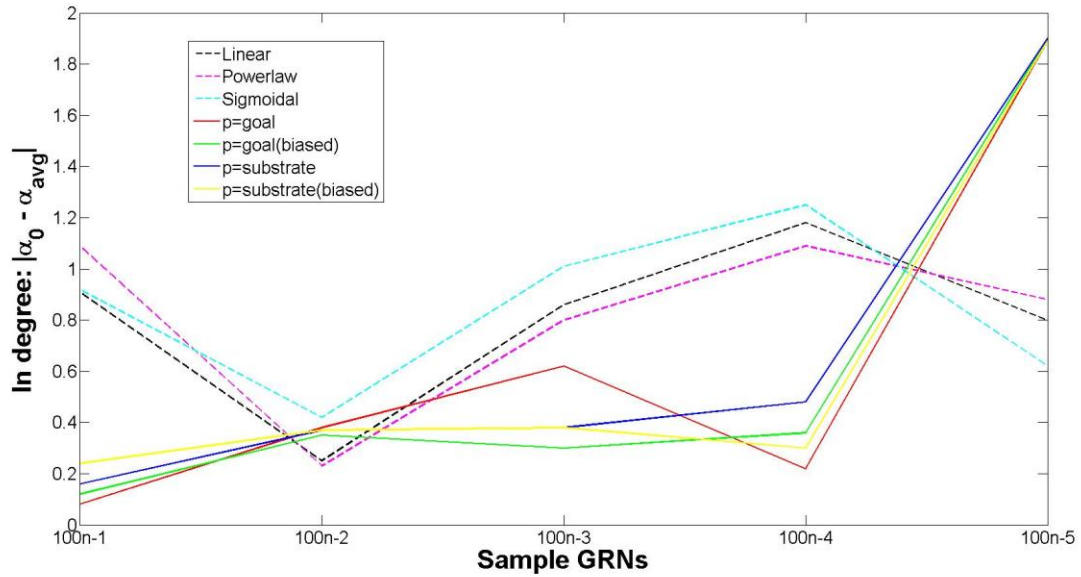


Figure (6-1) MLE difference for in degree distributions

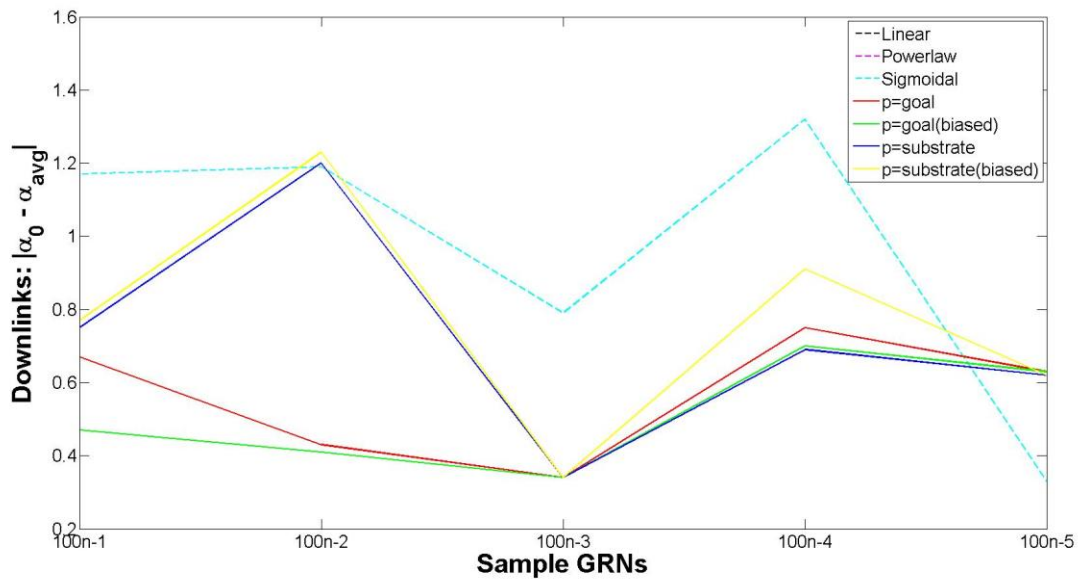


Figure (6-2) MLE difference for motif distributions

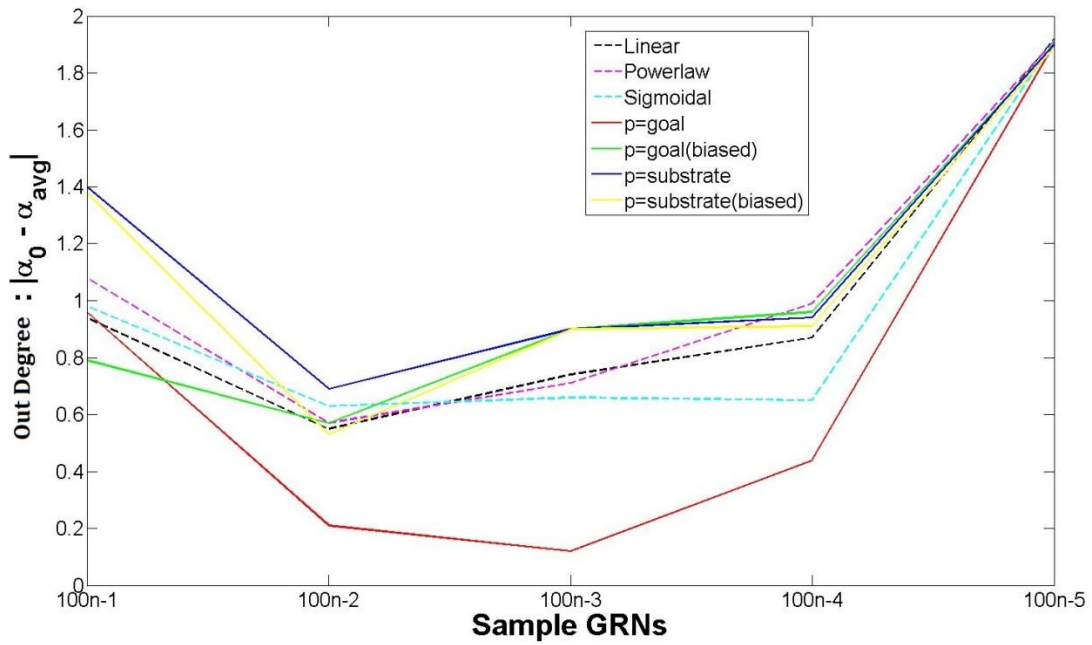


Figure (6-3) MLE difference for out degree distributions

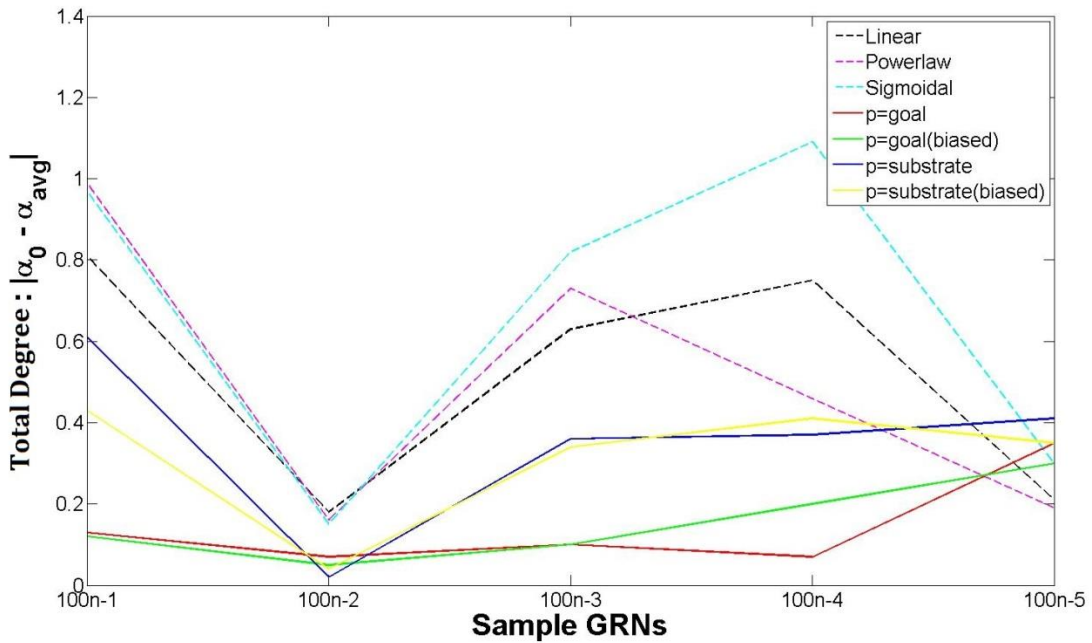


Figure (6-4) MLE difference for total degree distributions

Chapter Seven Growing the network using correct downlink attachment in a game based approach

Preserving the sequence of downlinks that have been correctly added to the grown substrate network would be used in specifying the most repeated subsequences of attached downlinks. This would help in recognizing the recurrent *patterns* of subsequences of downlink attachments and determining the categories of these patterns' sequences as rules of our network grow model. A game based model is designed to use the human choices of adding downlinks and connections instead of using the stochastic model described before. The purpose of the game is to both aid in education activities on complex networks as well as to assess if new rules emerge by exploiting the gamers' intelligence.

