Identification of genes involved in T-cell differentiation

Master's thesis
Payam Emami Khoonsari
Institute of Biomedical Technology
University of Tampere
December 2012

Acknowledgments

I would like to give my best thank to my supervisor Csaba Ortutay for being my mentor during my study in university of Tampere. I owe him a lot because of his guidance, patience and kindness he provided to me during my thesis. He helped me a lot in all stages of my study and the thesis. I also would like to thank Matti Nykter for reviewing and scoring my thesis.

I'm very grateful to Martti Tolvanen because of the excellent teaching, help and coordinating he provided to me during the master degree. He is one of the best bioinformaticians at IBT. I learned a lot from him.

I would like to thank and appreciate one of my best friends and one of the most brilliant students in the group, Harlan Barker, not only for the tips he gave me during my study, but also for reviewing my thesis and providing hints.

I would like to appreciate all the teachers and staff in IBT for providing a good environment for learning and doing research.

I would like to give my best thanks to my family especially my father and mother for all the goodness they have done to support and maintain my life and study.

At last but not the least, from the depths of my heart I thank my wife Elham for the support, love and motivation she gave to me.

Payam Emami Khoonsari

December, 2012

MASTER'S THESIS

Place:	UNIVERSITY OF TAMPERE Faculty of Medicine Institute of Biomedical Technology
Author:	Payam Emami Khoonsari
Title:	Identification of genes involved in T-cell differentiation.
Pages:	114 pages + appendix + figures
Supervisors:	Csaba Ortutay, Acting Professor
Reviewers:	Csaba Ortutay, Acting Professor, Matti Nykter, Professor
Date:	December 2012

Abstract

Background

T-cells are involved in many immune functions. Each function is carried out by specific subset of T-cells. All T-cell sub-types originate from one stem cell and the fate of each cell is dictated by its pattern of gene expression. The pattern of gene expression is the direct outcome of genetic regulatory network which can be visualized as a network containing nodes (genes) with edges (interaction) between them. Simulation of the dynamics of gene regulatory networks reveals several attributes of not only the network itself but also the pattern of gene expression of different developmental or differentiation processes. Since gene regulatory networks often include thousands of genes, the network has to shrink or contain only a small subset of possible states of the large networks can be explored through the simulation. Therefore, different methods are needed to extract information from gene regulatory network.

Methods

Central T-cell Network (the main gene regulatory network in T-cells) is used to start the simulation with customized random Boolean networks. Because of large scale of the network an initiative approach was used to reduce the number of possible states which were needed to be explored. Graph theory was used to find the attractors. GO analysis was used to find information in attractors. Clustering methods were applied on attractors in order to find groups of interesting gene states. Finally, data mining and microarray data analysis were utilized to verify the simulation system.

Results

Forty experiments resulted in 833 attractors (with period 2 or 4). GO analysis (performed on most frequent attractors) resulted in no significance in T-cell differentiation processes. Clustering methods classified each type of attractors to exactly two different clusters. The simulated gene expression divided the genes into 3 groups, and GO analysis did not show significance in any differentiation process. The result of gene expression ratio of CD4+ and CD8+ cells showed a significant difference between the microarray data experiments and simulated gene expression ratios. Finally, the result of data mining suggested that CD4+ cells were located in one of the clustered attractors.

Conclusion

A new environment was developed to simulate the dynamics of the gene regulatory network in T-cells. A novel approach was used to reduce the state space and in finding attractors. The resulting attractors were analyzed by several experiments. Although the genes involved in differentiation processes were distributed sporadically on the attractor clusters, CD4+ related genes were clustered in one group. This indicates the usability of the system for distinguishing different cell types. The result also indicates that the system can be used not only for T-cells but also for any biological network. A conclusion can be drawn that this new system is applicable for different networks but more experiments with different parameters are needed to verify the simulation system.

Table of Content

Abbreviation

Natural Killer	NK
T cell receptors	TCR
T helper cells type 1	TH1-Cells
Major Histocompatibility Complex	MHC
allergic airway disease	AAD
Follicular Helper CD4 T Cells	TFH
Cytotoxic T-cells	CTL
reactive oxygen species	ROS
Regulatory T-cells	Treg
Natural Killer T-cells	NKT
central memory T cells	TCM cells
effector memory T cells	TEM cells
Protein-Protein Interaction	PPI
Matthews's correlation coefficient	MCC
Central T cell Network	CTN
Cellular Automata	CA
Random Boolean Networks	RBN
classical RBN	CRBN
Asynchronous Random Boolean Networks	ARBNs
Deterministic Asynchronous Random Boolean Networks	DARBNs
Generalized Asynchronous Random Boolean Networks	GARBNs
Deterministic Generalized Asynchronous Random Boolean Networks	DGARBNs
Discrete Dynamical Networks	DDN
Probabilistic Boolean Networks	PBN
Pearson correlation coefficient	PCC
Rank correlation coefficient	RCC
support vectors machine	SVM
significance analysis of microarrays	SAM
false discovery rate	FDR
Gene ontology	GO
enrichment score	ES
Visualization and Integrated Discovery	DAVID
unweighted pair group method with arithmetic mean	UPGMA
effect components	EC
random components	RC

1. Introduction

The human body consists of several types of cell each of which performs different functions which are critical for survival of the living organism. The immune system which performs the role of attacking pathogens is one of the most crucial systems in almost all organisms. This system in humans consists of several types of cell which act together to ensure human survival. One the most interesting subsets of these cells are called T-cells. T-cells also have several subtypes which can be distinguished by their specific functions and pattern of gene expression.

Differentiation is the process of acquiring specialized functions from a general cell (stem cells). Since T-cells and many human cells are derived from the same stem cells, finding the fate of each cell in different conditions is of great importance. That is because cell differentiation usually takes place in different conditions and the differentiation itself is directly affected by the pattern of gene expression. Therefore, finding the genes involved in the differentiation process is a great help for predicting the fate of cells. By finding this pattern, not only can one predict cell conditions but also that information can be used for treatment of many diseases such as cancer. The pattern of genes expression is the result of genes regulatory network where there may be several thousand genes interacting in a highly interconnected network. The genes may regulate their descendants through a directed link and may be regulated by neighboring genes through undirected network genes interaction called co-regulatory network. The state of a gene in this network is not only dependent on the neighboring genes, but also on genes throughout the whole network. Because the number of expressed genes in each cell that participate in this network is often extremely large, computational methods are used to reduce the number of required genes for experimental analysis. Computational methods are also utilized to predict the fate of cells by the pattern of gene expression.

The main goal of this study is to find the genes involved in T-cells differentiation by simulation of dynamics of the most important gene regulatory network in the T-cells. Since the genes involved in T-cell differentiation (e.g. stem cells) are almost well characterized, the study focuses on finding not only finding genes involved in T-cell differentiation from the stem cells, but also the genes which cause T-cells to differentiate into sub-types. The ultimate goal of this thesis is to propose a new approach for simulating of dynamics of any experimentally pre-defined biological network and proposing a solution for limitations of other previously known approaches for simulation of specific and customized network. Since this network is not a usual network which is used by normal modeling techniques, special approaches were used to simulate and find information in order to find the genes which are involved in the differentiation process and also verifying the simulation system.

2. Literature review

2.1 Immune System

The aim of this chapter is to briefly introduce the mechanism in the immune system which is responsible for protecting the body from threats posed by pathogens. Pathogens can be regarded as any harmful microorganisms such as bacteria, viruses and parasites which frequently threaten the body and cause disease. For example, cholera, AIDS and trichinosis are caused by bacteria, viruses and parasites, respectively. After a pathogen threatens the body, the immune system has two major tasks; first, detection of the pathogen and second, elimination or killing of it. These two principal works are done by a multilayer, hierarchical system which consists of two major parts called innate and adaptive immune systems. Both of these systems are described in the next sections. The complete description of the immune system is out of the scope of this thesis so for preliminary introduction readers are advised to use available textbooks.

2.1.1 Innate immune system

Pathogens wanting to attack a body have to pass the first line of defense which is the skin. Skin provides a barrier to invading microbes and is usually only permeable through cuts or abrasions. Digestive and respiratory systems are also ways which microbes enter the body. Microbes which try to enter the body through nose or lung usually trigger sneezing which pushes them out of the body. The ones which try to enter through digestive system will face a strong acid that destroys many of them and even if they can survive they have to pass through the walls of the digestive system which makes many of them unable to enter the destination tissues. If a few pathogens can pass the first line of defense, they will face the second line of defense which is philological conditions such as temperature, PH and oxygen tension which limit the microbial activity and growth. The third layer is the innate immune system which is not specific to a particular pathogen (unlike the adaptive system). This system provides the first rapid attack against pathogens by several mechanisms such as complement, endocytic and phagocytic systems. The complement system has two major tactics which are called lysis and opsonization. Lysis is a process of rupturing a bacterial membrane resulting in the bacteria's destruction. In opsonization, bacteria are covered with complement allowing macrophages to easily detect them. Macrophages are cells which has several critical roles in the body, in immunity they engulf the detected bacteria (identified either through their receptor or complement) and destroy them. Another way of activating macrophages is by binding to cytokines which have a signaling role in the body. Cytokines are secreted by many cells (not only immune cells) in the body and they usually cause an inflammatory response in infected or damaged tissue. Inflammation in turn causes increased local blood flow (causes attracting more immune cells) and temperature (fever) which is beneficial by reducing activity of pathogens or increasing the intensity of adaptive immune response. The innate immune system has another weapon which is called interferon. These proteins are secreted when cells are infected by viruses and they inhibit viral replication and also activate Natural Killer (NK) cells. Natural Killer cells bind the normal cells and receive an inhibitory signal which keeps them inactive. This signal is produced by normal cells but infected cell cannot inhibit NK cells and they become activated and trigger apoptosis (programmed cell death) which kills the infected cells. Innate immunity is not only defending the body but giving adaptive immune system time to build up stronger response to pathogens. If pathogens can survive the innate system, they will face the final line of defense which is specifically built for each of them.

2.1.2 Adaptive immune system

As its name suggests, the adaptive immune system can learn to detect specific kinds of pathogens. When a pathogen which has been not encountered yet is detected by the immune system the learning process starts. It usually takes several days to a few weeks, to clear an infection and learn that specific pathogen. After that, if that pathogen is encountered again the second response will be very rapid and often with no indications of infection. Adaptive immunity consists of a type of white blood cells called a Lymphocyte. On the surface of Lymphocytes there are proteins called receptor which recognize and bind epitopes on the surface of pathogens. These receptors are specific to a few similar epitopes on different pathogens. They may differ between Lymphocytes but are identical on a single Lymphocyte, making each one specific to particular pathogen. A Lymphocyte gets activated only when it binds to a pathogen by high affinity and this makes them general to many similar pathogens. As mentioned, the adaptive immune system must first detect the pathogen and then learn and remember the pathogen for the next response. Both of these issues are addressed by a type of Lymphocyte called a B-cell. After activation, B-cells move to lymph nodes where the adaptation process occurs through cell division. B-cells are subject to mutation, termed somatic hypermutation. Each B-cell clone binds to captured pathogenic epitopes. If they have high affinity, they will be released and differentiate into plasma or memory B-cells. Otherwise they will die after a short period. Plasma B-cells are able to secrete antibodies which have critical roles in immunological defense either by marking pathogens (opsonization) which makes them easier targets for phagocytes (white blood cells) or by neutralizing them. Memory B-cells will proliferate after a successful response and in doing so they promote defense against type of epitopes they previously recognized for the second response. They are also able to act against similar pathogens and so similar infections can be treated in a short time. There is one important issue associated with the adaptation to pathogens through somatic hypermutation which is called autoimmunity. This happens when the immune system attacks self-cells. Handling of this problem is the responsibility of another type of Lymphocytes which are called T-Cells. This type of cell has many mechanisms which are described in the next section.

2.1.3 T-Cells

T-cells are so-called because they mature in the thymus. They can be distinguished from other lymphocytes by the presence of T cell receptors (TCR) on their surface. There are four sub-types of T-cells, namely, Helper, Cytotoxic, Memory, Regulatory and Natural Killer T-cells.

2.1.3.1 Helper T-cells

Helper T-cells can be distinguished from other sub-type by the presence of CD4 protein on their surface. They differentiate to six different sub-types, TH1, TH2, TH9, TH17, TH22, and TFH. The major role of these cells is providing assistance to other immune cells in biological processes. As discussed in the previous section, Helper T-cells help B-cells to solve the autoimmunity issue. Many of the self-epitopes are available in thymus and T-cells are exposed to them. If a T-cell is activated by binding to one the presented self-epitopes it will die by a process called negative selection. Those cells which survive in this process will leave the thymus and undertake their responsibilities. B-cells mature in bone marrow and this is not enough to ensure they are not self-reactive. That’s because of hyper-mutation process which may cause B-cells to be auto reactive. T-cells will help B-cells handle this problem by a process called co-stimulation. Intuitively, a specific subset of T-cells called T helper cells type 1 (TH1-Cells) are involved in this process. They are produced when naive T helper cells differentiate into TH1-Cells in presence of Interleukin 12 (IL-12) [1]. Specifically, B-cells have to receive two different signals to get activated. The first signal will be received when B-cells bind to the target of high binding affinity and the second signal will be emitted by TH1-Cells. After B-cells recognize and bind to their target. They engulf the pathogen peptide and present the peptide on their surface, using a molecule called Major Histocompatibility Complex (MHC). If TH1-Cells bind MHC on the surface of a B-cell it ensures that B-cell has selected a non-self-peptide, it can receive the second signal and become activated. In rare cases even TH-cells can be auto reactive and bind the self-peptides. So there is another co-stimulation process for TH1-cells in which they require two signals to be activated. One signal is provided by binding affinity threshold and another signal is given by cells of innate immune system.

TH2-cells are another type of helper T cell which stimulates macrophage when they are affected by bacteria in their vesicles. TH2-cells are produced by naive T helper cell differentiation in presence of Interleukin 4 (IL-4) [1] and they are involved in immune responses against intra-cellular pathogens whereas TH1-cells are involved against extra-cellular pathogens.

Naive T helper cells differentiate into T Helper-cell type 9 (TH9-cells) in presence Interleukin 9 (IL-4) [2]. TH9-cells play many roles in different diseases such as contributing to inflammation and allergic disease they also have a role in allergic airway disease (AAD) [3]. It also has been shown that they are involved in immunity against Helminth infections and intestinal parasites [4] [5].

T Helper-cells type 17 is established when TGF-β and IL-6 are presented in a cell [6]. The main function of this category of T helper cells is to clear pathogens which cannot be removed by other immune systems because their clearance needs strong inflammatory response [7]. Reportedly, TH17-cells also have important effects in many diseases, especially they play a “pro-inflammatory” role against autoimmune diseases but they also have role against fungi and parasites [8].

TH22-cells are another type of TH-cells which are expressed when naive TH-cells differentiate in presence of TNF-ɑ and IL-6 [9]. Increased innate immune response and regeneration are caused by IL-22 which is produced by TH22-cells in many tissues such as skin and liver [10].

The last type of TH-cells is called Follicular Helper CD4 T Cells (TFH) which are generated by TH-cells differentiation in presence of B-cell CLL/lymphoma 6 (Bcl6), IL-6, IL-21 and CXCL13. Similar to the TH-cell, TFH-cells provide help to B-cells (through direct physical interaction) and also allow them to form plasma and long-lived memory B-cells. GC TFH-cells have a role in regulation of B-cells differentiation [11] and they are also involved in autoimmune diseases such as systemic lupus erythematosus [12].

2.1.3.2 Cytotoxic T-Cells

Cytotoxic T-cells (CTL) are a class of T-cells which kill other infected cells. Their targets can be virus infected cells, cancer cells or the cells infected by intra-cellular pathogens. They can be distinguished from other T-cells by presence of CD8 molecule on their surface. They recognize and bind infected cells using TCRs which are specific to particular antigens that cause stimulation of CD8+ cells. As discussed in the previous section, MHC helps TH-cells to confirm B-cells have engulfed non-self-cells and then give the second signal to B-cells to become fully activated. There is another class of MHC called MHC I molecule. These molecules are present inside almost all of the cells in the body. When cells are infected by intra-cellular pathogens such as virus, they present inside peptides (antigens) on the surface of cells. CD8+ cells recognize and bind to the combination of MHC I and peptides (pMHC I) through a CD8 glycoprotein which also plays an important role for differentiation of naive T-cells to CD8+ cells. After binding to the target cells, they kill the infected cells by utilizing two mechanisms. First, when they bind to the cells, cytoplasmic granules are discharged and perforin molecules are injected into plasma membrane of the target cells. Next, granzymes enter the cells through pore created by the perforin injection. Granzymes which are serine proteases cut the peptide bonds and have two classes, A and B. Granzyme A goes to mitochondria and kills the cells by producing reactive oxygen species (ROS). Granzyme type B causes apoptosis by activation of the caspase cascade. The second mechanism of CTLs is based on Fas ligand (FASL) protein. FASL is a transmembrane protein which is expressed on the surface of Cytotoxic T-cells. When CTLs bind to the target, the interaction between FASL and FAS protein (which is expressed on the target surface) triggers apoptosis, killing the infected cell [13]. Cytotoxic T-cell activation is also regulated through two signals. The first signal is given by the cell receptor when bound to MHC I. The second signal is presented by a co-stimulatory mechanism, specifically by CD80 and CD86 proteins which are detected by CD25 protein on the T-cells surface. Similar to T helper cells, if the cell receives only one type of signal it will undertake apoptosis. This type of cell also plays an important role in many disorders such as hepatitis B [14] and autoimmune diseases [15]

2.1.3.3 Regulatory T-Cells

As discussed earlier, T-cells are exposed to most self-peptides to make sure that they are not auto-reactive and don't bind self-components. Although this process is very strict some auto-reactive cells may escape negative selection. Also, when a successful response is given to pathogens, effector T-cells have to become inactivated, otherwise they may hurt the body. Regulation of immune system cells is the major role of Regulatory T-cells. Regulatory T-cells (Treg) are regarded as a major regulator of the immune system. They have a pivotal role in preventing autoimmunity and diseases such as type 1 diabetes [16]. The most important factor for their development and function is forkhead box P3 (FOXP3) but they also express CD25 and cell surface CTLA-4. It also has been shown that Interleukin 2 (IL-2) is a critical factor involved in Tregs development and function. Tregs are divided into three sub-types called natural Treg cells (nTreg, also called CD4+CD25+), induced Treg cells (iTreg) and T-helper 3 (Th3) cells. nTregs are developed in the thymus and their major function is inhibiting other T-cells from binding to self-components. iTregs are thought to be derived from CD4+ T-cells. They are most abundant in mucosal surfaces but have variety of roles in different tissues. For example in Placenta, they prevent the mother's immune system from attacking the fetus. Th3 cells are regarded as mediators of oral tolerance and apply their effect by secreting transforming growth factor-β [17].

There are four mechanisms which are used by Tregs to perform their regulatory roles: suppression by inhibitory cytokines such as IL-10, IL-35 and TGFβ; cytolysis through secretion of granzyme A and perforin; metabolic disruption not only through depleting of IL-2 which T-cells need to survive but by expression of the CD39 and CD73 which in turn causes generation of adenosine and suppression of effector T-cells; finally they suppress other immune cells through modulation of dendritic-cells which affects T-cell activation [18]. Deficiency in regulatory T-cells may cause many diseases, especially autoimmune diseases. They play important roles in cancer, diabetes, infectious diseases [19] [20] [21].

2.1.3.4 Natural Killer T-cells

Natural Killer T-cells (NKT) are another type of immune cells which are characterized by having TCRs which recognize glycolipid antigen (presented by CD1d molecule) and also having characteristics of natural killer cells (NK) by possessing their receptors. Several researchers have provided different sub-types of NKT-cells but the main classification divides NKT-cells to two different groups called type I and type II. Type I natural killer T-cells express invariant Vα24Jα18 in human and type II presents more diverse Vα24Jα18. NKT-cells are thought to form a bridge between the innate and adaptive immune systems. On one hand, they present limited TCRs which are specific to lipids allowing them to recognize presented lipids which are not detectable by other members of adaptive immune system. But on the other hand, they have the characteristic of rapid response which is an attribute of innate immune system. They also play a regulatory role in which they call other immune cells such as members of innate system and CTLs and TH-cells in adaptive immune system. Upon activation, NKT-cells produce a substantial amount of TH1 and TH2 cytokines which in turn causes pro-inflammatory and immunosuppressive effects. NKT-cells also produce CD40L and FASL which are effector molecules and play a role in suppression of tissue destruction and autoimmunity.

Both types of NKT-cells have a variety of roles in disorders and infections such as bacterial and parasitic infections, endocrine diseases, neurologic and rheumatologic diseases, murine tumor, and type I diabetes [22] [23] [24].

2.1.3.5 Memory T-cells

As discussed earlier, when a body is re-infected by the same or similar pathogens, the immune response will be stronger and faster. This is because the immune system uses memory cells to identify pathogens which it has previously encountered. The mechanism is almost the same as that of B-cells, but in T-cells there are two type of memory cells called central memory T cells (TCM cells) and effector memory T cells (TEM cells) each of which can be either CD4+ or CD8+ cells. TCM cells express CCR7 and CD62L receptors and are located in secondary lymphoid. They do not have effector ability but when they face antigens they quickly proliferate and differentiate to effector T-cells so they have common ability with stem T-cells. TEM cells don’t express either of CCR7 or CD62L, but they can move to peripheral tissues and have effector activities [25]. They apply their effector role by secretion of cytokine [26]. The differentiation process of memory T-cells is not fully understood but they are two proposed models. In the first model which is also called linear model, memory T-cells are directly derived from effector T-cells (CD4+ and CD8+). At the earliest stage, naive T-cells differentiate to effector T-cells in presence of antigens. After clearance of pathogen, the majority of effector T-cells will die and the remaining cells go to the third and final stage which is differentiation to memory T-cells. But in certain cases such as presence of inflammatory milieu, T-cells can differentiate to memory type without going to the effector cells differentiation stage [27].

2.1.3.6 T-cells Differentiation

Cellular differentiation is a process by which general cells become more specialized cell types. The human genome contains more than 20,000 genes which code for proteins. Cells of any specific type normally express 10 to 20 percent of the whole genome’s coding genes. Expression of a particular sub-set of genome depends on several factors such as cell function or its environment and tissue. Each cell type is defined by a specific pattern of gene expression and switching between different gene expression patterns can often cause a transition between different cell types. Gene expression is the result of a gene regulatory network wherein a gene receives input signals and may be inhibited or expressed. There are several networks for different sets of functions in cells. The composite of all networks constitutes the regulatory network of the cell. The result of a genetic regulatory network is that some genes which are expressed in almost all of the cell types (housekeeping genes) and some genes are specific to different cells or different developmental stages (cell type specific genes) [28]. The cells which are capable of differentiating into all cell types are called totipotents and the cells which can differentiate into a slightly more limited scope, all except extraembryonic cell types are called pluripotents (stem cells). This cell type is able to produce other stem cell for maintaining their population and also generate more specialized stem cells (multipotents) such as blood stem cells which are themselves capable of producing red blood and white blood cells. Multipotents generate specialized cell types for different tissues or structures. The differentiation process starts from totipotent (such as zygote), passes pluripotent stage to multipotent and finally finishes at the most specific cell types.

The process of specialization of the T-cells starts from pluripotents stem cells where they differentiate into less general lymphoid stem cells (multipotents). Lymphoid cells become more specialized and divide into T, B and natural killer (NK) lymphocytes. NK cells a final stage of specialization and will not differentiate further. Naive B lymphocytes differentiate into plasma and memory B cells and T-lymphocyes divide into naive CD4+/CD8+, memory, regulatory, and natural killer T-cells. Naive T-lymphocyes will also differentiate into very specialized types of cells with the responsibility of acting against pathogens (figure 2.1).

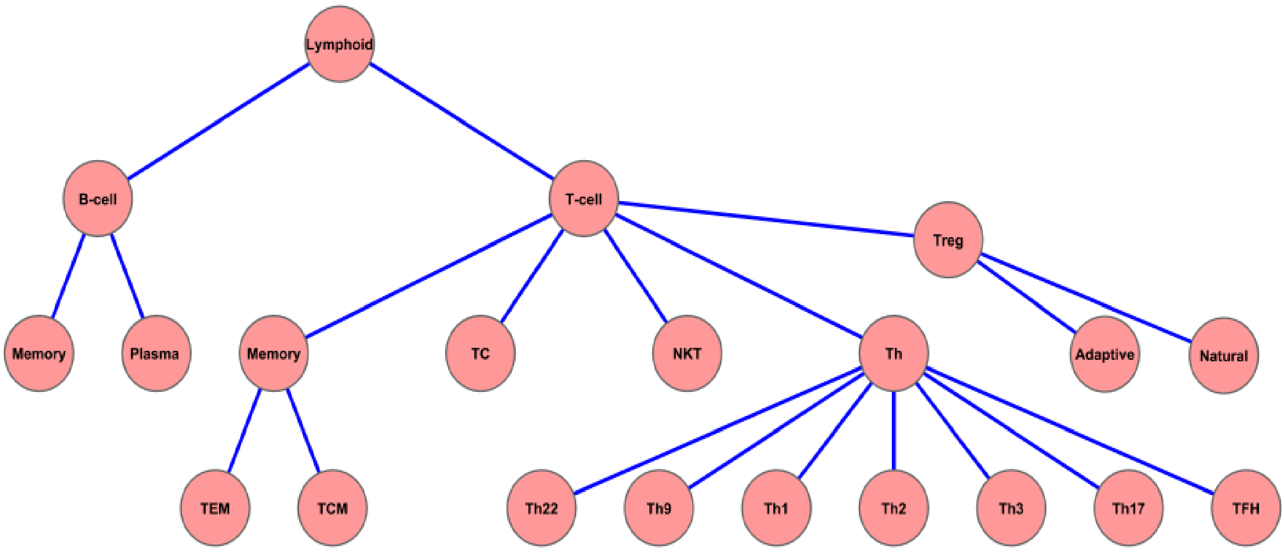


Figure 2.1. T-cell Differentiation

2.1.3.7 T-cells Main Regulatory Network

Since this study focuses on Protein-Protein Interaction (PPI) in T-cells, this section is devoted to the work that has been done by Gabriel Teku and others [29] where they characterized the most important gene interaction network in T-cells. PPI means contact between several proteins at their functional sites. PPIs can be part of cellular context, cell-type, and tissue specific. Study of dynamics and structures of PPIs can reveal many aspects of protein function and regulatory process in the body.

A PPI network can be represented by a set of nodes and connecting edges between them. Since they can be shown using a graph, there are several approaches for studying them by graph theory methods. As protein interactions in many organisms are extremely complex, in many cases study of the entire network is infeasible. Many articles have shown that sub-graphs in the complex networks may be related to specific biological processes, different transduction and even disease pathways [30] [31]. There are many techniques for identifying sub-graphs related to different cellular process or signal networks [32] [33]. Unfortunately, due to large scale of networks in many organisms, these approaches are infeasible in many cases. An interesting approach introduced by [33] identifies cell-type specific sub-graphs in the large network without having computational complexity problem (a description to the approach is given but for a comprehensive description sees [29]).

In this project the researchers obtained 1323 proteins from Immunome Knowledge Base

(IKB) [34] and the KEGG Pathways database [35]. After obtaining protein interactions from iRefIndex database [36], 353 proteins which did not have any interaction with other proteins in the data-set are removed. Links between proteins were supplemented by the Pearson correlation coefficient (see statistics section) of gene expression information from microarray studies of T-cells using many microarray data analysis techniques (see microarray data analysis section). After retrieving 22 time series experiments (containing 262 data-sets), they separated two-color, non-affymetrix experiments and the experiments for which raw data was not available into different groups. Afterwards, normalization methods for each of the experiments were performed (on each group with different methods) and effect size of the individual experiments was calculated. Outliers were removed using k-means and 19 remaining experiments were used to find gene correlation value. A series of steps were performed on each of the 19 experiments, except as noted otherwise: First, a Pearson correlation coefficient for gene pairs was computed. Second, Fisher's z transformation Reference Average Weighted Effect Size for gene pairs was. Then, Average Weighted Effect Size for all 22 data-sets was calculated resulting in 22 tables. Euclidean and weighted Manhattan distances algorithms were applied on the tables to find the outliers (using clustering methods). Finally, after removing 3 data sets, the gene correlation coefficient values were computed based on 19 remaining tables (for formula and complete description see the original article).

So far, a large network of 617 proteins and links between them (associated with gene expression correlation value) has been made. They performed network decomposition with an iterative algorithm that starts with PPI network and identifies a link with least weight (correlation value) and remove it. Afterward, the algorithm checks if there are disconnected nodes in the network and removes them. In the final step the algorithm measures four scores to find out if it can finish the process. The first score *E* measures the effectiveness of information transfer between nodes of a network. The *C* score is the global clustering coefficient which is the probability of adjacent nodes having the potential to form a loop. It is calculated by using two measures, number of triangles (three fully connected nodes) and number of triples (three nodes which are accessible from each other). Matthews's correlation coefficient (MCC) is also calculated for each cycle. All measures are plotted and a cut-off value is decided based on properties of the plot (such as elbow point or if the network is falling apart). The remaining network contains the most important proteins, or “Central T cell Network (CTN)”, in T-cells. To verify, gene ontology term enrichment analysis was performed and the result showed enrichment in T-cells related terms. They also mapped the network onto TCR, JAK-STAT and MAPK signaling pathways. The result presented that most of the proteins comprising the important pathway for signaling of T-cells are present in the network. Therefore, 1323 proteins and 2095 links were reduced to 254 proteins 196 links. This approach has potential problems. First, it needs several microarrays for the target tissue and each data set has to include at least three samples. Second, this approach needs at least one protein as the input and one output component in the essential signal transduction pathways. But regardless of these limitations this great reduction makes analysis of this network much easier such that it can be processed and validated by many algorithms such as Boolean networks.

2.2 Boolean Networks and dynamics of a system

Cellular Automata (CA) is an abstract computational system, a computational problem solver, and a tool which is a representation of dynamics in many fields such as physics and system biology. The concept behind CA is simple but it can be applied to extremely complex problems. A CA consists of an N-dimensional lattice of entities which are called cells (also called atoms). Each cell can only be in one of the finite states at discrete time *t.* At each time step *t+1*, the states of all the cells are updated according to an update rule which is based on the neighboring cells. So in each time step a new grid is produced. The pattern produced by this process can be further analyzed to find a possible solution for the problem. This is a very limited definition of CA. Cellular automata can be much more complex than has been described [37], but further explanation is out of the scope of this thesis. Instead, an extension CA which is called Random Boolean Networks (RBN) is described in the next section. RBNs have been adapted to biological system and they are generalization of CAs.

2.2.1 Random Boolean Networks

A Random Boolean Networks (RBN) was originally proposed by Kauffman [38] as a model of genetic regulatory network. Since the idea behind the model is based on Boolean attributes (such as function and value), this model is well-adapted to many real life systems because their states can be determined by threshold. Kauffman proposed that living systems are created based on randomly combined elements rather than with previously defined components. The classic RBN consists of *d* number of nodes (shown by *N*) with links between them where N= $\{n_1, n_2, n_3 \ldots n_{d\}.}$ Associated with each node n_i, is a value v_i which can be 1 (on) or 0 (off). A state of the system is determined by a vector of the all values called *V* where V= $\{v_1, v_2, v_3 \ldots v_n\}$. Considering these variables, the notion of dynamics can be applied to the system. Dynamics can be described simply as time. This means that if a variable (say *t*) is added to *V* then the state of a system at time *t* can be represented by V_t = $\{v_{1t}, v_{2\ t}, v_{3t} \ldots v_{nt}\}$ which denotes a vector of the state of all nodes at time *t*. The state of the system at time *t+1* is determined based on the state at time *t*. This can be formally described by $V_{t+1}=f(V_{t)}$ where *f* is a function which maps the previous state of the system to the current state. Intuitively, when a RBN is created, the connections between nodes are randomly generated with a pre-defined input degree (number of incoming connection for each node). Afterwards, a set of Boolean function is randomly generated and one function is assigned to each node (in random).Mathematically, a set of functions *F* are generated in which $F= \{f_1, f_{2,} f_{3\ ,\ldots,} f_n\}$. This Boolean function dictates the state of a node at time *t+1* based on the state of the neighbors of this node at time *t*. The two sets of connections and functions are only generated one time (before the simulation) and remain constant during the calculation. After the initialization, a state V_1 (at time 1) is generated randomly and the system keeps updating the state of the network until it finds a state which was previously encountered. In this step, the algorithm generates another state at random and does the same updating process. The algorithm finishes when all the possible states of the nodes are generated. The flowchart of this process is depicted in figure 2.2.

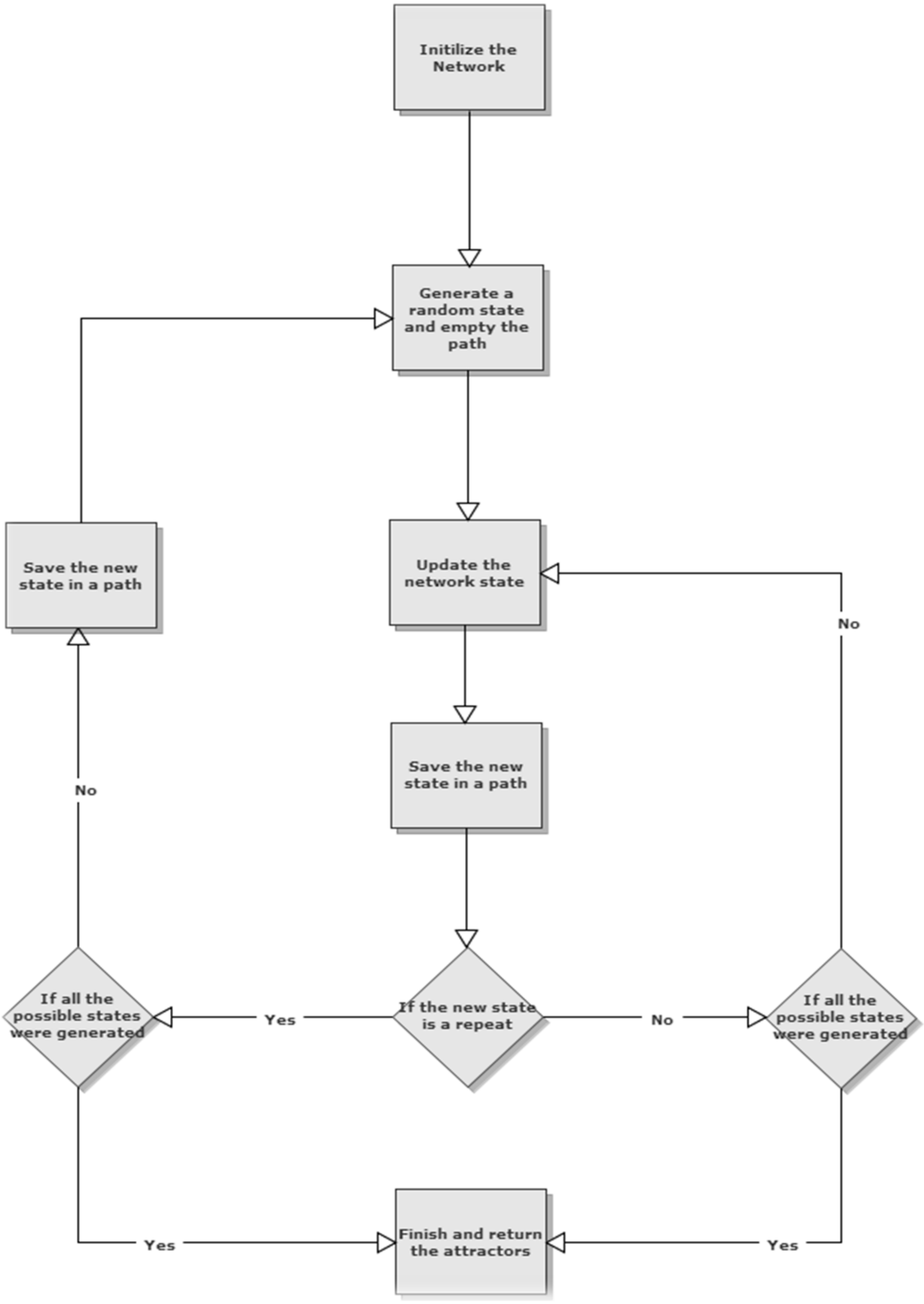

Initilize the Network
Generate a random state and empty the path
Update the network state
Save the new state in a path
Save the new state in a path
If the new state is a repeat
If all the possible states were generated
If all the possible states were generated
Finish and return the attractors
Yes
No
No
No
Yes
Yes

Figure 2.2. RBN simulation flowchart.

The network of states produced by the dynamics of system is called state transition network and as its name suggests, the system constantly updates its state until it is caught in loops. Since classical RBNs are deterministic and there are finite number of states 2^n (n number of nodes), the system eventually encounters repeated states. A state in RBN transition network can only have one successor but many predecessors and this causes system to get caught in loops which are repeated during the simulation. These loops are called attractors. When encountering a loop, there are two possibilities. First, the current state is repeated right after itself which means there are no states between the repeated states. In this case the period of the loop or attractor is one. The second possibility is when the period of attractor is more than one which means there are one or more states between two repeats. The first loop is called a singleton, or point attractor, and the second loop is called a cycle attractor. The sequence of states which end in an attractor is called the basin of attractor. These attractors are of the greatest importance and will be covered later in more detail.

