2009

Computational modeling of biochemical systems using cellular automata

Advait Apte
Virginia Commonwealth University

COMPUTATIONAL MODELING OF BIOCHEMICAL SYSTEMS USING CELLULAR AUTOMATA

A Dissertation submitted in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY at Virginia Commonwealth University.

by

ADVAIT ASHOK APTE
MASTER OF SCIENCE, VIRGINIA COMMONWEALTH UNIVERSITY, 2006
BACHELOR OF ENGINEERING, UNIVERSITY OF MUMBAI, INDIA, 2004

Director: STEPHEN S. FONG
ASSISTANT PROFESSOR CHEMICAL AND LIFE SCIENCE ENGINEERING

Virginia Commonwealth University
Richmond, Virginia
DECEMBER 2009

Acknowledgement

Four and half years and 103 credits ago I set out on a mission to become a scientist. Four and half years ago in the cool and comforting breeze of fall '05 I was searching for people who can show faith in me and give me the directions to complete this mission. I was quite fortunate to find right people at the right time in the right places. In a new and unknown world where losing track and getting lost was quite a possible destiny, these wonderful people helped me to stay on course with their continued guidance and invaluable advice. If I engage myself in creating a list of these friendly counselors and their counsels, a document bigger than this one would be required to be written. Hence I would mention a chosen few and leave the rest in a special place in my heart. To start with, I would like to sincerely thank Dr. Fong without whom this dream would have been unfulfilled. His continued support and encouragement to try new and different things without the fear of failure, is the single most factor responsible for such a diversification of applications of my research and broadening my perspective. I consider myself fortunate to be advised by him.

A special section of acknowledgements is reserved for Dr. Bonchev, who along with Dr. Kier introduced me to cellular automata and gave me right perspective into the world of solving complex biological problems with CA modeling technique. He has a lion's share in guiding my research work.

I would like to thank Dr. Cain, Dr. Kier, Dr. Peters and Dr. Yadavalli for being on my Ph.D. dissertation committee and taking keen interest in evaluating my research work.

My lab mates, Chris, Yu, Oscar, Seth, George and Niti, thank you guys for all your help over the years. All of you maintained the lab environment one of the most informal, productive and enjoyable.

I would like to express my special thanks to Sheena. The scope of your support and encouragement is beyond this acknowledgement so I will just say thank you for anything and everything.

Many thanks to Yonathan, the guys who is been around and taught me how to go around.

And what should I say about my parents? You were there for me every time I needed you throughout my life, caring, supporting and encouraging. You gave me the strength to overcome my failures without losing hope. You gave me the confidence to rise from these failures to achieve success. Unfortunately there is no word in English diction to accurately capture my emotions on how I feel about you, so I will just say…love you mom, love you dad.

To my parents

Anuradha Apte

&

Ashok Apte

who wished nothing else but to see me achieve this...

Table of Contents

List of Tables

List of Figures

Page

Abstract

COMPUTATIONAL MODELING OF BIOCHEMICAL SYSTEMS USING CELLULAR AUTOMATA

By Advait Ashok Apte, Ph.D.

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2009

Major Director: Stephen S. Fong
Assistant Professor, Chemical and Life Science Engineering

Biological systems exhibit complex behaviors through coordinated responses of individual biological components. With the advent of genome-scale techniques, one focus has been to develop methods to model interactions between components to accurately describe intact system function. Mathematical modeling techniques such as constraint-based modeling, agent-based modeling, cellular automata (CA) modeling and differential equation modeling are employed as computational tools to study biological phenomenon. We have shown that cellular automata simulations can be used as a computational tool for

predicting the dynamics of biological systems with stochastic behavior. The basic premise for the research was the observations made during a study of biologically important feed-forward motifs where CA simulations were compared with differential equation simulations. It was shown for classes of structural motifs with feed-forward architecture that network topology affects the overall rate of a process in a quantitatively predictable manner. The study which comprised of CA simulations compared with differential equation modeling show reasonable agreement in the predictability of system dynamics, which provided enough support to model biological systems at cellular level to observe dynamic system evolution. The great promise shown by CA simulations to model biochemical systems was then employed to elucidate evolutionary clues as to why biological networks show preference for certain types of motifs and preserve them with higher frequency during evolution. It was followed by modeling apoptotic networks to shed light on the efficacy of inhibitors and to model cellulose hydrolysis to evaluate efficiency of different enzyme systems used by cellulytic bacteria.

CHAPTER 1 INTRODUCTION

Over the past several years, systems biology has come up as a promising tool to help us understand the underlying principle of biological processes. This advance of systems biology was made possible by the invention of high throughput techniques like next generation sequencing [1-6] which have increased the ability to study highly complexed and otherwise complicated systems with ease. To understand the functions of intact biological systems it has become more important to study the organisms and their biological processes on systems level along with studying the isolated characteristics of individual components. A fundamental step in systems biology is to derive appropriate mathematical models for the purposes of analysis and design. For example, to synthesize a gene regulatory network, the derivation of a mathematical model is important in order to carry out *in silico* investigations of the network dynamics and to investigate parameter variations and robustness issues [7]. One of the most important areas of research in this regard is studying the network interactions between different metabolic, regulatory and signaling networks. Traditionally computational modeling is used to supplement experimental research. These days these models of interacting networks are gaining more significance in every area of science due to their increased accuracy and predictability and can be used to drive experimental work.

Essentially a model is an abstract representation of the system. It depicts the most important and interesting features of a real system. A computational model attempts to study complex behaviors using computer simulations by translating the physical phenomena into a system of equations which governs it. Currently there are many modeling approaches which include more traditional methods like differential equation modeling and few nontraditional ones like stoichiometric modeling and cellular automata or agent based modeling. As current dynamic models of biological systems often are developed with a tradeoff between detail and scalability, it becomes important to know the strength and weaknesses of each type of modeling in order to make correct choice of method to model the system under consideration. The major modeling approaches include differential equations modeling (DE), stoichiometric or constraint-based modeling and cellular automata (CA) or agent based modeling (ABM), though some hybrid approaches are also suggested and implemented [8-12].

1.1 Computational Modeling Methods Overview

The three major modeling methods described above will be discussed in this section stating their methodology and usefulness in modeling different types of systems.

1.1.1 Differential equation modeling (DE)

Differential equation modeling is one of the most commonly used methods of modeling biochemical reactions. To model these systems by ordinary differential equations (ODE), one needs to know the kinetic constants for a particular reaction. Only forward rate constants are required in case of reaction proceeding in only one direction while reverse rate constants are needed in case of reversible reactions. Using the kinetic rate constants and with the knowledge of interactions between different biochemical entities, a biochemical reaction can be converted in to differential rate equation. Such a system of differential equations, consisting one differential equation for each transformation, can represent the whole system. This system of differential equations can be solved using concentrations of its components, analytically, in case of linear systems or numerically, in case of nonlinear systems to evaluate overall system behavior at any given point in time. Different approaches like quasi-steady-state assumption and piece-wise linear approximations [13] are proposed to reduce the number of equations and simplify the system of equations. While using ODEs, the concentrations of different molecules are represented by continuous time variables but discrete time models can also be used to study biochemical networks [14, 15]. The discrete time approach is especially deemed useful to model systems consisting dynamics on different temporal scales [14]. Linear differential equations fail to capture the true nature of nonlinearity in a system generated by different interacting components. Hence such a system needs to be modeled using nonlinear differential equations [16, 17]. Examples of such systems include the Michaelis-

Menten kinetics [16] involving formation of transient enzyme-substrate complexes. The above mentioned DEs are deterministic. Inclusion of stochastic quantities in DEs leads to stochastic differential equation modeling which is an emerging field of mathematics. Usefulness of such system of modeling has been shown with examples from pharmacokinetics [18], population modeling [19] and biochemical reaction systems [20]. Hence, DE modeling offers a reliable and simple method to quickly model biochemical systems. Several limitations of this type of modeling are cited. DE models neglect cells' individuality by treating them as populations thereby failing to capture complex behavior such as spatially explicit heterogeneous dynamics taking place at individual cell level [10]. Also due to the cellular complexity it is difficult to capture the aggregate behavior mathematically without considerable simplification [10, 20]. To capture this complexity in whole cell models, introduction of many non-linear terms becomes necessary, making the DE model difficult to solve.