Establishing the rules of *purpose-guided* network growing models become necessary since not all networks were evolved using the concept of random connections (such as in *stochastic network formation* models [31], [32] and [30]). Models that use purpose-guided rules of growing networks have mostly focused on economic and social networks such as *network formation games* ([33], [34], [35],[36], [37], [38], [39], [40]). Herein, a complex network formation game is designed to grow complex large-scale networks using human intelligence as a mechanism of growing the network. Basically, a player is given two networks: goal network and substrate network which is a subnetwork of the goal network. The player has to grow the substrate network and required to match the goal network. In general, we used downlinks as a basic unit of the game, where the player has to select a node from the given substrate network's nodes that participate in the downlinks leading to the addition of correct nodes at the correct places.

A gaming software is designed to let a player grow the network starting with a specific number of nodes. We used small networks extracted from the online java software of GeneNetWeaver to use them as goal and substrate networks. Different sizes of networks were generated as shown in figure (7-3)

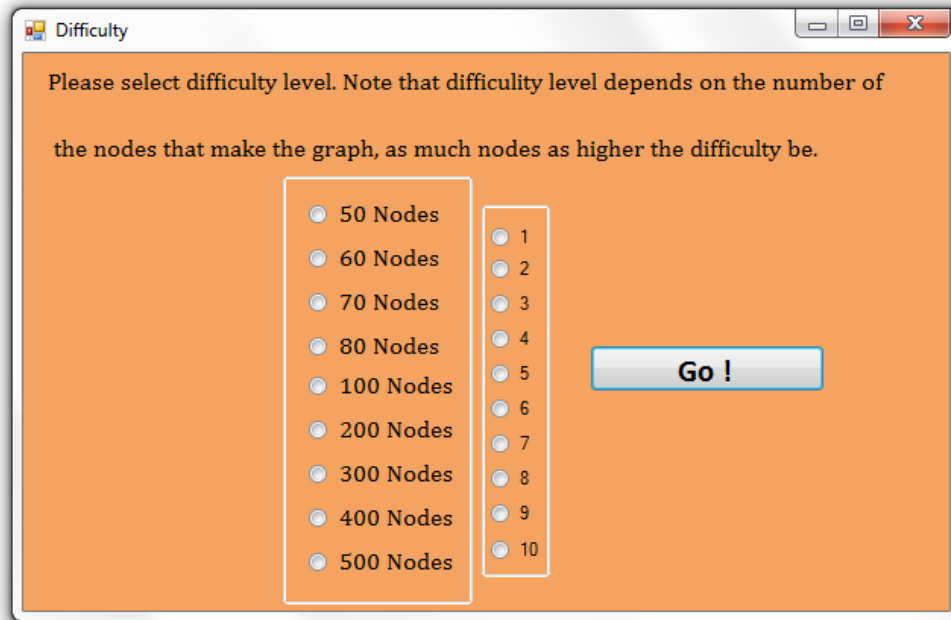


Figure (7-1) Gaming difficulty dialogue box.

The player has to select the size of the network by specifying the number of nodes existing in the network and then choose one of the several random networks of the same size. Once the player clicks the command button “Go” a new window pops up loading the selected network as a goal network and generate a smaller network called the “Substrate Network”. The substrate network will be loaded with transcription factor nodes only, as shown in figure (7-2), showing transcription factor t as red nodes and genes g as blue nodes.