2.2.2 Features of dynamical systems

Dynamical systems often can be classified using a measure called phase. Specifically, systems can be divided into ordered, chaotic, and critical phases. In the ordered regime, the system begins with an unstable pattern of nodes, but after some time it falls into the stable pattern in which simulation does not change the state of the system. It means, at the beginning, the nodes values are constantly changing and then the frequency of changes decreases until a stable pattern is found. After that, the system will not dramatically change its state. By contrast, in the chaotic phase the system tends to change constantly and will not stabilize its state. Critical or “edge of chaotic” happens when the system switches its state from ordered to chaotic. These regimes can be used to measure the stability of the system. For instance, in determining how sensitive the system is to the initial condition or how damage spreads across the whole state space. This can be done in several ways. For example, one can flip nodes values (one or more nodes) or changes the links between nodes or even change the Boolean function and observe how this change affects the system comparing with the normal conditions. It has been shown that, any changes to the normal conditions of the system will spread more rapidly and strongly when system is in the chaotic phase, and the reason for this relationship is that in the ordered phase, the stable states do not spread the change but in the chaotic regime spreading will occur much more easily. Therefore, chaotic systems are more sensitive to initial conditions, damage, and changes [39]. Living systems need both features to survive. They have to be stable in order to retain information and they need to be flexible to adapt to the changing environment. So the preference of the living organisms is to be in a the critical phase (or in the ordered state near the critical regime). Since RBNs are used to model dynamics of systems in living organisms, many studies have been done to find parameters involved in balancing the system between chaotic and ordered regimes (for a comprehensive review see [40]). It has been shown that phase transition between ordered and chaotic regimes depends on two parameters K and P (where K is input degree of nodes and P is used when the functions which are associated with nodes have probability of being one or zero). In general, it has been shown that when $K \leq 2$ the system is in ordered phase and when $K \geq 3$ the system tends to be in chaotic regime. Therefore one can assume the system will be in critical point when $K= 2$ [40].

2.2.3 Attractors in Boolean network

As mentioned earlier, attractors will be encountered in any network simulated using RBNs. Kauffman proposed that the state of the network in attractors is potentially related to cell types and the period of attractors (and basin) may be related to developmental process of cells. Many studies have approved his findings [41]. For example, [42] showed point attractors related to differentiation and apoptosis states of cells, and [43] wrote that singleton attractors are related to the steady state of cells. Several studies have also been done not only to find attractors but also on their properties, such as number, length, and their basins [44]. For example, there has been a great debate on the relationship between number of attractors and different parameters in network, such as number of nodes. For instance, Kauffman showed the total number of attractors can be √N where N is the number of nodes. Since the number of states in a network grows exponentially (2^N), there is no exact way for a complete statistical analysis of such networks. Basically, the problem of finding all of the attractors is NP-Hard [45]. So in many cases, researchers only explore small subset of the network (by sampling the state space) or only use small networks (less than 20 nodes). Although some of the attractors will be missed it can still be a reasonable approach given that living systems never try all possible configurations. The second case would be useful only for limited cases because many real biological networks are extremely complex.

There exist several algorithms which have been proposed to have lower computational complexity compared with the original problem. For example, [46] proposed an algorithm for finding singleton attractors. Using this algorithm, a partial state is extended toward a complete state. If the partial state cannot be part of an attractor, it would be left out. So this approach reduces the state space by removing the paths which are not potentially part of an attractor. They also extended this notion for finding small cycle attractors. In both of these cases the time complexity of the algorithm is not less than 1.13^N for singleton attractors (in real cases) and therefore 1.23^N for finding cycle attractors. So this method is only feasible when working on average size network.

The second algorithm was proposed by [47]. First, a propositional formula, for an unfolded transition relation of the network for *k* steps is generated and solved by using SAT-solver [48]. The satisfying assignment for this formula is an open path which is further expanded to an attractor. Next, *k* is increased for a different path with different length. Since SAT-solver has low time complexity for solving formulas with many variables, this approach works well for large networks but also has two limitations. First, its performance highly depends on length of the attractors and second, it cannot to be utilized in customized Boolean networks which their propositional formulas are not solvable by SAT-solver.

There are other approaches which have nearly the same performance, compared with the previously described approaches (for a complete review see [49]). For instance, [50] also uses partial state to find the attractors, but it is more efficient than [46]. But altogether, none of these algorithms are optimal and they cannot find all attractors in free-scale networks. Specially, when RBN is not deterministic, network analysis will be more difficult. Different types of RBN are covered in the next section.

2.2.4 Different types of RBNs

Random Boolean networks have several sub-types with regard to the updating rules. The previously introduced RBN is called classical RBN (CRBN) because it was the original model. In CRBN the state of the network at *t+1* is assigned by synchronously updating values of all the nodes at time *t*. The major criticism to CRBN is that they are not realistic because the states of all genes in living organisms do not change synchronously. This criticism caused invention of Asynchronous Random Boolean Networks (ARBNs). In ARBNs, the state of the network at time *t+1* is dictated by updating functions which update the value of only one node in at time *t*. Since this target node is selected randomly, ARBNs often behave stochastically and so only singleton attractors can be found in ARBNs and therefore there are no cyclic attractors [51]. Three other classes of CRBNS were proposed by [53] which are not regarded as new methods as only the parameters are changed in order to restrict or loosen the ARBNs domain. Deterministic Asynchronous Random Boolean Networks (DARBNs) are a type of ARBNs which do not update each node at random. Instead, selecting a node is restricted to two parameters *p* and *q* that indicate at what time a particular node can be selected and updated. A node is updated when “modulus of time *t* over *p* is equal to *q*”. If two nodes are selected at the same time, the first node will be updated and then the second node will be selected by taking the new state of the network into account. This modification enables ARBNs to work in a deterministic way. Generalized Asynchronous Random Boolean Networks (GARBNs) is a generalization of DARBNs in which any number of nodes for may be updated. Deterministic Generalized Asynchronous Random Boolean Networks (DGARBNs) are nearly the same as DARBNs, but when more than one node is selected at the same time, the nodes will be updated synchronously. Finally, there are Discrete Dynamical Networks (DDN) [54] which are assumed to be the most general of all types of Boolean networks. Figure 2.2.2 shows how Boolean networks are derived according to [53].

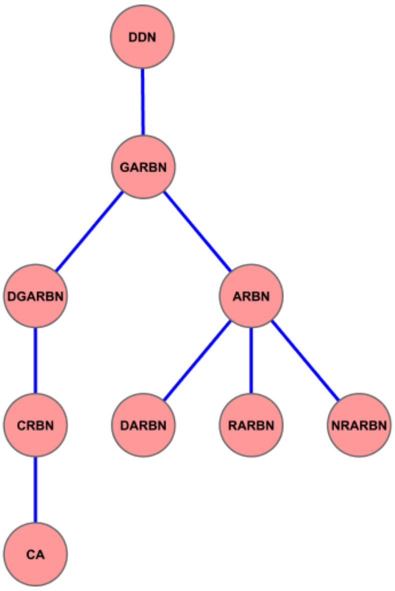


Figure 2.2.2. Boolean networks hierarchy.

2.2.5 Applications of RBNs.

Random Boolean networks have several applications not only in biology but also in many other computational sciences. They are used to investigate evolution through interpretation of how the networks themselves iteratively evolve [54]. Robotics and neural networks are two non-biological examples of applications of RNBs. As discussed earlier, the most important application of Boolean networks is to study genetic regulatory mechanisms in living organisms. A derivative of RBNs is called Probabilistic Boolean Networks (PBN) which has been successfully applied for finding gene functions [55]. Biological aspects of random Boolean networks can be assessed using several bioinformatics tools. In the next two sections two tools for extracting information from RBNs are described.

2.3 Microarray Data analysis

2.3.1 Introduction

Microarrays are one the most important tools for finding gene expression level in an organism. There are several types of microarray which are used for different purposes. For instance, gene expression profiling microarrays are used for finding expression level of several genes [56], comparative genomic hybridization is used when assessment of genome in different cells is needed [57], and SNP detection is used for identifying single nucleotide polymorphisms within or between populations. Microarray technology can be divided into two different devices, single and dual channel. In the single channel, only one cDNA is loaded and the array is used to find absolute levels of RNA expression. In dual-channel microarrays, two cDNAs from different conditions are loaded at the same time. The devices are utilized to determine relative levels of cDNA expression. Each of these technologies has advantages and disadvantages. The major advantages of single channel chips are that it does not suffer from contamination by a second sample. However, in dual-channel, comparison of the two sources (conditions) is easier but it is more error prone. The rest of this section is devoted to dual channel microarrays for gene expression profiling which is more related to the thesis but most of the ideas are applicable to single channel microarrays as well.

2.3.2 Microarray chip preparation

For preparing a chip, RNA molecules need to be extracted from two conditions. The first condition can be an infected a cell or tissue which has not been under normal conditions and the second condition a reference and it can be cells with normal condition (no diseases and no treatments). After that, an enzyme called reverse transcriptase is used to convert RNA to cDNA. Both cDNA molecules from two conditions are labeled by different colors (green for normal and red for the other condition). Next, samples are allowed to hybridize onto the same slide. This slide contains genomic DNA or a short stretch of oligonucleotide strands (corresponding to a gene) which are fixed to thousands of locations that are called spots. Each spot may contain millions of the same DNA fragment. cDNAs from samples will hybridize to their complimentary strands on each spot and the slide is then scanned by using a laser to detect the red and green dyes. The amount of emitted fluorescence color from the slide will show the relative level of gene expression in different samples. If a gene in normal condition is more expressed the spot will be green, if it is more abundant in the test condition the spot will be red, if the expression level is the same in both samples, the spot will be yellow, and if the gene is not expressed in either of samples the color will be black. After scanning, the system will generate a TIFF image which needs to be analyzed to determine gene expression levels for both samples.

2.3.3 Image analysis and data pre-processing

When the scanning process is done and an image is produced, the image needs be analyzed. The image processing can be divided into three different procedures. First, sub-arrays which contain a set of spots are identified. Next, spots are identified either by estimating locating of spot centers. Finally, the median background color intensity values are subtracted from the median value of pixels within a spot. Using this method, the values are less sensitive to anomalous fluorescence values.

When background subtracted value for all spots (for two channels) are detected, the relative expression ratio can be calculated using formula $E_i = \frac{R_i}{G_i}$ where R_i is median expression level for gene i in red channel after subtracting background intensity of the red channel and G_i is median expression level for gene i in green channel subtracted by background intensity of the green channel.

This expression ratio is highly sensitive when one has to compare up-regulate vs. down-regulated genes. This sensitivity is rooted in that facts that up-regulated genes are mapped between 1 and infinity but down-regulated genes lie between 0 and 1. Therefore, transformation is required to map both measures between comparable, and the same, scales. There are many transformation methods such as inverse transformation and log transformation. Inverse transformation converts an expression ratio into a fold change in which if expression ratio is less than 1, the folds change is multiplied by -1, otherwise the fold change remains the same. The pseudo-code for conversion is given below:

```
IF $E_i>=1$ THEN
        $E_i=E_i$
IF $E_i<1$
        $E_i=-1/E_i$
```

Using this approach, one can map fold change to the same interval. Log transformation has an advantage comparing inverse transformations, the capability of handling continuous space. Log transformation can simply be done by taking logarithm base 2 of the expression ratio. Using log transformation, one can treat equally differential up-regulation and down-regulation. The most important disadvantage of transformation method is that it completely removes information about expression level of genes.

2.3.4 Data normalization

The next step after transformation is normalization. Normalization involves removing systematic variation from the data [58] [59]. These errors can be detected when measuring genes which are supposed to have similar expressions (such as housekeeping genes which are considered to have expression ratio 1). Systematic variation can be caused by factors such as labeling inefficiency and mRNA materials.

Housekeeping gene expression is the first filter used to remove variation. This is done by isolating these genes and calculating normalization factor for the set of genes (which all have similar expression). Next, the calculated factor will be used for normalizing the other genes. There are several methods for normalizing the data. Total intensity normalization is used with the assumption that the number of RNAs in both samples is the same and also that the same number of molecules from both samples hybridizes on the chip. Considering this assumption, the normalization factor is calculated using the isolated gene set and intensities are rescaled. Mean centering normalization assumes that log ratio means of the gene set are equal to zero and a normalization factor is computed and intensities are rescaled using this factor. One of the most widely utilized methods is called locally weighted linear regression (lowess) normalization [60]. Using this method, one can assume that the dye bias is dependent on spot intensity. It fits a curve through all of the data points and adjusts the value of each point with regard to the curve. There are also other methods such as Quantile normalization or linear regression which are beyond the scope of the thesis.

2.3.5 Data analysis

Having transformed and normalized data, biological knowledge needs to be extracted from the data. This involves performing many experiments such as finding genes with different expression patterns, gene expression profile between two samples, or functional annotation etc. There are several methods for extracting information from microarray data and the selection of a method is highly dependent on the biological question which needs be answered.

Expression ratios can be represented as a matrix where each row represents expression ratios of a single gene in all samples and a column shows expression ratios for all of the genes in a single sample. Using the matrix, the problem of finding differences in expression profiles within the genes and between samples is converted to problem of finding distance or correlation between two vectors (either a genes vector of two samples or a samples vector of two genes). One of the most commonly available methods is called Euclidean distance. This method measures the square root of the sum of the squares of the distances between the values. There are other distance measurement methods such as Hamming distance and Minkowski distance which are less used compared to Euclidean distance (for formulas, see statistics section). A Pearson correlation coefficient (PCC) computes difference in shape of expression profile (not its magnitude). PCC produces a value in the range of -1 to 1. The value 1 means the genes (samples) have quite similar expression profiles and -1 means they have exactly opposite profiles. If there are no inferable relationships between the profiles PCC would produce zero. Often a cut-off value is considered for decrementing between similar and different profiles but choosing this value is highly dependent on the data-set. PCC is assumed to be a highly reliable measurement and is broadly used in microarray data analysis (For description and formula see statistics section). There are other measures such as Rank correlation coefficient (RCC), Shannon's entropy which are not covered in this text but readers are referred to related articles [61] [62]

Finding differences between two samples is not informative in many cases. That is because microarrays contain thousands of genes and many samples so finding distance between two samples, or two set of genes, is not an effective way to find patterns in data. Machine learning methods are often utilized to find patterns in data. Machine learning methods are divided to two categories, supervised and unsupervised methods. The first class is called supervised because external sources of information are used to classify genes to different classes. For example, one can gather information for expression of genes in different disease conditions and normal conditions and then assign the genes or sample to different groups. The unsupervised approach is used when there does not exist external information and data needs to be clustered only by considering relationships between data points. Compared with supervised approaches, unsupervised methods need more sophisticated techniques because in supervised approaches the system can be trained using available information but in unsupervised approaches there is no information to learn from. There are several algorithms for performing supervised analysis of microarray. Several machine learning methods such as support vectors machine (SVM), k-Nearest Neighbors, and Fisher Discriminant Analysis work perfectly on microarrays when external information is available. The aim of all of these methods is to learn from training data and decide on classification of unseen data (for complete descriptions see [63]). T-test and significance analysis of microarrays [64] (SAM) are also used to measure differentially expressed genes in several samples. Since thousands of genes may be assessed at the same time, multiple T-tests or hypothesis testing may produce error. Therefore, errors need to be handled, for example by false discovery rate (FDR) controlling [65]. Unsupervised clustering methods are classified to two general categories called hierarchical and non-hierarchical. Clustering methods are described in a separated section.

Analysis of microarrays often results in a huge number of genes which have to be biologically interpreted. There are several tools for annotating microarray analysis results. For example, some databases provide information regarding genes, and gene products, which can be used for gene enrichment analysis and gene set enrichment analysis. Both of these aspects are covered later in a separate section.

2.3.6 Software packages

Since microarrays are widely used, there are many software packages and websites available which are dedicated to performing specific analyses on chips. Some beneficial tools are described below:

Bioconductor [66] probably is the most well-known programming packages for performing different analyses on microarrays. Bioconductor is open source software which contains hundreds of packages for analysis and comprehension of genomic data and many microarrays methods have their own package. Because Bioconductor can be used in R programming environment, at least an intermediate knowledge of R is required to utilize this package. Gene Expression Pattern Analysis

Suite (GEPAS) [67] provides a web interface for doing many microarrays analysis from pre-processing to functional profiling. Tm4 [68] also provides a java application for performing different analysis. It offers tools for microarray data analysis, ranging from image processing to functional profiling of genes. Microarray data can be obtained from several sources. Gene Expression Omnibus (GEO) [69] and Arrayexpress [70] are among the biggest databases but there are also several other publicly available resources such as ArrayTrack, Stanford Microarray and MUSC databases. Most of the data are provided in two forms. One form is raw format which has been not processed and the second form is processed format (which is usually in matrix format) where pre-processing and normalization have been done on the data.

Microarrays data analysis is a very broad subject and several aspects of them, such as deferential analysis and comparing microarrays, are beyond the scope of this thesis. For more comprehensive introductions readers are referred to several available textbooks (see for example [64]).

2.4 Gene Ontology

2.4.1 Gene Ontology enrichment analysis

Many biological analysis tools such as microarrays produce large sets of gene (in different conditions or with different expressions). The next step is to determine if any subset of the genes is significant in any specific terms (e.g. biological functions). Since studying each gene individually is not feasible because of the huge number of genes or subjects under study, an automatic approach is needed to handle this problem. Gene Ontology [71] database provides centralized knowledge about known genes and allows researchers to annotate their genes list. Gene ontology (GO) provides relationships of biological terms to the genes which are known to be part of the terms. The database is regularly updated and new associations are created either manually by curators or automatically.

GO is organized in a hierarchical way by directed acyclic graph (DAG). Each term is related to its successors by either “part of” or “is a” relationships. Using DAG structure, each term can have zero or more children and one or more parents. GO terms are categorized into three groups called cellular component, molecular function and biological process. Cellular component shows the location where each gene (product) is acting. Molecular function represents the functions of gene products and biological process can be described as biological phenomena or events which are carried out by a set of molecular functions.

GO can be used in many approaches. The most common problem is that given a set of reference genes, a subset of genes which are under study has to be found using different methods such as clustering. This subset, and the reference set, has to be analyzed in order to find out if a set of terms which is most common is significantly different from others or it occurred only by chance. This process is a typical enrichment testing which asks if assigned to a particular GO term are over represented (not by chance), compared to genes assigned to that term in the reference set. Using hypothesis testing, the null hypothesis that there is no difference between genes associated to a particular term in test and reference sets. Therefore, alternative hypothesis can be “genes are either over or under represented in the target set”. A p-value is then calculated which shows how likely different terms are associated to the genes by chance.

Gene Set Enrichment Analysis [72] (GSEA) is another approach used to identify term significance in a subset of genes by taking into account the expression ratio of each gene in different conditions (samples). GSEA uses about 1300 gene sets from different databases such as pathways or GO. This is particularly useful when one wants to find out if a subset of genes is significantly different in distinct conditions. The mechanism of GSEA is more complicated than simple gene enrichment analysis. Four different inputs are given to the algorithm: a set of expression data for genes and samples, ranking procedure and profile of interest (such as sick and healthy), a set of independently derived genes (e.g. genes sharing the same GO category), and a control variable for the weight of the step. Next, an enrichment score (ES) is calculated by first rank ordering genes according to their correlation of expression in the selected profile (phenotypes). A cumulative sum over ranked genes is calculated in which if a gene is in the set the number is increased and otherwise it is decreased. The changes are weighted by gene correlation with selected profile. Finally, ES is calculated as the maximum deviation from zero. The significance of gene expressions is estimated by permutation of phenotypes and recalculation of ES score, resulting in a P-value. The next step compares the original ES score to the created distribution by the permutation process and adjusts the significance level to count for multiple hypothesis testing. This is done by normalizing ES scores for each gene set and will result in a normalized enrichment score (*NES*). The proportion of false positives is then controlled by calculating FDR of each NES.

2.4.2 GO Tools

Since GO itself is a database of genes and their associated terms, many online and standalone tools have been provided for doing different research on gene sets. As mentioned earlier, Bioconductor provides several tools for high-throughput genomic data such as microarrays and sequence. Bioconductor uses the R programming environment and contains more than 400 different packages. It provides annotation browser, term enrichment, text mining etc. Specifically, topGO [73] package provides semi-automated enrichment analysis for Gene Ontology by using several algorithms such as a combination of elim and weight algorithms which have been provided by [74]. Since Bioconductor is associated with R, basic knowledge of this programming language is required for using its packages.

The Database for Annotation, Visualization and Integrated Discovery (DAVID) [75] [76] is an online tool and web service which is widely used by many researchers. The major functions of DAVID are identifying enriched GO terms and enriched functional related gene groups, Clustering annotation terms, visualizing KEGG pathway. This tool has a friendly interface and also provides web service for programmatic access to the available tools.

G: Profiler [77] [78] provides an easy to use interface through 5 applications for analyzing Gene Ontology (GO) terms, KEGG and REACTOME pathways etc. G: Profiler supports 85 species such as mammals, fungi and plants. There are many other tools available for GO analysis but their functions are nearly the same as the tools mentioned. Regardless of their mechanism, these tools usually provide enrichment score and P-value for different subset of GO terms and associations. So researchers can select the most desirable result among enriched terms. Using this information, researchers determine if an interesting, or target subset, of genes grown in different conditions is associated with a particular process and how they affect state of the organism.

2.5 Clustering Methods

Machine learning methods are classified into two main categories called supervised and unsupervised methods. Given a problem without data for training, unsupervised approaches are used to find possible patterns in data. Specifically, clustering which are known as unsupervised methods are used to classify data into different groups (called clusters) based on the similarity between data points. Clustering methods also are divided into two categories called hierarchical and non-hierarchical methods. Hierarchical clustering approach behaves in two different ways with nearly the same mechanisms. Given a set of vectors, similarity between pairs of vectors is calculated by one of the distance measurement methods, resulting in a matrix which shows distance between pairs of vectors. In the bottom-up approach (which is also called agglomerative), the algorithm starts with an assumption that each data point (value) is one cluster itself. In each iteration, the two closest clusters are combined and replace the two participating clusters. This process continues until all of the data points belong to one common cluster. In contrast, using top-down approaches (which are also called divisive), all of the data points are assumed to be in a large cluster and then in each iteration this cluster is split to different clusters (based on similarity) until each point is in its own cluster. There are several methods for performing hierarchical clustering such as neighbor joining and unweighted pair group method with arithmetic mean (UPGMA). One the most important properties of these approaches are called linkage method. A linkage method is a way that the algorithm calculates distance between two clusters. There are several linkage methods, such as complete linkage, average linkage and single linkage. Complete linkage is measured by the maximum distance which can be found out between one point in one cluster and another point in other cluster. Average linkage is the measure of average distance between all the pairs in two clusters. Single linkage is defined by distance of the most similar (opposite to complete linkage) member between both clusters. Using different criteria, one may end up in different clusters and distances. Non-hierarchical approaches start with a predefined number of clusters in which each data point is (in general) randomly assigned to only one cluster. Given a criterion, the algorithm keeps moving points to a different cluster and optimizing this criterion. This process continues until no improvement can be made to the criterion. Since trying all possible assignments to the cluster is computationally infeasible (an NP-Hard problem), heuristic methods are employed to decrease the state space of the problem. Many devised algorithms, K-means, K-nearest neighbors, and many derivatives of k-means such as fuzzy k-means, are mostly used for computational biology analysis. Each algorithm can use many criteria such as Mahalanobis squared distance and Within-class dispersion. The first criterion maximizes Mahalanobis squared distance between groups and the second criterion minimizes determinant of within class dispersion matrix. UPGMA and K-means are described in the two next sections.

2.5.1 K-Means

K-means [79] is one of the non-hierarchical clustering methods with a simple mechanism. For using k-means, the number of clusters *K* (also called centroid) needs be defined before starting the algorithm. Having the number of clusters and a set of data points (vectors), initial place of centroids are defined. This is a very crucial step in k-means algorithm because the initial places of centroids have a dramatic effect on the output of the algorithm. Centroid values are often chosen randomly and far away from each other. For example, having two centroids, one can choose two data points which are far away from each other in random space and place the centroid exactly on these points. After initialization, each data point is assigned to only one centroid in which the distance between the point and corresponding centroid is minimized. When all of the data points are assigned, the new position of centroids is calculated as the mean of distances between the centroid and its associated data points. This process is repeated until no more change can be made to the location of the centroids. The result is one possible classification of data point to *k* clusters. Specifically, k-means aims to minimize the objective function below:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^j - c_j \|^2$$

This equation calculates the sum of distances between each data points and their corresponding clusters. K-means objects to minimization of this function, but this algorithm is not optimal. K-means has several limitations. The first problem is that the result of clustering is highly dependent on the initial placement of centroids and there is no exact way to find the optimal value for them. Another limitation is selecting the number of clusters. Because in many applications such as phylogenetic trees the number of clusters cannot be defined, applying k-means is not feasible or will not give a satisfactory result. There are a few methods such as elbow method which can be used to find one possible number of clusters, but manual inspection and a smooth curve after plotting *J* are required. Other limitations such as choosing reasonable distance measure and empty clusters need be considered when k-means is used. Considering the limitations, K-means always terminates by producing some clusters, but usability and correctness of the clusters must be evaluated manually.

2.5.2 UPGMA clustering method

UPGMA is the simplest method which performs hierarchical clustering. It starts with a distance matrix which contains distances between the objects under study. It finds the closest pair of objects in the table (say *m* and *n*), and combines them to form a new cluster (say *u*). Next, it eliminates the two combined objects and *u* is added to the table, objects making this cluster are connected to a point called a common ancestor. Distance between the new cluster and other object is computed by using one of mentioned linkage methods (usually average linkage). This process continues until the last two clusters are grouped. For instance, if *u= {m, n}* and the table contains another object say *a*, distance between *u* and *a* is calculated as follows (Using average linkage):

$$D(u,a)=\frac{D(m,a)+D(n,a)}{2}$$

D is a function which returns the distance between two objects. UPGMA produces a rooted tree in which each edge is associated with the average distance of two pairs making the corresponding branch. UPGMA is widely used in phylogenetics, but it has a drawback. It cannot handle molecular clock hypothesis. Therefore, the algorithm assumes the same evolutionary speed on all branches [80]. This problem is efficiently handled by other clustering algorithms such as neighbor joining method [81].

2.6 Graph Theory

2.6.1 Introduction

Graph theory is an efficient way of representing and manipulating of networks in computer science. Since many systems are characterized by their interacting components, graphs can be used in many applications such as decision making in machine learning, modeling biological network, and traffic controls. A graph consists of a set of nodes (vertexes) and a set of edges (links) which connect the nodes to each other. A graph can be directed or undirected which means if links between nodes have direction toward one of the nodes (or both of them). Formally, a graph *G* can be shown by a pair $G=(V,E)$ that consists of a set of nodes *V* in which $v_i \in V$ and a set of edge pairs in which $(v_i, v_j) \in E$.

An undirected graph is described by $(v_i, v_j) \wedge (v_j, v_i) \in E$ and $(v_i, v_j) = (v_j, v_i)$. Otherwise, it is a directed graph. In an undirected graph the degree of a node is the number of nodes which are connected to this node. In contrast, nodes in directed graphs have two different degrees which are called in-degree and out-degree. The out-degree of a node is the number of edges which points from the current node toward other nodes. The in-degree of a node is the number of edges which point to the current nodes. Formally, In-degree and out-degree of a node v_i are shown as follows:

$$\text{in-degree}: \textit{number of edges}(v_i, v_j) \in E$$
$$\text{out-degree}: \textit{number of edge}(v_k, v_i) \in E$$

There are several notations which need to be considered when dealing with graphs:

- Order of a graph is the number of nodes it has.
- Two nodes are said to be adjacent if there exists an edge in E which connects them.
- Adjacent nodes of v_i are its neighbors.
- A path in a graph is described by $P=(V_p,E_p)\subset G$ where $V_p=\{v_1,v_2,v_3,...,v_m\}$ (distinct nodes) and $E_p=\{(v_1,v_2),(v_2,v_3),...,(v_{m-1},v_m)\}$.
- A cycle is a path in which one node occurs twice. So a cycle has one additional edge (v_m,v_1) which connects the last node to the first node.
- A sub-graph of a directed graph is called a strongly connected component if there exists a directed path between any two nodes of the sub-graph.
- Graphs can be weighted such that with each edge is associated one value. This value is the weight of traversing between the nodes connected by the edge (other assumptions are possible).
- A directed graphs without cycles are called directed acyclic graph (DAG)

Finally, graphs can be represented using many approaches such as unordered edge sequence, adjacency arrays, adjacency lists, and adjacency matrices. Each approach has its own advantages for example, adjacency arrays are used for static graphs, adjacency lists are utilized for dynamic graphs and matrices are easy to read, and they are also used in many graph analysis applications. For example an adjacency matrix of an N ordered graph is given by $A=\lfloor a_{ij} \rfloor \in \mathbb{R}^{nxn}$ where:

$$a_{ij}=\begin{Bmatrix} 1\, if\,(v_i,v_j)\in E \\ 0 \quad \text{otherwise} \end{Bmatrix}$$

2.6.2 Graph analysis

Applicability of networks and trees in many practical applications is not a new concept. Graphs are broadly used in numerous applications, ranging from biological network (e.g. gene interactions) to making an intelligent decision by artificial neural networks. This diversity of applications causes several problems in this field. Some problems are only specific to a particular concept, and others are nearly common in many applications. Given a set of vertices, graphical enumeration is the problem of counting specific directed or undirected graphs [82]. The routing problem is probably the most common and the most studied problem in graph theory. Traveling salesman problem (TSP), Hamiltonian path, and shortest path problems are among the most well-known concepts in the routing problem. One of the most exciting areas in graph theory (which is related to path finding) is cycle detection in a given directed or undirected graph. Cycle detection can be handled by depth-first search (DFS) algorithm in which if DFS finds an edge to an observed vertex a cycle is found. Cycle detection is also efficiently handled by many topological sorting algorithms, but no algorithm exists to identify all the cycles in a network. Since cycles in a network have very interesting properties such as capturing the dynamics of that network, many algorithms have been proposed to find cycles in different types of graph [83] [84] [85] [86]. Tarjan proposed an efficient algorithm [83] for enumerating elementary circuits in a directed graph. An elementary circuit is a cycle in which no vertex appears twice but the first and last vertexes. The algorithm accepts an adjacency list *A* which is shown by *A(v)* which is a set of all the nodes which have edges to node *v*. The algorithm uses a point stack which denotes an elementary path. The elementary path starts with node s and since algorithm assumes nodes are numbered, every node on this path needs to satisfy $s \geq v$ condition. In additional to the stack, this algorithm utilizes a list which is called “marked list” which contains the nodes that are on the elementary path or the nodes that if all the paths passing from them to *s* intersects with *p* at any nodes other than *s*. Considering these concepts, the procedure starts by generating all the elementary paths which have *s* at the beginning and contain the nodes which are not numbered with an integer less than *s*. A cycle is found when the last node on the elementary path is adjacent to the first element of the path. The important point is that a node is only used in the path if it is not marked and also is not deleted from the stack. This algorithm has at least $O(n.\ e(c + 1))$ time complexity. Tarjan's algorithm is the successor of the algorithm presented by [86]. Another algorithm is presented by [86] and improved time complexity of Trajan’s and TIERNAN's algorithms to $O(n + e)$. Its mechanism is almost the same as [83] [85] but it adds two more features to these algorithms in order to improve accuracy and time complexity. First, to avoid duplicating cycles, vertex *v* is blocked when it is added to an elementary path starting with *s* and is held blocked until no path from *v* to *s* intersects the current path at no nodes (other than *s*). Second, a vertex will not become a root vertex on an elementary path unless it appears at least on one elementary circuit. This algorithm does not find cycles with period one (loops). These modifications make this algorithm faster than the previously mentioned approaches, but given a graph, there is no feasible solution for finding of all of the cycles, and this remains an open problem for further investigations.

2.7 Statistics

2.7.1 Hypothesis testing

Hypothesis testing is a popular method for finding out if there is enough evidence in favor of a hypothesis with regard to a parameter. A test consists of two hypotheses. The first hypothesis which is called the null hypothesis (*H0*) and means the favored assumption regarding the parameter and the second belief is called alternative hypotheses (*H1* or *HA*) and means an assumption that will be accepted if there exists enough evidence to the contrary of *H0*. There are four possibilities when one draws a conclusion regarding hypotheses. There are two types of error when applying hypothesis testing. The first and most important error occurs when the null hypothesis is correct but alternative hypothesis is favored. This error is called type I error and its probability is shown by Greek symbol α. The second error occurs when the alternative hypothesis is incorrectly rejected. This error is called type II error and its probability is shown by β. Other two possibilities occur when correct decision is made. The conditional equations for both of the errors are shown below:

$$\alpha = P\ (H1 \text{ is judged} \mid H0 \text{ is true})$$
$$\beta = P\ (H0 \text{ is judged} \mid H1 \text{ is true})$$

Clearly both of the probabilities are favored to be small, but it is a tradeoff between α and β. When α is decreased, β is naturally increased and vice versus. Hypothesis testing approach tries to keep α as small as possible (not too small such that β is increased). That is because the favored assumption is *H0* and decreasing α means more evidence is needed in order to reject *H0*. There are two commonly used values for α, 0.05 and 0.01. α is also called significance level which is used to find the rejection region for the null hypothesis. The mechanism of hypothesis testing begins with stating a hypothesis and conditions for the rejection. Normally a hypothesis is based on the difference between the sample mean and the population mean (other parameters can also be included). For instance, $H0 : \mu = \mu0$ can be a representation of equality between the sample mean (μ) and population mean (μ0) which means the null hypothesis is true when observed sample mean is equal to the population mean. Against *H0* is the alternative hypothesis which can be either one sided or two sided. In one sided experiments only one side of the mean rejects *H0* ($H1 : \mu < \mu0$ *or* $H2 : \mu > \mu0$). Two sided experiments only state that $H1 : \mu \neq \mu0$ which means if μ lies on either sides of μ0 the null hypothesis is rejected. The framework for comparing the means is provided by t-test:

$$t=\frac{\bar{x}-\mu 0}{\frac{s}{\sqrt{n}}}$$

Where $\bar{x}$, *s*, *n* are the sample mean, standard deviation and number of entities, respectively. If the original data is normally distributed, *t* can be either from t-distribution with *n-1* degree of freedom or not. If *t* comes from t- distribution, then *H1* would not be accepted. Otherwise, *H0* would be rejected, and this means that difference between population and sample means is large and cannot be described by t-distribution. This largeness can be indicated by α. Specifically, this cut off (c) is identified by critical t-values table with chosen α level and degree of freedom *n-1*. For instance, if *t* is greater or less than *c* (in two sided experiments), *H0* would be rejected. This means that *H0* is rejected with $a\times 100$ percent probability of making an error.

Results of the test are highly dependent on α value. It means that when decreasing α, it becomes more likely that the null hypothesis is accepted. P-value is the probability of wrongly rejecting *H0*. This value is the probability of observing T-test value more extreme than which was observed if the null hypothesis is true. So smaller the P-value, more likely *H0* is rejected. Usually 0.05 or less can be regarded as strong evidence against *H0*. Finally, there are several important aspects one has to consider for performing hypothesis testing:

- Data distribution has to be normal or at least it should not highly deviate from the normal distribution. If the data lightly deviates from normal, other testing formulas such as bootstrap testing can be considered. If the data is highly skewed then one might want to consider other parameters and not the mean.
- Power of the test is calculated by 1-β. This is the probability of rejecting *H0* when *H0* is actually false.
- Sample size (n) is a very important aspect of hypothesis testing. Usually larger the sample size, the more reliable the test would be. Retrieving more samples may be a difficult task in some problems. Therefore, many authors suggested a sample size of more than 30.
- Multiple hypothesis testing is a special case of normal test in which one wants to test different samples (different hypotheses). This means a test would be done several times. Since each experiment produces a small amount of error and because a test is taken several times, the probability of having one false in the tests can be much higher than one experiment. So one has to have control over different errors such as family wise error rate (FWER) or false discovery rate (FDR). This is especially used for microarray data analysis where thousands of genes are tested. So performing ordinary hypothesis tests will produce unreliable results.
- T-test is performed when population standard deviation is unknown, and z-test is performed when population standard deviation is known.

2.7.2 Pearson correlation coefficient

As mentioned earlier in microarray data analysis section, Pearson Correlation Coefficient (PCC) is used to measure the similarity between two vectors. In microarrays, these vectors can be gene expression values (normalized) between two different samples or expression ratio of two genes in different samples. Having two vectors which are denoted by $A=[a_{1,}a_{2,}a_{3.}..a_{N1}]$ and $B=[b_{1,}b_{2,}b_{3.}..b_{N2}]$, PCC is calculated as follows:

1. Means of two vectors are calculated. *M1* and *M2* are means of *A* and *B*, respectively.

$$M_1=\frac{\sum a_i}{N_1}$$

$$M_2=\frac{\sum b_i}{N_2}$$

2. The vectors are mean centered (minus sign is element-wise).

$$A_c=A-M_1$$
$$B_c=B-M_2$$

So

$$A_c=[a_{c_1},a_{c_2},a_{c_3},...a_{c_n}]$$
$$B_c=[b_{c_1},b_{c_2},b_{c_3},...b_{c_n}]$$

3. Element-wise multiplication of two vectors are computed as follows:

$$EM=\sum(a_{c_i}*a_{c_i})$$

4. Sum of each of the mean centered vector is calculated:

$$S_A=\sum a_{c_i}$$
$$S_B=\sum b_{c_i}$$

5. Finally, PCC is computed as follows:

$$PCC=\frac{EM}{S_A*S_B}$$

The last 3 steps actually calculate the cosine of the angle between two vectors. As previously mentioned, PCC produces a number which is an indication of the similarity between two vectors Some software packages also calculate a p-value which denotes how reliable the PCC is. There are other methods for finding out relationships between vectors such as Rank correlation coefficient (RCC) which are not covered in this thesis.

2.7.3 Distance between two vectors

Measuring distance between two sets of values is widely used in many applications such as phylogenetics. There are several methods for measuring the difference between two vectors such as rectilinear distance, Hamming distance and Euclidean distance.

Given two vectors $A=[a_1,a_2,a_3,...,a_n]$ and $B=[b_1,b_2,b_3,...,b_n]$, Euclidean distance is calculated as the square root of the sum of the square of difference between entities in each vector:

$$D(A,B)=\sqrt{\sum(a_i-b_i)^2}$$

Hamming distance is a special case of Euclidean distance where A and B only contain binary values. Both of these methods are derivatives of Minkowski distance which is calculated as follows:

$$D(A,B)=\sqrt[pow]{\sum(a_i-b_i)^{pow}}$$

As mentioned, applying distance methods [87] is very common in many areas such as clustering, retrieval problems, and phylogenetics and choice of the method often changes the result of the algorithms.