1.1.2 Stoichiometric modeling

Stoichiometric or constraint-based models [24-27] identify the allowable solution space by balancing internal fluxes of the metabolic network of an organism. This allowable solution space then can be searched using linear programming for solutions of interest, depending upon a well defined objective. The objective could be maximum growth on specific source or maximum production of a particular by-product [21-23]. Construction of a constraint-based model is dependent upon the metabolic fluxes of an organism. Metabolic network information such as metabolite concentrations, input-output fluxes, growth substrates and growth rates are required for building a comprehensive metabolic model. The metabolic fluxes are written as flux balance equations which is essentially a large scale mass balance of the entire metabolic system. Based on these reactions, a stoichiometric matrix can be formed such that $A = [aij]$, where $[ai]j$ is the concentration of j^{th} metabolite with sign, consumed (−) or produced (+), in i^{th} reaction. The model is then written in form of matrix equation and solved for optimizing a particular objective [24-29]. This method of modeling gives exact account of the metabolic capabilities of an organism. In this type of modeling method, models are often represented with set of linear algebraic equations which are easy to compute. Hence it can be scaled to model whole cell metabolism with reasonable computing time. It also eliminates the necessity to know all the kinetic constants for thousands of reactions while modeling whole cell metabolism. The usefulness of this modeling method has been shown to model *E.coli*. [30-32], *Saccharomyces cerevisiae* [33-37] and human [38] metabolic network reconstruction. On the negative side, this approach cannot be used to model dynamics of different cell processes, as it assumes a steady state for its calculations. It is not possible to model evolution of a system over time due to lack of the interactions of different dynamic components of the system.

1.1.3 Cellular Automata (CA) / Agent Based Modeling (ABM)

Cellular automata (CA) are a modeling tool, which presents dynamic systems as discrete in space, time, and state [40]. It has four basic components [41]:

- Grid
- Components
- Rules governing the behavior of components
- Initial conditions

A grid represents the simulation environment which is divided into cells. In biological terms, a grid can be thought of as the biological cell that contains proteins, organelles, metabolites etc. These biological entities make the components of the system. They are also referred to as agents giving the name, agent based modeling. Like the enzymes and metabolites, these agents move inside the grid. The movement and interactions between different agents are defined by rules governing the system. The rules represent the biochemical interactions between different metabolites, proteins and enzymes. These interactions are governed by probabilities which define the efficacy of one component separating from, joining with or interacting with other components. The movement of components in the grid is random and they can move to one of the adjoining sites, their von Neumann neighborhood, with equal probability. An abstraction of this process can be seen in Figure 1.

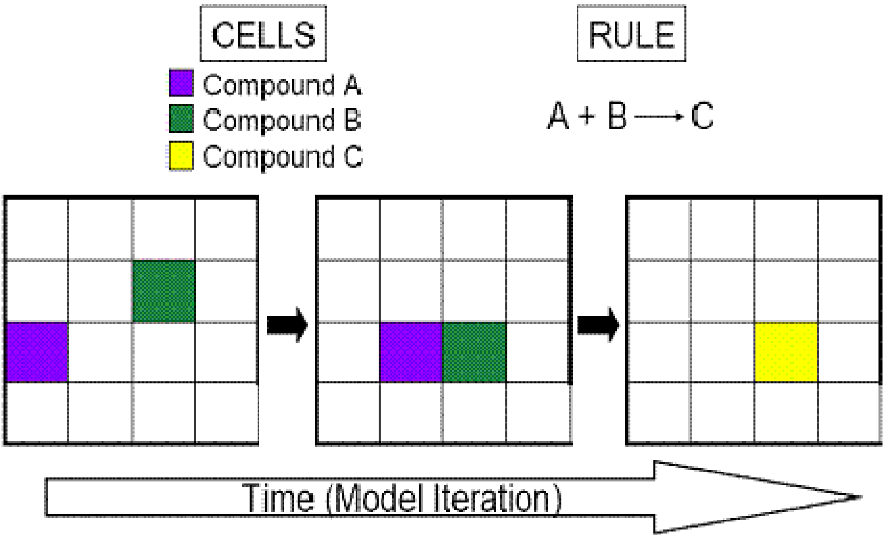


Figure 1. Graphical illustration of a CA simulation grid. The elements A and B while moving over a N x N grid follow the simple deterministic rule A + B = C to produce C given both occupy neighboring sites To illustrate the rules or local relationships governing the system, following simple biochemical equation can be converted to a CA system as shown in Figure 2

$$E+S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E+P$$

Hence a CA model, starting from initial conditions iterates over the simulation for specified number of repetitions. Similar to the above mentioned kinetics of enzyme mediated biochemical reaction, other biochemical reactions can be converted to CA for building the rules of CA system.

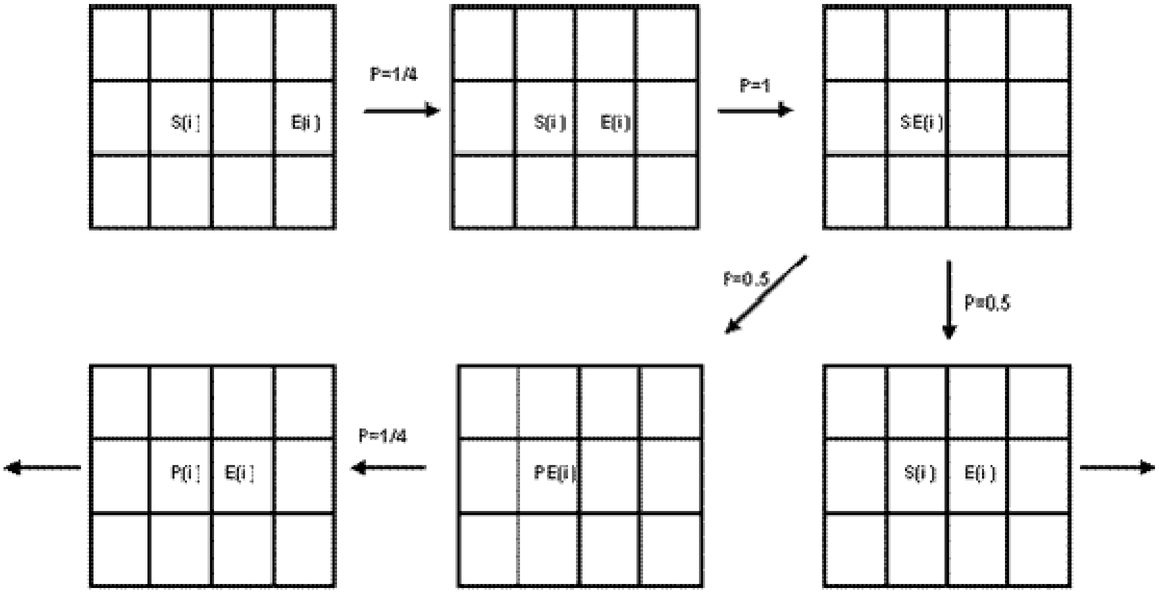


Figure 2. Cellular automata representation of a simple biochemical reaction mediated by enzyme E, where substrate S is converted into the product P [16].