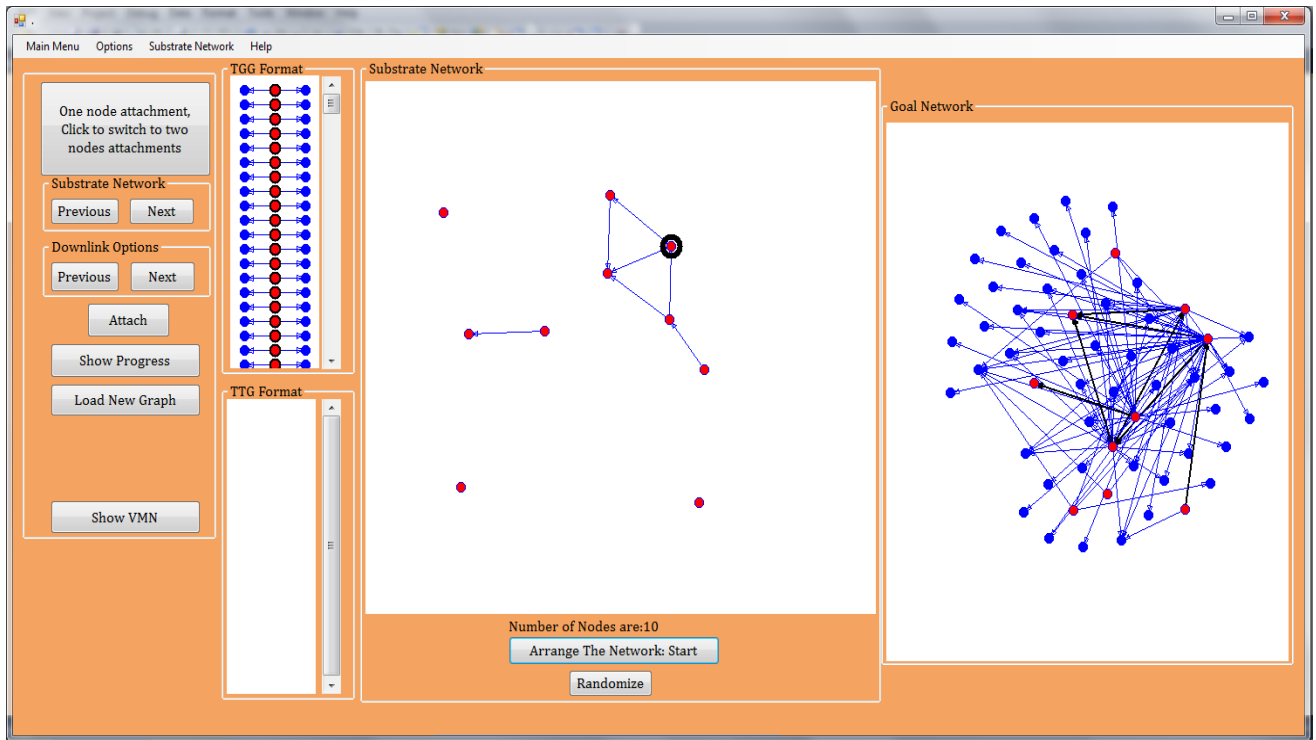


Figure (7-2) Basic interface of the game

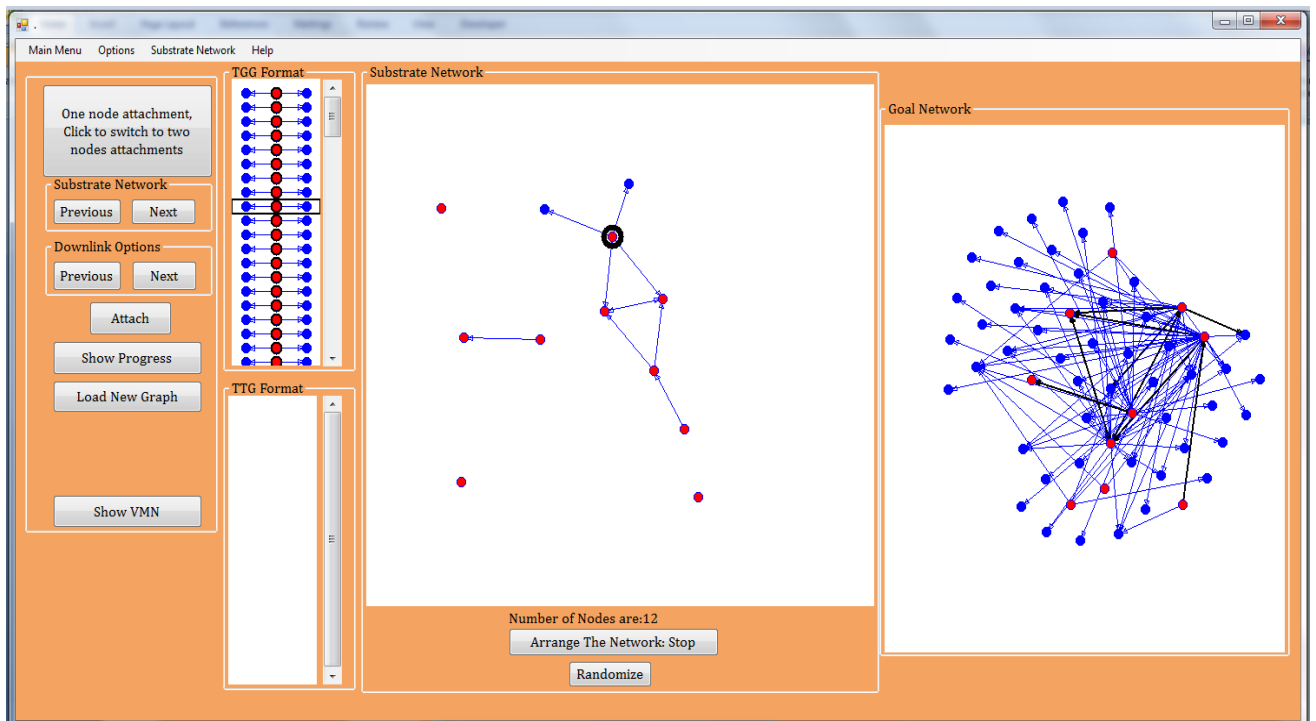