Objectives

The main purpose of this study is to find the genes which are involved in T-cell differentiation, but the ultimate goal of this project is to provide a general framework for simulation of gene co-regulatory networks. In order to solve the problems, following analysis need to be done:

- Defining an approach in order to simulate gene regulatory networks.
- Defining an updating function for determining state of a gene in different times.
- Finding initial values in order to reduce the space and time complexity of the simulation.
- Finding different parameters for random and effect part of the updating function in which the randomness and effect of neighboring genes are balanced.
- Performing simulation for a reasonable number of runs.
- Detecting attractors using graph theory algorithms.
- Analyzing most frequent and probable attractors in order to find T-cell differentiation related genes.
- Clustering attractors and analyzing different clusters.
- Finding gene expressions in different clusters, identifying differentially expressed genes, and analyzing them in order to find T-cell differentiation related genes.
- Verifying the system by comparing simulated gene expressions with actual microarrays experiments.

3. Materials & Methods

As previously described, "the Central T cell Network, CTN" which contains the most important genes and their interactions in T-cells was proposed by [29]. This network consists of 254 genes (nodes) with 196 links (edges) with average node degree 1.5, and the network is separated into 61 clusters where the largest one has 73 nodes and 74 links, and the smallest ones have two nodes and one link (table A1.1 and figure A1.1 in appendix 1). With each edge is associated a correlation coefficient value of expression between two genes which are linked by that edge (table A1.2 in appendix 1). The network assumes no directions on edges, so the result is a co-regulatory network which means if two genes are connected they may regulate each other depending on whether each gene is expressed or not. Expression of one gene is not only dependent on the expression of its neighbors, but also on the correlation between the neighbors of the gene. Simulation of the dynamics of such a network needs special treatment not only because one has to consider the correlation values, but also stochasticity of the cell environment needs to be taken into account. This is because a cell may behave in a way which is unknown or it may have different states in different conditions. For simulating the system behavior, a modified version of random Boolean network is used to model dynamics of the system. Because RBN and its derivatives and similar methods such as PBN, work only on systems with Boolean functions with predefined level of stochasticity, they cannot handle special networks with correlation values. The next section provides the modified version of RBN which is used to handle these problems.

3.1 Setting up the environment

For initializing an RBN, one needs to randomly assign connection between nodes. Because CTN network has its own connections, this step is skipped. RBNs also need randomly assigned Boolean function to each node. Since the network provided by [29] is characterized by correlation values between genes, non-Boolean functions need be considered for finding the state of each gene at different time.

Considering these problems, dynamics of CTN can be simulated by using the following system:

Similar to RBNs, a state of the system at time *t+1* is shown by a vector of 254 genes where each gene can take value 1 or 0 which means gene is expressed (on) or not expressed (off), respectively. So the state of the network *S* at time *t* is shown by S_t as follow:

v_1	v_2	v_3	...	v_{254}

Where $v_i = \{0,1\}$. Dynamics of the network are shown by the state transition network where each node is one possible state of the network at time *t* followed by its descendant which is the state of the network at time *t+1.*

$$S_t = \{v_1(t), v_2(t), v_3(t), ..., v_{254}(t)\}$$
$$\downarrow$$
$$S_{t+1} = \{v_1(t+1), v_2(t+1), v_3(t+1), ..., v_{254}(t+1)\}$$

Value of a node v_i at time t+1 is a function of its neighbors at time *t*:

$$S_{t+1} = \{f_t(v_1), f_t(v_2), f_t(v_3), ..., f_t(v_{254})\}$$

The functions which undertake mapping from S_t to S_{t+1} have principle differences to RBN's functions in which instead of having several functions which are randomly assigned to each gene, a general function is used to perform the mapping. This function consists of two parts which are named the effect and the random components (EC and RC, respectively).

$$f_t(v_i)=EC(v_i)+RC(v_i)$$

The effect component applies the effect of neighboring genes on the current gene in which if a neighbor is *on* at time *t*, it has a positive effect and otherwise its effect is negative. Neighbors apply their effect on gene *i* by using their expression correlation coefficient value with the gene *i*. Specifically, effect component (EC) is calculated as follows:

$$EC(v_i)=\sum (r_{ij}\times s_{ij}(t))$$

Where r_{ij} is the correlation value between neighboring gene j and gene *i*. $s_{ij}(t)$ is computed as follow:

$$s_{ij}(t)=\begin{Bmatrix}1 & if\ v_j(t)=1\\ -1 & if\ v_j(t)=0\end{Bmatrix}$$

This equation means if *jth* neighbor of node *i* has value 1 at time *t*, $s_{ij}(t)$ is 1, meaning that this neighbor has a positive effect on the current gene and promotes expression of the gene *i*. If the neighboring gene has value of zero, -1 is assigned to $s_{ij}(t)$ which means that neighbor has an inhibitory effect on gene *i* and suppresses expression of the gene. So the effect of the EC is the sum of positive or negative effect which neighboring genes apply on the current gene *i* and determine whether the current gene being expressed or inhibited. Effect component regulates current gene only based on the correlation values. Using this part alone, one completely ignores the possible error in correlation values. Also, the stochasticity of the cell environment is completely ignored. This causes the dynamics to flow deterministically, and contrasts with all of the proposed methods for simulation of system dynamics. To handle this problem, the random component (RC) is added to the equation. RC is meant to add a controlled random effect to EC, causing the system not behaving fully deterministic. RC is computed as follow:

$$RC=C_i\times P\times R$$

Where $C_i = \sum_{j=1}^{L(neighbors)} r_{ij}$ is the sum of all correlation values between the current gene *i* and all of its neighbors (indexed by *j*). *R* is a random number in the range of [-1, 1] which shows if the random component has negative or positive effect and also indicates the extent of this effect. *R* is reproduced for each gene when updating the state of the network. *P* is power of the random component which remains constant in the simulation process and indicates how much the random component can affect the whole equation. This number is in the range of [0, 1] where 1 means that the system may behave quite random (depending on R), and 0 means that there is no randomness in the system.

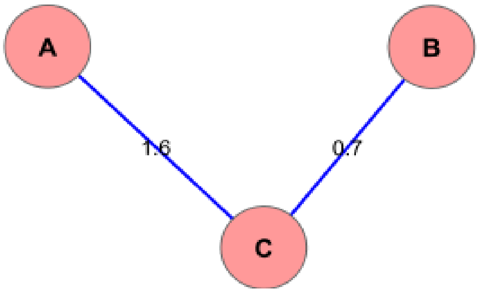
The following equation is used to find the state of a gene (node) at time *t+1*:

$$f_t(v_i)=\sum (r_{ij}\times s_{ij}(t))+(C_i\times P\times R)$$

Since this equation only shows the effect of neighbors on the current node, the binary state of a gene at time *t* is a computed as follows:

$$v_i(t)=\begin{Bmatrix}1 & if\ f_t(v_i)>0\\ 0 & if\ f_t(v_i)\leqslant 0\end{Bmatrix}$$

For example, considering following network where with each link is associated one correlation value.



State of the network at time t is shown by $S_t = \{A, B, C\}$. If a state at time 1 is assumed to be $S_1 = \{1, 0, 1\}$ and P is 0.1, state of the network at time 2 (t+1) is computed as follows (R is generated randomly for each equation):

$$v_a = (1\times 1.6) + (1.6\times 0.1\times 0.4324) = 1.669184$$
$$v_B = (1\times 0.7) + (0.7\times 0.1\times (-1)) = 0.63$$
$$v_c = ((1\times 1.6) + (-1\times 0.7)) + ((1.6+0.7)\times 0.1\times 0.32) = 0.9736$$

As it is shown, all of the equations resulted in numbers greater than 0, so the state of the network at time 2 (t+1) is $S_2 = \{1, 1, 1\}$.

Considering this function and the PPI, dynamics of the network can be simulated by RBN. Using RBN, the system has to produce all of the possible states of the network which in this case exists. Since this is a very large number and cannot be handled by even the fastest supercomputers, one has to figure out an approach to reduce the state space to a reasonable size. Another issue is selecting the *P* parameter which controls the power of the random component. This parameter is crucial for the system to work optimally. For example, if *P* is highly increased in the example (say to 1), the state of the node *b* will be switched to off (0). These two problems are addressed in the next two sections.

3.2 Simulation

In order to simulate dynamics of a network, random Boolean networks need to generate all the possible states of the network. Biological networks are often extremely large and impossible to be directly simulated using RBNs. This is because even considering only the binary state of the genes, 2^N possible states have to be generated, and this number is often very large and cannot be handled using current computing power. As mentioned, several algorithms have been proposed to handle this problem. Having a reasonable number of genes (nodes), they are able to find the attractors of large networks. The CTN network consists of 254 nodes and so for the purpose of simulation one needs to generate 2^{254} states. Since this number is large, state space has to be reduced. As described in the random Boolean section, the sampling approach is an option which picks a subset of states, but it cannot guarantee if one can select a good subset which is true in nature. Also because the number of the states is very large, a reasonable (computationally applicable) subset of states is still too small to be a representative of the whole network. Other proposed algorithms also do not work in case of CTN because of two main reasons. First, they need Boolean functions for each node. Second, The CTN functions are not stable. This means that because of the random component, it is possible that the network flow changes direction at any time. Specifically and in contrast with RBNs, each state of CTN transition network can have more than one parent and several children. This may remind the asynchronous RBNs, but they are two different subjects. In ARBNs, changing direction is predictable, because the state of the parents are regarded as stable (switching one gene always results in one specific state so in case of selecting one gene for updating, there are *n* possible next states which the current state can flow to) but in the case of CTN, randomness (e.g. error possibility) is added to the state of the parents and the CTN network can behave in completely unpredictable manner. So it is not possible to use ARBNs approaches to simulate the network.

As Kauffman proposed, biological systems often do not explore all of their possible states (e.g. 2^{254} possible state of genes in T-cells). Therefore, if one assumes that a state of a network is already known, the process can be started from the known state and then network dynamics flows as many states as possible to generate more states starting from the initial condition. The best way to find the initial state is to utilize microarray gene expression data and make a consensus state between different samples for all genes.

From 22 microarray time series, which contain 353 data sets and were retrieved by [29], nine time series are isolated (table 4.1). The isolation is based on whether they contain all the genes in CTN or not. The isolated experiments contain 47 samples (All of the experiments were previously normalized by Bioconductor and Robust Multi-Arrays algorithm). For each sample, a histogram diagram is made (figures 3.1 and 3.2 shows examples of both distributions). Considering the histograms, normally distributed samples and those without normal distribution are separated into two different sets of samples. Mean and median of each sample (based only on CTN gene expression) are calculated (difference between the mean and the median is shown in figures A1.3 and A1.4 in appendix 1 for normal and non-normal distribution, respectively) and it is assumed that if a gene has expression greater than the mean or median it is expressed or on (1) and otherwise it is off (0). The calculations result to 4 vectors of length 254. The consensus vector (between 4 vectors) is calculated as follow:

1) Align 4 vectors (A, B, C, D) which are corresponding to (Boolean transformed) mean of the data with normal distribution, mean of the data without normal distribution, median of the data with normal distribution and mean of the data without normal distribution, respectively.
2) An empty vector (say T) of 254 lengths is created in which each cell is corresponding to a gene.
3) $T_i = \left\{ \begin{matrix} 1 & if\,(A_i + B_i + C_i + D_i) > 2 \\ 0 & if\,(A_i + B_i + C_i + D_i) < 2 \end{matrix} \right\}$ Where *i* is the index of each cell in the vectors.

   If the result is equal to 2, the most frequent number (1 or 0) in all vectors is selected for that place.

The result of these steps is a vector of one and zeros. This vector is used as the initial value for starting the simulation.

Table 3.1 Isolated time series.

Database ID	Title
E-MEXP-549	Transcription profiling time series of gene expression following irradiation to identify P53 activity [89]
GSE2770	Transcriptional profiles of Th cells induced to polarize Th1 and Th2 direction in the presence or absence of TGF [90]
GSE7497	Influence of TGF on human resting CD4+ T cells [91]
GSE11755	Gene expression profiling in pediatric meningococcal sepsis reveals

Table 3.1 Isolated time series.

Database ID	Title
	dynamic changes in NK-cell and cytotoxic molecules [92]
GSE12079	Molecular profiling of CD3- CD+ T cells from patients with the lymphocytic variant of hypereosinophilic syndrome [93]
GSE14330	Comparison of stable human Treg and Th clones by transcriptional profiling- experiment I [94]
GSE15928	Influence of anti-CD25 mAb on the transcriptome of activated Peripheral blood mononuclear cell (PBMC) [95]
GSE24634	Expression data from developing regulatory T cells [96]
GSE27291	Expression data from human TCRV(+) T lymphocytes [97]

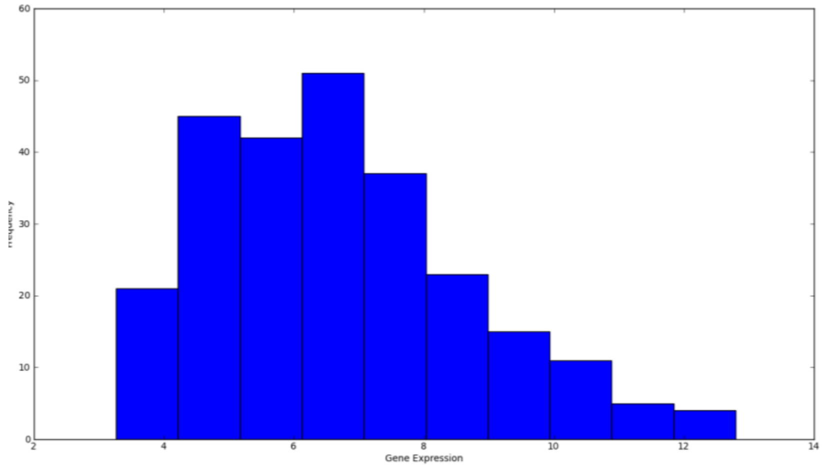


Figure 3.1. An example of normally distributed microarray data which is used to find initial values.

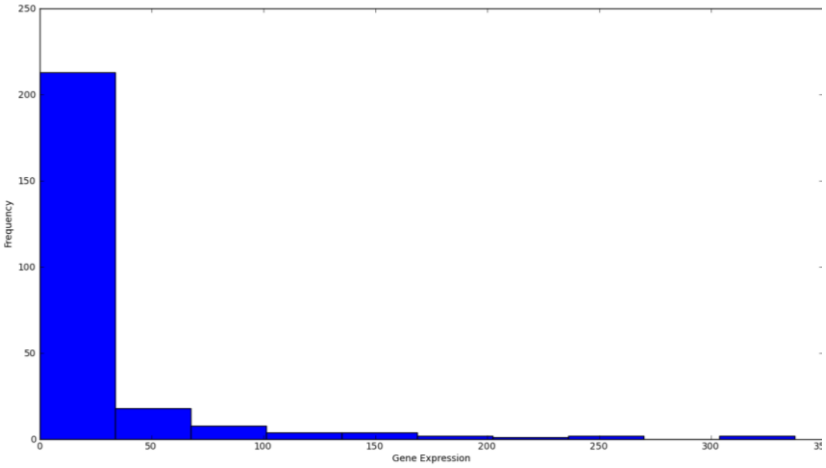


Figure 3.2. An example of non-normally distributed microarray data which is used to find initial values.

As briefly described, RBNs start with a randomly generated state and then keep updating this state until a state appears twice (it goes back and generates another random state) or all of the possible states are generated (which is the end simulation). Considering the proposed system for CTN without the RC in the formula, the system starts with the initial values and keeps updating the state (creating state transition network) until a state appears again. At this point, the algorithm stops and the system process will be terminated because as previously mentioned, the algorithm assumes that CTN has one initial state and so no states are randomly generated. Another reason is that in CRBNs (or CTN without RC component), the updating function is deterministic and this means that a state in the state transition network can have only one child but many parents. So, if the system visits a state twice, it means that if the latest visited state is updated again, the system will pass the same path between the previously encountered repeats. For example, suppose the initial state is called S_1 and the system keeps updating the sequence of states after S_1 regardless of the encountering repeats. If in this process, a state called S_r appears twice, a path between two states S_r will be explored repeatedly. This can be depicted as follow:

$$S_1 \to S_2 \to S_3 \to \dots \to \underbrace{\underset{\substack{\Uparrow \\ \text{Repeat}}}{S_r} \to S_{r+1} \to S_{r+2} \to \dots \to \underset{\substack{\Uparrow \\ \text{Repeat}}}{S_r}}_{\substack{\text{A path between a repeated state} \\ \text{(say path A)}}} \to S_{r+1} \underbrace{\to S_{r+2} \to \dots \to S_r \to S_{r+1} \to S_{r+2} \to \dots \to S_r \to \dots}_{\substack{\text{Path A is repeated until stopping} \\ \text{the algorithm}}}$$

This process is the same as RBNs. Once a repeat is encountered no more new states (previously not encountered) will be produced, this is what a single iteration of random Boolean network updating algorithm does. This means that no more variation is added to the state transition network and the produced network has only one attractor and its basin.

When the complete equation (EC+RC) is taken into account, the condition is completely changed. Since the equation has randomness, it cannot guarantee that a state has only one child. So a state can have more than one parent and also more than one child. This means that if a repeat is encountered in the updating process, the system will not keep updating the repeated state, it will always get caught on the same loop (path). This is because the randomness in the equation may cause the system to change its direction at any time with some probability. This process is shown below:

$$S_1 \to S_2 \to S_3 \to \dots \to \underbrace{\underset{\substack{\Uparrow \\ \text{Repeat}}}{S_r} \to S_{r+1} \to S_{r+2} \to \dots \to \underset{\substack{\Uparrow \\ \text{Repeat}}}{S_r}}_{\substack{\text{A path between a repeated state} \\ \text{(say path A)}}} \to \underbrace{S_{r+i} \to S_{r+i+1} \to \dots \to S_I}_{\substack{\text{Different path from A} \\ \text{Until threshold I}}}$$

As it is obvious from the figure above, the system is not caught in a loop, and it easily passes the repeat and goes until the defined number of states is generated (shown by *I*). In the process starting from state S_1 to S_I, there may be several repeated states and several unique states. The number of repeats and unique states directly depends on parameters *P* and *I* in which if both of parameters are increased, the number of unique and repeats have the potential to be increased. This is because when *I* is increased, the system produces more states and when *P* is increased, the system tends to update the states more randomly. Because of the huge number of possible state, encountering repeats is rare. Thus, a good balance between *I* and *P* is critical for the system to be able to produce reasonable variation for covering of the state space as well as steady states which are indications of some behavior of the system (see next section). When choosing *P* and *I*, the algorithm finds *I* possible states starting from one initial value. The pseudocode of the algorithm is as follow:

```
function simulation (I, P)
    begin
        array initialValues[254]=initializer()
        array tempValues= initialValues
        dynamic array states[]
        states.append(tempValues)
        For 1 to I do
            begin
                tempValues=updateState( tempValues, P)
                states.append( tempValues)
            end

    Return ( states)
    end
```

The pre-defined initial state is assigned to a variable, and this variable is added to the list of produced states. In the loop, the algorithm updates the current value with an updating function, assigns the current variable to the next state of the system, and finally saves it to the list of produced states. This process is repeated *I* times and after that the function terminates and returns a sequence of generated states (a transition list instead of transition network). This list will then be used to finding attractors and in making the state transition network.

3.3 Finding Parameters

The power of the RC parameter (P) and the number of iterations (I) are very important for the algorithm to output a better result. Both of the parameters have their own impact on the number of unique states. Obviously when the number of iterations is increased, the system tends to generate more states, but this cannot be done without having a good *P* value. The problem of finding the number of iterations is solved using 3 different approaches:

1. The simplest and the most straightforward solution is to let the system progress as much as the computational power allows. Using this approach, the system keeps updating the states until the computer system terminates the process. So the system produces as many states as possible.
2. The second approach is letting the system be completely random (P=1) and then performing several experiments while increasing *I* in each experiment. The objective will be to see if there is an elbow point in the plotted number of unique states in which the system behaves in a different way compared to other experiments. Using this mechanism, it is possible to find out what potential the system has for producing more states while increasing the number of iterations. Due to randomness in the system, this approach can be unfeasible. This is because the system may not end up

with a normal distribution shape. So detecting an elbow point is a difficult task. To address this problem, several experiments are done (with the same range of iterations), where in each iteration, the standard deviation between the experiments is plotted. This measure represents whether the pattern of produced states is stable in many different experiments. The lowest standard deviation (or an elbow point) is selected as the number of iterations.

3. The last approach is nearly the same as the second approach with the minor difference that P is increased in each experiment. Therefore, both *I* and *P* are changed at the same time and the resulting number unique states are plotted in each iteration. Since there are several possible ways to combine different *Ps* and *Is*, this way may not result in a reasonable conclusion. The next chapter shows that using this approach, the curves are not plotted in a sensible way, and the behaviors of the system may be quite unpredictable. So this approach is not recommended for finding parameters.

Having the number of iterations, the power of the random component is computed using an arbitrary initial value (a very small value is preferred). The number needs be small enough (while not zero) such that the number of produced unique states is constant in several experiments. After that, *P* is increased, and the experiments are performed again. This process is repeated until the system starts to produce variation (not constant number unique states in all the experiments). When the system produces variations, the number of unique states is plotted and the same approach which is used to find iteration number (the second method for parameter *I*) is used to find a suitable power of the random component. It is very important to have a *P* value which produces a good variation and also prevents the system from being completely random.

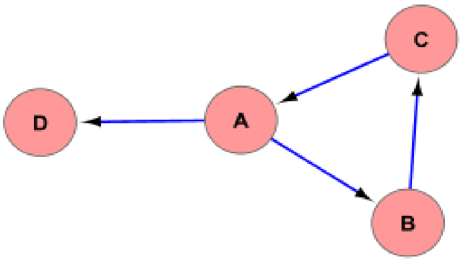
3.4 Finding Attractors

Because of randomness, and also non-Boolean functions associated with each node, none of the described algorithms will work for finding attractors. This is because the state transition network may change its direction at any time and so become completely unpredictable. The proposed approach for finding network dynamics (starting from an initial value) results in a sequence of states called a state transition list or sequence. The list may contain several repeats and many unique states depending on the parameters *I*, *R* and *P*. For making a network from this list the following approach is used:

1) Find unique states in the network and number them (e.g. zero to *N*).
2) Generate a set of *N* nodes, one node for each isolated state in the step 1.
3) For each node v_i, explore the state list from the beginning and set a directed edge which goes from v_i to a node v_j which is a state that appears exactly after v_i on the list.

For example, consider the following state list and the corresponding network:

$$N=\{A,B,C,D\}$$



At the first stage, a node is generated and named *A*. The array is explored to find the locations of *A* and a node which appears exactly after the occurrence of *A* is selected and a directed edge is drawn which points from *A* to the selected nodes (*B* and *D*). This process is repeated for every unique node in the list. The pseudocode of the algorithm is as follow: (adjacency matrix representation is used for making a graph):

```
function graph_maker (list)
    begin
        uniqueStates= retriveUnique(list)
        uniqueStatesLength= length (uniqueStates)
        matrix adjacecnyMatrix[uniqueStatesLength] [uniqueStatesLength]
        for i= 1 to  uniqueStatesLength do
            for j= 1 to  uniqueStatesLength do:
                adjacecnyMatrix[i][j]=0

        for each  item in  uniqueStates do
            begin
            indexesOfUniqueState= locations(item, list)
            for each index in indexesOfUniqueState do
                if(index!=  length( list))
                    begin
                        itemAfterUnique= list[ index + 1]
                        indexOfChild=                locattions(itemAfterUnique,
uniqueStates)
                        indexOfParent= locattions(item, uniqueStates)
                        adjacecnyMatrix[ indexOfParent] [indexOfChild]=1
                    end
            end
    return adjacecnyMatrix

function retriveUnique(list)
    begin
    vector uniqueStates
        for each item in list
            if item is not in uniqueStates
                uniqueStates.append(item)
    return  uniqueStates
```

At first, an adjacency matrix with dimensions of the length of the unique states list is generated and all the cells are set to zero. For each state *v* in the unique states list, if the state is not the last state of the transition states list, the location of the state *w* (in the unique states list) which occurs exactly after *v* (in the state transition list) is retrieved and the cell with row index of location of *v* and column index of location of *w* in the unique states is set to 1. Therefore, if a cell *c* with indexes *i* and *j* is set to 1, there is a directed link from *ith* node to *jth* node (in the unique state list) in the state transition network. The state transition matrix is then converted to an adjacency list and passed to the algorithm proposed by [86]. This algorithm finds elementary circuits of a graph which are attractors of the state transition network. Since the algorithm ignores the singleton attractors, the state transition matrix is utilized to find singleton attractors in which if a cell with indexes *i* and *i* (the same row and column) is 1, this is a self-link between the *ith* state of the unique states list and itself. The output the process is sets of indexes of nodes in the unique state list in which a set starts and ends with the same state, and a sequence of states is in between that is the period of an attractor (regarding singleton attractors, the set contains only two states which are the same). So, the generated sets can be converted back to sets of vectors. Each set is an attractor and each vector contains states of all the genes in the network at a specific time. The final step is to extract information from the attractors. The next section covers the methods which are used to find out if attractors are biologically attractive.

3.5 Analyzing Attractors

The resulting attractors from the previous step need to be analyzed to find out if they are related to specific biological processes. In the rest of this chapter, it is assumed that an attractor is a set of (the same length) vectors. A vector is a set of binary numbers and each number shows state of a gene at the corresponding period. For example, assuming the gene regulatory network (say *G*) contains 4 nodes *A*, *B, C,* and *D* and the state transition graph has one cycle (attractor), an attractor *L* with period 4 and state of the genes at different period *i* can be shown as follow:

$$G=\{A,B,C,D\}$$
$$L_1=\{1,0,1,1\}$$
$$L_2=\{0,0,1,1\}$$
$$L_3=\{1,0,0,0\}$$
$$L_4=\{1,0,1,1\}$$

So the states of the genes in this attractor are as follow:

$$A=\{1,0,1,1\}$$
$$B=\{0,0,0,0\}$$
$$C=\{1,1,0,1\}$$
$$D=\{1,1,0,1\}$$

This study proposes three approaches to investigate biological roles of states of different genes in attractors.

3.5.1 Different state of genes and GO analysis

The simplest and the most straightforward analysis is performing one experiment and isolating the attractors. Next, the states of each gene in different attractors are extracted, and different gene sets with different patterns of state are analyzed separately. Three distinct patterns of gene states can be defined:

- Genes which are constantly one (1) in an attractor.
- Genes which are constantly zero (0) in an attractor.
- Genes which switch their states in the period of an attractor, so they have both zeros and ones in their state pattern.

Considering 3 groups of genes, if the number of genes is small, one can manually look for the role of each gene in the cells and find out if most of the genes in each group have the same biological function (or if they are in the same pathway). If the number of genes is large, gene ontology term enrichment is performed by taking a whole set of genes as the reference and each mentioned group as the target. Using this approach, it can be seen what process in each group of genes is most likely to be involved in desired processes.

3.5.2 Attractors frequency

The simulation process has reasonable randomness and it also covers only a small subset of all the possible states in the state transition network, especially in large gene interaction networks. The result of any one experiment is not statistically significant and also is highly likely to be random. A sensible solution is to perform many experiments and analyze the attractors which are present in most of the experiments. As mentioned earlier, many authors proposed that more than 30 samples are regarded as significant. Considering this fact, *P* and *I* are set based on more than 30 experiments. Using the parameters, more than 30 simulations are performed, and attractors are extracted for all of the experiments. Afterward, the most frequent attractors are found as follows:

1) Attractors are unfolded, such that, periods (vectors) in an attractor are joined into a larger vector which consists of the state of the genes in one attractor. For instance, considering following attractor:

$$L_1=\{1,0,1,1\}$$
$$L_2=\{0,0,1,1\}$$
$$L_3=\{1,0,0,0\}$$
$$L_4=\{1,0,1,1\}$$

Unfolded version of the attractor is:

$$L_U=\{\underbrace{1,0,1,1}_{L_1},\underbrace{0,0,1,1}_{L_2},\underbrace{1,0,0,0}_{L_3},\underbrace{1,0,1,1}_{L_4}\}$$

2) All the unfolded attractors are stored in an array. The array is then explored from the beginning such that the first attractor *A* is selected, and the array is explored again (excluding the selected item) and for each attractor if its similarity with *A* is greater than a threshold (95%), the attractor counter is increased. Since there may be several types of attractors (different lengths), there are two possible ways for defining the similarity between two attractors:

   a) Separate attractors based on their periods and compare an attractor only with the attractors which have the same period. Because the attractors are all 1 and 0 and the have the same length, either Hamming distance or Euclidean distance is utilized to measure the distance between two attractors.

   b) The more challenging option is to use all of the attractors for comparison. So, there is no separation, and an attractor is compared to all the other attractors either with or without the same period. This is done by using the algorithm presented in [100].

Both approaches have some advantages. The first approach takes into account the period of the attractors, so it assumes that different attractors with different period may have different biological roles. The second approach focuses on the state of genes and ignores the period of the attractor. Using this approach, one can only see if genes involved in the cycles are also involved in the specific biological functions. Regardless of the type of similarity measure, the result is a vector of frequency of attractors in all experiments. The vector of attractors is sorted based on the frequency vector and the most frequent attractors are analyzed using the described methods such as GO. The pseudocode of the algorithm for finding the most frequent attractor is as follows:

```
function mostFrequent(attractorsVector, treshold)
    begin
        int frequencies [length(attractorsVector) = 0
            for i=1 to  length(attractorsVector)
                for j=1 to  length(attractorsVector)
                    if(i!=j)
                        if(similarity( attractorsVector[i],attractorsVector[j])>threshold))
                            frequencies[i]+=1
        ascendingsSortBasedOn( attractorsVector, frequencies)
        return ( attractorsVector)
```

3.5.3 Attractor clustering

Large numbers of attractors are often generated by performing several experiments. Focusing only on a few of the most frequent attractors may miss important states of the genes in other attractors. Attractor clustering provides a solution for focusing on the state of genes in all attractors. Using either of the similarity measurement approaches (in this case Euclidean distance), the distance between each pair of attractors is calculated, and the result is stored in a distance matrix. Using a distance matrix, a clustering method (UPGMA) is used to cluster the attractors into different groups. This method generates a tree (or more depending on similarity measure) which starts at a root and divides the attractors into the largest subgroups. Each subgroup has its own root which again separates its attractors into different groups. These sub-trees continue until terminating at the leaves which are the attractors. The pseudocode for preparing distance matrix is as follow:

```
function distanceMatrix(attractorsList)
        begin
                uniqueStates=retreiveUniqe( attractorsList)
                matrix DMatrix=[length( uniqueStates)][length( uniqueStates)]
                for i=1 to length( uniqueStates)
                        for j=1 to length( uniqueStates)
                                if(i==j)
                                        DMatrix[i][j]=0
                                else
                                        DMatrix[i][j]=similarity( uniqueStates[i], uniqueStates[j])
                                        DMatrix[j][i]=similarity( uniqueStates[j], uniqueStates[i])
        tree=UPGMA(DMatrix)
```

The variable *tree* is the clustering result produced by UPGMA method. The tree has a one sub variable denoted as *child*. This variable shows of descendants of the current root. For example, considering the variable *tree* in the pseudocode, tree->child[1] is the first cluster under the main root of the tree. This sub-cluster may be a root itself or it may be a leaf. Having a tree, this text proposes two methods for further investigations:

1. Bottom-Up approach: This approach is similar to breadth first search (BFS). The algorithm starts from the main root and traverses its children. The attractors classified under each child are stored. This means that the largest sub groups of attractors are separated. The percent of simulated gene expression (percent of being one) under each group are calculated. Only differentially expressed genes are selected and the $\log_2$ ratio of their expression is calculated and the genes are sorted based on the ratio. Considering the sorted gene expression, the genes which are mostly on (1) in one group and off (0) in the other cluster are regarded as one group and the genes which have the same situation regarding the other group are also selected and finally the rest of genes which is not highly overexpressed in one of the clusters are grouped. GO analysis is performed, taking all the genes in all the groups as the reference and each group as the target. If a satisfactory result is not seen, breadth first search goes one step further, and the process regards each child of the main root as a root, and the whole process is repeated for that child. The only difference is that one can use the whole gene set under the first root of the tree or use the genes under each root as a reference. This process is repeated either until a satisfactory result is seen (e.g. two groups which are significant in T-cell differentiation of CD8+ and CD4+) or no more clusters are left.

2. Top-Down approach: Instead of focusing on the state of genes in attractors and finding out the possible biological process of different clusters, the top-down approach assumes that the clusters are related to predefined cell subtypes (e.g. CD8+ and CD4+ in case of T-cells). Two microarrays (table 3.2) were extracted and normalized using the methods presented in section 2.3. Log_2 ratios of percentage of gene expressions in different clusters are extracted (the same as the previous method). Log_2 ratios of the same genes in different subtypes of the T-cells are also calculated from the microarray experiments. The result is two sets of vectors, one set for the simulation and the other set for each of the actual experiments. Each set may contain one or more vectors, depending on the clustering result and defined number of subtypes. PCC of each pair of vectors (in two different sets) is calculated. If the PCC between two vectors is high enough, it is either the sign of involvement of the gene states in two different clusters for cell differentiation (or their processes) or indication of a reasonable relationship between simulated and experimental gene expressions. Specifically, the expression ratio of gene *k* between clusters *i* and *j* is shown by $r(k)_{ij}$ and the experimental ratio of gene *k* between two cell subtypes *i* and *j* is represented by $e(k)_{ij}$. PCC is calculated between two vectors of simulated and experimental ratios of *N* genes:

$$R=\{r(1)_{ij}, r(2)_{ij}, r(3)_{ij}, \dots, r(N)_{ij}\}$$
$$E=\{e(1)_{ij}, e(2)_{ij}, e(3)_{ij}, \dots, e(N)_{ij}\}$$

Table 3.2. Microarrays used for the ratio of gene expression comparison.

ID	Title
GSE14926	Analysis of the impact of the method of cell selection on the gene expression profile of human CD4 and CD8 T cells [98]
GSE16130	Gene expression of TCR-alpha/beta CD4- CD8- human T cells [99]

These two methods can also complement each other. For example, if PCC between two clusters is high compared with other clusters, one can do the first bottom-up approach only for the interesting cluster.

3.6 Simulation Environment

Results of the computations are stored in a SQL database for further analysis and reference. Since performing several experiments with the large number of iterations is computationally extensive, four programs were developed using Java, C++, python and R programming languages. Due to well adapted nature Java language to database systems, a Java program is used to initialize the SQL database with interaction data, initial values and parameters for each experiment. Since C++ is fast for several types of calculation especially on supercomputers and for highly parallel algorithms, the C++ program performs the simulation (using parallelization of the present algorithm) and passes the data to the java program to store the data to the database. Python has several libraries for working on strings and vectors; therefore, a program developed in this language is used to analyze the attractors. The result is passed to R for further statistical and microarray analysis. The next section provides the results of performing the simulation on the T-cell PPI.

4. Result

4.1 Initial values

Using the described method in the section 3, the initial states of 254 genes are calculated. The resulting vector was an array of length 254 in which each cell corresponds to a gene and can take value either one or zero, meaning that the gene is on or off, respectively (153 genes were expressed and 101 genes were not expressed).

4.2 Finding parameters

Considering *P* is set to one, 40 different experiments (statistically significant) were performed, and the number of unique states was plotted. Figure 4.1 shows 4 examples of 40 experiments. The number of iterations ranges from 10000 to 70000 (the maximum iterations the computer allows) with step size 5000. As the figure shows, behavior of the system to produce the states is random. This situation was the same for all 40 experiments. Therefore, stable patterns were not generated in the experiments. The number of unique states may have several peaks or no peaks with minor fluctuations. The system is supposed to produce more states when *I* is increased, but in many figures (as the curve shows) when the number of iterations is increased, number of unique states is decreased. But even this pattern is not stable because there are several peaks at different (and even far away) number of iterations. For example, there are 3 peaks at 15000, 30000 and 50000 iterations in experiments number 3 (in figure 4.1). The only observable pattern in the data is that when number of iterations is increased, the system produces fewer jumps between different numbers of unique states. Because the system is supposed to produce more states (and with higher jumps) when *I* is increased, and this is not apparent in the diagrams and the curves are smoother in a large number of iterations, it is assumed that the generated number of states in large iterations are more desirable. But the behavior of the system is still random and there is no pattern between the iteration number and number of produced states. So, $\log_2$ of standard deviation (SD) of each iteration number within all the experiments is computed (figure 4.2) in order to see which number of iterations has fewer fluctuations.