This and similar types of rule-based modeling have been of great use in capturing the spatial dynamics of a system. It has been successfully implemented in simulating biological systems [10, 42-45], biochemical systems [16, 46-49], and physical systems [50-52], to name a few. Primary motivations for using cellular automata to model biochemical systems are as follows.

1. Flexibility: CA offers the possibility for the components to assume characteristics and behavior of any real biological entity, defining relationships between components by assigning probabilities to these interactions. These probabilities of breaking, joining and transition defined for each pair of components will dictate the possibility of the biochemical reactions taking place. Hence it offers flexibility to model biological systems at any level starting from interactions between molecules,

metabolites, proteins, and regulatory, metabolic and signaling networks, individual cells and even on organ level.

2. Dynamic nature: CA is a dynamic method of simulating biological systems. Unlike other modeling systems, CA represents a biological system with its components actually moving, resembling real biological entities. It endorses individual characteristics to these biological entities rather than treating them as populations. Hence the dynamics of the system become more realistic, eliminating the need for assumptions like quasi steady state.
3. Spatial considerations: Other modeling methods fail to consider the spatial effects by not having the components of the system interact in space. CA models address this issue by having a simulation environment mimicking the real biological cellular "space" where effects of physicality are taken into consideration while simulating biological systems. This can have profound effects on the evolution of system over time, e.g., density of the simulation environment. A sparsely populated environment will make the movement of the system components far easier than a densely packed surrounding. Such situations can significantly change the behavior of biological systems where different entities like substrates and enzymes need to "find" each other before the reaction can take place. Such considerations are possible while modeling a system using CA.
4. Statistically quantitative: CA model offers statistically quantitative information about the system as the biological system under consideration is simulated many times to average out the effects of nonlinearity.

5. Emergent behavior: One of the strengths of CA modeling is its ability to show emergent behavior of a system over time. The interactions between different components of the system can give rise to unpredicted phenomena which are otherwise counter-intuitive and impossible to decipher by traditional modeling methods.
6. Reduced complexity of modeling: To model a system using CA, one only needs to know about the interactions between different components of the system. The need for reaction rate constants is eliminated due to probabilistic nature of CA modeling. Each and every interaction can be defined using probability Complex and nonlinear events such as movement of cellular components and formation and dissociation of enzyme-substrate complexes can be modeled by assigning predefined probability for these interactions.

Some disadvantages of CA modeling do exist.

1. Computationally intensive: Due to the fact that CA simulations require multiple runs to calculate the standard deviation for each simulations in an effort to average out deviations, considerable amount of time is required as the complexity of the system increases. This can be attributed to the need to move hundreds of components simultaneously and execute the rules of interactions for each component of the system individually.
2. No structural considerations: With CA it is almost impossible to give structural considerations to the variety and details of components such as proteins and metabolites. CA has a rather non-structural approach towards modeling individual

structures as it treats or assumes fairly simple and basic structures e.g. square for components. Though the facility to treat each edge of this square differently to mimic the presence and absence of active sites on the proteins exists, it can not come close to what molecular modeling methods have to offer.

3. Scalability: Following on the computationally intensive nature of CA simulations, current computational infrastructure does not allow scaling a CA model to whole-cell level. Even though thousands of different reactions and reactants can be defined due to availability of whole genome information, it will be difficult to obtain any useful information or behavior due to overwhelming computational memory requirements.

These advantages of CA modeling have made it the most suitable approach to model different biological systems presented in this dissertation.

1.2 Dissertation overview

The work included in this dissertation can be described as exploratory and method centric research. It is exploratory because it started with the use of cellular automata as a tool to explore the dynamics of small 3 to 4 node motifs in an effort to build a library of dynamic characteristic of such motifs which can be used later to build larger and more complex network motifs, dynamics for which can be predicted using the small motifs it is made up of. This search led to the development of cellular automata as a method to simulate biochemical networks to decipher their dynamics. This research is method-centric because it follows the full life cycle of method development process, starting with development itself, followed by method validation and finally leading to the method's real world applications. The method in discussion here is Cellular Automata, or CA, to simulate biochemical reaction networks and most of the research presented in this dissertation revolves around this notion. Chapter 2 discusses the process of method development, validation and application of CA to model small node motifs. Chapters 3 and 4 highlight some of the strengths of CA modeling when applied to real-world problems in biology. The detailed development of cellular automata as a computational tool is described is chapter 2. In the first half of the same chapter, CA is established as a useful and valid tool for such simulations. The concept of isodynamicity, which was discovered in this study, was then taken further and has formed the basis for the later half of the chapter. In this chapter, different patterns among small node motifs were identified, and higher prevalence of faster and more robust motifs in the process of evolution of organisms was proposed and shown using different organizational levels of interacting pathways. The usefulness of CA was shown again in modeling FAS-L activated apoptosis in cells, which is the subject of chapter 3 of this dissertation. Strategies like dual inhibition of apoptosis inhibitors, modeled using CA showed agreement with published experimental work and proved the potential of using CA for prediction of cancer drug targets. In chapter 4, agent based modeling (ABM), was used to model cellulose hydrolysis by cellulytic organisms. Promising results showing the dependence of particular cellulase system's efficiency on underlying cellulases of the enzyme systems were obtained. Chapter 5 shows the process of algorithm development targeted at detecting mutations in organisms grown under different conditions and for several generations. This work was not based on CA but an individual project in itself, hence categorized as a separate chapter. The research work from past four and half years was summarized and conclusions are presented in chapter 6.

CHAPTER 2 CELLULAR AUTOMATA SIMULATION OF TOPOLOGICAL EFFECTS ON THE DYNAMICS OF NETWORK MOTIFS

2.1 Introduction

The present study focuses on the qualitative and quantitative characterization of a variety of network motifs with different architecture, and the relationship between topology and overall process rate in network motifs. As described in the introduction, a variety of stochastic simulation methods exist that can be used to model networks dynamics [53-56]. In this study, cellular automata [57] were selected as a promising method for dynamic modeling of all possible topologies of network motifs due to their flexibility, robustness and accuracy. Widely applied in a variety of areas of science and technology, the cellular automata method recently showed great promise for modeling dynamics of complex biological systems [47,48, 51 58, 59]. In modeling biochemical processes, the general method used by Kier and Cheng was followed [48, 52]. Specifically, this study examines the purely topological effects on motif productivity. Thus, the cellular automata simulations were conducted in a manner so as to keep all probabilistic parameters constant.

This effectively "freezes" the process stochasticity, making these simulations *de facto* non-stochastic. Results of cellular automata topodynamic pattern simulations were verified with those obtained by parallel ODE and non-linear differential equations simulations.

2.2 Methods

2.2.1 Parameter selection for simulations of three node motifs

The study of motif topology started with the simulations of several three and four node network motifs with different topologies using CA to gain understanding of their dynamics. Starting from relatively simple structures, more complex motifs which contain feedback, feed-forward and bi-parallel edges were studied. A series of 22 different structures having three nodes was simulated. Network structures with four nodes include simulations of 32 linear, 20 star, 28 cycles, 21 branched cycles, and 20 bi-cyclic and 25 tri-cyclic. Over all, I simulated 7 different series and 168 different network structures using CA. All the network structures are simulated using five different sets of interaction probabilities, 0.01, 0.03, 0.09, 0.27 and 0.81 which have a scaling factor of 3.The matrix size of CA grid used is 100 X 100. Different amounts of conversion from initial concentration to final concentration used for comparison range from 25 % to 100 %, depending on the motif conformation. The speed of conversion is measured in number of iterations required for conversion from source node to target node.