Figure (7-3) Selected t node with the applicable downlinks

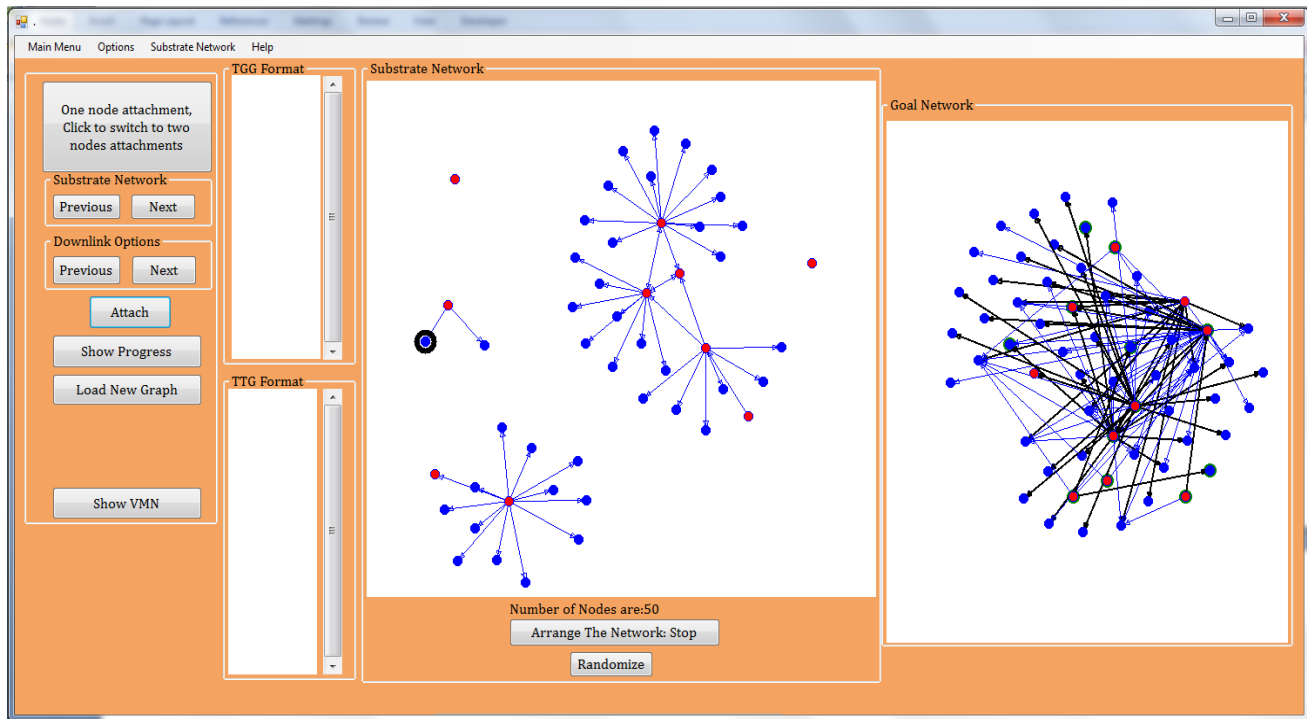


Figure (7-4) Substrate network after adding several downlinks.