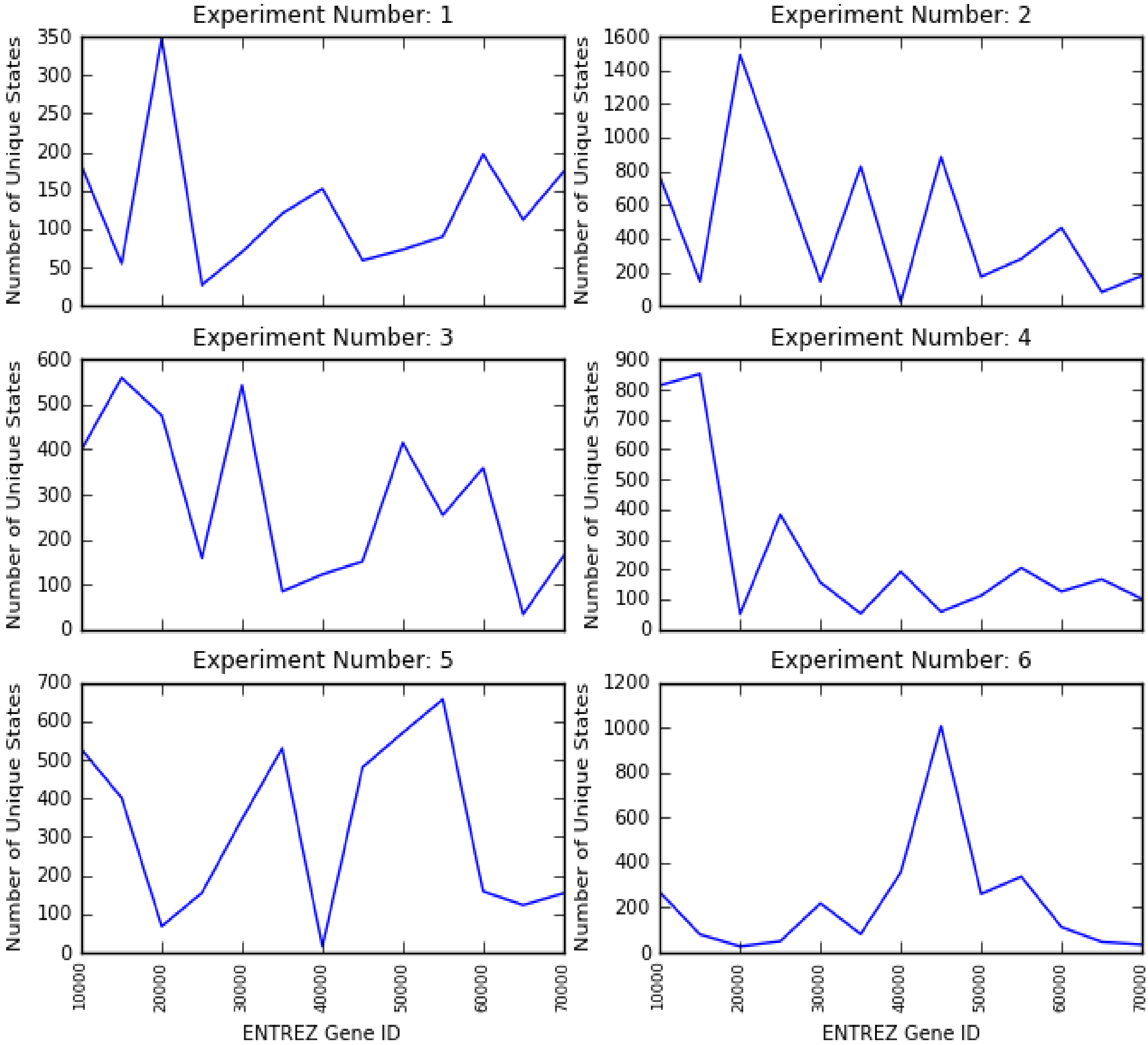


Figure 4.1. Unique States Produced By Different Iterations.

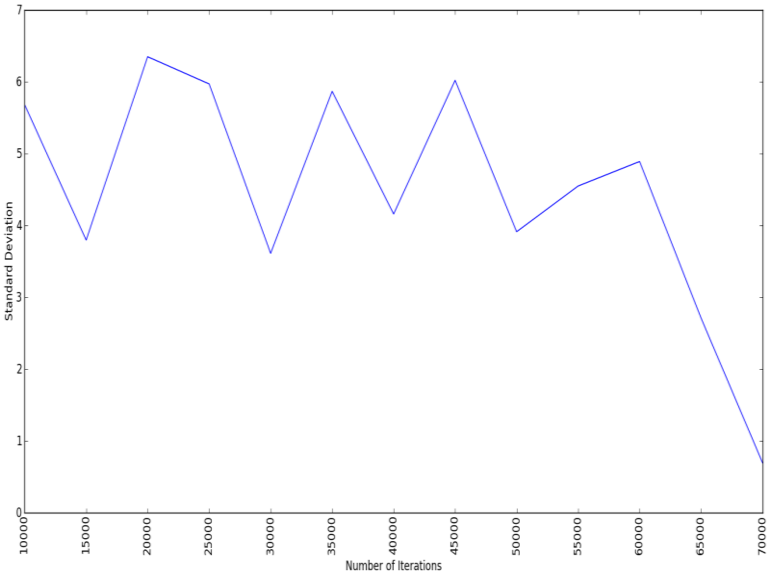


Figure 4.2. Standard Deviation in all experiment with different iterations.

The SD values from 10000 to 50000 show a stable pattern in which SD increases and decreases within two different iterations with small size. This pattern stops at 60000 iterations value and the SD suddenly drops. It is assumed that the behavior of the system changes at that point. So, the 60000 iterations level which was selected, and the same process was repeated for different *P* parameters and 60000 iterations. The only difference is that instead of plotting different *P* in several number of iterations, a constant number of iterations (60000) is selected and the plot is drawn by taking into account the number of unique state in each of the experiments (figure 4.3).

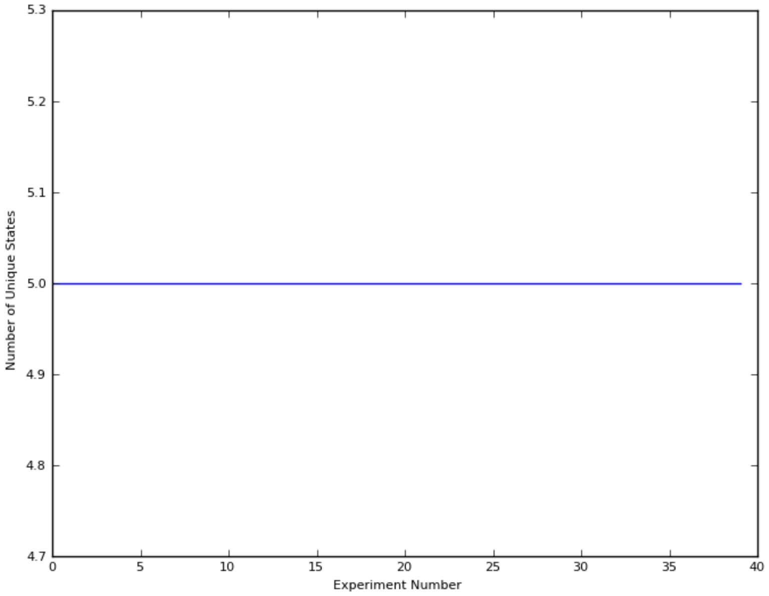


Figure 4.3 Number of unique state (P is less that 0.006)

Figure 4.3 shows that the number of unique states is constant when *P* is very small. This is because when P is small, the random component is almost zero. Therefore, RC does not affect the equation and the system produces the same states for the genes at different times. Since small step size does not have a remarkable effect on the number of the states, the step was selected to be 0.05 and the system was set to start from 0.01 and go to an arbitrary number 0.3, and number of generated unique states was plotted. As it is obvious in figure 4.4, the behavior of the system is totally random in 40 different experiments (only one result for *P* 0.1 is shown, but the situation is the same in other *P* values).

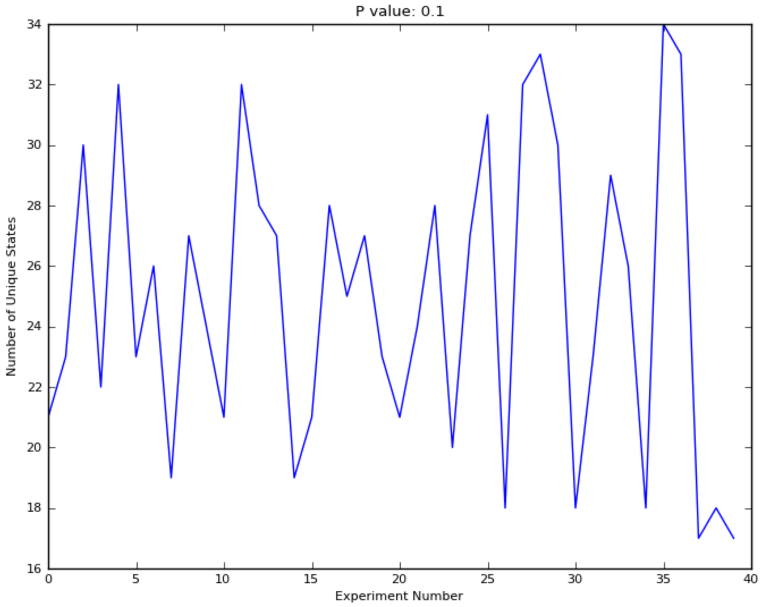


Figure 4.4. Number of unique state produced with 60000 iterations and P parameter=0.1.

Since no steady pattern was observed when performing 40 different experiments with different *P* parameters starting from 0.01 to 0.3, standard deviation is calculated for each *P* in 40 experiments (figure 4.5).

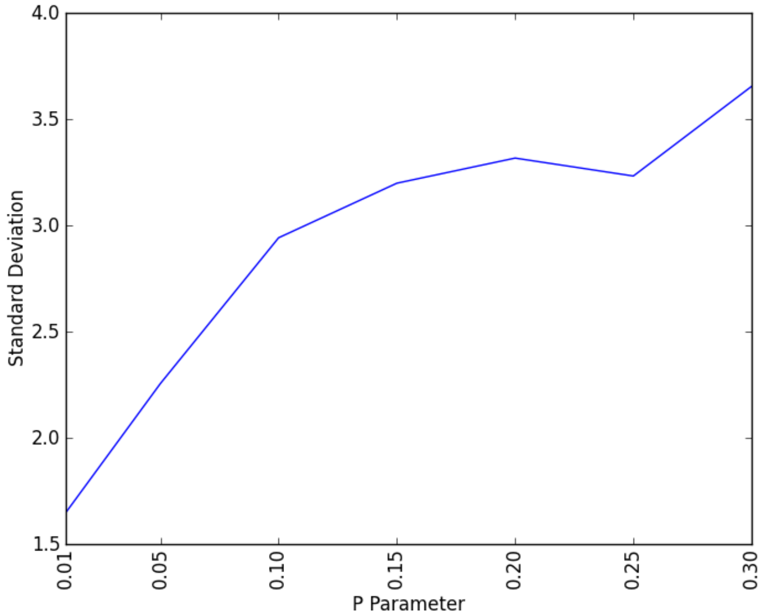


Figure 4.5. Standard Deviation for different standard deviation for different P parameter.

As it is shown in the figure 4.5, randomness is increased when *P* is increased. This result is exactly the behavior that is expected from the system with different *P* values. This is because large *P* adds more randomness to the equation and the system produce larger deviations. Two interesting points have the potential to be good *P* parameters. Parameter 0.1 can be selected because at that point, the curve starts to behave differently, and 0.25 can be chosen because the standard deviation is decreased. Forty experiments were performed (*P* and *I* were set to 0.1 and 60000, respectively), and the result was 40 different networks. Figure 4.6 shows an example of the state transition network (a directed graph) for one of the experiments. As it became apparent, there are several attractors (either singleton or cyclic) in the network. The networks were then analyzed in order to identify the attractors.

4.3 Attractors Analysis

Eight hundred and thirty three (833) attractors were identified. There were two types of attractor with 2 or 4 periods. The frequency of each attractor was computed within all the experiments. The most frequent attractors were selected (with 26 and 21 repeats with periods 4 and 2, respectively). Since networks with less than 4 nodes are not interesting (due to the fact that they may not be informative), only genes in sub-networks with more than 4 nodes were considered (135 genes). The states of the selected genes were extracted. Since only two attractors with period 2 and 4 were chosen, the state of a gene in an attractor is a vector (with length the same as attractor period number) in which each entity in the vector is the state of the gene at a specific time within the attractor period. Genes with three different patterns of state in two different attractors were separated. The result was 10 genes with a pattern of only “1” (arrays of genes were always expressed), 49 genes with a pattern of combined number of “1” and “0” (arrays of genes were sometimes expressed and sometimes not), and 76 genes with a pattern of only “0” (arrays of genes that were not expressed). The pattern of expression in attractors with period 4 was: 10 genes with “1”, 50 genes with combined, and 75 genes with “0”. The percentage of states of each gene in the other (831) attractors verified that the result was completely consistent. GO term enrichment analysis was performed on the 3 sets of gene for each attractor. Table 4.1 and 4.2 show the top hit (complete table is in appendix 2) of GO analysis for attractors with period 2 and 4, respectively.

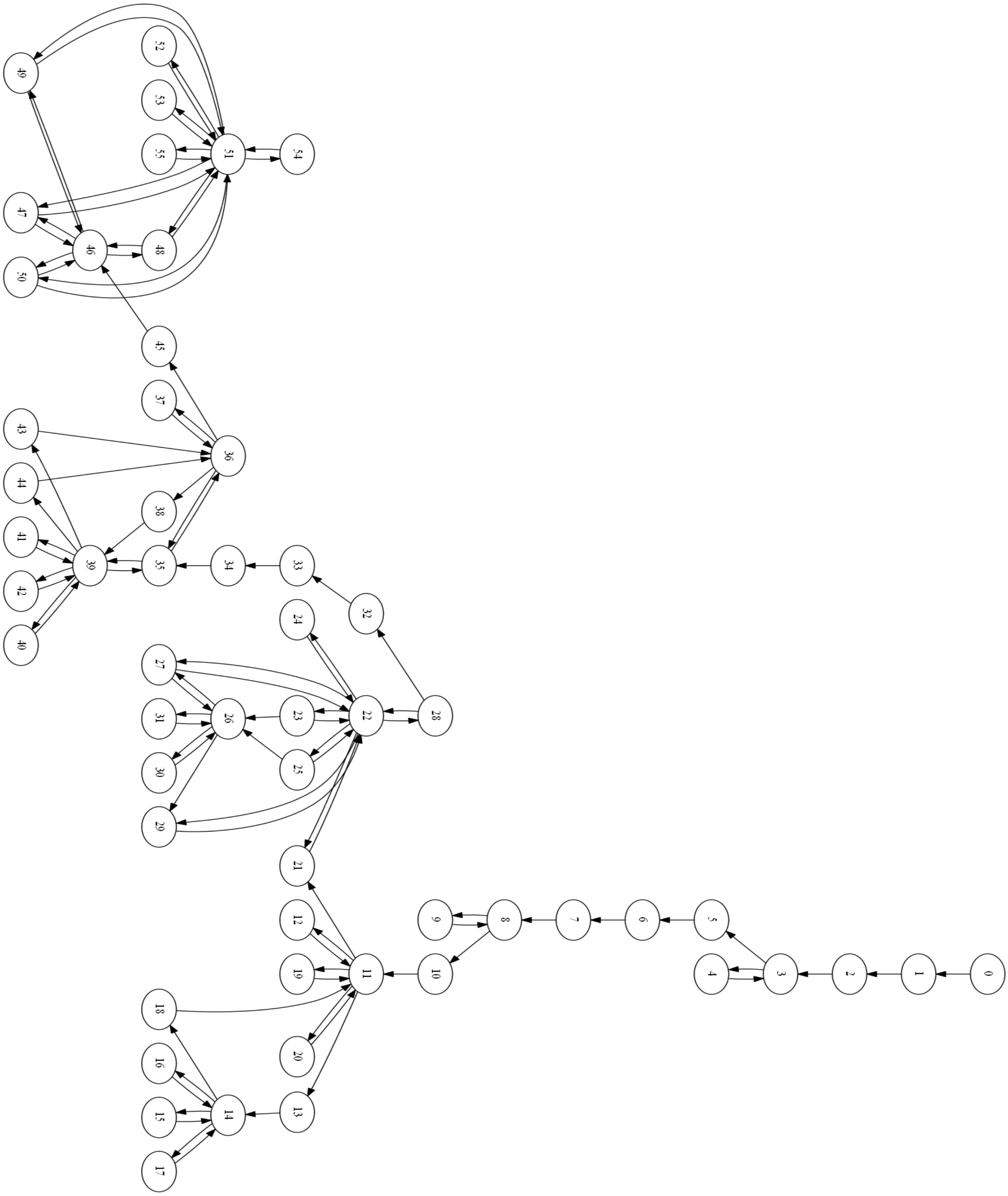


Figure 4.6. State transition network when P is set to 0.1 and number of iterations is 60000.

Table 4.1. Go Analysis for attractor with period 2.

Term	Count	%	P-Value	Genes	List Total
Only Ones Pattern					
regulation of system process	4	40.0	.0162	2771, 5781, 1956, 5604	9
Only Zeros Pattern					
Leukocyte transendothelial migration	20	27.03	.0030	5747, 8503, 5175, 7409, 7414, 3683, 1499, 3383, 5290, 4478, 387, 3689, 2185, 71, 81, 60, 5294, 5829, 5295, 5296	72
Combined Pattern					
protein kinase cascade	22	48.89	.0142	7535, 6714, 8517, 1147, 6774, 5618, 6772, 6464, 7124, 5058, 5608, 8651, 7189, 6416, 3480, 1399, 2688, 2885, 8737, 4793, 3654, 10746	50

Table 4.2 Go Analysis for attractor with period 4.

Term	Count	%	P-Value	Genes	List Total
Only Ones Pattern					
regulation of system process	4	40.0	0.0162	2771, 5781, 1956, 5604	9
Only Zeros Pattern					
Leukocyte transendothelial migration	20	27.78	0.0024	5747, 8503, 5175, 7409, 7414, 3683, 1499, 3383, 5290, 4478, 387, 3689, 2185, 71, 81, 60, 5294, 5829, 5295, 5296	71
Combined Pattern					
regulation of kinase activity	14	28.57	0.0122	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44

The results indicate that none of the groups is significant in T-cell differentiation. Although some processes are well described by different groups of genes, the result is not verification of significance in T-cell differentiation terms. Therefore, 833 attractors were separated into 2 sets of 568 and 265 attractors with 2 and 4 periods, respectively. UPGMA clustering was performed on each set; the resulting trees are shown in figures 4.7 and 4.8. At the main root of the both trees, attractors are classified into two different clusters (a small and a large cluster). One branch of each main root has two children (inner top and bottom branches), but the other branch has more children and its size is different between the two figures. The number of ones and zeros in the attractors clustered under the main branches of the two clustering results was calculated. The percentage of "ones" is calculated for each gene, and the genes which were oppositely (with 5% error compensation) expressed in two main clusters was selected, and $\log_2$ ratio was computed between each pair of the same genes in two main clusters. The resulting vector (of ratios) was sorted, and the gene expression (percentage) was ordered based on the sorted ratio vector (figures 4.9 and 4.10).

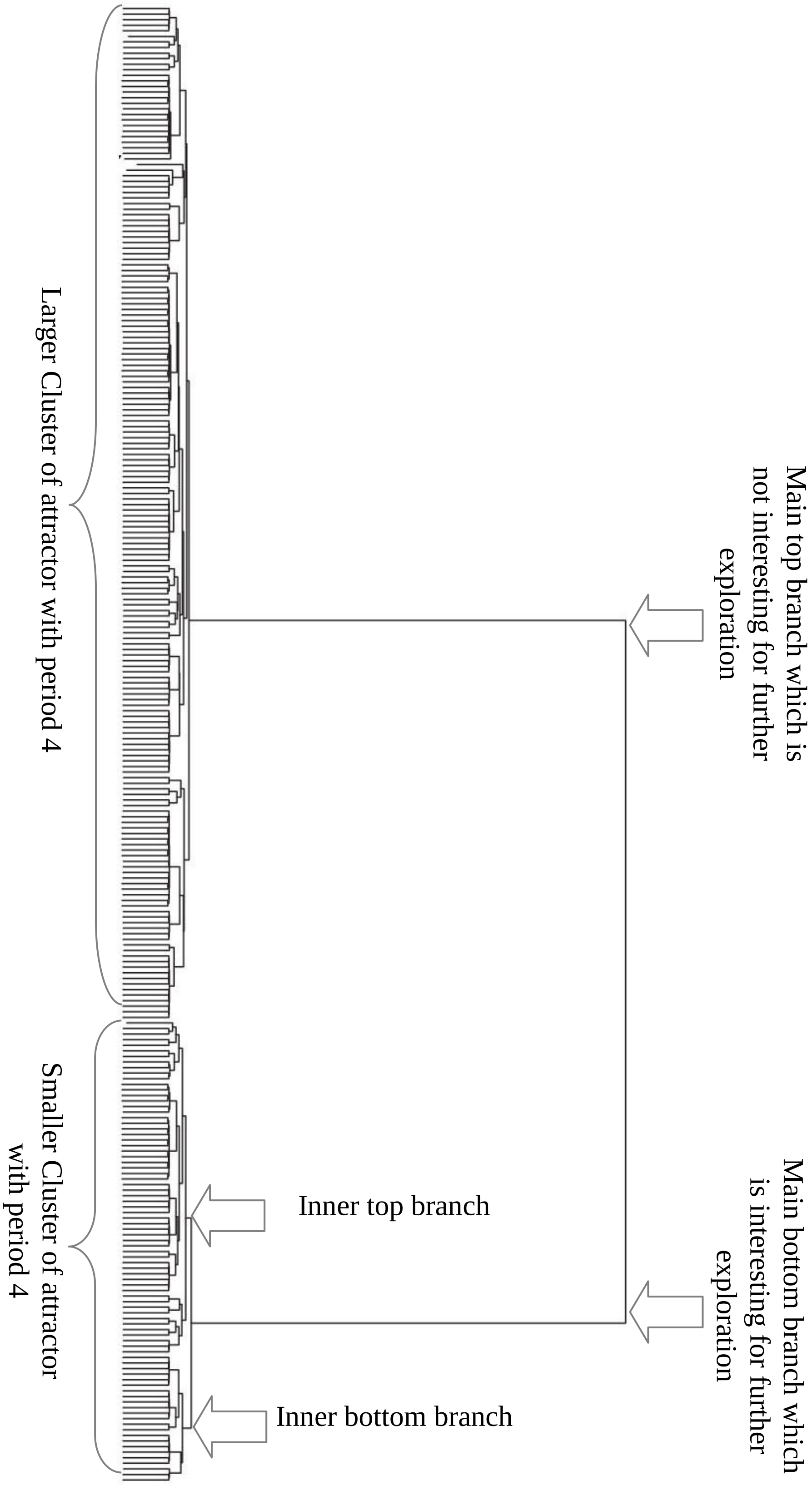

Larger Cluster of attractor with period 4
Smaller Cluster of attractor with period 4
Inner top branch
Inner bottom branch
Main top branch which is not interesting for further exploration
Main bottom branch which is interesting for further exploration

Figure 4.7. Clustering result of period 4 attractors using UPGMA method.

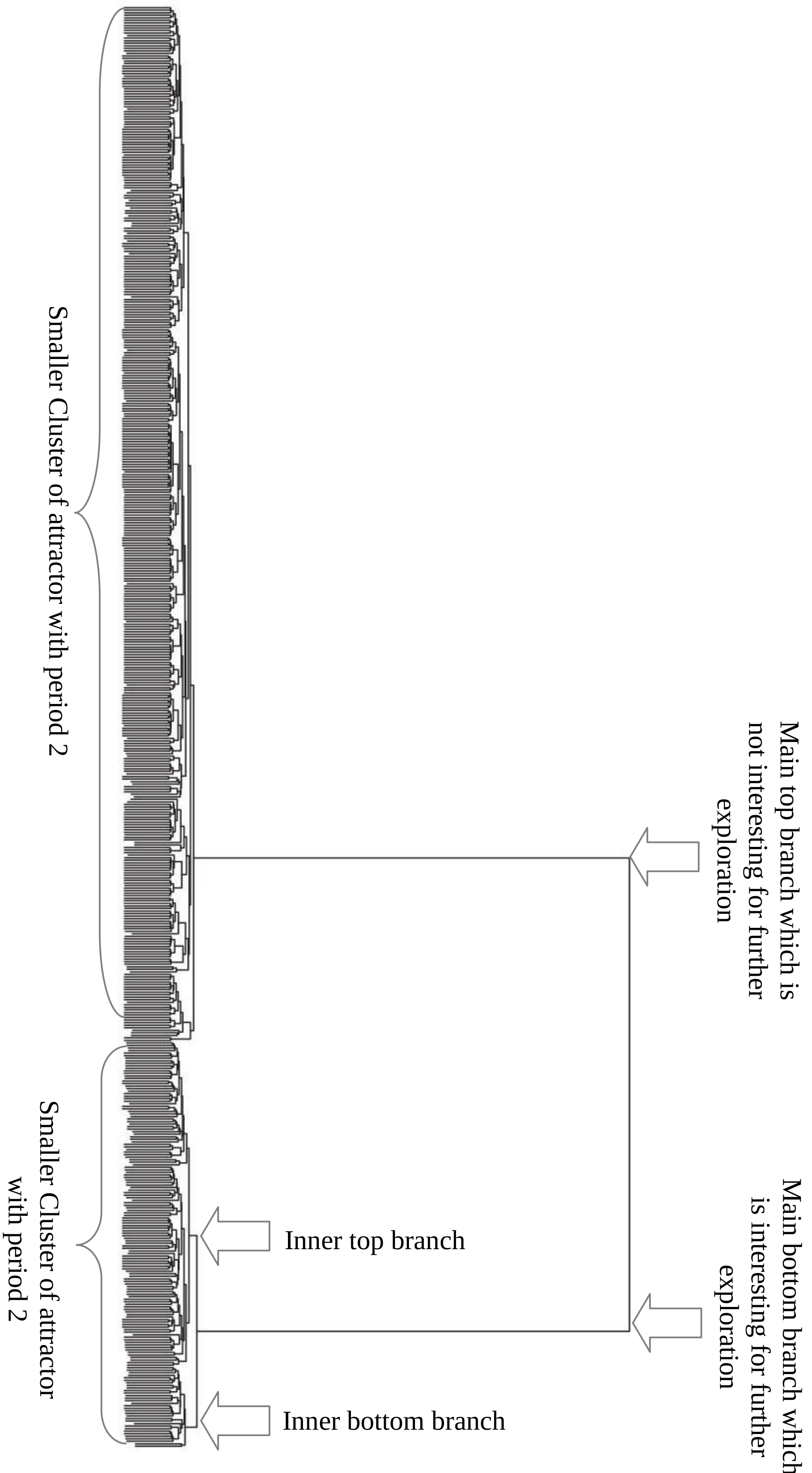

Smaller Cluster of attractor with period 2
Smaller Cluster of attractor with period 2
Main top branch which is not interesting for further exploration
Inner top branch
Main bottom branch which is interesting for further exploration
Inner bottom branch

Figure 4.8. Clustering result of period 2 attractors using UPGMA method.

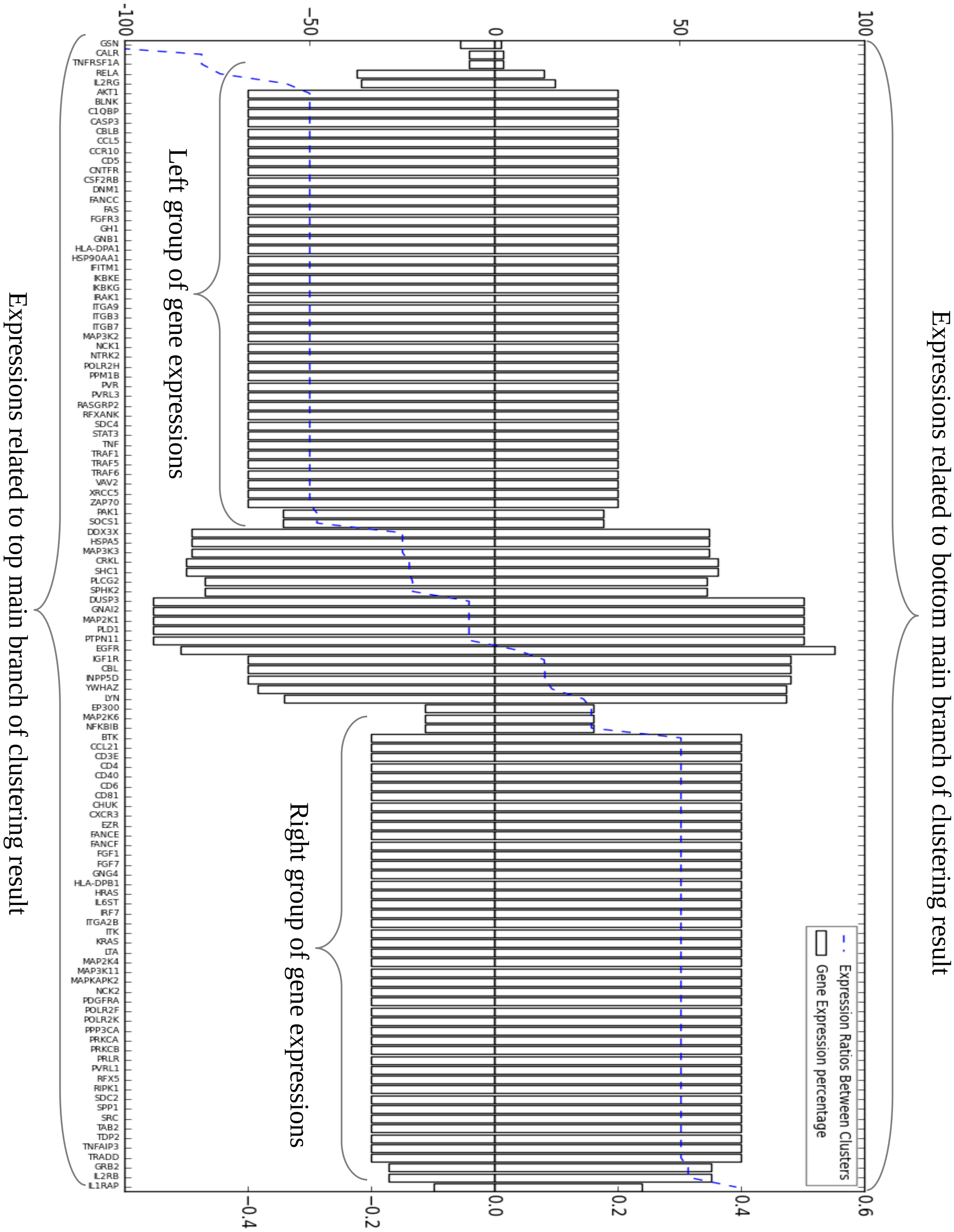


Figure 4.9. Gene expression percentage shows 3 separate groups of genes in different clusters in attractor with period 2.

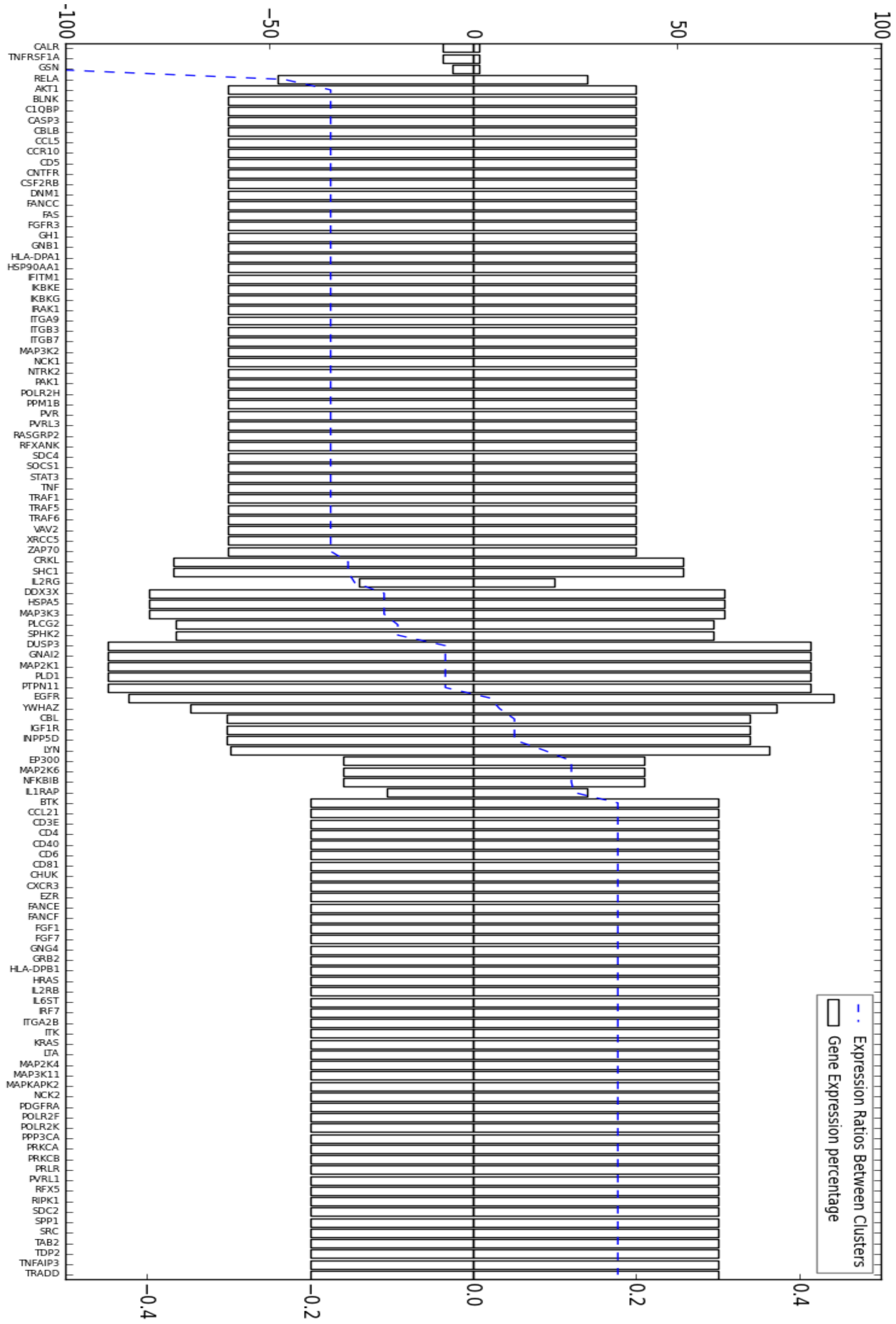


Figure 4.10. Gene expression percentage shows 3 separate groups of genes in different clusters in attractor with period 4.

Figure 4.9 and 4.10 clearly show that the genes are exactly grouped into 3 different categories (the ratio describes the behavior of the expression fluctuation). The left group contains the genes which are highly expressed in main top clusters of both the clustering trees. The center group contains the genes with shared expressions between two clusters and the right group shows the genes which are highly expressed in the bottom clusters of the clustering trees. Considering the three groups for each set of clustered attractors, GO analysis was performed on each group, taking the whole gene set (all the genes in figures 4.9 and 4.10) as background and each of the three groups as the target. The results (the top hit, complete tables are in appendix 2) are shown in tables 4.3 and 4.4.

Table 4.3. GO analysis for three groups of clustered attractors with period 2.

Term	Count	%	P-Value	Genes	List Total
Right Group					
p38 MAPK Signaling Pathway	7	14	0.0651	6416, 3265, 2885, 8737, 9261, 5608, 8717	35
Center Group					
cell fraction	9	50.0	0.0019	3480, 2771, 5781, 5336, 5337, 5604, 7534, 56848, 4067	16
Left Group					
disulfide bond	21	42.0	0.0688	811, 3113, 25945, 8517, 1271, 2934, 3561, 7124, 3680, 3690, 1439, 921, 2261, 3695, 7132, 2826, 6352, 355, 2688, 5817, 4915	50

Table 4.4 GO analysis for three groups of clustered attractors with period 4.

Term	Count	%	P-Value	Genes	List Total
Right Group					
p38 MAPK Signaling Pathway	7	14	0.0651	6416, 3265, 2885, 8737, 9261, 5608, 8717	35
Center Group					
cell fraction	9	47.37	0.0033	3480, 2771, 5781, 5336, 5337, 5604, 7534, 56848, 4067	17
Left Group					
leukocyte homeostasis	5	10.2	0.0932	8517, 355, 2176, 207, 836	49

Results of GO analysis show no significance in T-cell differentiation terms, but the right groups of gene expression, in both clustered attractors with different periods, mostly contain the genes which are related to CD4+ T-cells. This group of genes is related to the bottom clusters of UPGMA trees. Both of these clusters in the two trees contain two inner branches (inner top and bottom branches). So, it is more sensible to explore how these branches compare to groups under main top roots of the trees. The main bottom branches of both trees were explored one step further, and the same process (calculating simulated gene expressions and ratios) were repeated but this time between two inner top and bottom branches of the main bottom branches of both trees. As the figures 4.11 and 4.12 show, the genes are separated into three groups of expressions (expressed in the bottom or top inner clusters and a group which is expressed in almost both of the clusters).