These small motifs were analyzed by measuring the effect of adding/removing edges or changing topology on the number of iterations required for S→ T conversion. Out of the 7 series mentioned above, detailed analysis of structures with 3 nodes was performed to evaluate if dynamics of these small motifs can be predicted using CA. The motifs in this series were classified in two different types, one input - two outputs (S1T2) and two inputs - one output (S2T1). The starting concentration of inputs or "substrates" was taken as 100 CA cells and edges or "enzymes" as 20. All other intermediate nodes had initial concentration of 0. Analysis was performed by choosing two arbitrary interaction probabilities, 0.09 and 0.81. The grid size of CA matrix is kept constant throughout the simulations at 100 X 100. These structures are broadly divided into three main patterns, adding a reverse edge, adding a forward edge or feed-forward and adding a backward edge or feed-back. The 22 structures were categorized in above mentioned patterns and compared on the basis of difference in number of iterations, where a positive value indicates acceleration while a negative being deceleration.

2.2.2 Parameter selection for simulations of feed-forward motifs

To generate accurate and reproducible results, several series of tests were performed to optimize the basic CA parameters to be used before proceeding with the detailed study on feed-forward motifs. Due to the statistical nature of the CA modeling, a large number of simulation runs should be performed to obtain statistically-meaningful results. The optimal number of runs must be large enough to provide reliable statistics, and at the same time not excessively large so as to minimize the computation time. The influence of the number of runs on the number of iterations needed to reach 100 % conversion of the source substrate into the target product, was examined (Table 1).

The tests were performed at different degrees of lattice occupancy (termed *lattice density*) and with different number of nodes (3, 6 and 9) in the feed-forward FFA series shown in Figure 3. Results for 50 runs differed from those obtained at 100, 250 and 1000 runs (Table 1); the latter three were practically the same and within the range of standard deviation observed. Therefore, 100 runs were selected to perform the basic feed-forward modeling.

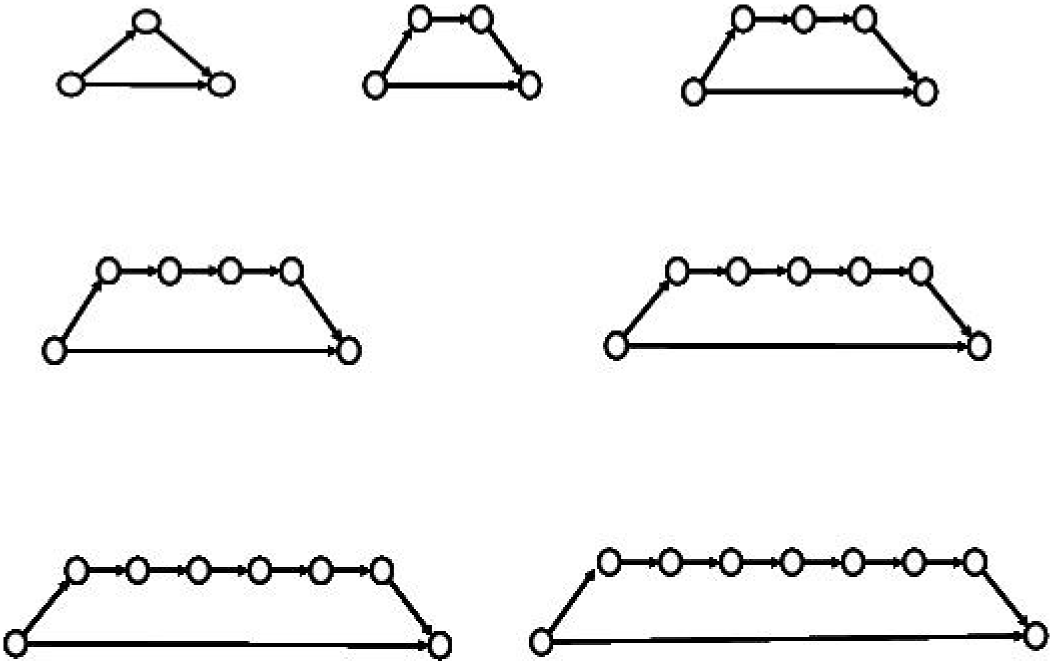


Figure 3. The members of the examined primary feed-forward motif series FFA.

Table 1

Cellular Automata simulation of the dynamics* of feed-forward series FFA shown in Fig. 4

Number of Nodes	Lattice Density %	Number of Runs				SD** average	SD %
		50	100	250	1000		
3	3.6	447	469	458	460	7	1.51
	10	143	145	144	144	8	5.56
	60	21	20	21	20	8	39.0
6	3.6	939	895	893	906	8	0.89
	10	287	293	290	288	10	3.45
	60	41	42	41	41	9	21.6
9	3.6	1430	1425	1426	1422	2	0.14
	10	458	460	456	458	2	0.44
	60	65	65	65	65	9	13.9

* Dynamics is expressed as the average number of iterations needed for 100% conversion of 100 source substrate cells into the target product. It is modeled at different number of nodes, different lattice densities, and different number of cellular automata runs.

** Standard deviation

Another parameter investigated was the CA lattice density (the number of cells per unit area). A high density can impede the free cell motion and the results obtained in different runs could diverge considerably. That was confirmed in tests showing over 25-fold increase in the standard deviation of the number of iterations when the lattice density was increased within the series 1.0, 3.6, 5, 10, 20, 40, and 60%. On the other extreme, a very low density (See Figure 4) would unnecessarily prolong the time for attaining a steady state. For these reasons the 3.6% lattice density (corresponding to 100x100 cells lattice with a total of 360 cells occupied by substrates and enzymes) as a constant parameter for the detailed study of the dynamics of feed-forward motifs was selected. This required resizing the lattice for each of the FF motifs examined. All lattice sizes used were within the range of 100x100 to 220x220 cells.