Downlink based network growing means that the network is created and increased by adding more nodes and edges to it, with the use of downlinks instead of a single node. Since the substrate network started with very little number of nodes that are weakly connected, a player has to select the right downlink to be added to the best place. To achieve that purpose, players can select the substrate node that might play a critical role in selecting the best incoming downlink with the potential correct nodes to be added. By selecting that node, a list of related downlinks will appear each highlighting the selected node from a list of all the applicable downlinks of the goal network. Moreover, the player is able to look for downlinks that have one and two existing nodes in the substrate network. Therefore, two basic modes of selection are available:

- 1- Downlinks with the selected one node, labeled as one node attachment mode.
- 2- Downlinks with the selected two nodes, labeled as two nodes attachment mode.

Moreover, a player is provided with another two lists of downlinks for simplifying the game playing experience:

- 1- Downlinks with *tgg* format.
- 2- Downlinks with *ttg* format.

Later on, the player would select one downlink from the available downlink list and press “attach”. The process of attachment will exclude the existing shared nodes and add the new applicable nodes with their edges. For example, if the player selects one node mode then two new nodes will be added and if two nodes mode is selected then only one new node is added; for both cases, two edges is added. Nodes and edges are added to the substrate network using the same structure as they appear in the goal network. Specifically, every node in the substrate network and in the goal network has a unique identifier. Note that we do not consider 3-node attachments in the game as that is straightforward to achieve for the gamers. The player has different ways of growing the network, such as connecting the t nodes using two nodes attachment mode or simply adding nodes with one node attachment mode. The complexity of adding downlink to the substrate network using this manner has mainly two issues. *The first issue* is when the player has selected the two nodes mode. In this mode, the applicable lists of *tgg*, *ttg* downlinks will appear in two colors: red for transcription factors and blue for genes as mentioned earlier. The player has to highlight two nodes in the substrate network and then in the shown lists. The player is equipped with the selection of the first *-main* node. The player does not know which second highlighted node in the downlink lists is exactly the second node in the substrate network. The player has to appropriately place the selection over the correct nodes and then press “Attach”. If the player makes a wrong selection, the program will add the node to the right place and show the edge in red color in the goal network to mean it was a wrong addition.

If the player makes a correct addition then the added edges would be black. The remaining untouched edges are displayed in blue. *The second issue* is specifying the best sequence of attachment patterns and downlink-downlink combinations to grow the network to match the goal network. Should the player initially finish all the one node modes and then start the two nodes mode or vice versa? Or is there a specific pattern 1/2 node modes and downlink types that result in the best grown network?

We will be able to see the progress of the players as well as their understanding for the best option to choose to grow the network. By the end of the game, a sequence of types of downlinks and modes that have been added to the substrate network is stored and returned for further analysis. We have used the following formats to save the sequence of attachments:

Mode	Type of downlink	Char
One node mode attachment	<i>tgg</i>	<i>a</i>
Two nodes mode attachment	<i>tgg</i>	<i>b</i>
One node mode attachment	<i>ttg</i>	<i>c</i>
Two nodes mode attachment	<i>ttg</i>	<i>d</i>

Table (7-1) Different modes of attachment's characters.

To properly evaluate the usefulness of this gaming model, we need to generate sufficient statistics by letting thousands of gamers play the network growing game. Generating these statistics is planned as future work for us in this area.

Chapter Eight Conclusions and Further Work

In this thesis, we have presented a directed complex network growing algorithm using the concept of motif additions. Specifically, while existing algorithms in this are grow undirected networks using the preferential attachment model or directed networks using the modified preferential attachment scheme and different attachment kernels, they fail to produce good correspondence in the motif distributions observed in real-world biological networks. Our goal in this thesis was to propose a new paradigm, that of network growing using motifs, which serve as the building blocks of complex networks. In this thesis, we have only considered downlink motifs and also the type of nodes (transcription factor or gene) they contain as downlinks seem to cover most of the nodes and edges in a GRN. Our proposed algorithm can be improved upon by considering other motif structures into the network growing algorithm. We have also implemented a complex network growing game which also considers downlink additions to gain insights on network growing using the intelligence of the gamers.

In the following, we will highlight some immediate future works planned from this thesis:

8.1. Identifying the best initial substrate network structure to be grew

In this proposed study, we will generate several subnetworks from a given set of goal networks and grew these substrates to the size of the goal network. Then we will assess what features of the substrate topologies play a critical role in specifying the best initial substrate network structure, such as the number of t and g nodes, the values of several network statistics and so on. We plan to generate several networks (100, 200, up to 500 nodes each) and extract several hundred subnetworks with different percentages and apply our downlink based growing

model. These statistics will be collected and a support vector machine model (SVM) will be built to identify the factors that increase or decrease the similarity between the grown network and the corresponding goal network.