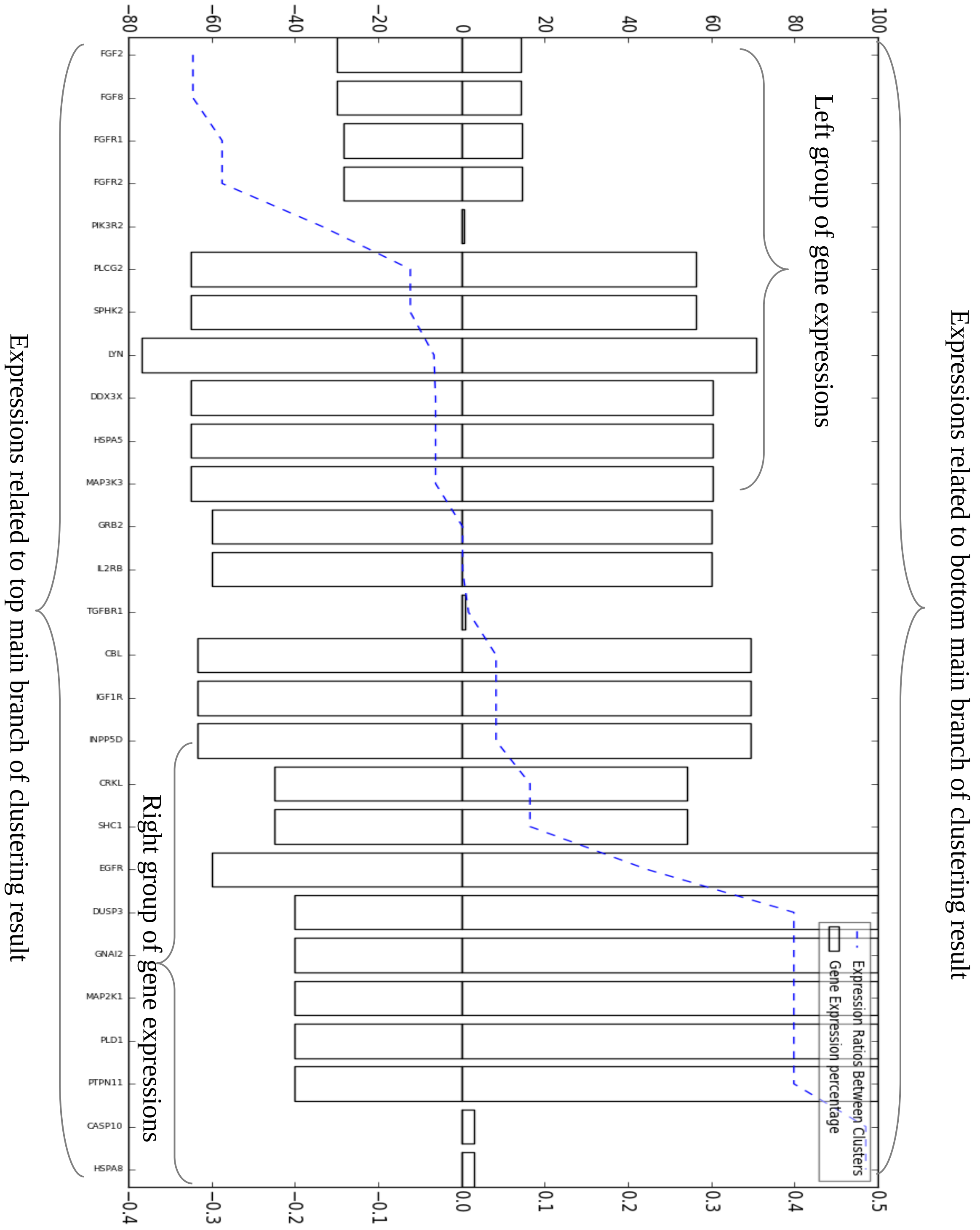


Figure 4.11. Gene expression percentage shows 3 separate groups of genes in two inner clusters in attractor with period 2.

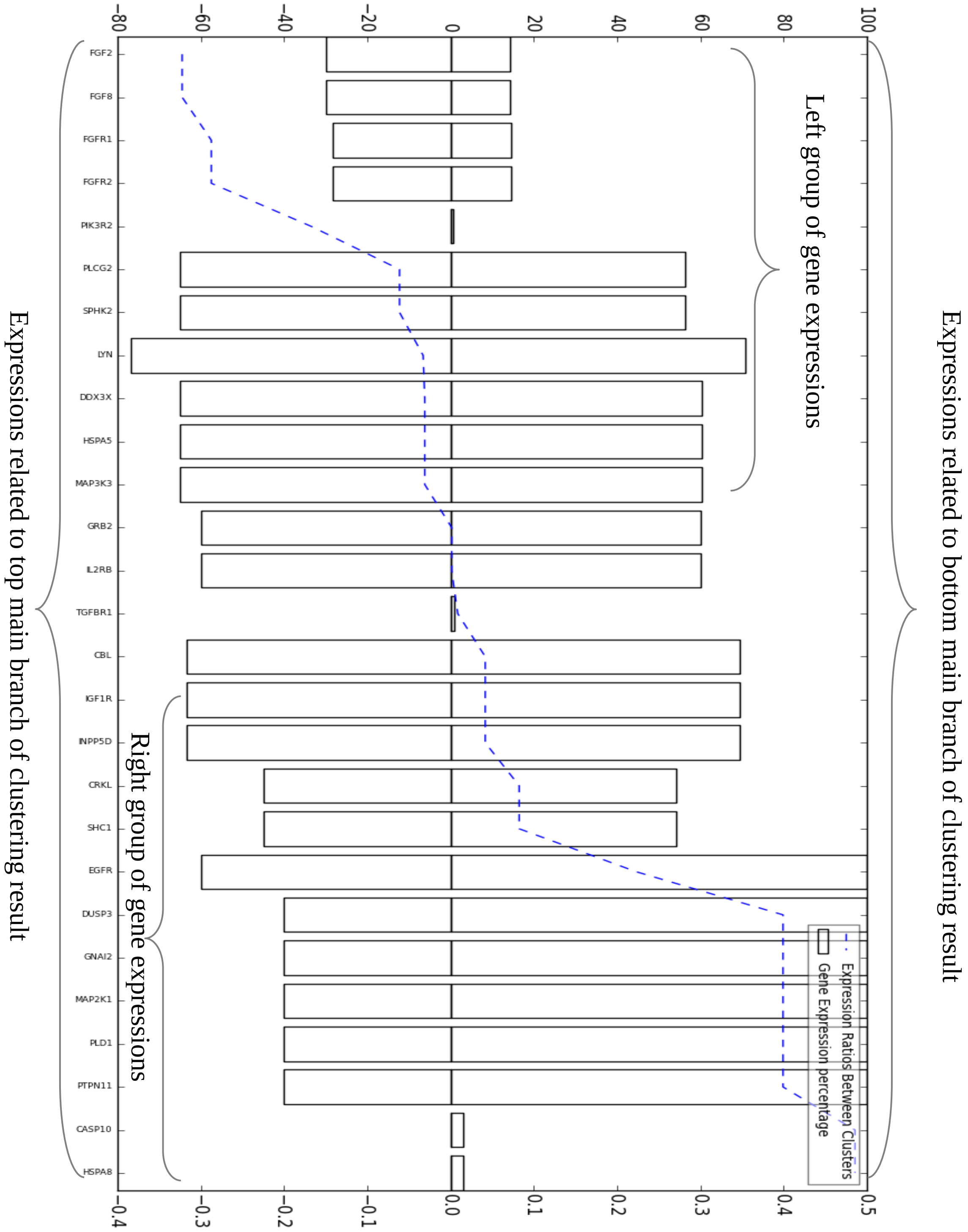


Figure 4.12. Gene expression percentage shows 3 separate groups of genes in two inner clusters in attractor with period 4.

GO analysis was performed on all the three groups, taking whole genes (in figure 4.11 and 4.12) as background set and each of three groups as the target. The result of GO analysis is shown in tables 4.5 and 4.6 (only the top hit is shown. Complete table is appendix 2).

Table 4.5. GO analysis for three groups of sub-cluster of attractors with period 2.

Term	Count	%	P-Value	Genes	List Total
Right Cluster					
cell fraction	7	58.33	0.0673	2771, 5781, 5336, 5337, 5604, 56848, 4067	11
Center Cluster					
SH3 domain binding	2	66.67	0.1453	867, 3635	3
Left Cluster					
signal	6	50.0	0.1440	3560, 2263, 2253, 3309, 7046, 2260	12

Table 4.6. GO analysis for three groups of sub-cluster of attractors with period 4.

Term	Count	%	P-Value	Genes	List Total
Right Cluster					
cytoplasm	6	46.15	0.2116	5781, 5337, 6464, 867, 3312, 3635	13
Center Cluster					
receptor	3	100.0	0.0598	2885, 3560, 7046	3
Left Cluster					
fibroblast growth factor receptor signaling pathway	4	36.36	0.1039	2247, 2263, 2253, 2260	10

The result of GO analysis does not show significant terms in T-cell differentiation. Therefore, the Top-Down method was used to find the relationship between the ratio of simulated gene expression and microarray data. The result of simulated ratio of gene expressions in both analyses on the main and inner branches of the tree were plotted against the actual ratio of expression between all the genes in the simulation data set expressed in CD8+ and CD4+ T-cells (figures 4.13-4.20). CD8+ and CD4+ T-cells were selected because more data are available for those genes and also nearly all the CD4+ T-cells related genes are clustered in only this group. PCC between the resulted ratio of CD8+ and CD4+ cells gene expression (which is an array of float numbers as the same as simulated ratio of gene expression) and simulated ratios were calculated for each analysis. As figures 4.13 to 4.20 show, no significant similarity is observable between microarray experiments and different simulations. This clearly shows that the actual experiments (in-vitro) are completely different from the simulated experiments. Although the ratios (solid and dashed) curves behave in two totally different manners, the amount of changes (or behavior of the curves) is similar in a limited number of genes such as IL2RB gene. Since these similarities are quite rare (may be by random), one can assumes that the behavior is quite different.

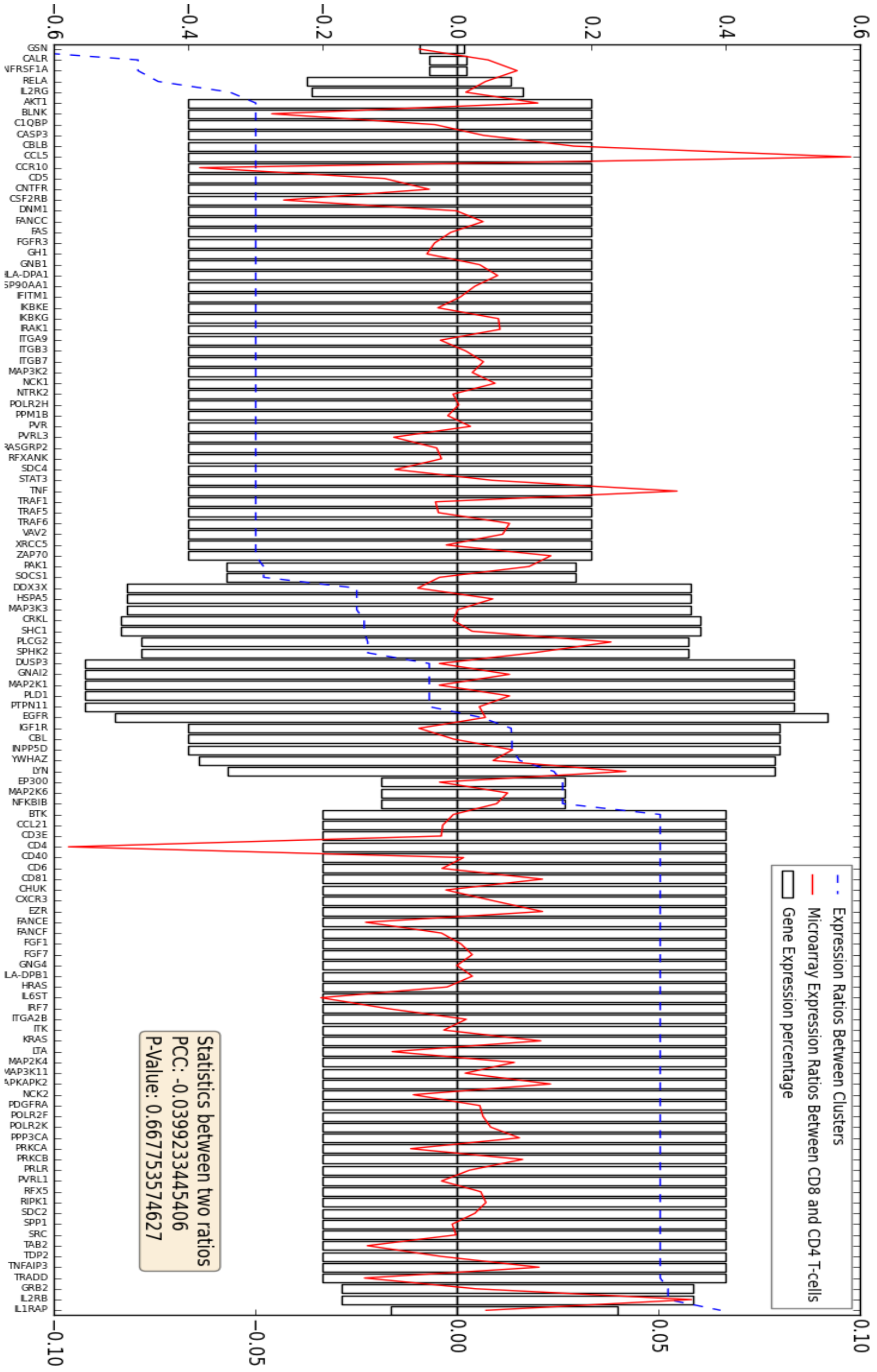


Figure 4.13. Comparison between simulated expression ratio of genes in two groups of the main root of clustered attractors with period 2 and experimental microarray data in GSE14926 time series shows completely different behavior in in-vitro and simulated gene expressions.

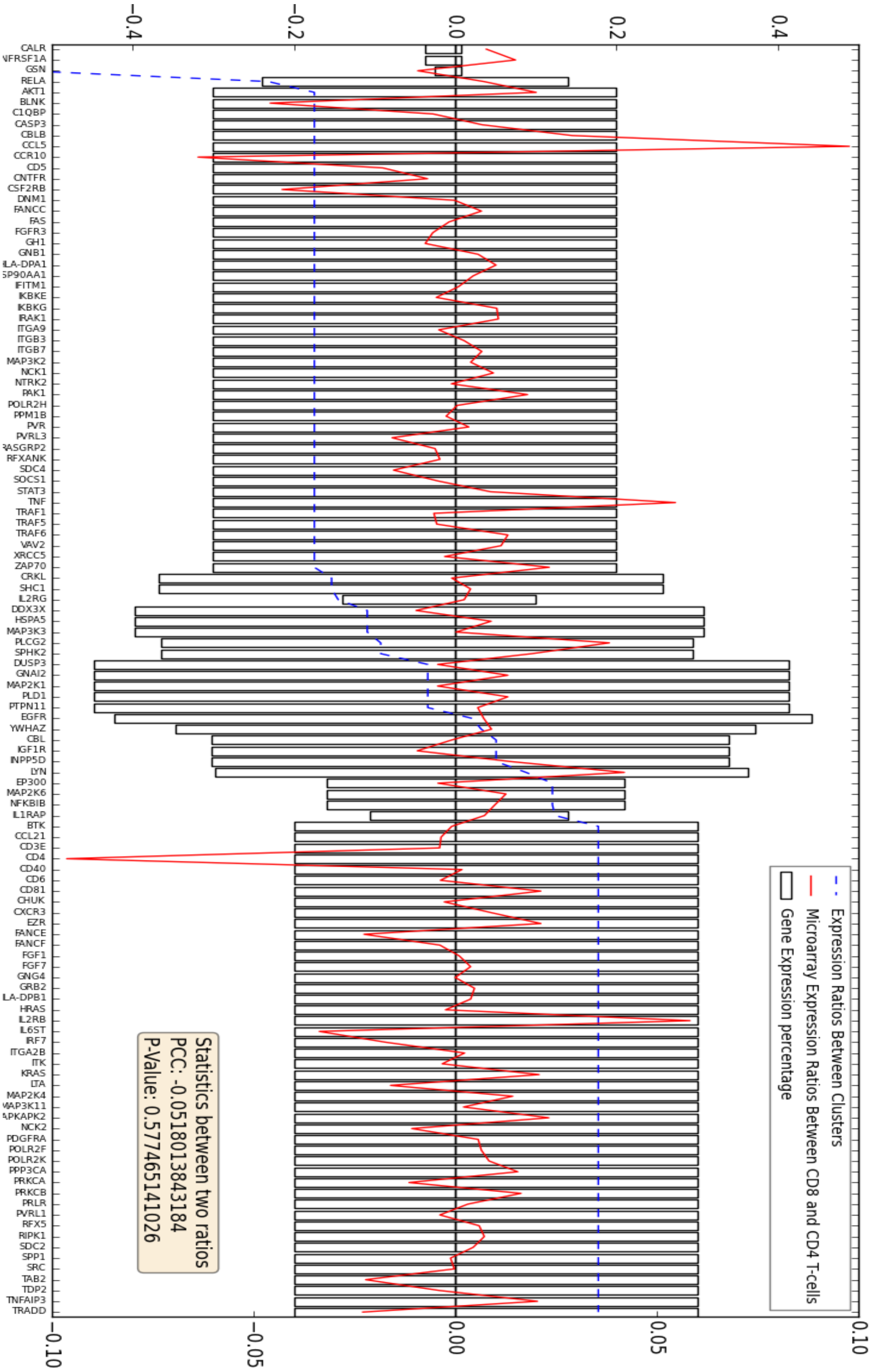


Figure 4.14. Comparison between simulated expression ratio of genes in two groups of the main root of clustered attractors with period 4 and experimental microarray data in GSE14926 time series shows completely different behavior in in-vitro and simulated gene expressions.

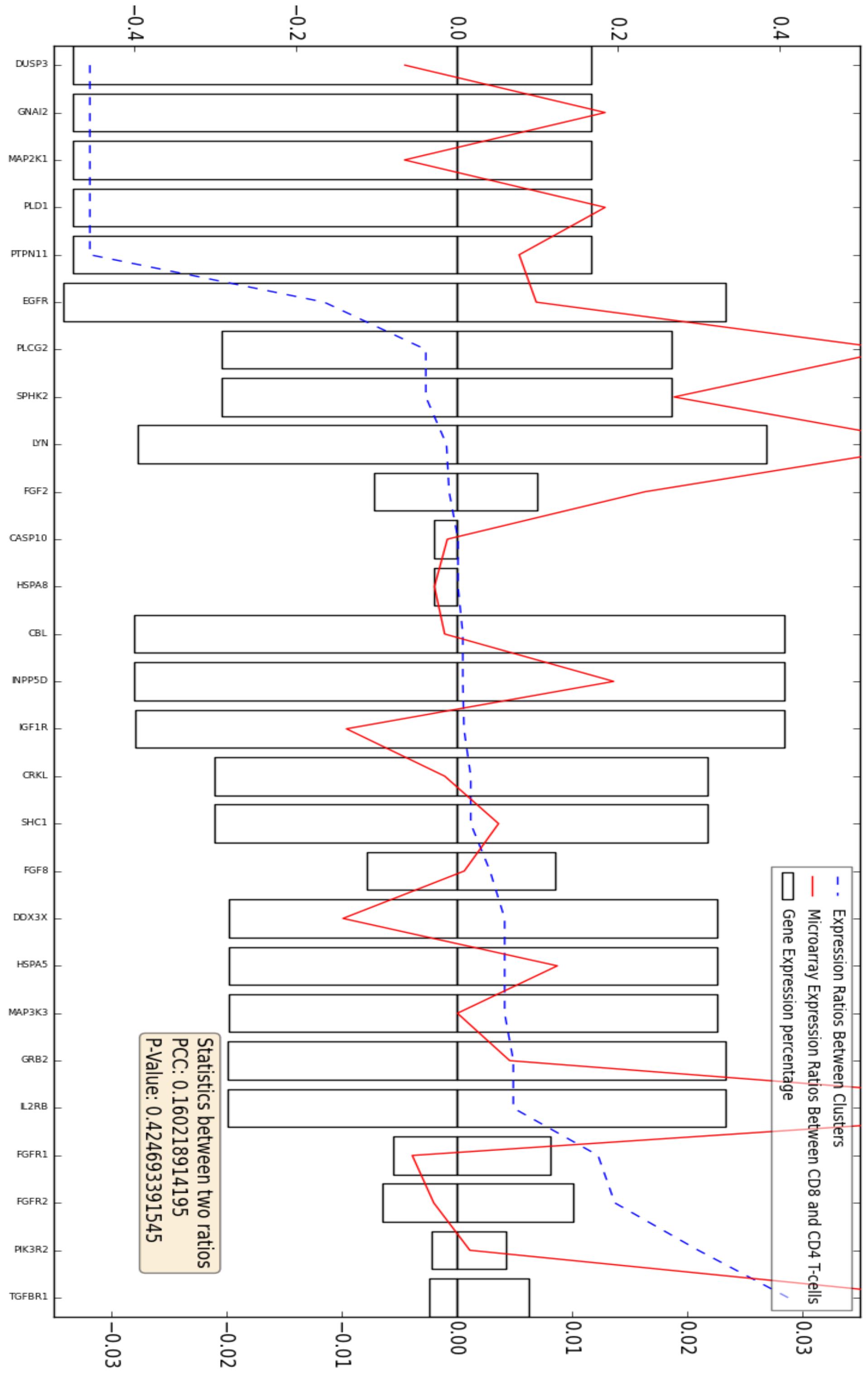


Figure 4.15. Comparison between simulated expression ratio of genes in two groups of the first right root of clustered attractors with period 2 and experimental microarray data in GSE14926 time series shows completely different behavior in in-vitro and simulated gene expressions.

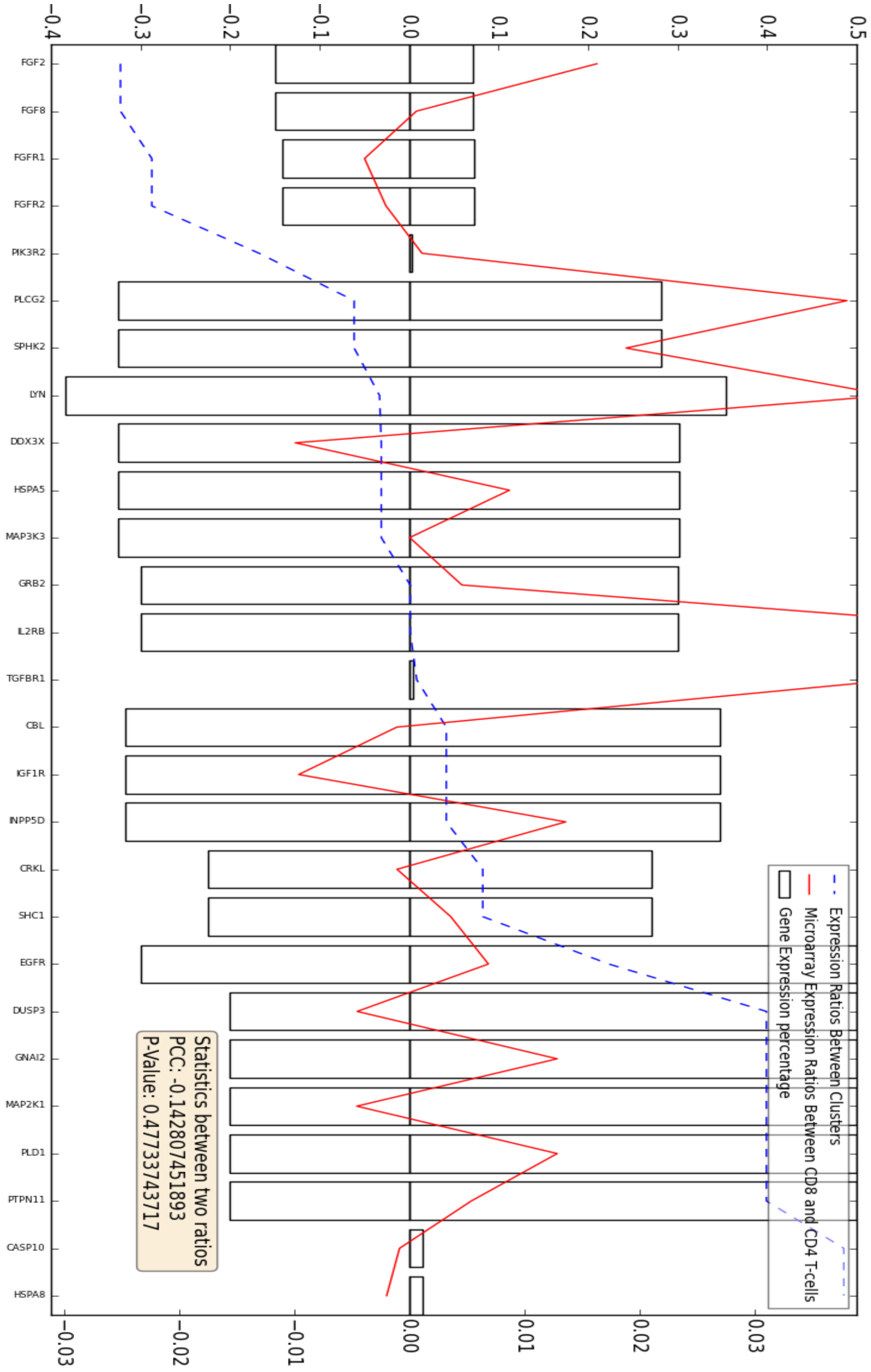


Figure 4.16. Comparison between simulated expression ratio of genes in two groups of the first right root of clustered attractors with period 2 and experimental microarray data in GSE14926 time series shows completely different behavior in in-vitro and simulated gene expressions.

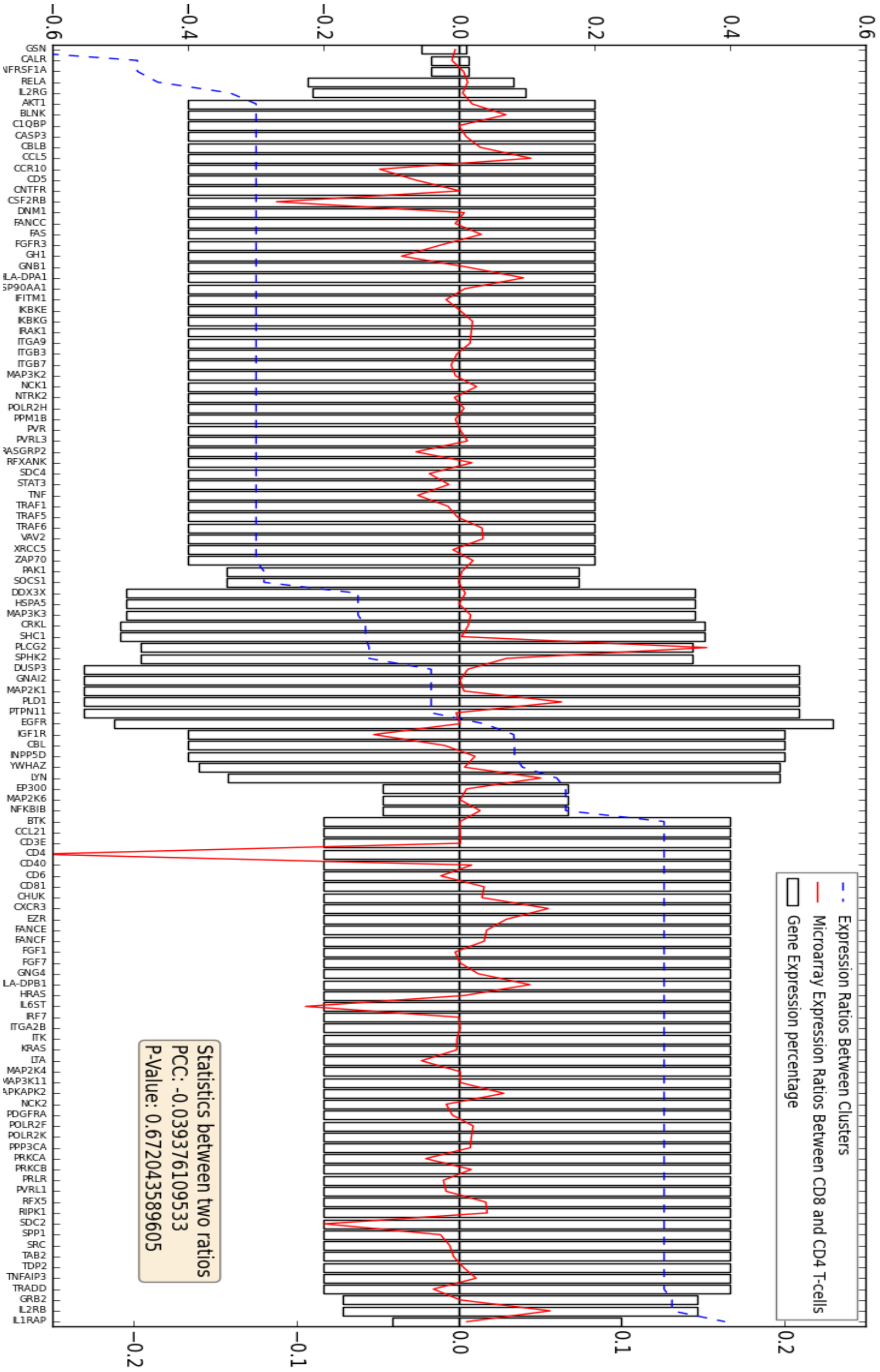


Figure 4.17. Comparison between simulated expression ratio of genes in two groups of the main root of clustered attractors with period 2 and experimental microarray data in GSE16130 time series shows completely different behavior in in-vitro and simulated gene expressions.

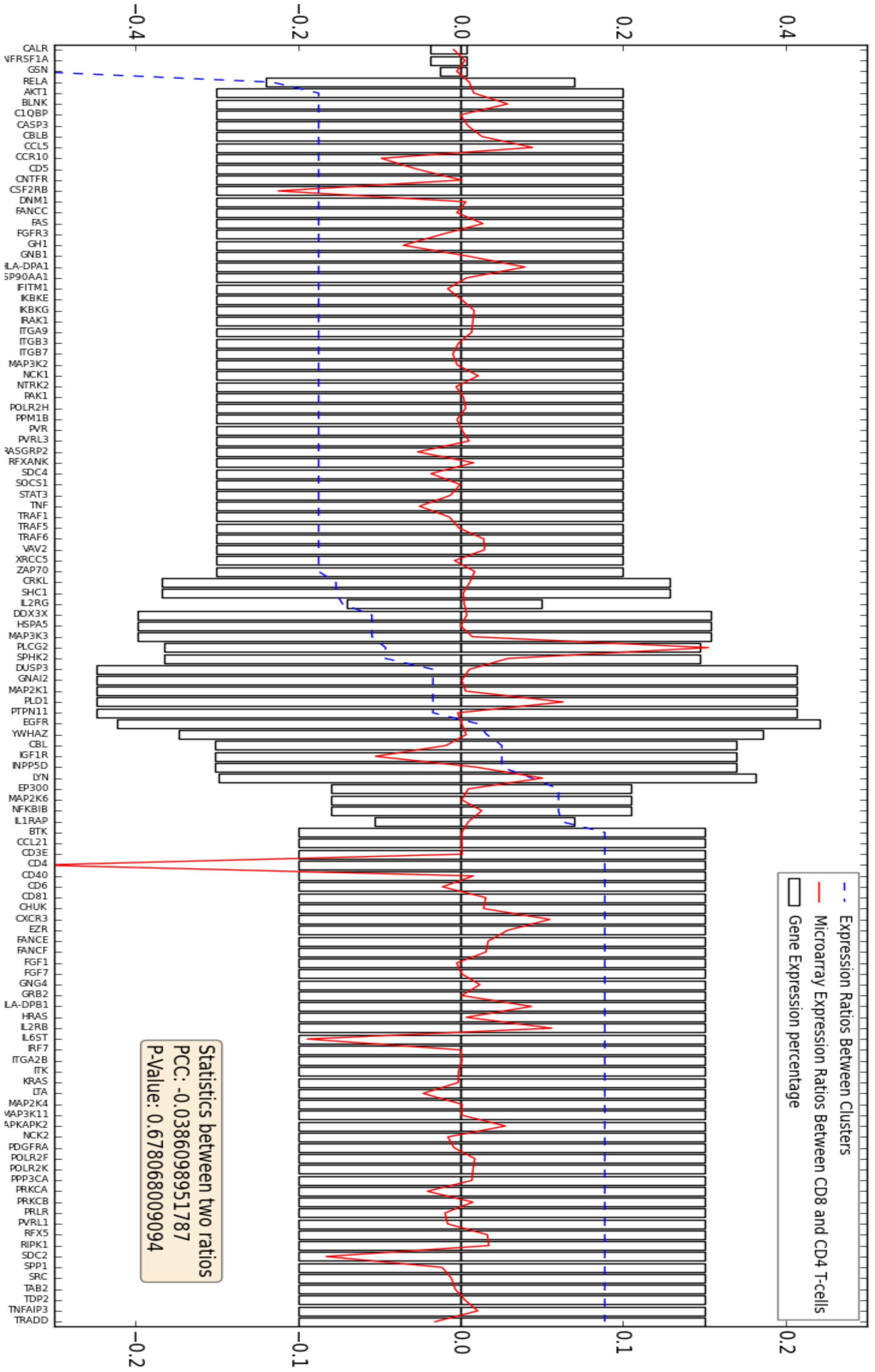


Figure 4.18. Comparison between simulated expression ratio of genes in two groups of the main root of clustered attractors with period 4 and experimental microarray data in GSE16130 time series shows completely different behavior in in-vitro and simulated gene expressions.

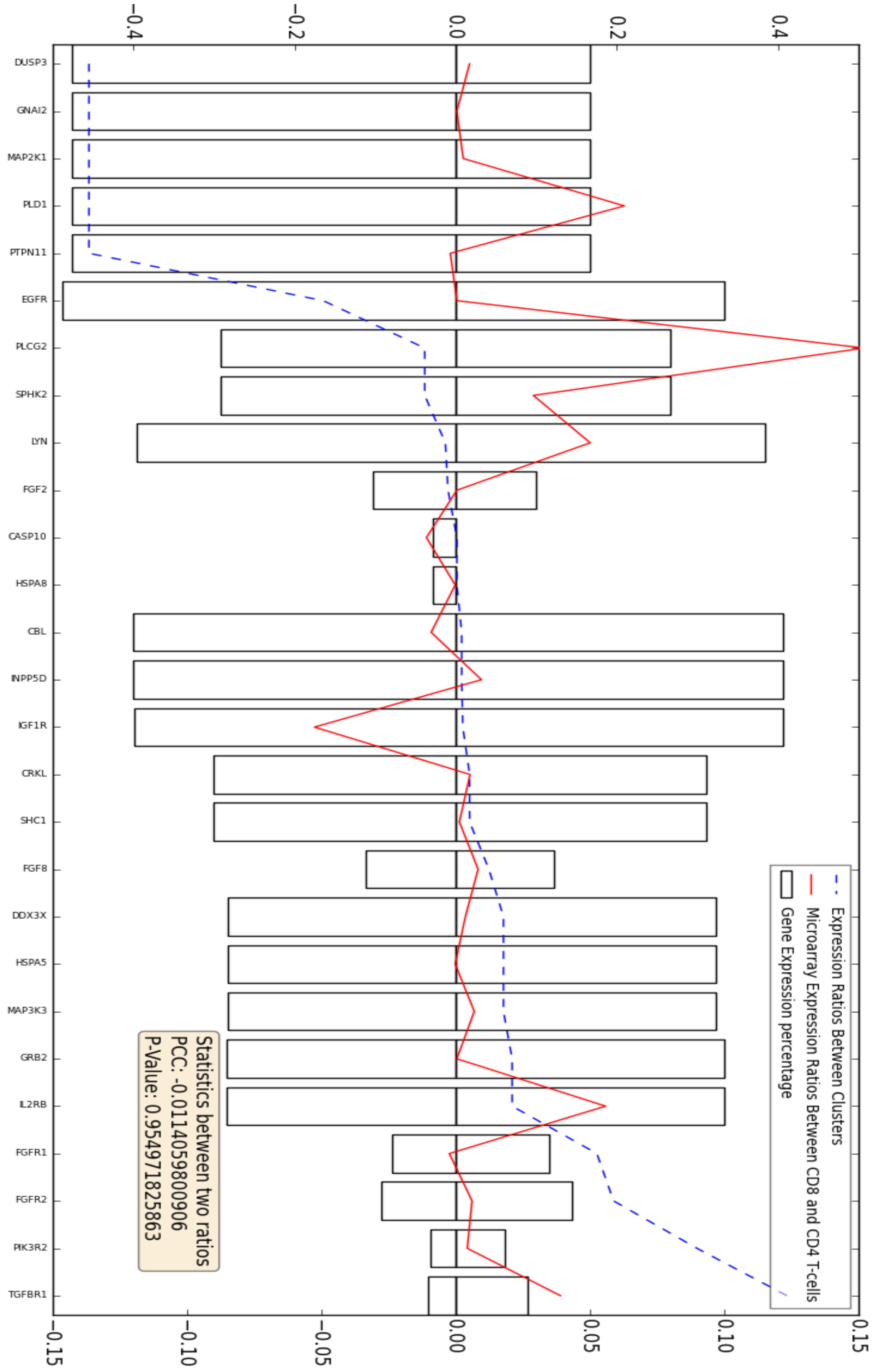


Figure 4.19. Comparison between simulated expression ratio of genes in two groups of the first right root of clustered attractors with period 2 and experimental microarray data in GSE16130 time series shows completely different behavior in in-vitro and simulated gene expressions.