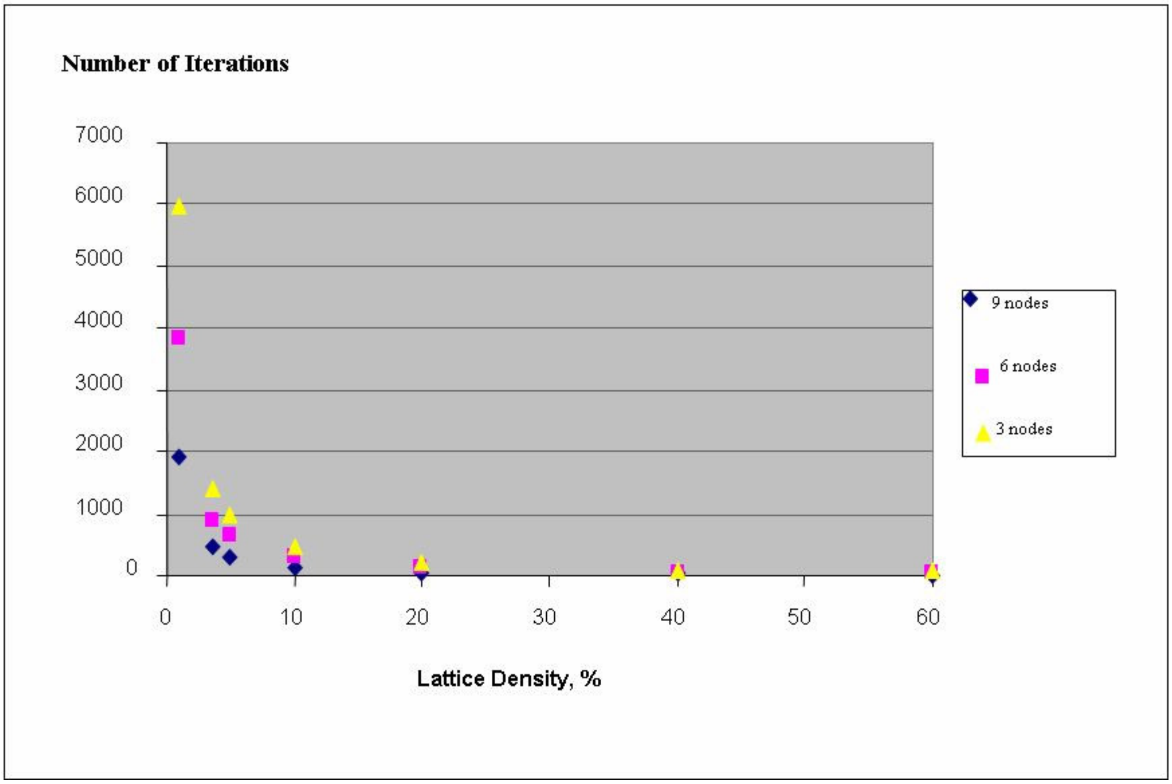


Figure 4. The overall rate of the feed-forward process of the source substrate conversion into the target product in the Feed-Forward series A (FFA) decreases rapidly with lattice density. The number of iterations, needed to attain a steady-state in Series 1, 2 and 3, correspond to the number of feed-forward nodes (Fig. 4) equal to 3, 6, and 9, respectively. All data are averaged over 100 runs simulation

More detailed simulations were run to quantitatively capture the manner in which the feed-forward dynamics vary with different lattice density, *D*, and the number of nodes, *V*, in the FFA-series. Equation (2), relating these quantities, was derived from the data of Table 1:

$$I = 640.5VD^{-1.0862} - 8.281\ln(D) + 36.08 \qquad (2)$$

The equation derived shows considerable acceleration of the FF processes with the increase in density in agreement with the law of mass action, since density is proportional to a substrate concentration modeled as number of cells per unit lattice area.

In deriving Equation (2) the linear dependence of the number of iterations on the number of FF loop nodes was used: I = aV + b, which for V = 3, 6, and 9 nodes were obtained with correlation coefficient R^2 = 0.9961, 0.9988, and 0.9998, respectively. The regressions best expressing the dependence of a and b on the lattice density *D* were

$$a = 640.5D^{-1.0862} \text{ and } b = -8.281\ln(D) + 36.08 \qquad (3)$$

with R^2 equal to 1.0000 and 0.9901, respectively.

The last parameter to select was the number of source cells (*N*). The number of cells of all other pathway constituents was kept equal to 100 cells. The enzymes associated with each of the biochemical reaction steps were kept equal to 20 cells. All enzyme activities were also kept equal (by applying the same CA probabilistic rule) in order to extract the purely topological effects on the network dynamics. Since the concentration, simulated as number of cells per unit lattice, was kept constant, *N* itself should not influence the correlation coefficient of the linear model relating the number of nodes in the pathway to the rate of the source-to-outcome conversion. As shown in Table 2, while the correlation coefficient remains within the same range, the increase in the number of source cells in the feed-forward motif reduces the model standard deviation at the cost of considerable increase in the number of iterations needed to reach a steady-state. As a reasonable compromise between a low standard deviation and too many iterations, detailed study in the next section was performed at N = 500, a parameter value that enabled us to attain steady-state with an average standard deviation of 0.31% for less than 10,000 iterations.

Table 2.

Influence of the size of the feed-forward motif on the dynamics of 100% conversion of the source substrate into target product

N*	Iterations Range	R^2	SD %
100	469 – 1425	0.9960	0.99
300	1603 – 4473	0.9994	0.76
500	2966 – 8444	0.9996	0.31
700	4435 – 13282	0.9989	0.28
900	6397 – 20526	0.9931	0.19

*N – Number of source nodes; R^2 – correlation coefficient; SD % – relative standard deviation

The feed-forward motif FFA shown in Fig. 3 may be termed the *primary feed-forward* series (PFF) to be distinguished from some more complex variations on the same feed-forward pattern. One may also consider double, triple, etc., FF motifs, which include secondary, tertiary, etc. feed-forward edges. Such structural motifs with more complex topologies may be obtained by merging two or more primary FF motifs. This approach may be of interest in predicting dynamic patterns in larger networks as combinations of well established patterns in small subnetworks (motifs). Several such complex cases of feed-forward patterns of PFFs (Fig. 5) were selected for this study.

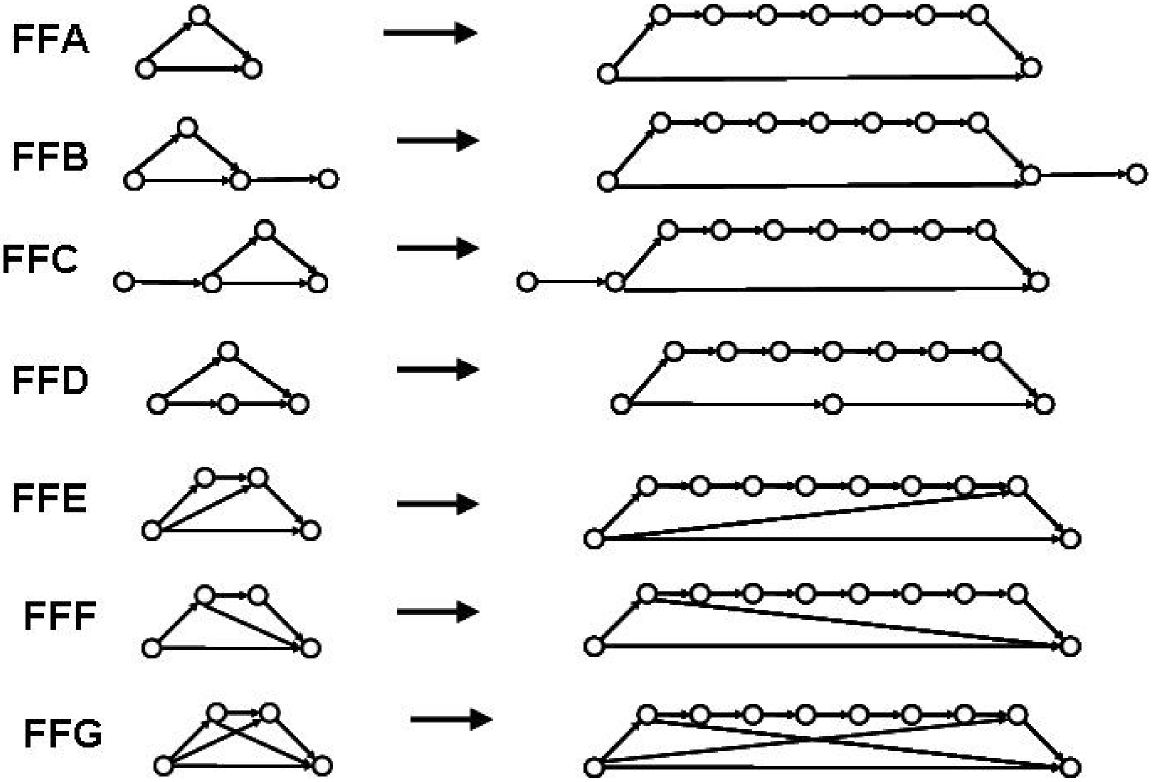


Figure 5. Feed-forward motifs with different network topologies. The first and the last member of the examined seven feed-forward motifs series are shown only. The series include all structures with intermediate number of nodes (FFA series had networks with 3 to 9 nodes; all other series had 4–10 nodes).