8.2. Identify the most common pattern of attachment using LCS.

In this work, we want to identify the most recurrent sequence of downlink attachment modes (1, 2, 3 node attachments) and sequence of attachment patterns (ttg, tgg, tt) by using our complex network growing game. We will perform several simulations on several goal networks with different sets of percentages of subnetworks. In each simulation we will record the following:

- 1- String to record the type of the selected substrate downlink to attach to.
- 2- String to record the type of the new downlink.
- 3- String to record the pattern of attachment.
- 4- String to record the number of shared nodes.

Then we would use the largest common subsequence algorithm among each of them to find the most frequent series of growing actions dependently,

8.3. Grow the network using different set of downlink centralities.

Note that only degree centralities were implemented in the VMN of the substrate network for identifying the downlink to attach to. We plan to use different centralities as a downlink selection method from the VMN of the substrate. These centralities include betweenness, closeness, eigenvector, Katz, and many others.

References

- [1] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, “Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles,” *PLoS biology*, vol. 5, no. 1, p. e8, 2007.
- [2] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano, “Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context,” *BMC bioinformatics*, vol. 7, no. Suppl 1, p. S7, 2006.
- [3] Q. Li, B. Zhang, Z. Fan, and A. V. Vasilakos, “Dynamics in small worlds of tree topologies of wireless sensor networks,” *Journal of Systems Engineering and Electronics*, vol. 23, pp. 325 – 334, June 2012.
- [4] C. Bensong, G. N. Rouskas, and R. Dutta, “Clustering methods for hierarchical traffic grooming in large-scale mesh wdm networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, pp. 502– 515, August 2010.
- [5] I. Shmulevich and E. R. Dougherty, *Probabilistic Boolean Networks – The Modeling and Control of Gene Regulatory Networks*. SIAM, 2010.
- [6] J. Feng, J. Jost, and M. Qian, *Networks: from biology to theory*. Springer, 2007.
- [7] H. Kitano, “Biological robustness,” *Nat Rev Genet*, pp. 826–837, November 2004.
- [8] H. Kitano, “Towards a theory of biological robustness,” *Mol Syst Biol* 3, September 2007.

- [9] R. J. Prill, P. A. Iglesias, and A. Levchenko, “Dynamic properties of network motifs contribute to biological network organization.,” *PLoS Biol.*, November 2005.
- [10] S. A. Kauffman, *The origins of order: self-organization and selection in evolution*. Oxford University Press, 1993.
- [11] U. Alon, “Network motifs: theory and experimental approaches,” *Nat. Rev. Genet.*, vol. 8, no. 6, pp. 450–461, 2007.
- [12] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network motifs: Simple building blocks of complex networks,” *Science* 25, vol. 298, pp. 824–827, October 2002.
- [13] S. Magnan and U. Alon, “Structure and function of the feed-forward loop network motif,” *Proc. Natl. Acad. Sci. USA*, October 2003.
- [14] K. Wu and C. V. Rao, “The role of configuration and coupling in autoregulatory gene circuits.,” *Molecular microbiology*, vol. 75, pp. 513– 527, Jan. 2010.
- [15] U. Alon, *An Introduction to Systems Biology Design Principles of Biological Circuits*. July 2006.
- [16] J.-R. Kim, Y. Yoon and K.-H. Cho, Coupled feedback loops form dynamic motifs of cellular networks, *Biophys J.*, 94 (2008), 359–365.
- [17] Y. keun Kwon and K. hyun Cho, “Boolean dynamics of biological networks with multiple coupled feedback loops,” 2007.