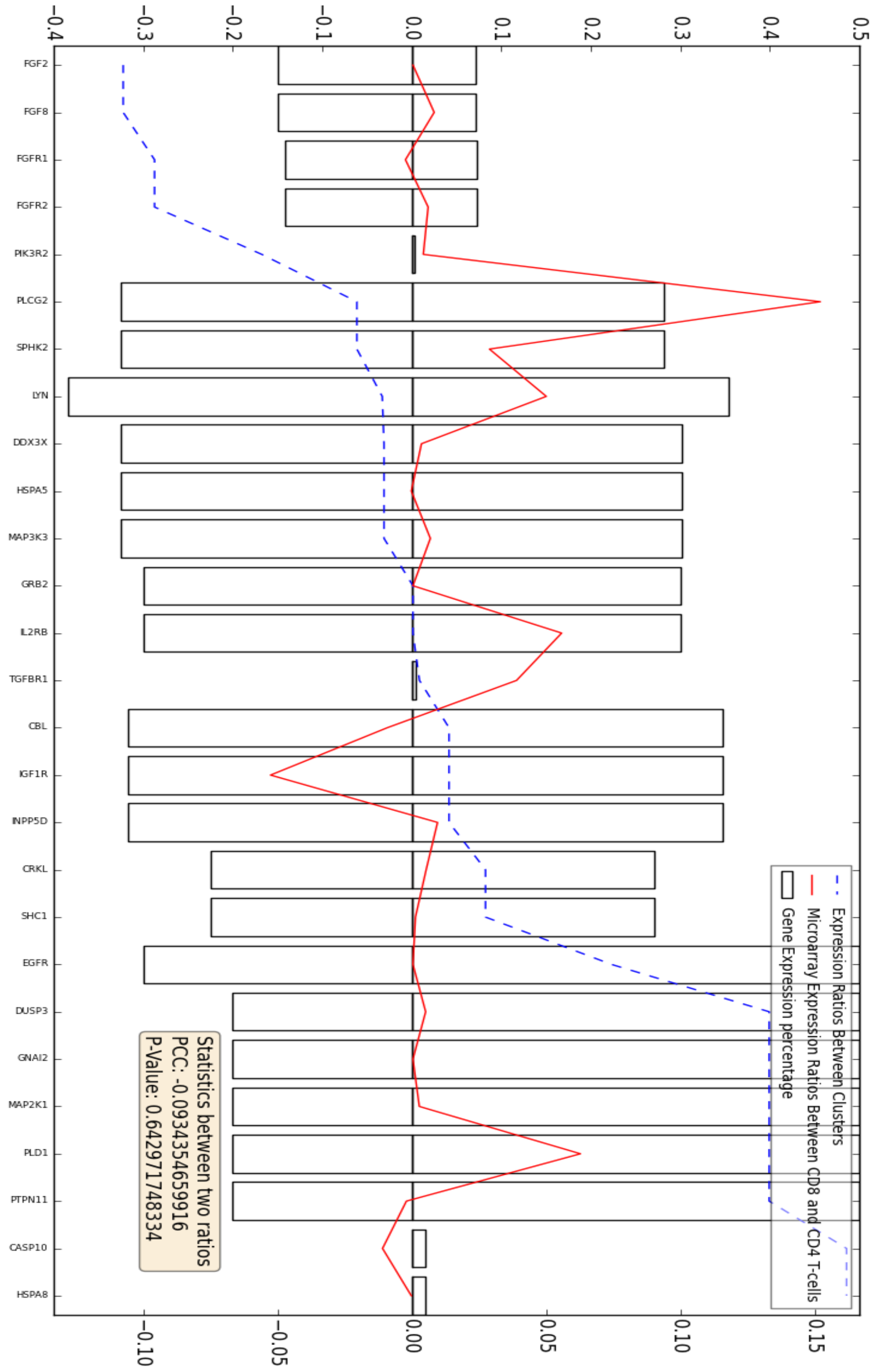


Figure 4.20. Comparison between simulated expression ratio of genes in two groups of the first right root of clustered attractors with period 2 and experimental microarray data in GSE16130 time series shows completely different behavior in in-vitro and simulated gene expressions.

Since the actual experiments are different from the simulations, a system was developed to exhaustively find information for each gene in the set (using data mining on articles and GO terms as supplementary data in which whole Homo sapiens genome was selected as the background and all the differentially expressed genes were chosen as the target). Table 4.7 shows the genes (in CTN) which are significant in T-cell differentiation (complete table is in appendix 2 and contains genes which are involved in more general differentiation processes). The result shows that CD4+ cell related genes are exactly grouped in under main bottom branches of the UPGMA trees (figure 4.7 and 4.8). But regardless of this group, most of the genes involved in differentiation processes are almost evenly spread over different clusters and sub-clusters (figures 4.9 and 4.10) and those with related (or same) functions are clustered in different groups. Also, some of the genes such as TP53 are not presented in gene expression ratio experiments due to the fact that they were not differentially expressed in different clusters (Or they are involved T-cell differentiation and not T-cell sub-types differentiation).

Table 4.7. Genes related to T-cell differentiation.

ENTREZ ID	Gene Name	Function	P-value
916	CD3E	T cell differentiation in the thymus	4.79E-006
7157	TP53	T cell differentiation in the thymus	4.79E-006
1499	CTNNB1	T cell differentiation in the thymus	4.79E-006
920	CD4	lymphocyte differentiation	4.52E-008
3932	LCK	lymphocyte differentiation	4.52E-008
5295	PIK3R1	lymphocyte differentiation	4.52E-008
3635	INPP5D	regulation of lymphocyte differentiation	2.79E-005
3575	IL7R	regulation of T cell differentiation in the thymus	0.0993
3561	IL2RG	regulation of alpha-beta T cell differentiation/regulation of T cell differentiation	0.0231/0.0993
6850	SYK	regulation of T cell differentiation in the thymus/positive regulation of gamma-delta T cell differentiation	0.0231/0.0013
7535	ZAP70	regulation of alpha-beta T cell differentiation	0.0231
5788	PTPRC	positive regulation of gamma-delta T cell differentiation	0.0013
6776	STAT5A	positive regulation of gamma-delta T cell differentiation	0.0013

5. Discussion

To be able to detect the fate of stem cells it is critical to analyze the gene expression pattern in each cell type. This is because different expression patterns are normally the indications of different cell types. In-vitro identification of cell types is a laborious task. This is because for performing different experiments, sufficient data for each cell line are needed and, also several methods such as transplantation experiments and molecular manipulation techniques, for cell fate determination [101][102][103]. The most important factors involved in cell differentiation are the interactions within and between the cells. Interactions between different cells determine how a cell will differentiate. Inside the cell, the gene regulatory network produces different patterns of gene expression which are specific to different cell types or different developmental processes. The patterns indicate the function of different cell types. The gene regulatory network normally acts on two types of genes, called housekeeping genes and cell specific genes. Cell specific genes are the genes which are involved in cell differentiation and are specific to each cell type. Immunological cell types (T and B cells) are the cells which are critical in the attack against pathogens. Finding the genes which are involved in T and B cells differentiations can be useful in several biological problems. This is because these cell types, especially T-cells, have several sub-types and each of them has different functions. So a disorder in the differentiation process can completely knockout one function of the immune system and cause different diseases.

Several methods have been proposed in order to separate the cell specific genes from the housekeeping genes and also detect which genes are mostly involved in cell differentiation by simulation of genetic regulatory networks. Boolean networks are the most widely used method for simulation of the genetic regulatory network. Kauffman [38] proposed that randomly combined elements in a gene regulatory network cause different gene expression patterns. Attractors resulted from a Kauffman network are assumed to be related to cell types and developmental processes. Several studies have approved Kauffman's findings, but many studies have also proposed that the RBN network completely ignores stochasticity of the genetic regulatory network and proposed other Boolean networks which take into account a probability of cell stochastic behavior.

Because gene interaction networks often have a very large magnitude, simulation of their dynamics is often unfeasible with greedy approaches. Heuristics methods are used to reduce the state space of such networks, but because there are many types of network (often with different definitions) in the human cells, these methods are not able to be applied on all the possible networks. So for handling this limitation, network decomposition methods are applied on large networks in order to find the most important sub-networks. The study done by [29] reduces the huge network of gene interactions in T-cell to the smaller scale, but this network is still too large to be simulated by RBNs. Heuristics methods are also not applicable on this network. This limitation is due to differences between CTN features and normal Boolean networks:

- The network is defined by experimental data and not randomly generated components.
- There is only one function (with different parameters) in CTN network. This function is not randomly assigned to each gene.
- The function used to update the network is not a pure Boolean function and the states of genes are affected by the correlation coefficient values of the neighboring genes with the current gene.
- The function is not deterministic.

To solve these problems, this study proposed a customized version of RBN by which the

CTN network can be simulated, and attractors can be found. This approach not only takes into account the deterministic nature of gene regulation but the stochasticity of gene interactions. Using this approach, not only the random Boolean network approach is performed but also included are the advantages of RBN's derivatives such as asynchronous RBB. The main advantage of this system is its flexibility of being applicable on other regulatory networks such as B-cell network. This is because by changing the value of parameter, one can easily switch between different simulation models which are suitable for distinct networks. The system is also significantly faster than RBNs. This is because the system explores only one initial state which is experimentally defined. The other benefit of this approach is the applicability of this method on networks with thousands of genes. The reason is that even huge networks are converted to small graphs (depending on parameters) and so graph theory algorithms can be easily applied on the networks.

The main goal of this thesis is to identify genes related to T-cell differentiation. To fulfill this requirement, attractors detection was performed on several networks, but the resulting attractors did not show reasonable significance in T-cell differentiation related functions. Attractors clustering and gene expression simulation were performed. The resulted clusters were very interesting because some of the genes related to CD4+ T-cells were clustered in only one cluster of attractors (regarding their expression profiles). The main problem is that other genes involved in T-cell differentiation were almost evenly separated between attractors. To find out if the simulated expression values are the same as the experimental data, the ratio of simulated gene expressions was plotted against the real data but no reasonable relationships were observed between the simulated and experimental ratios.

These failures do not mean that the system works in not a sensible manner. The result of clustering clearly shows that the system generated reasonable number of clusters which can be related to T-cell sub-types. For example, the cluster related to CD4+ cells was divided into several sub-clusters which can be regarded as CD4+ sub-types. So the failures may be related to the fact that for finding genes related to T-cell differentiation, this study only used data in publicly available databases. Lack of sufficient data and lab experiments may cause the system to fail to find interesting genes.

The proposed method has three major limitations. First, in order to verify the system and find interesting genes, sufficient biological data and published articles are needed. If the amount of information is not sufficient, the analysis of attractors is not possible. The second problem is that there is not a direct and exact way to set the parameters. Choosing parameters is highly dependent on type of network, computational power, and expected number of cycles in network. Having a large random component may cause the system to behave completely randomly and having varying numbers of iteration will generate completely different numbers of attractors. So a good balance between the two parameters is essential for the system to simulate the network in a reasonable way. The last problem is that using initial values, only a small (but most interesting) subset of states and corresponding attractors can be extracted.

There are several ways to expand and verify this system. The most effective method to verify the findings is to measure different types of T-cell in actual (in-vitro) experiments. If the result of the laboratory experiments shows that the simulated gene expression is the same as the actual experiments (with only the CTN genes) it a strong verification of the system. In another method, the clustered genes in the simulation can be verified in the lab by performing microarray data analysis and observing if the same genes are clustered also in actual experiments. Another good approach to compare the behavior of simulation with actual experiments is performing gene knockout (or finding experiments where gene knockout was previously performed on any genes involved in CTN). This method can be quite expensive because many experiments are needed to verify different simulations (with different parameters). If the changes in two experiments are the same, a conclusion can be drawn that the simulation works correctly, with regard to in-vitro experiments. Since performing microarray experiments (and gene knockout) are quite expensive, performing more data mining and using more resources are recommended for verifying the system or reducing the number of genes which are required to be evaluated experimentally. Another potential approach is to combine the regulatory model to a metabolic network (involved in T-cells) and perform gene knockout (or without gene knockout) to find out if the result of metabolic network simulation has common features with in-vivo experiments (or if interesting changes in the metabolic network occur). If the mentioned method cannot be used to verify the system and find differentiation related genes, the nature of simulation can be modified by two methods. First, one can use different parameters and try to find a good balance between power of random component of the presented updating function and number of iterations. By finding new parameters, the result of the simulation may be dramatically changed and the outcome might be more informative. Since finding parameters can be quite challenging, a possible way to run the simulation is to remove the random component and use slightly customized RBN (only without randomly assigned functions and links) and let the dynamics flow (the same as RBN process) until a reasonable number of attractors is found. This technical solution will change the nature of the problem because the random component is supposed to be a compensation for microarray analysis and gene expression profiling errors (and stochastic environment of the cells). Since gene expressions directly define how neighboring genes can affect each other, removing the random component means that the extracted interaction between genes is 100% reliable which is not true in real life applications. Another approach is performing further network decomposition and repeating the whole experiment for each resulting sub-network and coupling the results. Since decomposition methods often greatly shrink the network, normal RBN experiments can be performed on the smaller network. This means that initial states are generated randomly and not by using available microarray data. Finally, the most challenging approach is to remove the correlation values from CTN network and treat the network as a pure RBN with the only difference being that the network links were previously defined. This approach can be applied by considering all possible values of a gene which can be generated by its neighbors and produce all possible Boolean functions for each gene. These functions are assigned to (randomly, in case of having many functions) the corresponding gene or are shuffled between all the genes in CTN. Having Boolean functions, available methods can be applied to find the attractors and extract information from them. This approach also has a limitation. There are several possible values that can be produced for the random component. Therefore, one has to define one (or several) cut off points for P parameter, in the updating function, which may cause the neighbors to have the negative or positive effect on the current genes. Since the state of the neighboring genes is also indicated by its neighbors, two possible states (one and zero) for the genes have to be considered. The result of this approach may be huge number of possible functions for each gene. If the functions are randomly assigned to the genes, not only two random parameters are considered (one for P and one for probability of distributing the functions), but also one gene may be always off or on regardless of actual expressions of its neighbors. There are many other approaches which can be used to change the simulation system or the network and find interesting genes (e.g CA or Bayesian network) but because of the unique attributes of this network, most of the solutions need to be highly modified to become applicable for CTN.

As mentioned, this new method was tested on a specific T-cell gene regulatory network, but since the random Boolean networks are a solid framework for simulation of any biological network and the modifications made to RBNs are quite flexible, this approach can be further utilized as a very general type of RBN which is not only a system for simulating any biological network with many attributes, but it is also a reasonable way to add experimentally verified biological data to the simulation. So this framework has the advantages of both the gene regulatory network simulation and experimentally verified regulatory networks. Considering these features and performed experiments, a good potential target for the next simulation is B-cells. The benefits to B-cells are that they are very well defined and also have fewer sub-types compared with T-cells. Thus it is much more straightforward to extract information using the presented system.

6. Conclusion

This study shows how to simulate a gene regulatory network where the effects of genes on one another are associated with the correlation coefficient values. A new system was proposed in order to simulate co-regulatory networks in T-cells using a unique updating function and a customized version of a random Boolean network. Initial values were defined in order to reduce space and time complexity of the algorithm. Two parameters related to the updating function were set in which there is a reasonable balance between randomness and effect of genes on each other in the network. The simulations were performed for a statistically significant number of runs, the attractors were identified, and most frequent cycles were analyzed in order to find the genes which are related to T-cell differentiation processes. But no significant terms associated with T-cell differentiation were detected. Since this might be because of loss of information (because only most frequent attractors were used), in order to find a solution to reduce data loss, attractors were clustered and analyzed to find simulated gene expression. The genes were grouped by their patterns of expression, and each group was analyzed by GO and data mining techniques, but none of the groups were significant in all the T-cell differentiation terms. To treat the gene expression more naturally, the simulated gene expression ratios were compared to the actual gene expression ratios in different microarray samples related to T-cell sub-types. The result of most of the experiments shows that the proposed system for simulating of the central T-cell network was not enable to correctly separate the genes involved in T-cell differentiation. But the result of clustering was very promising. Because, one cluster only contains CD4+ T-cells related genes, one can draw a conclusion that the system correctly classified and detected the genes related to CD4+ cells. As the results indicated, the simulator system has good potential to be applicable on a variety of regulatory networks but for producing more sensible results, one has to perform more experiments and verifications with more data.

Reference

1. Kikkawa, E., M. Yamashita, M. Kimura, M. Omori, K. Sugaya, C. Shimizu, T. Katsumoto, M. Ikekita, M. Taniguchi, T. Nakayama. (2002). Th1/Th2 cell differentiation of developing CD4 single-positive thymocytes. Int. Immunol. 14:943-951.
2. Stassen M , Schmitt E , Bopp T .(2012). From interleukin-9 to T helper 9 cells . Ann N Y Acad Sci .1247:56–68

3. Jabeen R, Kaplan MH. (2012). The symphony of the ninth: the development and function of Th9 cells. Curr Opin Immunol.24(3):303-7. Epub 2012 Feb 22.

4. Veldhoen M, Uyttenhove C, van Snick J, Helmby H, Westendorf A, Buer J et al.(2008). Transforming growth factor-beta 'reprograms' the differentiation of T helper 2 cells and promotes an interleukin 9-producing subset. Nat Immunol; 9: 1341–1346.

5. Cortelazzi C, Campanini N, Ricci R, De Panfilis G.(2012). Inflammed skin harbours Th9 cells. Acta Derm Venereol. doi: 10.2340/00015555-1408.

6. Korn T, Bettelli E, Oukka M, Kuchroo VK. (2009). IL-17 and Th17 Cells.Annu Rev Immunol.27:485-517.

7. Acosta-Rodriguez EV, Rivino L, Geginat J, Jarrossay D, Gattorno M, Lanzavecchia A, Sallusto F, Napolitani G.(2007). Surface phenotype and antigenic specificity of human interleukin 17-producing T helper memory cells. Nat Immunol.8(6):639-46.

8.Kurts C.(2008) .Th17 cells: a third subset of CD4+ T effector cells involved in organ-specific autoimmunity.Nephrol Dial Transplant.23(3):816-9. Epub 2007 Nov 28.

9.Eyerich S, Eyerich K, Pennino D, Carbone T, Nasorri F, Pallotta S, Cianfarani F, Odorisio T, Traidl-Hoffmann C, Behrendt H, Durham SR, Schmidt-Weber CB, Cavani A. (2009). Th22 cells represent a distinct human T cell subset involved in epidermal immunity and remodeling. J Clin Invest.119(12):3573-85. doi: 10.1172/JCI40202.

10. Zhang N, Pan HF, Ye DQ.(2011). Th22 in inflammatory and autoimmune disease: prospects for therapeutic intervention. Mol Cell Biochem.353(1-2):41-6. Epub 2011 Mar 8.

11. Crotty S.(2011). Follicular helper CD4 T cells (TFH). Annu Rev Immunol.29:621-63.

12. Vinuesa CG, Tangye SG, Moser B, Mackay CR.(2005). Follicular B helper T cells in antibody responses and autoimmunity. Nat Rev Immunol.5(11):853-65.

13. Alberts B, Johnson A, Lewis J, et al. (2002). Molecular Biology of the Cell.4th edition. New York: Garland Science.

14. Matteo Iannacone, Giovanni Sitia & Luca G Guidotti.(2006). Pathogenetic and antiviral immune responses against hepatitis B virus. Future Virology. Vol. 1, No. 2, Pages 189-196 , DOI 10.2217/17460794.1.2.189 (doi:10.2217/17460794.1.2.189)

15. Neumann H, Medana IM, Bauer J, Lassmann H.(2002).Cytotoxic T lymphocytes in autoimmune and degenerative CNS diseases. Trends Neurosci. 25(6):313-9.

16. Shevach EM, DiPaolo RA, Andersson J, Zhao DM, Stephens GL, Thornton AM.(2006).The lifestyle of naturally occurring CD4+ CD25+ Foxp3+ regulatory T cells.Immunol Rev.212:60-73.

17. Carrier Y, Yuan J, Kuchroo VK, Weiner HL.Th3 cells in peripheral tolerance. I. (2007) . Induction of Foxp3-positive regulatory T cells by Th3 cells derived from TGF-beta T cell-transgenic mice.J Immunol.178(1):179-85.

18. Vignali DA, Collison LW, Workman CJ. (2008) .How regulatory T cells work. Nat Rev Immunol. 8(7):523-32.

19. Sakaguchi S, Yamaguchi T, Nomura T, Ono M. (2008). Regulatory T cells and immune tolerance. Cell. 133(5):775-87.

20. Belkaid Y, Rouse BT. (2005). Natural regulatory T cells in infectious disease. Nat Immunol. 6(4):353-60.

21. Holaday BJ, Pompeu MM, Jeronimo S, Texeira MJ, Sousa Ade A, Vasconcelos AW, Pearson RD, Abrams JS, Locksley RM. (1993). Potential role for interleukin-10 in the immunosuppression associated with kala azar. J Clin Invest. 92(6):2626-32.

22. Ken-ichiro Seino and Masaru Taniguchi.(2005). NKT Cells: A Regulator in Both Innate and Acquired Immunity. Curr. Med. Chem. – Anti-Inflammatory & Anti-Allergy Agents, 4, 59-64 59

23. Godfrey DI, Hammond KJ, Poulton LD, Smyth MJ, Baxter AG. (2000) .NKT cells: facts, functions and fallacies. Immunol Today.21(11):573-83.

24. Terabe M, Berzofsky JA. (2008). The role of NKT cells in tumor immunity.Adv Cancer Res.101:277-348.

25. V Kalia, S Sarkar, TS Gourley, BT Rouse, R Ahmed.(2006). Differentiation of memory B and T cells. Current opinion in immunology 18 (3), 255-264

26. Brigitta Stockinger, Christine Bourgeois, George Kassiotis. (2006) . CD4+ memory T cells: functional differentiation and homeostasis. DOI: 10.1111/j.0105-2896.2006.00381.x

27. Kaech SM, Wherry EJ, Ahmed R.(2002).Effector and memory T-cell differentiation: implications for vaccine development. Nat Rev Immunol.2(4):251-62.

28. DeLeon SBT, EH Davidson; Gene regulation: Gene control network in development. Annual Review of Biophysics and Biomolecular Structure36:191-212, 2007 Ben-Tabou De-Leon, S.; Davidson, E. (2007). "Gene regulation: gene control network in development". Annual review of biophysics and biomolecular structure 36 (1):

191.doi:10.1146/annurev.biophys.35.040405.102002. PMID 17291181.

29. Gabriel Teku, Csaba Ortutay, Mauno Vihinen.Immunome Protein Interaction Decomposition Reveals T Cell Function Subnetworks.

30. Chen J, Yuan B. (2006) Detecting functional modules in the yeast protein-protein interaction network. Bioinformatics 22(18): 2283-2290.

31. Padiadpu J, Vashisht R, Chandra N. (2010) Protein-protein interaction networks suggest different
targets have different propensities for triggering drug resistance. Syst Synth Biol 4(4): 311-322.

32. Rui-Sheng W, Reka A. (2011) Elementary signaling modes predict the essentiality of signal transduction network components. BMC Syst Biol 5(44).

33. Luo F, Yang Y, Chen CF, Chang R, Zhou J, et al. (2007) Modular organization of protein interaction networks. Bioinformatics 23(2): 207-214.

34. Ortutay C, Vihinen M. (2009) Immunome knowledge base (IKB): An integrated service for immunome research. BMC Immunol 10: 3.

35. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M. (2012) KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res 40(1): D109-14.

36. Razick S, Magklaras G, Donaldson IM. (2008) iRefIndex: A consolidated protein interaction database with provenance. BMC Bioinformatics 9: 405.

37. Ilachinski, Andrew (2001). Cellular Automata: A Discrete Universe. World Scientific. ISBN 9789812381835.

38. Kauffman, S. A. (1969). Metabolic stability and epigenesist in randomly constructed genetic

39. Kauffman, S. A. (2000). Investigations. Oxford University Press

40. Gershenson, C. (2004). Introduction to Random Boolean Networks In Bedau, M., P. Husbands, T. Hutton, S. Kumar, and H. Suzuki (eds.) Workshop and Tutorial
Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX). pp. 160-173.

41.Huang S, Ingber DE.(2000). Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. Exp
Cell Res. 261(1):91-103.

42. Huang S.(1999).Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery.J Mol Med (Berl).77
(6):469-80.

43. Devloo V, Hansen P, Labbé M.(2003). Identification of all steady states in large networks by logical analysis. Bull Math Biol.65(6):1025-51.

44. M. Aldana, S. Coppersmith, and L. P. Kadanoff.(2003).Boolean dynamics with random coupling. in Perspectives and Problems in Nonlinear Science,
Springer Applied Mathematical Sciences Series. Springer.

45. Akutsu T, Kuhara S, Maruyama O, Miyano S.(1998). A System for Identifying Genetic Networks from Gene Expression Patterns Produced by Gene Disruptions and
Overexpressions. Genome Inform Ser Workshop Genome Inform.9:151-160.

46.Shu-Qin Zhang , Morihiro Hayashida , Tatsuya Akutsu , Wai-Ki Ching , Michael K. (2007) . Ng Algorithms for finding small attractors in boolean networks, EURASIP
Journal on Bioinformatics and Systems Biology, p.4-4, [doi>10.1155/2007/20180]

47. Elena Dubrova , Maxim Teslenko.(2011). A SAT-Based Algorithm for Finding Attractors in Synchronous Boolean Networks. IEEE/ACM Transactions on Computational Biology
and Bioinformatics (TCBB), v.8 n.5, p.1393-1399.

48. A. Biere, A. Cimatti, E. Clarke, M. Fujita, and Y. Zhu.(1999). Symbolic model checking using sat procedures instead of bdds. Design Automation Conference. Proceedings. 36th,
pp. 317–320.

49. Tatsuya Akutsu, Morihiro Hayashida, and Takeyuki Tamura.(2008). Algorithms for Inference, Analysis and Control of Boolean Networks. AB '08 Proceedings of the 3rd
international conference on Algebraic Biology
Pages 1 - 15

50. Irons, D.J.(2006). Improving the efficiency of attractor cycle identification in Boolean network models. Physica D, 217: 7-21

51. Harvey, I. and T. Bossomaier.(1997). Time Out of Joint: Attractors in Asynchronous Random Boolean Networks. In Proceedings of the Fourth European Conference on
Artificial Life (ECAL97), P. Husbands and I. Harvey (Eds.). pp. 67-75.

52. Gershenson C.(2002) .Classification of random boolean networks. In: Standish R.K., et al., editors. Proceedings of the 8th International Conference on Artificial
Life. MIT Press; p. 1-8

53. Wuensche, A.(1998). Discrete Dynamical Networks and their Attractor Basins. Complexity International 6.

54. Lemke, N., Mombach, J. C. M., and Bodmann, B. E. J.(2002). A numerical investigation of adaptation in populations of random Boolean networks.Physica A,301(1-
4):589–600.

55. I. Shmulevich, E. R. Dougherty, W. Zhang.(2002). From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, Proc. IEEE 90 (11) 1778–
1792

56. Adomas A, Heller G, Olson A, Osborne J, Karlsson M, Nahalkova J, Van Zyl L, Sederoff R, Stenlid J, Finlay R, Asiegbu FO.(2008).Comparative analysis of transcript
abundance in Pinus sylvestris after challenge with a saprotrophic, pathogenic or mutualistic fungus. Tree Physiol. 28 (6): 885–897. PMID 18381269.

57. Pollack JR, Perou CM, Alizadeh AA, Eisen MB, Pergamenschikov A, Williams CF, Jeffrey SS, Botstein D, Brown PO.(1999). Genome-wide analysis of DNA copy-number
changes using cDNA microarrays. Nat Genet 23 (1): 41–46. doi:10.1038/14385.

58. Yang YH, Dudoit SD, Luu P.(2001). Speed TP: Normalization for cDNA Microarray Data.In SPIE BioE.

59. Park T, Yi S-G, Kang S-H, Lee S-Y, Lee Y-S, Simon R.(2003). Evaluation of normalization methods for microarray data. BMC Bioinformatics; 4: 33.

60. Cleveland, W.S.(1979). Robust locally weighted regression and smoothing scatterplots. J. Amer. Stat. Assoc. 74, 829–836.

61. Myers, Jerome L.; Well, Arnold D.(2003). Research Design and Statistical Analysis (2nd ed.), Lawrence Erlbaum, pp. 508, ISBN 0-8058-4037-0.

62. Ihara, Shunsuke.(1993).Information theory for continuous systems. World Scientific. p. 2. ISBN 978-981-02-0985-8.

63. Hastie, T., Tibshirani, R. and Friedman, J.(2001). The Elements of Statistical
Learning: Data Mining, Inference, and Prediction. Springer, New York.

64. CSC. DNA Microarray Data Analysis, second edition.

65. Y. Benjamini and Y. Hochberg.(1995). Controlling the false discovery rate: a
practical and powerfull approach to multiple testing, J. R. Statist. Soc.
B 57(1), pp. 289-300.

66. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, et al.(2004). Bioconductor: Open software development for computational biology and bioinformatics. Genome Biol 5(10): R80.

67. Montaner, D., Tárraga, J., Huerta-Cepas, J., Burguet, J., Vaquerizas, J.M., Conde, L., Minguez, P., Vera, J., Mukherjee, S., Valls, J., Pujana, M.A.G., Alloza, E.,
Herrero, J., Al-Shahrour, F., and Dopazo, J. (2006) Next station in microarray data analysis: GEPAS. Nucleic Acids Research, 34 (Web Server issue): W486-W491.

68. Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, Sturn A, Snuffin M, Rezantsev A, Popov D, Ryltsov A,
Kostukovich E, Borisovsky I, Liu Z, Vinsavich A, Trush V, Quackenbush J.(2003). TM4: a free,

open-source system for microarray data management and analysis.
Biotechniques. 34(2):374-8.

69. Barrett, T., Troup, D.B., Wilhite, S.E., Ledoux, P., Evangelista, C., Kim, I.F., Tomashevsky, M., Marshall, K.A., Phillippy, K.H., Sherman, P.M., et al.(2011).
NCBI GEO: archive for functional genomics data sets–10 years on. Nucleic Acids Res. 39 (Database issue), D1005–D1010.

70. Parkinson et al. (2010) ArrayExpress update—an archive of microarray and high-throughput sequencing-based functional genomics experiments. Nucl. Acids Res., doi:
10.1093/nar/gkq1040. Pubmed ID 21071405.

71. The Gene Ontology Consortium: Gene Ontology: tool for the uni_cation of biology.
Nature Genetics Volume 25 May 2000

72.Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. (2005) .Gene set enrichment
analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A;102(43):15545-50.

73. Adrian Alexa and Jorg Rahnenfuhrer (2010). topGO: topGO: Enrichment
analysis for Gene Ontology. R package version 2.6.0.

74. Alexa, A., Rahnenfuhrer, J., and Lengauer, T. (2006). Improved scoring of functional groups from gene expression data by decorrelating go graph structure.
Bioinformatics (Oxford, England), 22:1600{1607. 10.1093/bioinformatics/btl140.

75. Huang DW, Sherman BT, Lempicki RA.(2009). Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. Nature Protoc.4(1):44-57.

76. Huang DW, Sherman BT, Lempicki RA.(2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids
Res.37(1):1-13.

77. J. Reimand, M. Kull, H. Peterson, J. Hansen, J.(2007). Vilo: g:Profiler -- a web-based toolset for functional profiling of gene lists from large-scale experiments
NAR 35 W193-W200

78. J. Reimand, T. Arak, J.(2011). Vilo: g:Profiler -- a web server for functional interpretation of gene lists Nucleic Acids Research 2011; doi: 10.1093/nar/gkr378

79. J. B. MacQueen. (1967). Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical
Statistics and Probability, Berkeley, University of California Press, 1:281-297

80. Michener, C.D., Sokal, R.R. (1957). A quantitative approach to a problem of classification. Evolution, 11:490–499.

81. Saitou N, Nei M.(1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution, volume 4, issue 4, pp.
406-425.

82. Harary, Frank; Palmer, Edgar M. (1973). Graphical Enumeration. Academic Press . ISBN 0-12-324245-2.

83. R. TARJAN.(1973). Enumeration of the elementary circuits of a directed graph, this Journal. pp. 211-216.

84. H. WENBLAX.(1972). A new search algorithm for finding the simple cycles of a finite directed graph.
Assoc. Comput. Mach., pp. 43-56.

85. J. C. TIERNAN.(1970). An efficient search algorithm to find the elementary circuits of a graph. 13. Comm. ACM, pp. 722-726.

86. D.B. Johnson.(1975). Finding all the elementary circuits of a directed graph. SIAM J. Comput., v. 4, pp. 77-84.

87. Cha S. (2007). Comprehensive survey on distance/similarity measures between probability density functions. Int J Math Models Meth Appl Sci. 1:300–307.

89. Barenco M, Tomescu D, Brewer D, Callard R, Stark J, et al. (2006) Ranked prediction of p53 targets using hidden variable dynamic modeling. Genome Biol 7(3): R25.

90. Lund R, Aittokallio T, Nevalainen O, Lahesmaa R. (2003) Identification of novel genes regulated
by IL-12, IL-4, or TGF- during the early polarization of CD4+ lymphocytes. J Immunol 171(10): 5328-5336.

91. Classen S, Zander T, Eggle D, Chemnitz JM, Brors B, et al. (2007). Human resting CD4+ T cells
are constitutively inhibited by TGF under steady-state conditions. J Immunol 178(11): 6931-6940.

92. Emonts M. (2008) Polymorphisms in immune response genes infectious diseases and autoimmune
diseases. (Doctoral Dissertation). Erasmus University of Rotterdam, Rotterdam.

93. Ravoet M, Sibille C, Gu C, Libin M, Haibe-Kains B, et al. (2009) Molecular profiling of CD3-CD4+ T cells from patients with the lymphocytic variant of hypereosinophilic syndrome reveals targeting of growth control pathways. Blood 114(14): 2969-2983.

94. Stockis J, Fink W, Francois V, Connerotte T, de Smet C, et al. (2009) Comparison of stable human
treg and th clones by transcriptional profiling. Eur J Immunol 39(3): 869-882.

95. Richter GH, Mollweide A, Hanewinkel K, Zobywalski C, Burdach S. (2009) CD25 blockade

protects T cells from activation-induced cell death (AICD) via maintenance of TOSO expression. Scand J Immunol 70(3): 206-215.

96. Prots I, Skapenko A, Lipsky PE, Schulze-Koops H. (2011) Analysis of the transcriptional program
of developing induced regulatory T cells. PLoS One 6(2): e16913.

97. Pont F, Familiades J, Dejean S, Fruchon S, Cendron D, et al. (2012) The gene expression profile of
phosphoantigen-specific human T lymphocytes is a blend of T-cell and NK-cell signatures Eur

98. Le Dieu R, Mitter R,Gribben JG (2009) Analysis of the impact of the method of cell selection on the gene expression profile of human CD4 and CD8 T cells

99. Crispín JC, Tsokos GC. Human TCR-alpha beta+ CD4- CD8- T cells can derive from CD8+ T cells and display an inflammatory effector phenotype. J Immunol 2009 Oct
1;183(7):4675-81. PMID: 19734235

100. John W. Ratcliff and David Metzener. July (1988) . Pattern Matching: The Gestalt Approach, Dr. Dobb's Journal, page 46.

101. Shen-Orr SS, Tibshirani R, Khatri P, Bodian DL, Staedtler F, Perry NM, Hastie T, Sarwal MM, Davis MM, Butte AJ. (2010). Cell-type-specific gene expression
differences in complex tissues. Nat. Methods;7:287-289.

102. Xi H, et al. (2007). Identification and characterization of cell type-Specific and ubiquitous chromatin regulatory structures in the human genome. PLoS
genetics.3:e136.

103. Artavanis-Tsakonas S, Rand MD, Lake RJ: Notch signaling: cell fate control and signal integration in development. Science 1999, 284:770-776.Artavanis-Tsakonas,
S.; Rand, M. D.; Lake, R. J. (1999). "Notch Signaling: Cell Fate Control and Signal Integration in Development". Science 284 (5415): 770–6. Bibcode1999Sci...284..770A.

Appendix 1

Tables A1.1 and A1.2 show the data related to the CTN network (figure A1.1). Figures A1.2 and A1.3 show difference between mean and median of microarray data (with different distribution) used for finding initial values.