Two versions of CA model results were compiled: 50% conversion and 100% conversion. These two categories of results represent the number of CA iterations needed for 50% and 100% conversion of the source substrate into the target product. This number serves as an inverse measure of the FF motif dynamics (the larger the number of iterations needed for arriving at a steady-state, the slower the overall process). As seen in Table 3, the linear regression models obtained for 50% conversion have lower correlation coefficients (R^2 = 0.9796 to 0.9980) and considerably higher standard deviations (SD = 1.70 to 4.14%), as compared to the corresponding models with 100 % conversion (R^2 =

0.9990 to 0.9996 and SD = 0.25 to 0.50%). The subsequent analyses are based on the results obtained with 100% conversion only.

Table 3

Linear dependence of the overall rate of feed-forward motifs on the number of motive nodes

Series	N	Iterations range	Regression	R^2	SD %
FFA-50	3–9	804 – 2160	219.00 N + 227.14	0.9796	4.14
FFB-50	4–10	1735 – 3736	320.18 N + 483.46	0.9966	1.43
FFC-50	4–10	2043 – 4530	402.36 N + 481.36	0.9980	1.71
FFD-50	4–10	1375 – 3428	329.93 N + 148.50	0.9939	2.38
FFE-50	4–10	1080 – 2247	202.93 N + 258.64	0.9874	2.62
FFF-50	4–10	785 – 1574	133.36 N + 235.07	0.9981	1.70
FFG-50	4–10	1955 – 4104	365.61 N + 395.89	0.9900	2.78
FFA-100	3–9	2966 – 8444	920.75 N + 187.50	0.9996	0.31
FFB-100	4–10	4187 – 9044	800.93 N + 998.07	0.9994	0.29
FFC-100	4–10	4542 – 10565	993.71 N + 556.00	0.9990	0.25
FFD-100	4–10	3291 – 8067	802.64 N + 89.357	0.9993	0.34
FFE-100	4–10	3505 – 7599	683.57 N + 766.14	0.9994	0.33
FFF-100	4–10	2408 – 4927	418.21 N + 705.07	0.9991	0.50
FFG-100	4–10	2721 – 5086	397.61 N + 1121.8	0.9991	0.42

The overall rate is measured by the number of iterations needed for 50% and 100% conversion of the source substrate into the target product. Seven series (Fig. 6) of feed-forward motifs with different topology, constant number of source cells, *N* = 500, and constant lattice density, *D* = 3.6%, are studied.

2.3 Results and Discussion

2.3.1 Simulations of three node motifs

Following observations were made after simulating the series of three node networks.

Pattern 1: Adding a reverse edge.

A. The reversal of any step of consecutive reaction in a set of two reactions producing the same product slows down the production of final product.

B. The reversal of a step in a parallel reaction accelerates the production of alternative product.

C. In a feed-forward triangle with one reverse step, reversal of any of the two indirect steps considerably decelerates the production of final product.

D. Reaction decelerates if new edge added reverses the feed-forward edge.

Pattern 2: Adding a forward edge (feed-forward)

A. Adding the forward edge accelerates the production of final product except when two different reactions produce the same product.

B. Adding the forward edge accelerates the production of final product in feed-back networks where the added edge reduces feed-back

Pattern 3: Adding a backward edge (feed-back)

A. Adding a backward edge always decelerates the process in consecutive two-step reaction. The strongest deceleration occurs when a second decelerating step is already present in the network.

These preliminary observations, also confirmed by analyzing different topologies of four node motifs, formed the basis to study the effects of topology on dynamics of networks. The first step was to study biologically significant feed-forward topology in an effort to validate CA as a method to simulate dynamics of network motifs. This validation was achieved by comparing CA simulation results with a traditional modeling method, DE modeling, with promising results to strengthen use of CA as a dynamic modeling tool. The central focus of this analysis was to study how network topology affects the dynamics of processes in different feed-forward motifs. In order to ensure that the networks analyzed were comparable to enable the identification of stable structure–dynamics patterns, it was assumed that (i) the rate constants for all processes are equal, (ii) the initial conditions are chosen such that the source (S) is initially five times larger than each of the other species, and (iii) all enzyme activities are constant and equal. A chart containing all ten motifs having four nodes was constructed (Fig. 6) and their mutual transformations (15 additions of an edge with the formation of a new cycle, and three link direction reversals connecting feed-forward motifs with bi- and tri-parallel ones), and performed both CA and ODE linear and nonlinear modeling. Each network gives rise to a system of four linear ODEs, which can be solved explicitly. In the nonlinear case we performed numerical simulation with both irreversible and reversible first reaction steps. In all versions of the ODE models the networks were ranked according to their 90% conversion times. The numerical ODE values stand for the time measured in arbitrary units needed for a 90% overall conversion of the source substrate *S* into the target product *T*.