- [18] M. Isalan, C. Lemerle, K. Michalodimitrakis, C. Horn, P. Beltrao, E. Raineri, M. Garriga-Canut, and L. Serrano, “Evolvability and hierarchy in rewired bacterial gene networks,” *Nature*, vol. 452, pp. 840–845, Apr. 2008.
- [19] M. Mayo, A. F. Abdelzaher, E. J. Perkins, and P. Ghosh. “Motif participation by genes in *E. Coli* transcription”. *Frontiers in Physiology*. August 2012.
- [20] P. L. Krapivsky, S. Redner, and F. Leyvraz, “Connectivity of Growing Random Networks,” *Physical Review Letters*, vol. 85, pp. 4629–4632, Nov. 2000.
- [21] S. Redner, How popular is your paper? An empirical study of citation distribution, *Eur. Phys. J. B* 4,131 (1998).
- [22] Pimm, S. L., 1991, *The balance of Nature* (university of Chicago, Chicago).
- [23] Abello, J., P. M. Pardalos, and M. G. C. Resende, 1999, in *External Memory Algorithms*, edited by J. Abello and J. Vitter, *DIMACS Series in Discrete Mathematics Theoretical Computer Science* (American Mathematical Society), p. 119.
- [24] Newman, M. E. J. (2010). *Networks: An Introduction*. New York: Oxford University Press.
- [25] Pellegrini Matteo, Haynor David, Johnson JM: Protein interaction network, *Expert Rev Proteomics* 2004, 1(2).
- [26] Jeong, H., B. Tombor, R. Albert, Z. N. Oltvai and A. –L. Barabasi: The large scale organization of metabolic networks, 2000, *Nature (London)* 407, 651.
- [27] Crombach, A., and Hogeweg, P. (2008). Evolution of evolvability in gene regulatory networks. *PLoS Comput. Biol.* 4, e1000112. doi: 10.1371/journal.pcbi.1000112
- [28] Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002). Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* 31, 64–68.

- [29] Jeong, H., Néda, Z., and Barabási, A. L. (2003). Measuring preferential attachment in evolving networks. *Europhys. Lett.* 61, 567–572.
- [30] Barabási, A.-L., and Albert, R. (1999). Emergence of scaling in random networks. *Science* 286, 509–512.
- [31] Bollabas, B. 2001. *Random Graphs*. Cambridge University Press.
- [32] Watts, D. and Strongatz, S. 1998. Collective dynamics of small-world networks. *Nature* 393, 409–410.
- [33] Tardos, E. and Wexler, T. 2007. Network formation games and the potential function method. In *Algorithmic Game Theory*. Cambridge University Press, 487–513.
- [34] Jackson, M. O. 2005. A survey of models of network formation: stability and efficiency. In *Group Formation in Economics: Networks, Clubs and Coalitions*. Cambridge University Press.
- [35] Fabrikant, A., Luthra, A., Maneva, E., Papadimiriou, C., And Shenker, S. 2003. On a network creation game. In *Principles of Distributed Computing (PODC)*.
- [36] Borgs, C., Chayes, J., Ding, J., and Lucier, B. 2011. The hitchhiker’s guide to affiliation networks: A game-theoretic approach. In *Innovations in Theoretical Computer Science (ITCS)*.
- [37] Albers, S., Eilts, S., Even-Dar, E., Mansour, Y., and Roddity, L. 2006. On Nash equilibria for a network creation game. In *Symposium on Discrete Algorithms (SODA)*.
- [38] Brautbar, M. and Kearns, M. 2011. A clustering coefficient network formation game. In *Symposium on Algorithmic Game Theory (SAGT)*.
- [39] Even-Dar, E., Kearns, M., and Suri, S. 2007. A network formation game for bipartite exchange economies. In *Symposium on Discrete Algorithms (SODA)*.
- [40] Even-Dar, E., and Kearns, M. 2006. A small world threshold for economic network formation. In *Neural Information Processing Systems (NIPS)*.

- [41] Clauset A., Shalizi C. R., Newman M. E. J. (2009). Power-law distributions in empirical data. *SIAM Rev.* 51, 661–703. doi: 10.1137/070710111.
- [42] Hoogenboom J. P., den Otter W. K., Offerhaus H. L. (2006). Accurate and unbiased estimation of power-law exponents from single-emitter blinking data. *J. Chem. Phys.* 125, 204713.
- [43] Schaffter, T., Marbach, D., and Floreano, D. (2011). GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* 27, 2263–2270.
- [44] Choe, H, Ghosh, P and Das, S, 2010. QoS-Aware Adaptive Data Reporting in Wireless Sensor Networks, Elsevier Computer Communications, 33:11, pp. 1244-1254.
- [45] Choe, H, Ghosh, P and Das, S, 2009. Cross-Layer Design for Adaptive Data Reporting in Wireless Sensor Networks. The 1st International Workshop on Information Quality and Quality of Service for Pervasive Computing (IQ2S 2009), held in conjunction with Percom 2009, TX, USA, pp. 1-6.
- [46] Ghosh, S, Ghosh, P, Basu, K and Das, S, 2005. GaMa: An Evolutionary Algorithmic Approach for the Design of Mesh-Based Radio Access Networks. 30th IEEE Conference on Local Computer Networks (LCN), Sydney, pp. 374-381.