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
60	ACTB	actin, beta
71	ACTG1	actin, gamma 1
81	ACTN4	actinin, alpha 4
387	RHOA	ras homolog gene family, member A
778	CACNA1F	calcium channel, voltage-dependent, L type, alpha 1F subunit
867	CBL	Cas-Br-M (murine) ecotropic retroviral transforming sequence
916	CD3E	CD3e molecule, epsilon (CD3-TCR complex)
919	CD247	CD247 molecule
920	CD4	CD4 molecule
924	CD7	CD7 molecule
947	CD34	CD34 molecule
1072	CFL1	cofilin 1 (non-muscle)
1399	CRKL	v-crk sarcoma virus CT10 oncogene homolog (avian)-like
1759	DNM1	dynamin 1
2033	EP300	E1A binding protein p300
2185	PTK2B	PTK2B protein tyrosine kinase 2 beta
2207	FCER1G	Fc fragment of IgE, high affinity I, receptor for; gamma polypeptide
2212	FCGR2A	Fc fragment of IgG, low affinity IIa, receptor (CD32)
2534	FYN	FYN oncogene related to SRC, FGR, YES
2688	GH1	growth hormone 1
2934	GSN	gelsolin
3055	HCK	hemopoietic cell kinase
3113	HLA-DPA1	major histocompatibility complex, class II, DP alpha 1
3115	HLA-DPB1	major histocompatibility complex, class II, DP beta 1
3383	ICAM1	intercellular adhesion molecule 1
3385	ICAM3	intercellular adhesion molecule 3
3480	IGF1R	insulin-like growth factor 1 receptor
3551	IKBKB	inhibitor of kappa light polypeptide gene enhancer in B-cells, kinase beta
3556	IL1RAP	interleukin 1 receptor accessory protein
3575	IL7R	interleukin 7 receptor
3635	INPP5D	inositol polyphosphate-5-phosphatase, 145kDa
3683	ITGAL	integrin, alpha L (antigen CD11A (p180), lymphocyte function-

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
		associated antigen 1; alpha polypeptide)
3689	ITGB2	integrin, beta 2 (complement component 3 receptor 3 and 4 subunit)
3716	JAK1	Janus kinase 1
3791	KDR	kinase insert domain receptor (a type III receptor tyrosine kinase)
3932	LCK	lymphocyte-specific protein tyrosine kinase
3937	LCP2	lymphocyte cytosolic protein 2 (SH2 domain containing leukocyte protein of 76kDa)
4261	CIITA	class II, major histocompatibility complex, transactivator
4478	MSN	moesin
4690	NCK1	NCK adaptor protein 1
4793	NFKBIB	nuclear factor of kappa light polypeptide gene enhancer in B-cells inhibitor, beta
5156	PDGFRA	platelet-derived growth factor receptor, alpha polypeptide
5159	PDGFRB	platelet-derived growth factor receptor, beta polypeptide
5175	PECAM1	platelet/endothelial cell adhesion molecule
5290	PIK3CA	phosphoinositide-3-kinase, catalytic, alpha polypeptide
5294	PIK3CG	phosphoinositide-3-kinase, catalytic, gamma polypeptide
5295	PIK3R1	phosphoinositide-3-kinase, regulatory subunit 1 (alpha)
5296	PIK3R2	phosphoinositide-3-kinase, regulatory subunit 2 (beta)
5494	PPM1A	protein phosphatase, Mg2+/Mn2+ dependent, 1A
5580	PRKCD	protein kinase C, delta
5588	PRKCQ	protein kinase C, theta
5590	PRKCZ	protein kinase C, zeta
5595	MAPK3	mitogen-activated protein kinase 3
5608	MAP2K6	mitogen-activated protein kinase kinase 6
5618	PRLR	prolactin receptor
5777	PTPN6	protein tyrosine phosphatase, non-receptor type 6
5788	PTPRC	protein tyrosine phosphatase, receptor type, C
5970	RELA	v-rel reticuloendotheliosis viral oncogene homolog A (avian)
6198	RPS6KB1	ribosomal protein S6 kinase, 70kDa, polypeptide 1
6464	SHC1	SHC (Src homology 2 domain containing) transforming protein 1
6714	SRC	v-src sarcoma (Schmidt-Ruppin A-2) viral oncogene homolog (avian)
6774	STAT3	signal transducer and activator of transcription 3 (acute-phase response factor)
6776	STAT5A	signal transducer and activator of transcription 5A
6850	SYK	spleen tyrosine kinase
7046	TGFBR1	transforming growth factor, beta receptor 1

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
7157	TP53	tumor protein p53
7189	TRAF6	TNF receptor-associated factor 6
7297	TYK2	tyrosine kinase 2
7409	VAV1	vav 1 guanine nucleotide exchange factor
7410	VAV2	vav 2 guanine nucleotide exchange factor
7535	ZAP70	zeta-chain (TCR) associated protein kinase 70kDa
8503	PIK3R3	phosphoinositide-3-kinase, regulatory subunit 3 (gamma)
22918	CD93	CD93 molecule
708	C1QBP	complement component 1, q subcomponent binding protein
3320	HSP90AA1	heat shock protein 90kDa alpha (cytosolic), class A member 1
8717	TRADD	TNFRSF1A-associated via death domain
23118	TAB2	TGF-beta activated kinase 1/MAP3K7 binding protein 2
51567	TDP2	tyrosyl-DNA phosphodiesterase 2
468	ATF4	activating transcription factor 4 (tax-responsive enhancer element B67)
1387	CREBBP	CREB binding protein
2885	GRB2	growth factor receptor-bound protein 2
3560	IL2RB	interleukin 2 receptor, beta
3561	IL2RG	interleukin 2 receptor, gamma
4790	NFKB1	nuclear factor of kappa light polypeptide gene enhancer in B-cells 1
4794	NFKBIE	nuclear factor of kappa light polypeptide gene enhancer in B-cells inhibitor, epsilon
5058	PAK1	p21 protein (Cdc42/Rac)-activated kinase 1
6772	STAT1	signal transducer and activator of transcription 1, 91kDa
6773	STAT2	signal transducer and activator of transcription 2, 113kDa
6778	STAT6	signal transducer and activator of transcription 6, interleukin-4 induced
7185	TRAF1	TNF receptor-associated factor 1
7188	TRAF5	TNF receptor-associated factor 5
8651	SOCS1	suppressor of cytokine signaling 1
8737	RIPK1	receptor (TNFRSF)-interacting serine-threonine kinase 1
10379	IRF9	interferon regulatory factor 9
1654	DDX3X	DEAD (Asp-Glu-Ala-Asp) box polypeptide 3, X-linked
1845	DUSP3	dual specificity phosphatase 3
1956	EGFR	epidermal growth factor receptor
2771	GNAI2	guanine nucleotide binding protein (G protein), alpha inhibiting activity polypeptide 2
3309	HSPA5	heat shock 70kDa protein 5 (glucose-regulated protein, 78kDa)
4215	MAP3K3	mitogen-activated protein kinase kinase kinase 3

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
5337	PLD1	phospholipase D1, phosphatidylcholine-specific
5604	MAP2K1	mitogen-activated protein kinase kinase 1
5781	PTPN11	protein tyrosine phosphatase, non-receptor type 11
6416	MAP2K4	mitogen-activated protein kinase kinase 4
7534	YWHAZ	tyrosine 3-monooxygenase/tryptophan 5-monooxygenase activation protein, zeta polypeptide
10746	MAP3K2	mitogen-activated protein kinase kinase kinase 2
1432	MAPK14	mitogen-activated protein kinase 14
5601	MAPK9	mitogen-activated protein kinase 9
7186	TRAF2	TNF receptor-associated factor 2
1439	CSF2RB	colony stimulating factor 2 receptor, beta, low-affinity (granulocyte-macrophage)
3695	ITGB7	integrin, beta 7
5579	PRKCB	protein kinase C, beta
608	TNFRSF17	tumor necrosis factor receptor superfamily, member 17
4049	LTA	lymphotoxin alpha (TNF superfamily, member 1)
7124	TNF	tumor necrosis factor
7133	TNFRSF1B	tumor necrosis factor receptor superfamily, member 1B
7187	TRAF3	TNF receptor-associated factor 3
10456	HAX1	HCLS1 associated protein X-1
2175	FANCA	Fanconi anemia, complementation group A
3717	JAK2	Janus kinase 2
5335	PLCG1	phospholipase C, gamma 1
2782	GNB1	guanine nucleotide binding protein (G protein), beta polypeptide 1
2786	GNG4	guanine nucleotide binding protein (G protein), gamma 4
958	CD40	CD40 molecule, TNF receptor superfamily member 5
7520	XRCC5	X-ray repair complementing defective repair in Chinese hamster cells 5 (double-strand-break rejoining)
1000	CDH2	cadherin 2, type 1, N-cadherin (neuronal)
1002	CDH4	cadherin 4, type 1, R-cadherin (retinal)
2176	FANCC	Fanconi anemia, complementation group C
2178	FANCE	Fanconi anemia, complementation group E
2188	FANCF	Fanconi anemia, complementation group F
811	CALR	calreticulin
843	CASP10	caspase 10, apoptosis-related cysteine peptidase
3312	HSPA8	heat shock 70kDa protein 8
7132	TNFRSF1A	tumor necrosis factor receptor superfamily, member 1A

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
1237	CCR8	chemokine (C-C motif) receptor 8
6361	CCL17	chemokine (C-C motif) ligand 17
7412	VCAM1	vascular cell adhesion molecule 1
1147	CHUK	conserved helix-loop-helix ubiquitous kinase
3265	HRAS	v-Ha-ras Harvey rat sarcoma viral oncogene homolog
3654	IRAK1	interleukin-1 receptor-associated kinase 1
3665	IRF7	interferon regulatory factor 7
5495	PPM1B	protein phosphatase, Mg2+/Mn2+ dependent, 1B
7128	TNFAIP3	tumor necrosis factor, alpha-induced protein 3
8517	IKBKG	inhibitor of kappa light polypeptide gene enhancer in B-cells, kinase gamma
4267	CD99	CD99 molecule
5478	PPIA	peptidylprolyl isomerase A (cyclophilin A)
4035	LRP1	low density lipoprotein receptor-related protein 1
5609	MAP2K7	mitogen-activated protein kinase kinase 7
23542	MAPK8IP2	mitogen-activated protein kinase 8 interacting protein 2
3674	ITGA2B	integrin, alpha 2b (platelet glycoprotein IIb of IIb/IIIa complex, antigen CD41)
3690	ITGB3	integrin, beta 3 (platelet glycoprotein IIIa, antigen CD61)
355	FAS	Fas (TNF receptor superfamily, member 6)
5578	PRKCA	protein kinase C, alpha
5048	PAFAH1B1	platelet-activating factor acetylhydrolase 1b, regulatory subunit 1 (45kDa)
5049	PAFAH1B2	platelet-activating factor acetylhydrolase 1b, catalytic subunit 2 (30kDa)
1437	CSF2	colony stimulating factor 2 (granulocyte-macrophage)
4254	KITLG	KIT ligand
2826	CCR10	chemokine (C-C motif) receptor 10
6366	CCL21	chemokine (C-C motif) ligand 21
3134	HLA-F	major histocompatibility complex, class I, F
3135	HLA-G	major histocompatibility complex, class I, G
6890	TAP1	transporter 1, ATP-binding cassette, sub-family B (MDR/TAP)
695	BTK	Bruton agammaglobulinemia tyrosine kinase
3702	ITK	IL2-inducible T-cell kinase
29760	BLNK	B-cell linker
1271	CNTFR	ciliary neurotrophic factor receptor
3572	IL6ST	interleukin 6 signal transducer (gp130, oncostatin M receptor)
2833	CXCR3	chemokine (C-X-C motif) receptor 3

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
6352	CCL5	chemokine (C-C motif) ligand 5
2247	FGF2	fibroblast growth factor 2 (basic)
2253	FGF8	fibroblast growth factor 8 (androgen-induced)
2260	FGFR1	fibroblast growth factor receptor 1
2263	FGFR2	fibroblast growth factor receptor 2
1499	CTNNB1	catenin (cadherin-associated protein), beta 1, 88kDa
5747	PTK2	PTK2 protein tyrosine kinase 2
5829	PXN	paxillin
7414	VCL	vinculin
2246	FGF1	fibroblast growth factor 1 (acidic)
2252	FGF7	fibroblast growth factor 7
2261	FGFR3	fibroblast growth factor receptor 3
5879	RAC1	ras-related C3 botulinum toxin substrate 1 (rho family, small GTP binding protein Rac1)
7454	WAS	Wiskott-Aldrich syndrome (eczema-thrombocytopenia)
5817	PVR	poliovirus receptor
5818	PVRL1	poliovirus receptor-related 1 (herpesvirus entry mediator C)
25945	PVRL3	poliovirus receptor-related 3
2784	GNB3	guanine nucleotide binding protein (G protein), beta polypeptide 3
2785	GNG3	guanine nucleotide binding protein (G protein), gamma 3
567	B2M	beta-2-microglobulin
912	CD1D	CD1d molecule
2872	MKNK2	MAP kinase interacting serine/threonine kinase 2
3727	JUND	jun D proto-oncogene
5594	MAPK1	mitogen-activated protein kinase 1
1365	CLDN3	claudin 3
5566	PRKACA	protein kinase, cAMP-dependent, catalytic, alpha
5923	RASGRF1	Ras protein-specific guanine nucleotide-releasing factor 1
967	CD63	CD63 molecule
7076	TIMP1	TIMP metallopeptidase inhibitor 1
821	CANX	calnexin
3688	ITGB1	integrin, beta 1 (fibronectin receptor, beta polypeptide, antigen CD29 includes MDF2, MSK12)
868	CBLB	Cas-Br-M (murine) ecotropic retroviral transforming sequence b
4915	NTRK2	neurotrophic tyrosine kinase, receptor, type 2
8440	NCK2	NCK adaptor protein 2
7430	EZR	ezrin

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
9641	IKBKE	inhibitor of kappa light polypeptide gene enhancer in B-cells, kinase epsilon
207	AKT1	v-akt murine thymoma viral oncogene homolog 1
4296	MAP3K11	mitogen-activated protein kinase kinase kinase 11
9261	MAPKAPK2	mitogen-activated protein kinase-activated protein kinase 2
4067	LYN	v-yes-1 Yamaguchi sarcoma viral related oncogene homolog
5336	PLCG2	phospholipase C, gamma 2 (phosphatidylinositol-specific)
56848	SPHK2	sphingosine kinase 2
3985	LIMK2	LIM domain kinase 2
56288	PARD3	par-3 partitioning defective 3 homolog (C. elegans)
836	CASP3	caspase 3, apoptosis-related cysteine peptidase
5530	PPP3CA	protein phosphatase 3, catalytic subunit, alpha isozyme
7096	TLR1	toll-like receptor 1
7184	HSP90B1	heat shock protein 90kDa beta (Grp94), member 1
3845	KRAS	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog
10235	RASGRP2	RAS guanyl releasing protein 2 (calcium and DAG-regulated)
1843	DUSP1	dual specificity phosphatase 1
3308	HSPA4	heat shock 70kDa protein 4
930	CD19	CD19 molecule
933	CD22	CD22 molecule
462	SERPINC1	serpin peptidase inhibitor, clade C (antithrombin), member 1
710	SERPING1	serpin peptidase inhibitor, clade G (C1 inhibitor), member 1
716	C1S	complement component 1, s subcomponent
2147	F2	coagulation factor II (thrombin)
975	CD81	CD81 molecule
8519	IFITM1	interferon induced transmembrane protein 1 (9-27)
1441	CSF3R	colony stimulating factor 3 receptor (granulocyte)
5319	PLA2G1B	phospholipase A2, group IB (pancreas)
5993	RFX5	regulatory factor X, 5 (influences HLA class II expression)
8625	RFXANK	regulatory factor X-associated ankyrin-containing protein
921	CD5	CD5 molecule
923	CD6	CD6 molecule
2155	F7	coagulation factor VII (serum prothrombin conversion accelerator)
2159	F10	coagulation factor X
7035	TFPI	tissue factor pathway inhibitor (lipoprotein-associated coagulation inhibitor)
1513	CTSK	cathepsin K

Table A1.1. CTN gene names.

ENTREZ ID	Gene Symbol	Gene Name
3827	KNG1	kininogen 1
7408	VASP	vasodilator-stimulated phosphoprotein
8976	WASL	Wiskott-Aldrich syndrome-like
5435	POLR2F	polymerase (RNA) II (DNA directed) polypeptide F
5437	POLR2H	polymerase (RNA) II (DNA directed) polypeptide H
5440	POLR2K	polymerase (RNA) II (DNA directed) polypeptide K, 7.0kDa
2783	GNB2	guanine nucleotide binding protein (G protein), beta polypeptide 2
55970	GNG12	guanine nucleotide binding protein (G protein), gamma 12
2165	F13B	coagulation factor XIII, B polypeptide
2266	FGG	fibrinogen gamma chain
3680	ITGA9	integrin, alpha 9
6696	SPP1	secreted phosphoprotein 1
3952	LEP	leptin
3953	LEPR	leptin receptor
6383	SDC2	syndecan 2
6385	SDC4	syndecan 4

Table A1.2. Interactions between genes with the correlation value associated with each link.

ENTREZ ID	ENTREZ ID	Correlation Value
5970	5590	0.46000000000
5608	5970	0.45657651194
6778	4790	0.45681529800
5435	5437	0.45767202200
2782	2786	0.45847250161
5437	5440	0.45889674271
5618	7410	0.45967703649
3654	1147	0.46057684487
6714	6850	0.46064545328
6361	7412	0.46114884339
5580	2185	0.46121095289
3556	5295	0.46180330658
4261	5595	0.46226712328
836	5530	0.46316475956
5588	5590	0.46425962369
5295	5494	0.46447517025
2261	2252	0.46470102610
1437	4254	0.46475417377
710	2147	0.46518930757
2784	2785	0.46520743280
5048	5049	0.46526526037
8651	3560	0.46599275992
7534	10746	0.46611556733
8717	3320	0.46624854020
2147	462	0.46626926935

Table A1.2. Interactions between genes with the correlation value associated with each link.

ENTREZ ID	ENTREZ ID	Correlation Value
5777	7409	0.46747676193
1387	6778	0.46857846542
7454	5879	0.46895315439
3556	6774	0.47057973520
7534	4215	0.47071834851
5580	5494	0.47117585819
3932	867	0.47129688147
5970	2033	0.47139273512
2826	6366	0.47161452921
2165	2266	0.47257125416
5580	947	0.47382419437
3680	6696	0.47576917714
3654	3665	0.47619507533
3952	3953	0.47636508611
3717	5335	0.47805337238
6773	6778	0.47809497813
8517	7128	0.47868866878
3689	2185	0.48014405296
5594	3727	0.48213580836
7046	5295	0.48238743412
7188	8737	0.48270544097
1843	3308	0.48320801048
7189	6714	0.48414770168
5618	2688	0.48436113313
3113	3115	0.48698597825
1432	5601	0.48801756250
10456	7133	0.49023385246
3654	8517	0.49137184938
916	4690	0.49369892750
5478	4267	0.49376195737
4915	8440	0.49403333758
958	7520	0.49517065629
1956	1845	0.49780468911
1000	1002	0.49875261951
567	912	0.50580011454
916	7535	0.50725879931
8651	2885	0.50783603708
2534	3791	0.50887365853
22918	4478	0.51010887650
7132	843	0.51096480545
5159	8503	0.51253645773
6890	3135	0.51279532390
23542	5609	0.51286976864
3561	2885	0.51334493338
3683	3383	0.51528074421
5970	4793	0.51711552781
3312	811	0.51743070863
3937	5295	0.51889063252
4067	5336	0.51919156878
1399	3635	0.52272575723
2534	5788	0.52635860174
9641	7430	0.52732581112
933	930	0.52808436213

Table A1.2. Interactions between genes with the correlation value associated with each link.

ENTREZ ID	ENTREZ ID	Correlation Value
1956	2771	0.52824794972
23118	708	0.52876717830
5777	81	0.53170015294
7124	7133	0.53291727916
10235	3845	0.53351957647
7414	5747	0.53390957288
2534	7297	0.53520204583
4067	56848	0.53543050773
1759	6714	0.53662895930
5588	3551	0.53762455857
6772	6773	0.53851702383
5818	25945	0.53879432139
920	6850	0.54073743189
975	8519	0.54174562687
387	4478	0.54194917228
3480	6774	0.54386528533
5159	6776	0.54517732389
6464	867	0.54522048951
5156	6774	0.54620466413
1147	8517	0.54894043826
2260	2253	0.54898024423
7046	5296	0.54979563641
5058	2885	0.55070379221
8976	7408	0.55247887298
2934	81	0.55538083661
7414	1499	0.55815014321
2176	2188	0.56160898876
1441	5319	0.56207995630
6773	10379	0.56591024397
56288	3985	0.56699908185
4790	4794	0.57644262701
29760	3702	0.57744031617
5604	1956	0.57920553589
7414	5829	0.57966292124
3561	6772	0.58083814772
4035	23542	0.58279686205
5175	3055	0.58496403318
3932	5777	0.58738637753
920	7535	0.58972924724
5777	2185	0.58975445077
2263	2253	0.58988690909
6416	10746	0.59058953048
1956	5781	0.59067003192
3932	919	0.59138055213
8737	4790	0.59826867796
3654	3265	0.59912823790
7187	608	0.59957936137
7534	3309	0.60391766936
5594	2872	0.60483514294
7187	7133	0.60595223106
3385	4478	0.61407436662
7132	3312	0.61547233341
5595	6198	0.62191358516

Table A1.2. Interactions between genes with the correlation value associated with each link.

ENTREZ ID	ENTREZ ID	Correlation Value
3827	1513	0.62253771742
3683	3385	0.62557963253
5777	5175	0.62695716706
207	4296	0.62701724011
2159	7035	0.62763938765
1956	5337	0.63330715488
7186	5601	0.63875214992
1399	867	0.64284379666
1237	6361	0.64533200287
868	8440	0.64785994323
2534	3575	0.64957430332
2247	2260	0.65037119197
6383	6385	0.65197935503
8625	5993	0.65262617863
3572	1271	0.65409243141
2934	5294	0.65429458656
3932	5788	0.65749776877
1387	468	0.65877207172
2261	2246	0.65882887477
6464	3480	0.66034871062
207	9261	0.66109903401
7534	1956	0.66138950012
3688	821	0.66148448324
2207	6850	0.66207684507
51567	3320	0.66792776280
5580	60	0.67613121679
5588	7409	0.67678076667
5788	5595	0.67716154489
7185	8737	0.67779453985
5566	5923	0.67909139678
5566	1365	0.68155813877
2176	2178	0.68398261080
5595	7157	0.68505802974
5495	1147	0.68733033407
5777	7535	0.70340785580
7409	5618	0.70587630326
3716	5788	0.70628944810
2783	55970	0.71148467701
920	3113	0.71230060096
2833	6352	0.71408670050
355	5578	0.72348454708
5818	5817	0.72902466416
919	6776	0.74049645412
710	716	0.75261034194
2175	5335	0.75294636207
2155	7035	0.75465179429
921	923	0.76247382103
5290	5295	0.77074553925
5579	3695	0.77497434106
4049	7124	0.78857869455
29760	695	0.79610275380
2534	778	0.80645648920
3683	3689	0.83095423110

Table A1.2. Interactions between genes with the correlation value associated with each link.

ENTREZ ID	ENTREZ ID	Correlation Value
7184	7096	0.83589657996
7534	1654	0.84597414386
2212	6850	0.85900946708
23118	3320	0.85952105149
7076	967	0.86502129994
5295	924	0.92351951820
3690	3674	0.92732190288
1072	60	0.93174768195
1439	5579	0.95831887372
3932	5588	1.00425820442
71	60	1.00777555854
6890	3134	1.11125400284

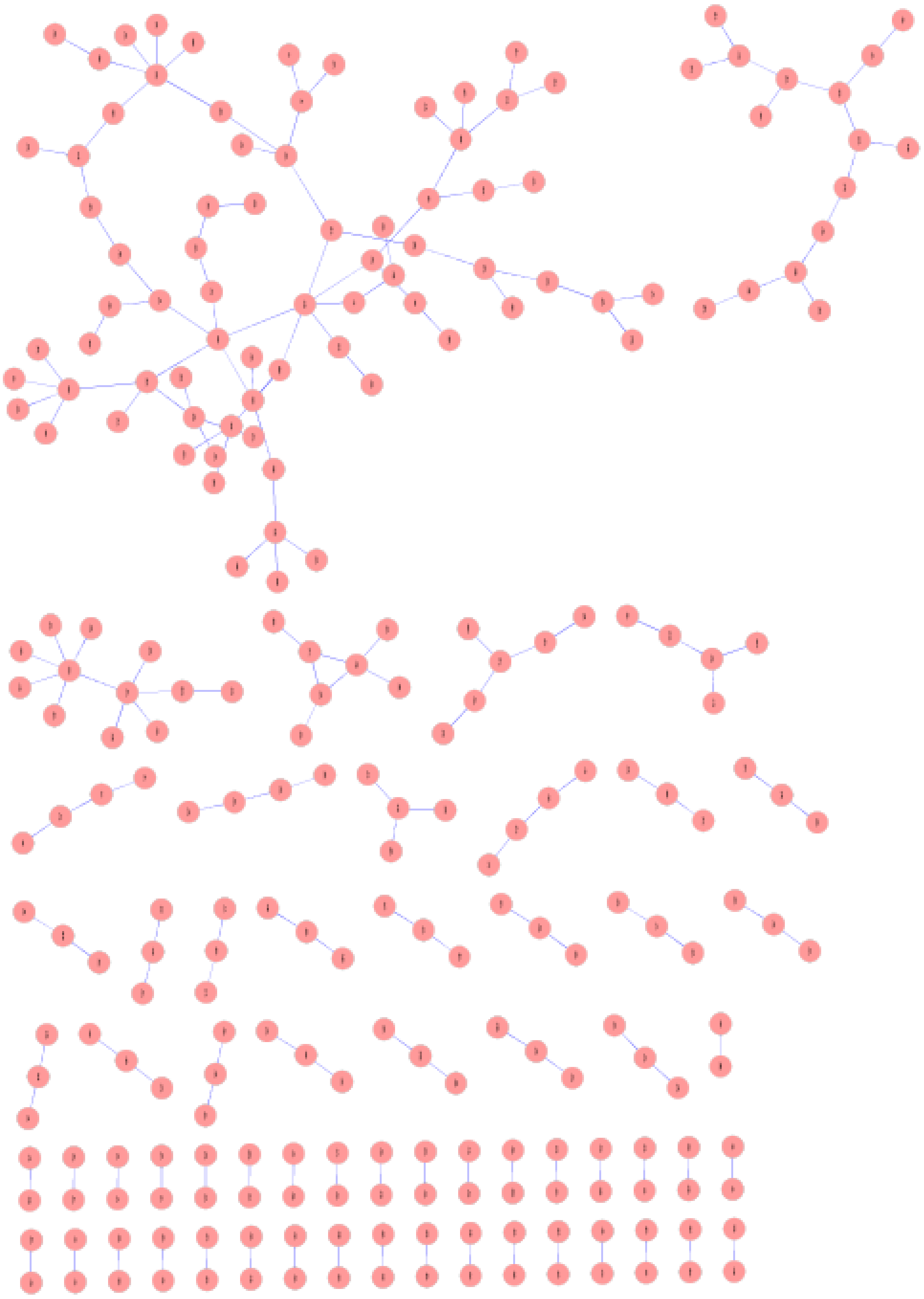


Figure A1.1. The Central T-cell Network (CTN) which contains 256 nodes and 196 edges.

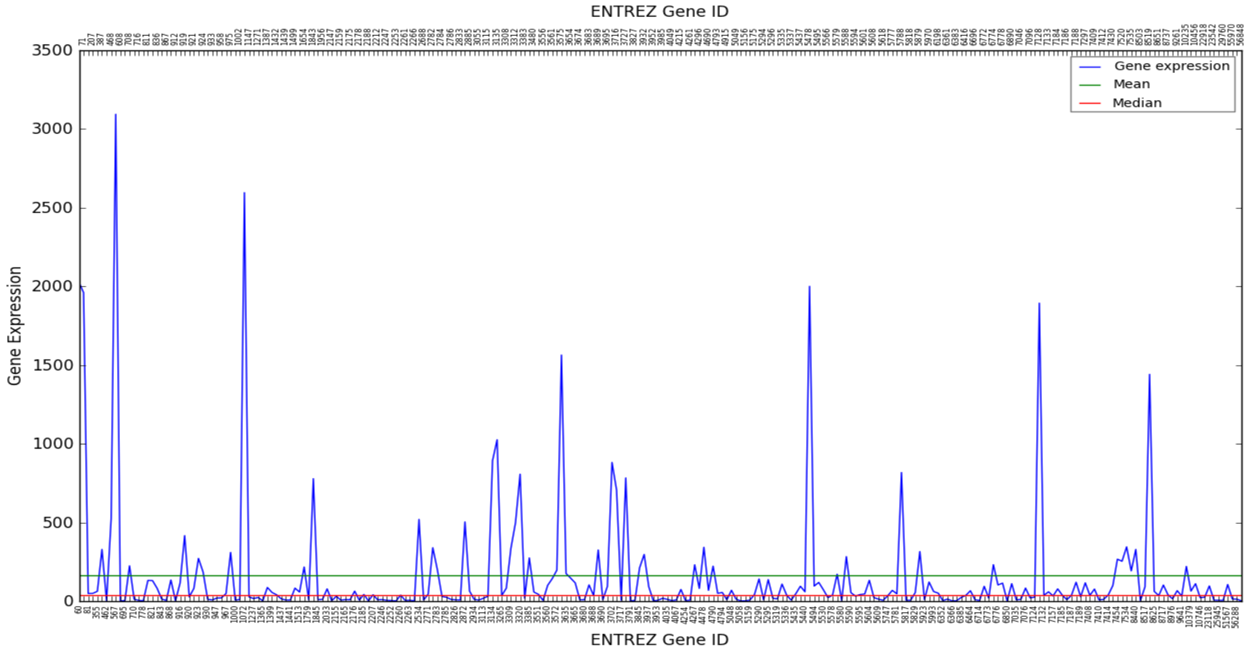


Figure A1.2. Difference between mean and median in non-normally distributed microarray data for CTN genes.

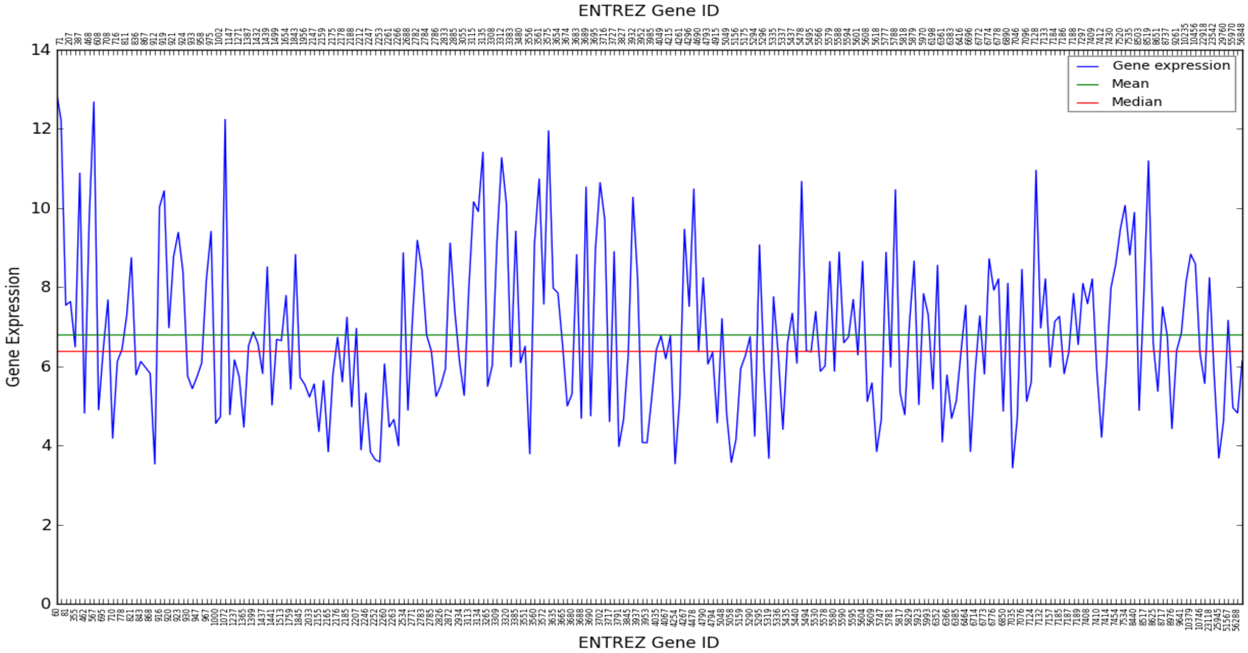


Figure A1.3. Difference between mean and median in normally distributed microarray data for CTN genes.

Appendix 2

This section contains the result of the GO and exhaustive analysis on different sets of data. Tables A2.1 and A2.2 contain the analysis for the most frequent attractors, Tables A2.3 to A2.6 show the result of Go analysis on different clusters and sub-clusters. Table A2.7 show the result of data mining method combined with Go analysis.

Table A2.1. GO Analysis for attractor with period 2, complete table.

Term	Count	%	P-Value	Genes	List Total
Only Ones Pattern					
regulation of system process	4	40.00	0.0162	2771, 5781, 1956, 5604	9
MAPKKK cascade	5	50.00	0.0211	2771, 5781, 1956, 4215, 5604	9
vesicle	4	40.00	0.0271	1956, 5337, 7534, 3309	9
protein tyrosine phosphatase activity	3	30.00	0.0414	5781, 5604, 1845	10
cell fraction	5	50.00	0.0469	2771, 5781, 5337, 5604, 7534	9
hydrolase	4	40.00	0.0570	5781, 1654, 5337, 1845	10
cell projection morphogenesis	3	30.00	0.0624	5781, 1956, 5604	9
cell part morphogenesis	3	30.00	0.0624	5781, 1956, 5604	9
neuron projection morphogenesis	3	30.00	0.0624	5781, 1956, 5604	9
GnRH signaling pathway	4	40.00	0.0714	1956, 5337, 4215, 5604	10
cytoplasmic vesicle	3	30.00	0.0933	1956, 7534, 3309	9
cytoplasmic membrane-bounded vesicle	3	30.00	0.0933	1956, 7534, 3309	9
Map Kinase Inactivation of SMRT Corepressor	2	20.00	0.0989	1956, 5604	6
Only Zero Pattern					
Leukocyte transendothelial migration	20	27.03	0.0030	5747, 8503, 5175, 7409, 7414, 3683, 1499, 3383, 5290, 4478, 387, 3689, 2185, 71, 81, 60, 5294, 5829, 5295, 5296	72
Regulation of actin cytoskeleton	25	33.78	0.0082	2147, 2934, 7414, 2253, 3683, 1072, 2247, 387, 81, 60, 5829, 5159, 5747, 8503, 2263, 7409, 2260, 5290, 4478, 3689, 5595, 71, 5294, 5295, 5296	72
cell adhesion	10	13.51	0.0231	3689, 5175, 7414, 3385, 22918, 3683, 5829, 947, 3383, 1499	74
cytoskeleton	11	14.86	0.0421	4478, 387, 3689, 2934, 7414, 71, 3683, 60, 5829, 1499, 1072	74
carbohydrate binding	8	10.81	0.0533	811, 462, 2247, 2263, 22918, 947, 5788, 2260	69
membrane	35	47.30	0.0599	7414, 3385, 22918, 3683, 924, 947, 3383, 387, 919, 2207, 608,	74

Table A2.1. GO Analysis for attractor with period 2, complete table.

Term	Count	%	P-Value	Genes	List Total
				7046, 3575, 5580, 5159, 3055, 5747, 5175, 2263, 3791, 778, 3551, 3716, 7133, 7132, 2260, 4478, 3689, 468, 2534, 2185, 10456, 2212, 3932, 5788	
cation homeostasis	8	10.81	0.0635	811, 2147, 3791, 2185, 778, 3932, 7157, 5788	72
cell adhesion	15	20.27	0.0728	5175, 7414, 3385, 22918, 3683, 947, 3383, 1499, 4478, 6850, 387, 3689, 2185, 5829, 5788	72
biological adhesion	15	20.27	0.0728	5175, 7414, 3385, 22918, 3683, 947, 3383, 1499, 4478, 6850, 387, 3689, 2185, 5829, 5788	72
cell motion	18	24.32	0.0784	5747, 7414, 3791, 3683, 947, 1072, 3383, 4478, 6198, 6850, 2247, 3689, 2534, 2185, 71, 60, 7046, 5159	72
Combined Pattern					
protein kinase cascade	22	48.89	0.0142	7535, 6714, 8517, 1147, 6774, 5618, 6772, 6464, 7124, 5058, 5608, 8651, 7189, 6416, 3480, 1399, 2688, 2885, 8737, 4793, 3654, 10746	45
regulation of kinase activity	14	31.11	0.0155	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
regulation of transferase activity	14	31.11	0.0155	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
positive regulation of transferase activity	14	31.11	0.0155	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
positive regulation of kinase activity	14	31.11	0.0155	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
GO:0042981~regulation of apoptosis	21	46.67	0.0180	3265, 4049, 6714, 8517, 5618, 6772, 3560, 5970, 7124, 916, 5608, 7189, 3480, 7188, 7410, 5590, 8737, 7128, 7185, 3654, 3635	45
I-kappaB kinase/NF-kappaB cascade	7	15.56	0.0204	7189, 8517, 1147, 8737, 6772, 3654, 4793	45
regulation of phosphorus metabolic process	16	35.56	0.0253	920, 8517, 5618, 6464, 7124, 5058, 916, 5608, 8651, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45

Table A2.1. GO Analysis for attractor with period 2, complete table.

Term	Count	%	P-Value	Genes	List Total
regulation of phosphorylation	16	35.56	0.0253	920, 8517, 5618, 6464, 7124, 5058, 916, 5608, 8651, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
regulation of phosphate metabolic process	16	35.56	0.0253	920, 8517, 5618, 6464, 7124, 5058, 916, 5608, 8651, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
regulation of cell death	21	46.67	0.0256	3265, 4049, 6714, 8517, 5618, 6772, 3560, 5970, 7124, 916, 5608, 7189, 3480, 7188, 7410, 5590, 8737, 7128, 7185, 3654, 3635	45
regulation of programmed cell death	21	46.67	0.0256	3265, 4049, 6714, 8517, 5618, 6772, 3560, 5970, 7124, 916, 5608, 7189, 3480, 7188, 7410, 5590, 8737, 7128, 7185, 3654, 3635	45
mutagenesis site	22	48.89	0.0286	7535, 920, 3265, 8517, 1147, 6772, 2033, 3560, 5970, 7124, 5058, 867, 5608, 1759, 7189, 3480, 5590, 2885, 8737, 7128, 4793, 3654	45
regulation of protein kinase activity	13	28.89	0.0299	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 5590, 2688, 8737, 3654, 10746	45
positive regulation of protein kinase activity	13	28.89	0.0299	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 5590, 2688, 8737, 3654, 10746	45
positive regulation of catalytic activity	15	33.33	0.0310	920, 8517, 5618, 6464, 6772, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
p38 MAPK Signaling Pathway	7	15.56	0.0343	6416, 3265, 2885, 8737, 6464, 6772, 5608	39
intracellular signaling cascade	27	60.00	0.0391	7535, 4049, 3265, 8517, 1147, 6774, 5618, 6464, 6772, 7124, 8651, 7189, 3480, 6416, 8737, 3654, 3635, 6714, 5058, 5608, 1399, 7410, 5590, 2688, 2885, 4793, 10746	45
response to cytokine stimulus	8	17.78	0.0402	6714, 6774, 5156, 8737, 6772, 5970, 8651, 3654	45
positive regulation of programmed cell death	11	24.44	0.0428	7189, 8517, 6714, 4049, 7410, 8737, 6772, 7124, 916, 5608, 3635	45
positive regulation of apoptosis	11	24.44	0.0428	7189, 8517, 6714, 4049, 7410,	45

Table A2.1. GO Analysis for attractor with period 2, complete table.