2.3.2 Simulations of feed-forward motifs

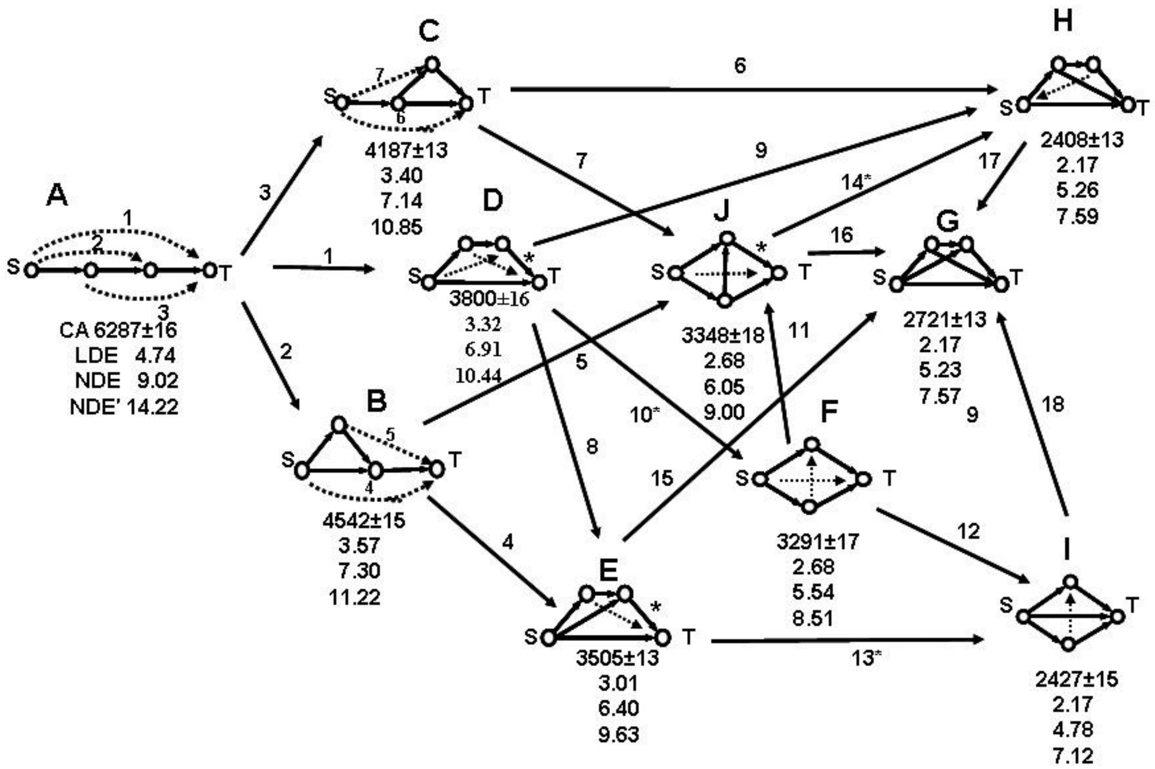


Figure 6. Chart of all ten four-node motifs (eight feed-forward and **F**– bi-parallel) of conversion of the source node S to target node T: **A** – linear, **B**, **C**, **D**, **E**, **G** and **H** – feed-forward (FF), **F** and **J** – bi-parallel FF, and *I*– tri-parallel FF. The motifs are ordered according to their dynamic efficacy in producing the target product with a highest rate, assessed by decreasing number of iterations, and decreasing time, as produced by the linear (LDE) and nonlinear ODE models for 90% S → T conversion. The nonlinear irreversible and reversible ODE times are denoted by NDE and NDE', respectively. The mutual transformations of the structures shown include 15 additions of another feed-forward link, whereas those marked by an asterisk (**D** → **F**, **E** → **I**, and **J** → **H** conversions) include a reversal of a single link direction.

The comparison of the efficacy of performance of the ten four-node networks (Fig. 6) shows that CA and linear ODE order the motifs in the same way with a minor exception. Namely, CA ranks structures **H** and **I** with the same highest conversion rate

(2408±13 and 2427±15, respectively), since the iteration numbers overlap within the range of their standard deviations. The linear ODE also ranks **H** and **I** as the fastest four-node topologies, adding a third structure **G**.

The nonlinear ODE models with reversible and irreversible first steps produce identical ordering of the ten structures. It coincides with the ordering of the first seven structures, described above by CA and linear ODEs, while suggesting that network **I** is the fastest, **G** and **H** have a very close performance, **H** is shown as slightly slower than **G**.

The topological analysis of the nine networks revealed some useful patterns of their dynamics. Although the networks analyzed here are relatively simple, they could be of use when analyzing local topology in large complex networks. Several of the observed topodynamic patterns are described below.

2.3.2.1 Dynamic Feed-Forward Pattern 1 (DFFP1):

The shorter the graph distance d(S→T) between the source node and the target node in a feed-forward motif, the higher the overall conversion rate:

$$\mathbf{A(d=3) < B, C\ (d=2) < D, E, G, H, I\ (d=1)} \qquad (4)$$

The bi-parallel motifs **F** and **J** do not obey this rate inequality.

2.3.2.2 Dynamic Feed-Forward Pattern 2 (DFFP2):

The shorter the average path length *l*(S→T) between the source node and the target node in a feed-forward motif, the higher the overall conversion rate:

$$\mathbf{A\ (\mathit{l}{=}3) < B,\ C\ (\mathit{l}{=}2.5) < D,\ E,\ G,\ H\ (\mathit{l}{=}2) < I\ (\mathit{l}{=}5/3)} \tag{5}$$

Accounting for all S→ T paths is a slightly more sensitive pattern, which singles out network **I** as the most efficient four-node structure, in agreement with the result obtained by the nonlinear ODE model. The bi-parallel motifs **F** and **J** do not obey this feed-forward topodynamic trend, which is more important in larger networks where the number of S → T paths increases rapidly.

2.3.2.3 Dynamic Feed-Forward Pattern 3 (*Isodynamicity*):

Some feed-forward motifs with different topology produce the same overall S → T conversion rate by the CA and linear ODE models:

$$\mathbf{CA:} \qquad \mathbf{H\ (2408{\pm}13) = I\ (2427{\pm}15)} \tag{6}$$

The CA and the nonlinear ODE simulations showed these two motifs with different, although relatively close efficacy, the structure **J** being the slower one:

$$\mathbf{CA:} \quad \mathbf{J\ (3348{\pm}18) > F\ (3291{\pm}17)} \tag{7}$$

The property of isodynamicity is a surprisingly novel network pattern, which could warrant further detailed studies.

2.3.2.4 Dynamic Feed-Forward Pattern 4 (DFFP4):

Any ring closure of a linear chain of conversion of a source substrate *S* to the target product *T* accelerates the transformation. Acceleration of the process is strongest when the feed-forward link directly connects the substrate to the target and is the smallest when the link connects the substrate with an intermediate product (Figs. 7, 8).

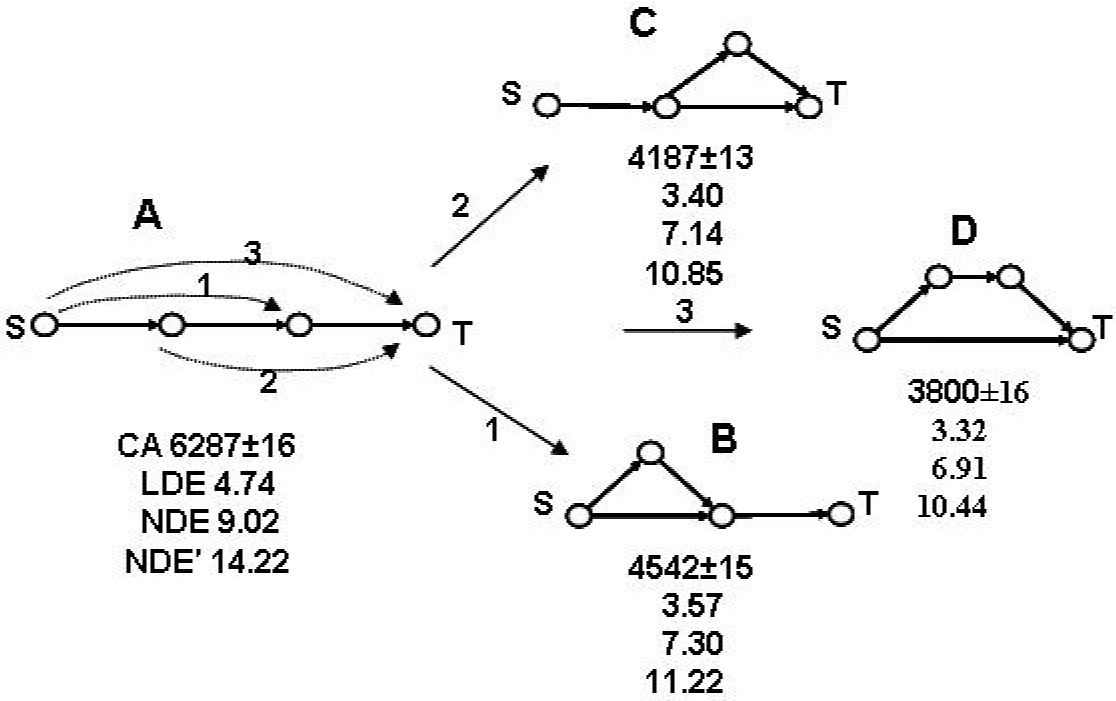
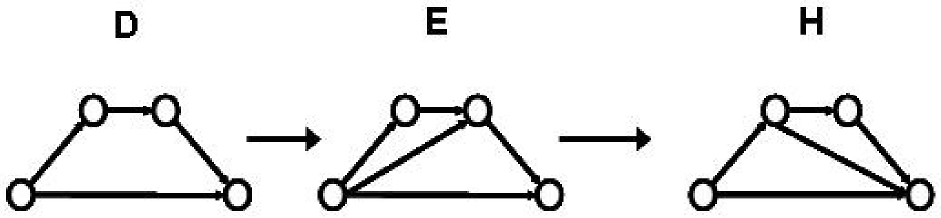


Figure 7. Topological feed-forward transformations (1, 2, and 3) always accelerate processes described as a linear chain of events. Different mechanisms of ring closure are shown, the fastest topology being the one with a direct Source → Target feed-forward link. The linear and nonlinear irreversible and reversible ODE times are denoted by LDE, NDE and NDE', respectively.