Term	Count	%	P-Value	Genes	List Total
				8737, 6772, 7124, 916, 5608, 3635	
positive regulation of cell death	11	24.44	0.0428	7189, 8517, 6714, 4049, 7410, 8737, 6772, 7124, 916, 5608, 3635	45
positive regulation of molecular function	17	37.78	0.0432	920, 8517, 5618, 6464, 2033, 6772, 5970, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	45
Endocytosis	10	22.22	0.0482	7189, 3480, 3265, 6714, 5590, 5156, 3561, 3560, 867, 1759	45
molecular adaptor activity	5	11.11	0.0515	6714, 1399, 2885, 6464, 4690	45
protein complex biogenesis	13	28.89	0.0664	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	45
positive regulation of cell communication	13	28.89	0.0664	7535, 920, 4049, 6714, 3265, 5970, 7124, 916, 7189, 7188, 5590, 2688, 8737	45
macromolecular complex subunit organization	13	28.89	0.0664	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	45
macromolecular complex assembly	13	28.89	0.0664	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	45
protein complex assembly	13	28.89	0.0664	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	45
NF-kB Signaling Pathway	8	17.78	0.0688	7189, 8517, 1147, 8737, 7128, 5970, 7124, 3654	39
response to steroid hormone stimulus	8	17.78	0.0703	6714, 6774, 2688, 5156, 2033, 5970, 7124, 8651	45
cytokine-mediated signaling pathway	8	17.78	0.0703	6774, 8737, 6772, 5970, 3560, 7124, 8651, 3654	45
NOD-like receptor signaling pathway	7	15.56	0.0791	7189, 8517, 1147, 7128, 5970, 7124, 4793	45
positive regulation of signal transduction	12	26.67	0.0798	7189, 7535, 920, 7188, 3265, 6714, 4049, 2688, 8737, 5970, 7124, 916	45
Cytosolic DNA-sensing pathway	6	13.33	0.0920	8517, 1147, 8737, 5970, 4793, 3665	45
zinc-finger	8	17.78	0.0993	7189, 7188, 8517, 5590, 7410, 2033, 7128, 867	45

Table A2.2. GO Analysis for attractor with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
Only One Pattern					
regulation of system process	4	40.00	0.0162	2771, 5781, 1956, 5604	9
MAPKKK cascade	5	50.00	0.0211	2771, 5781, 1956, 4215, 5604	9
vesicle	4	40.00	0.0271	1956, 5337, 7534, 3309	9
protein tyrosine phosphatase activity	3	30.00	0.0414	5781, 5604, 1845	10
cell fraction	5	50.00	0.0469	2771, 5781, 5337, 5604, 7534	9
hydrolase	4	40.00	0.0570	5781, 1654, 5337, 1845	10
neuron projection morphogenesis	3	30.00	0.0624	5781, 1956, 5604	9
cell projection morphogenesis	3	30.00	0.0624	5781, 1956, 5604	9
cell part morphogenesis	3	30.00	0.0624	5781, 1956, 5604	9
GnRH signaling pathway	4	40.00	0.0714	1956, 5337, 4215, 5604	10
cytoplasmic vesicle	3	30.00	0.0933	1956, 7534, 3309	9
cytoplasmic membrane-bounded vesicle	3	30.00	0.0933	1956, 7534, 3309	9
Map Kinase Inactivation of SMRT Corepressor	2	20.00	0.0989	1956, 5604	6
Only Zero Pattern					
Leukocyte transendothelial migration	20	27.78	0.0024	5747, 8503, 5175, 7409, 7414, 3683, 1499, 3383, 5290, 4478, 387, 3689, 2185, 71, 81, 60, 5294, 5829, 5295, 5296	71
Regulation of actin cytoskeleton	25	34.72	0.0062	2147, 2934, 7414, 2253, 3683, 1072, 2247, 387, 81, 60, 5829, 5159, 5747, 8503, 2263, 7409, 2260, 5290, 4478, 3689, 5595, 71, 5294, 5295, 5296	71
cell adhesion	10	13.89	0.0182	3689, 5175, 7414, 3385, 22918, 3683, 5829, 947, 3383, 1499	72
cytoskeleton	11	15.28	0.0331	4478, 387, 3689, 2934, 7414, 71, 3683, 60, 5829, 1499, 1072	72
disease mutation	26	36.11	0.0461	843, 2147, 6772, 2934, 7414, 2253, 1499, 1387, 81, 60, 7046, 3575, 2263, 778, 7157, 2260, 7132, 5290, 462, 3689, 4261, 71, 710, 3932, 5295, 5788	72
carbohydrate binding	8	11.11	0.0484	811, 462, 2247, 2263, 22918, 947, 5788, 2260	68
cation homeostasis	8	11.11	0.0580	811, 2147, 3791, 2185, 778,	71

Table A2.2. GO Analysis for attractor with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
				3932, 7157, 5788	
cell adhesion	15	20.83	0.0632	5175, 7414, 3385, 22918, 3683, 947, 3383, 1499, 4478, 6850, 387, 3689, 2185, 5829, 5788	71
biological adhesion	15	20.83	0.0632	5175, 7414, 3385, 22918, 3683, 947, 3383, 1499, 4478, 6850, 387, 3689, 2185, 5829, 5788	71
cell motion	18	25.00	0.0666	5747, 7414, 3791, 3683, 947, 1072, 3383, 4478, 6198, 6850, 2247, 3689, 2534, 2185, 71, 60, 7046, 5159	71
calcium ion binding	12	16.67	0.0859	811, 716, 2147, 2934, 6772, 6773, 6776, 778, 6778, 81, 22918, 3683	68
cell motility	14	19.44	0.0922	5747, 3791, 947, 3383, 1072, 4478, 6198, 6850, 3689, 2247, 2534, 2185, 7046, 5159	71
localization of cell	14	19.44	0.0922	5747, 3791, 947, 3383, 1072, 4478, 6198, 6850, 3689, 2247, 2534, 2185, 7046, 5159	71
cell migration	14	19.44	0.0922	5747, 3791, 947, 3383, 1072, 4478, 6198, 6850, 3689, 2247, 2534, 2185, 7046, 5159	71
mTOR signaling pathway	7	9.72	0.0931	5290, 6198, 8503, 5595, 5294, 5295, 5296	71
calcium ion homeostasis	7	9.72	0.0971	811, 2147, 3791, 2185, 778, 3932, 5788	71
di-, tri-valent inorganic cation homeostasis	7	9.72	0.0971	811, 2147, 3791, 2185, 778, 3932, 5788	71
metal ion homeostasis	7	9.72	0.0971	811, 2147, 3791, 2185, 778, 3932, 5788	71
cellular cation homeostasis	7	9.72	0.0971	811, 2147, 2185, 778, 3932, 7157, 5788	71
Combined Pattern					
regulation of kinase activity	14	28.57	0.0122	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
positive regulation of kinase activity	14	28.57	0.0122	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
positive regulation of transferase activity	14	28.57	0.0122	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44

Table A2.2. GO Analysis for attractor with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
regulation of transferase activity	14	28.57	0.0122	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
regulation of phosphate metabolic process	16	32.65	0.0196	920, 8517, 5618, 6464, 7124, 5058, 916, 5608, 8651, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
regulation of phosphorus metabolic process	16	32.65	0.0196	920, 8517, 5618, 6464, 7124, 5058, 916, 5608, 8651, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
regulation of phosphorylation	16	32.65	0.0196	920, 8517, 5618, 6464, 7124, 5058, 916, 5608, 8651, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
positive regulation of protein kinase activity	13	26.53	0.0242	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 5590, 2688, 8737, 3654, 10746	44
regulation of protein kinase activity	13	26.53	0.0242	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 5590, 2688, 8737, 3654, 10746	44
protein kinase cascade	21	42.86	0.0259	7535, 6714, 8517, 1147, 6774, 5618, 6464, 7124, 5058, 5608, 8651, 7189, 6416, 3480, 1399, 2688, 2885, 8737, 4793, 3654, 10746	44
regulation of apoptosis	20	40.82	0.0329	3265, 4049, 6714, 8517, 5618, 3560, 5970, 7124, 916, 5608, 7189, 3480, 7188, 7410, 5590, 8737, 7128, 7185, 3654, 3635	44
Endocytosis	10	20.41	0.0412	7189, 3480, 3265, 6714, 5590, 5156, 3561, 3560, 867, 1759	44
regulation of programmed cell death	20	40.82	0.0449	3265, 4049, 6714, 8517, 5618, 3560, 5970, 7124, 916, 5608, 7189, 3480, 7188, 7410, 5590, 8737, 7128, 7185, 3654, 3635	44
regulation of cell death	20	40.82	0.0449	3265, 4049, 6714, 8517, 5618, 3560, 5970, 7124, 916, 5608, 7189, 3480, 7188, 7410, 5590, 8737, 7128, 7185, 3654, 3635	44
molecular adaptor activity	5	10.20	0.0473	6714, 1399, 2885, 6464, 4690	44
mutagenesis site	21	42.86	0.0482	7535, 920, 3265, 8517, 1147, 2033, 3560, 5970, 7124, 5058, 867, 5608, 1759, 7189, 3480, 5590, 2885, 8737, 7128, 4793, 3654	44

Table A2.2. GO Analysis for attractor with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
protein complex biogenesis	13	26.53	0.0550	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	44
macromolecular complex assembly	13	26.53	0.0550	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	44
positive regulation of cell communication	13	26.53	0.0550	7535, 920, 4049, 6714, 3265, 5970, 7124, 916, 7189, 7188, 5590, 2688, 8737	44
macromolecular complex subunit organization	13	26.53	0.0550	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	44
protein complex assembly	13	26.53	0.0550	6714, 3265, 5618, 3560, 916, 3480, 5590, 2885, 3556, 3654, 7185, 3665, 4690	44
intracellular signaling cascade	26	53.06	0.0579	7535, 4049, 3265, 8517, 1147, 6774, 5618, 6464, 7124, 8651, 7189, 3480, 6416, 8737, 3654, 3635, 6714, 5058, 5608, 5590, 7410, 1399, 2688, 2885, 4793, 10746	44
NF-kB Signaling Pathway	8	16.33	0.0592	7189, 8517, 1147, 8737, 7128, 5970, 7124, 3654	38
response to steroid hormone stimulus	8	16.33	0.0621	6714, 6774, 2688, 5156, 2033, 5970, 7124, 8651	44
positive regulation of catalytic activity	14	28.57	0.0636	920, 8517, 5618, 6464, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
positive regulation of signal transduction	12	24.49	0.0673	7189, 7535, 920, 7188, 3265, 6714, 4049, 2688, 8737, 5970, 7124, 916	44
NOD-like receptor signaling pathway	7	14.29	0.0709	7189, 8517, 1147, 7128, 5970, 7124, 4793	44
positive regulation of molecular function	16	32.65	0.0787	920, 8517, 5618, 6464, 2033, 5970, 7124, 5058, 5608, 7189, 7410, 5590, 2688, 8737, 3654, 10746	44
Cytosolic DNA-sensing pathway	6	12.24	0.0837	8517, 1147, 8737, 5970, 4793, 3665	44
I-kappaB kinase/NF-kappaB cascade	6	12.24	0.0864	7189, 8517, 1147, 8737, 3654, 4793	44
zinc-finger	8	16.33	0.0884	7189, 7188, 8517, 5590, 7410, 2033, 7128, 867	44
RIG-I-like receptor signaling pathway	8	16.33	0.0954	7189, 8517, 1147, 8737, 5970,	44

Table A2.2. GO Analysis for attractor with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
				7124, 4793, 3665	
positive regulation of apoptosis	10	20.41	0.0997	7189, 8517, 6714, 4049, 7410, 8737, 7124, 916, 5608, 3635	44
positive regulation of cell death	10	20.41	0.0997	7189, 8517, 6714, 4049, 7410, 8737, 7124, 916, 5608, 3635	44
positive regulation of programmed cell death	10	20.41	0.0997	7189, 8517, 6714, 4049, 7410, 8737, 7124, 916, 5608, 3635	44

Table A2.3. GO analysis for three groups of clustered attractors with period 2, complete table.

Term	Count	%	PValue	Genes	List Total
Right Cluster					
p38 MAPK Signaling Pathway	7	14.00	0.0651	6416, 3265, 2885, 8737, 9261, 5608, 8717	35
Long-term potentiation	6	12.00	0.0961	3265, 5579, 5578, 2033, 3845, 5530	47
Center Cluster					
cell fraction	9	50.00	0.0019	3480, 2771, 5781, 5336, 5337, 5604, 7534, 56848, 4067	16
positive regulation of cell motion	5	27.78	0.0068	3480, 1956, 5337, 5604, 4067	17
Fc gamma R-mediated phagocytosis	7	38.89	0.0069	1399, 5336, 5337, 5604, 56848, 4067, 3635	18
phosphoprotein	17	94.44	0.0144	5781, 1956, 6464, 4215, 5604, 867, 56848, 3309, 2771, 3480, 1399, 5336, 1654, 5337, 7534, 4067, 3635	18
regulation of cell motion	5	27.78	0.0204	3480, 1956, 5337, 5604, 4067	17
Neurotrophin signaling pathway	7	38.89	0.0213	1399, 5781, 5336, 6464, 4215, 5604, 7534	18
hydrolase	6	33.33	0.0214	5781, 5336, 1654, 5337, 1845, 3635	18
phosphorus metabolic process	10	55.56	0.0226	3480, 2771, 5781, 1956, 6464, 4215, 5604, 4067, 1845, 3635	17
phosphate metabolic process	10	55.56	0.0226	3480, 2771, 5781, 1956, 6464, 4215, 5604, 4067, 1845, 3635	17
MAPKKK cascade	7	38.89	0.0245	2771, 1399, 5781, 1956, 6464, 4215, 5604	17
IGF-1 Signaling Pathway	4	22.22	0.0253	3480, 5781, 6464, 5604	11
membrane fraction	6	33.33	0.0279	3480, 2771, 5336, 5337, 56848, 4067	16

Table A2.3. GO analysis for three groups of clustered attractors with period 2, complete table.

Term	Count	%	PValue	Genes	List Total
insoluble fraction	6	33.33	0.0279	3480, 2771, 5336, 5337, 56848, 4067	16
SH2 domain	7	38.89	0.0312	1399, 5781, 5336, 6464, 867, 4067, 3635	18
cell part morphogenesis	4	22.22	0.0337	3480, 5781, 1956, 5604	17
positive regulation of cell migration	4	22.22	0.0337	3480, 1956, 5337, 5604	17
neuron projection development	4	22.22	0.0337	3480, 5781, 1956, 5604	17
cell projection morphogenesis	4	22.22	0.0337	3480, 5781, 1956, 5604	17
neuron projection morphogenesis	4	22.22	0.0337	3480, 5781, 1956, 5604	17
Sprouty regulation of tyrosine kinase signals	4	22.22	0.0412	1956, 6464, 5604, 867	11
intracellular signaling cascade	12	66.67	0.0422	3480, 2771, 1399, 5781, 5336, 1956, 5337, 6464, 4215, 5604, 4067, 3635	17
phosphatase activity	4	22.22	0.0440	5781, 5604, 1845, 3635	18
regulation of hormone levels	3	16.67	0.0496	5781, 7534, 4067	17
cell activation during immune response	3	16.67	0.0496	5336, 7534, 4067	17
regulation of calcium ion transport	3	16.67	0.0496	2771, 5336, 4067	17
leukocyte activation during immune response	3	16.67	0.0496	5336, 7534, 4067	17
cellular component morphogenesis	4	22.22	0.0539	3480, 5781, 1956, 5604	17
positive regulation of locomotion	4	22.22	0.0539	3480, 1956, 5337, 5604	17
cell morphogenesis	4	22.22	0.0539	3480, 5781, 1956, 5604	17
neuron development	4	22.22	0.0539	3480, 5781, 1956, 5604	17
perinuclear region of cytoplasm	4	22.22	0.0587	5337, 5604, 3309, 4067	16
protein tyrosine phosphatase activity	3	16.67	0.0597	5781, 5604, 1845	18
insulin receptor binding	3	16.67	0.0597	3480, 5781, 6464	18
Insulin signaling pathway	5	27.78	0.0601	1399, 6464, 5604, 867, 3635	18
protein complex binding	5	27.78	0.0619	3480, 5781, 6464, 7534, 4067	18
nucleotide-binding	9	50.00	0.0694	3480, 2771, 1956, 1654, 4215, 5604, 56848, 3309, 4067	18
neuron differentiation	4	22.22	0.0788	3480, 5781, 1956, 5604	17
regulation of cell migration	4	22.22	0.0788	3480, 1956, 5337, 5604	17
hsa05214:Glioma	5	27.78	0.0814	3480, 5336, 1956, 6464, 5604	18
regulation of metal ion transport	3	16.67	0.0910	2771, 5336, 4067	17
gland development	3	16.67	0.0910	3480, 1399, 1956	17

Table A2.3. GO analysis for three groups of clustered attractors with period 2, complete table.

Term	Count	%	PValue	Genes	List Total
leukocyte mediated immunity	3	16.67	0.0910	7534, 4067, 3635	17
negative regulation of cell death	6	33.33	0.0910	3480, 5336, 1956, 7534, 56848, 3309	17
negative regulation of programmed cell death	6	33.33	0.0910	3480, 5336, 1956, 7534, 56848, 3309	17
nucleotide binding	9	50.00	0.0926	3480, 2771, 1956, 1654, 4215, 5604, 56848, 3309, 4067	18
purine nucleotide binding	9	50.00	0.0926	3480, 2771, 1956, 1654, 4215, 5604, 56848, 3309, 4067	18
ribonucleotide binding	9	50.00	0.0926	3480, 2771, 1956, 1654, 4215, 5604, 56848, 3309, 4067	18
purine ribonucleotide binding	9	50.00	0.0926	3480, 2771, 1956, 1654, 4215, 5604, 56848, 3309, 4067	18
SM00252:SH2	6	33.33	0.0954	1399, 5781, 5336, 6464, 4067, 3635	16
SH2 motif	6	33.33	0.0966	1399, 5781, 5336, 6464, 4067, 3635	18
atp-binding	8	44.44	0.0979	3480, 1956, 1654, 4215, 5604, 56848, 3309, 4067	18
Long-term depression	4	22.22	0.0985	3480, 2771, 5604, 4067	18
Left Cluster					
disulfide bond	21	42.00	0.0688	811, 3113, 25945, 8517, 1271, 2934, 3561, 7124, 3680, 3690, 1439, 921, 2261, 3695, 7132, 2826, 6352, 355, 2688, 5817, 4915	50
disulfide bond	21	42.00	0.0688	811, 3113, 25945, 8517, 1271, 2934, 3561, 7124, 3680, 3690, 1439, 921, 2261, 3695, 7132, 2826, 6352, 355, 2688, 5817, 4915	50

Table A2.4 GO analysis for three groups of clustered attractors with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
Right Cluster					
p38 MAPK Signaling Pathway	7	14.00	0.0651	6416, 3265, 2885, 8737, 9261, 5608, 8717	35
Long-term potentiation	6	12.00	0.0961	3265, 5579, 5578, 2033, 3845, 5530	47
Center Cluster					

Table A2.4 GO analysis for three groups of clustered attractors with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
cell fraction	9	47.37	0.0033	3480, 2771, 5781, 5336, 5337, 5604, 7534, 56848, 4067	17
positive regulation of cell motion	5	26.32	0.0087	3480, 1956, 5337, 5604, 4067	18
Fc gamma R-mediated phagocytosis	7	36.84	0.0097	1399, 5336, 5337, 5604, 56848, 4067, 3635	19
phosphoprotein	18	94.74	0.0101	5781, 1956, 6464, 3561, 4215, 5604, 867, 56848, 3309, 2771, 3480, 1399, 5336, 5337, 1654, 7534, 4067, 3635	19
regulation of cell motion	5	26.32	0.0257	3480, 1956, 5337, 5604, 4067	18
hydrolase	6	31.58	0.0278	5781, 5336, 1654, 5337, 1845, 3635	19
Neurotrophin signaling pathway	7	36.84	0.0289	1399, 5781, 5336, 6464, 4215, 5604, 7534	19
IGF-1 Signaling Pathway	4	21.05	0.0337	3480, 5781, 6464, 5604	12
MAPKKK cascade	7	36.84	0.0338	2771, 1399, 5781, 1956, 6464, 4215, 5604	18
phosphorus metabolic process	10	52.63	0.0367	3480, 2771, 5781, 1956, 6464, 4215, 5604, 4067, 1845, 3635	18
phosphate metabolic process	10	52.63	0.0367	3480, 2771, 5781, 1956, 6464, 4215, 5604, 4067, 1845, 3635	18
insoluble fraction	6	31.58	0.0374	3480, 2771, 5336, 5337, 56848, 4067	17
membrane fraction	6	31.58	0.0374	3480, 2771, 5336, 5337, 56848, 4067	17
cell projection morphogenesis	4	21.05	0.0400	3480, 5781, 1956, 5604	18
cell part morphogenesis	4	21.05	0.0400	3480, 5781, 1956, 5604	18
neuron projection development	4	21.05	0.0400	3480, 5781, 1956, 5604	18
neuron projection morphogenesis	4	21.05	0.0400	3480, 5781, 1956, 5604	18
positive regulation of cell migration	4	21.05	0.0400	3480, 1956, 5337, 5604	18
SH2 domain	7	36.84	0.0418	1399, 5781, 5336, 6464, 867, 4067, 3635	19
phosphatase activity	4	21.05	0.0517	5781, 5604, 1845, 3635	19
Sprouty regulation of tyrosine kinase signals	4	21.05	0.0543	1956, 6464, 5604, 867	12
regulation of hormone levels	3	15.79	0.0558	5781, 7534, 4067	18
cell activation during immune response	3	15.79	0.0558	5336, 7534, 4067	18
regulation of calcium ion transport	3	15.79	0.0558	2771, 5336, 4067	18
leukocyte activation during immune	3	15.79	0.0558	5336, 7534, 4067	18

Table A2.4 GO analysis for three groups of clustered attractors with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
response					
cell morphogenesis	4	21.05	0.0636	3480, 5781, 1956, 5604	18
neuron development	4	21.05	0.0636	3480, 5781, 1956, 5604	18
cellular component morphogenesis	4	21.05	0.0636	3480, 5781, 1956, 5604	18
positive regulation of locomotion	4	21.05	0.0636	3480, 1956, 5337, 5604	18
insulin receptor binding	3	15.79	0.0667	3480, 5781, 6464	19
protein tyrosine phosphatase activity	3	15.79	0.0667	5781, 5604, 1845	19
perinuclear region of cytoplasm	4	21.05	0.0699	5337, 5604, 3309, 4067	17
Insulin signaling pathway	5	26.32	0.0730	1399, 6464, 5604, 867, 3635	19
intracellular signaling cascade	12	63.16	0.0750	3480, 2771, 1399, 5781, 5336, 1956, 5337, 6464, 4215, 5604, 4067, 3635	18
protein complex binding	5	26.32	0.0751	3480, 5781, 6464, 7534, 4067	19
neuron differentiation	4	21.05	0.0923	3480, 5781, 1956, 5604	18
regulation of cell migration	4	21.05	0.0923	3480, 1956, 5337, 5604	18
nucleotide-binding	9	47.37	0.0978	3480, 2771, 1956, 1654, 4215, 5604, 56848, 3309, 4067	19
Glioma	5	26.32	0.0981	3480, 5336, 1956, 6464, 5604	19
Left Cluster					
leukocyte homeostasis	5	10.20	0.0932	8517, 355, 2176, 207, 836	49

Table A2.5. GO analysis for three group of sub-cluster of attractors with period 2, complete table.

Term	Count	%	PValue	Genes	List Total
Right Cluster					
signal	6	50.00	0.1440	3560, 2263, 2253, 3309, 7046, 2260	12
signal peptide	6	50.00	0.1440	3560, 2263, 2253, 3309, 7046, 2260	12
magnesium	4	33.33	0.1692	4215, 2263, 7046, 2260	12
Center Cluster					
SH3 domain binding	2	66.67	0.1453	867, 3635	3
Left Cluster					
cell fraction	7	58.33	0.0673	2771, 5781, 5336, 5337, 5604, 56848, 4067	11
positive regulation of molecular function	7	58.33	0.0792	2771, 5781, 5336, 2247, 1956, 5604, 56848	12

Table A2.5. GO analysis for three group of sub-cluster of attractors with period 2, complete table.

Term	Count	%	PValue	Genes	List Total
regulation of phosphorylation	7	58.33	0.0792	2771, 5781, 2247, 1956, 5604, 56848, 4067	12
regulation of phosphate metabolic process	7	58.33	0.0792	2771, 5781, 2247, 1956, 5604, 56848, 4067	12
regulation of phosphorus metabolic process	7	58.33	0.0792	2771, 5781, 2247, 1956, 5604, 56848, 4067	12

Table A2.6. GO analysis for three group of sub-cluster of attractors with period 4, complete table.

Term	Count	%	P-Value	Genes	List Total
Right Cluster					
cytoplasm	6	46.15	0.2116	5781, 5337, 6464, 867, 3312, 3635	13
phosphatase activity	4	30.77	0.2162	5781, 5604, 1845, 3635	13
compositionally biased region:Pro-rich	4	30.77	0.2162	6464, 5604, 867, 3635	13
cellular component morphogenesis	4	30.77	0.2391	3480, 5781, 1956, 5604	13
cell projection morphogenesis	4	30.77	0.2391	3480, 5781, 1956, 5604	13
cell morphogenesis	4	30.77	0.2391	3480, 5781, 1956, 5604	13
neuron projection morphogenesis	4	30.77	0.2391	3480, 5781, 1956, 5604	13
cell part morphogenesis	4	30.77	0.2391	3480, 5781, 1956, 5604	13
GO:0044057~regulation of system process	4	30.77	0.2391	2771, 5781, 1956, 5604	13
Center Cluster					
receptor	3	100.00	0.0598	2885, 3560, 7046	3
Left Cluster					
fibroblast growth factor receptor signaling pathway	4	36.36	0.1039	2247, 2263, 2253, 2260	10
nucleoside binding	7	63.64	0.1988	1654, 4215, 2263, 56848, 3309, 4067, 2260	11
adenyl ribonucleotide binding	7	63.64	0.1988	1654, 4215, 2263, 56848, 3309, 4067, 2260	11
adenyl nucleotide binding	7	63.64	0.1988	1654, 4215, 2263, 56848, 3309, 4067, 2260	11
ATP binding	7	63.64	0.1988	1654, 4215, 2263, 56848, 3309, 4067, 2260	11
purine nucleoside binding	7	63.64	0.1988	1654, 4215, 2263, 56848, 3309, 4067, 2260	11
atp-binding	7	63.64	0.1988	1654, 4215, 2263, 56848, 3309, 4067, 2260	11

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
29760	B-cell linker	leukocyte differentiation lymphocyte differentiation B cell differentiation
912	CD1d molecule	leukocyte differentiation lymphocyte differentiation T cell differentiation
916	CD3e molecule, epsilon (CD3-TCR complex)	leukocyte differentiation lymphocyte differentiation T cell differentiation T cell differentiation in the thymus
920	CD4 molecule	leukocyte differentiation lymphocyte differentiation T cell differentiation
958	CD40 molecule, TNF receptor superfamily member 5	Th1/Th2 Differentiation
923	CD6 molecule	T-cell differentiation
1387	CREB binding protein	negative regulation of cell differentiation stem cell differentiation
355	Fas (TNF receptor superfamily, member 6)	leukocyte differentiation lymphocyte differentiation T cell differentiation T cell differentiation in the thymus regulation of lymphocyte differentiation regulation of myeloid cell differentiation
3717	Janus kinase 2	cell morphogenesis involved in differentiation myeloid cell differentiation neuron differentiation positive regulation of cell differentiation cell morphogenesis involved in neuron differentiation
4254	KIT ligand	regulation of myeloid leukocyte differentiation positive regulation of myeloid leukocyte differentiation neural crest cell differentiation positive regulation of cell differentiation regulation of melanocyte differentiation positive regulation of melanocyte differentiation regulation of myeloid cell differentiation positive regulation of myeloid cell differentiation mesenchymal cell differentiation regulation of pigment cell differentiation positive regulation of pigment cell differentiation
5747	PTK2 protein tyrosine kinase 2	cell morphogenesis involved in differentiation regulation of cell morphogenesis involved in differentiation central nervous system neuron differentiation

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
		neuron differentiation negative regulation of cell differentiation regulation of neuron differentiation cell morphogenesis involved in neuron differentiation
7076	TIMP metallopeptidase inhibitor 1	myeloid cell differentiation erythrocyte differentiation
7189	TNF receptor-associated factor 6	leukocyte differentiation myeloid leukocyte differentiation myeloid cell differentiation myeloid dendritic cell differentiation
7520	X-ray repair complementing defective repair in Chinese hamster cells 5 (double-strand-break rejoining)	positive regulation of cell differentiation stem cell differentiation hemopoietic stem cell differentiation
1002	cadherin 4, type 1, R-cadherin (retinal)	cell morphogenesis involved in differentiation regulation of cell morphogenesis involved in differentiation neuron differentiation positive regulation of cell differentiation regulation of neuron differentiation cell morphogenesis involved in neuron differentiation
811	calreticulin	negative regulation of cell differentiation regulation of neuron differentiation negative regulation of neuron differentiation
1499	catenin (cadherin-associated protein), beta 1, 88kDa	cell morphogenesis involved in differentiation leukocyte differentiation regulation of myeloid leukocyte differentiation negative regulation of myeloid leukocyte differentiation glial cell differentiation lymphocyte differentiation T cell differentiation regulation of epithelial cell differentiation positive regulation of epithelial cell differentiation regulation of chondrocyte differentiation negative regulation of chondrocyte differentiation T cell differentiation in the thymus muscle cell differentiation myoblast differentiation negative regulation of cell differentiation positive regulation of cell differentiation regulation of myeloid cell differentiation negative regulation of myeloid cell differentiation regulation of osteoblast differentiation positive regulation of osteoblast differentiation regulation of osteoclast differentiation negative regulation of osteoclast differentiation mesenchymal cell differentiation
6352	chemokine (C-C motif) ligand 5	regulation of myeloid leukocyte differentiation positive regulation of myeloid leukocyte differentiation positive regulation of cell differentiation regulation of myeloid cell differentiation

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
		positive regulation of myeloid cell differentiation regulation of osteoclast differentiation positive regulation of osteoclast differentiation
4261	class II, major histocompatibility complex, transactivator	CD4 T cell differentiation
1437	colony stimulating factor 2 (granulocyte-macrophage)	leukocyte differentiation myeloid leukocyte differentiation regulation of foam cell differentiation positive regulation of foam cell differentiation myeloid cell differentiation myeloid dendritic cell differentiation positive regulation of cell differentiation
1147	conserved helix-loop-helix ubiquitous kinase	leukocyte differentiation myeloid leukocyte differentiation myeloid cell differentiation osteoclast differentiation
2247	fibroblast growth factor 2 (basic)	glial cell differentiation positive regulation of cell differentiation
3635	inositol polyphosphate-5-phosphatase, 145kDa	regulation of myeloid leukocyte differentiation negative regulation of myeloid leukocyte differentiation regulation of granulocyte differentiation negative regulation of granulocyte differentiation regulation of B cell differentiation positive regulation of B cell differentiation negative regulation of cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation regulation of myeloid cell differentiation negative regulation of myeloid cell differentiation positive regulation of myeloid cell differentiation regulation of erythrocyte differentiation positive regulation of erythrocyte differentiation regulation of monocyte differentiation negative regulation of monocyte differentiation regulation of neutrophil differentiation negative regulation of neutrophil differentiation regulation of osteoclast differentiation negative regulation of osteoclast differentiation
3688	integrin, beta 1 (fibronectin receptor, beta polypeptide, antigen CD29 includes MDF2, MSK12)	leukocyte differentiation lymphocyte differentiation B cell differentiation cardiac cell differentiation muscle cell differentiation negative regulation of cell differentiation striated muscle cell differentiation cardiac muscle cell differentiation
3561	interleukin 2 receptor, gamma (severe combined immunodeficiency)	alpha-beta regulatory T cell differentiation alpha-beta regulatory T cell differentiation

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
		regulation of T cell differentiation in the thymus positive regulation of T cell differentiation in the thymus alpha beta T cell differentiation alpha beta T cell differentiation regulation of B cell differentiation positive regulation of B cell differentiation regulation of T cell differentiation positive regulation of T cell differentiation regulation of regulatory T cell differentiation positive regulation of regulatory T cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation regulation of alpha-beta T cell differentiation positive regulation of alpha-beta T cell differentiation
3572	interleukin 6 signal transducer (gp130, oncostatin M receptor)	positive regulation of cell differentiation regulation of osteoblast differentiation positive regulation of osteoblast differentiation
3575	interleukin 7 receptor	leukocyte differentiation lymphocyte differentiation T cell differentiation regulation of T cell differentiation in the thymus positive regulation of T cell differentiation in the thymus regulation of T cell differentiation positive regulation of T cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation
3727	jun D proto-oncogene	osteoblast differentiation positive regulation of cell differentiation regulation of osteoblast differentiation positive regulation of osteoblast differentiation
3952	leptin	sex differentiation central nervous system neuron differentiation neuron differentiation positive regulation of cell differentiation regulation of myeloid cell differentiation positive regulation of myeloid cell differentiation female sex differentiation
3932	lymphocyte-specific protein tyrosine kinase	leukocyte differentiation lymphocyte differentiation T cell differentiation
5594	mitogen-activated protein kinase 1	negative regulation of cell differentiation
1432	mitogen-activated protein kinase 14	chondrocyte differentiation positive regulation of cell differentiation regulation of myeloid cell differentiation positive regulation of myeloid cell differentiation regulation of erythrocyte differentiation positive regulation of erythrocyte differentiation

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
5601	mitogen-activated protein kinase 9	regulation of foam cell differentiation positive regulation of foam cell differentiation positive regulation of cell differentiation
5604	mitogen-activated protein kinase kinase 1	epidermal cell differentiation neuron differentiation keratinocyte differentiation epithelial cell differentiation positive regulation of cell differentiation
4790	nuclear factor of kappa light polypeptide gene enhancer in B-cells 1	regulation of foam cell differentiation positive regulation of foam cell differentiation positive regulation of cell differentiation
5295	phosphoinositide-3-kinase, regulatory subunit 1 (alpha)	leukocyte differentiation lymphocyte differentiation B cell differentiation
5336	phospholipase C, gamma 2 (phosphatidylinositol-specific)	mature B cell differentiation during immune response follicular B cell differentiation mature B cell differentiation leukocyte differentiation lymphocyte differentiation B cell differentiation
5818	poliovirus receptor-related 1 (herpesvirus entry mediator C)	cell morphogenesis involved in differentiation neuron differentiation cell morphogenesis involved in neuron differentiation
5788	protein tyrosine phosphatase, receptor type, C	leukocyte differentiation lymphocyte differentiation T cell differentiation T cell differentiation in the thymus regulation of B cell differentiation regulation of T cell differentiation positive regulation of T cell differentiation regulation of gamma-delta T cell differentiation positive regulation of gamma-delta T cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation
387	ras homolog gene family, member A	regulation of cell morphogenesis involved in differentiation negative regulation of cell differentiation positive regulation of cell differentiation regulation of neuron differentiation negative regulation of neuron differentiation positive regulation of neuron differentiation
6696	secreted phosphoprotein 1	osteoblast differentiation regulation of cell morphogenesis involved in differentiation negative regulation of cell differentiation regulation of neuron differentiation
6776	signal transducer and activator of transcription 5A	natural killer cell differentiation leukocyte differentiation regulation of myeloid leukocyte differentiation

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
		positive regulation of myeloid leukocyte differentiation sex differentiation lymphocyte differentiation T cell differentiation regulation of epithelial cell differentiation regulation of natural killer cell differentiation positive regulation of natural killer cell differentiation T cell differentiation in the thymus regulation of B cell differentiation positive regulation of B cell differentiation regulation of T cell differentiation positive regulation of T cell differentiation regulation of gamma-delta T cell differentiation positive regulation of gamma-delta T cell differentiation negative regulation of cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation regulation of myeloid cell differentiation negative regulation of myeloid cell differentiation positive regulation of myeloid cell differentiation regulation of erythrocyte differentiation negative regulation of erythrocyte differentiation female sex differentiation male sex differentiation regulation of mast cell differentiation positive regulation of mast cell differentiation
6850	spleen tyrosine kinase	leukocyte differentiation lymphocyte differentiation T cell differentiation regulation of B cell differentiation positive regulation of B cell differentiation regulation of T cell differentiation positive regulation of T cell differentiation regulation of gamma-delta T cell differentiation positive regulation of gamma-delta T cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation alpha-beta T cell differentiation regulation of alpha-beta T cell differentiation positive regulation of alpha-beta T cell differentiation
7046	transforming growth factor, beta receptor 1	neuron differentiation regulation of epithelial cell differentiation negative regulation of epithelial cell differentiation negative regulation of cell differentiation regulation of endothelial cell differentiation negative regulation of endothelial cell differentiation
7124	tumor necrosis factor (TNF superfamily, member 2)	leukocyte differentiation myeloid leukocyte differentiation regulation of myeloid leukocyte differentiation myeloid cell differentiation

Table A2.7. Genes related to differentiation process.

ENTREZ ID	Gene Name	Process
		osteoclast differentiation regulation of myeloid cell differentiation regulation of osteoclast differentiation
7157	tumor protein p53	leukocyte differentiation lymphocyte differentiation B cell differentiation T cell differentiation T cell differentiation in the thymus negative regulation of cell differentiation
207	v-akt murine thymoma viral oncogene homolog 1	positive regulation of cell differentiation regulation of fat cell differentiation positive regulation of fat cell differentiation
5970	v-rel reticuloendotheliosis viral oncogene homolog A (avian)	regulation of Schwann cell differentiation positive regulation of Schwann cell differentiation regulation of chondrocyte differentiation positive regulation of chondrocyte differentiation positive regulation of cell differentiation regulation of glial cell differentiation positive regulation of glial cell differentiation
4067	v-yes-1 Yamaguchi sarcoma viral related oncogene homolog	glial cell differentiation myeloid cell differentiation erythrocyte differentiation positive regulation of cell differentiation oligodendrocyte differentiation
7412	vascular cell adhesion molecule 1	leukocyte differentiation lymphocyte differentiation B cell differentiation
7535	zeta-chain (TCR) associated protein kinase 70kDa	leukocyte differentiation lymphocyte differentiation T cell differentiation T cell differentiation in the thymus regulation of T cell differentiation positive regulation of T cell differentiation positive regulation of cell differentiation regulation of lymphocyte differentiation positive regulation of lymphocyte differentiation alpha-beta T cell differentiation regulation of alpha-beta T cell differentiation positive regulation of alpha-beta T cell differentiation