Series	I(FFA)	>	I(FFE)	>	I(FFF)
Iterations	3800±16		3505±13		2408±13
LDE	3.32		3.01		2.17
NDE	6.91		6.40		5.26
NDE′	10.44		9.63		7.59

Figure 8. Adding a second feed-forward edge always accelerates the processes in a feed-forward motif. There is no such general pattern for the addition of a third feed-forward link (Compare **E** → **G** to **H** → **G**). The linear and nonlinear irreversible and reversible ODE times are denoted by LDE, NDE and NDE', respectively.

$$\mathbf{A < B < C < D} \qquad (8)$$

The ring-closures described by this pattern are shown in Fig. 8 with serial numbers 1, 2, and 3. The generality of this topology-dynamics relationship was verified for the entire series FFA, FFB, and FFC (Fig. 5) having up to ten motif nodes. In all cases, the standard deviation the number of CA iterations was found to be more than two orders of magnitude smaller than that number. This pattern goes beyond the simple topological patterns 1 and 2 shown above, which cannot discriminate between FFB and FFC series.

2.3.2.5 Dynamic Feed-Forward Pattern 5 (DFFP5):

Adding a second feed-forward edge (*double feed-forward motif*), between any pair of nodes in the longer path of the FF loop, speeds up the dynamics of the source substrate conversion into the target product (Fig. 8).

$$\mathbf{D} < \mathbf{E} < \mathbf{H} \tag{9}$$

These inequalities for the number of iterations, illustrated in Fig. 8 with four node motifs, were verified and found valid with no exceptions for all sizes of the three FF series examined (four to ten loop nodes). Comparing the FFF and FFE series, one may generalize that the acceleration of substrate-to-target conversion is higher when the second FF-link starts in a node located on the longer source-target path and ends into the target node, rather than when it starts in the source node and ends in another node before the target one. Since the structures of the triple-feed-forward motif FFG (see graph **G** in Fig. 5) combine the CA trends of the FFF and FFE series, the acceleration in this series is intermediate between those of FFE and FFF. However, the ODE models do not confirm this result with the linear model showing G and H to be isodynamic, whereas the two nonlinear models show G as slightly more efficient than H. Therefore, adding a third feed-forward link does not necessarily result in acceleration and no stable trend exists

2.3.2.6 Dynamic Feed-Forward Pattern 6 (DFFP6):

Reversing the direction of one or more links in a feed-forward motif to turn it into a bi-parallel and tri-parallel one increases the network efficacy. (Figs.6 and 9).

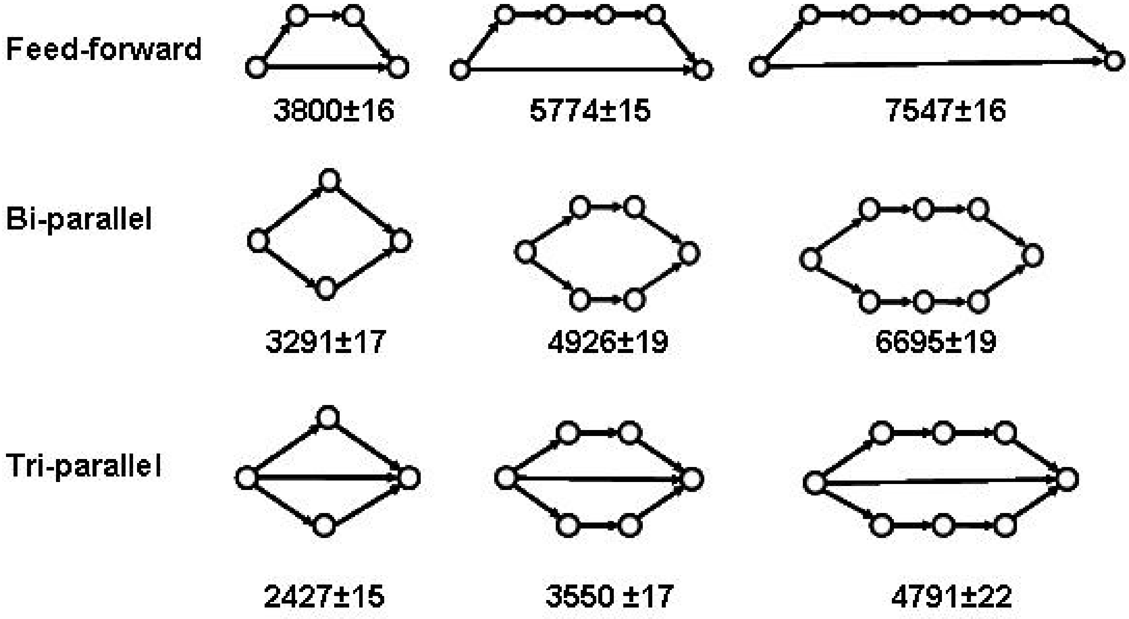


Figure 9. At the same number of nodes, the feed-forward motif is slower than the bi-parallel motif. The topology producing the fastest dynamics is that of the tri-parallel motif **I**.

$$\textbf{Feed-Forward} < \textbf{Bi-Parallel} < \textbf{Tri-Parallel} \qquad (10)$$

Three such conversions:

$$\mathbf{D < F, E < I, J < H} \qquad (11)$$

are shown in Fig. 6, where they are denoted by asterisks.

3.2 Patterns of Topological Effect on the Network Motif Performance

The overall reaction rate in a biochemical network motif was assessed by using cellular automata (CA) [61, 62, and 63] and ordinary differential equations (ODE) [47, 64] simulations. The rate was assessed as the time (ODE method) and the number of iterations (CA method) needed for 90% conversion of the source node concentration into the target. Several patterns of topological effects accelerating the motif performance were identified. The motifs analyzed below are denoted by their ID numbers, as introduced by Uri Alon and coworkers [65].

2.3.3 Simulation and classification study of small node motifs

CA was established as a valid method to perform dynamic modeling of biochemical network motifs by producing close results to DE modeling. Many patterns of small node motifs were identified in the feed forward motif study. Hence it was important to systematically classify these patterns according to their dynamic performance, in search of a universal trend in the organization of larger biochemical networks. The first and foremost thought was to query different organizational levels of different species to see if some of these patterns are more common than other. The motivation for undertaking such kind of study was that each different pattern of network motifs, offers different benefits. E.g. the different isodynamic networks found while studying feed forward mechanism might offer a choice of networks differing in cost and redundancy. Also different topologies create an ordering of network motifs based on dynamic characteristics of network, making some motifs performing faster than others, when it comes to converting a source substrate into a target product. Hence many different patterns of small node motifs were identified and studied, both using CA and ODE

2.3.3.1 The Average Path Length Pattern:

The shorter the average path length L between the source node and the target node (S→T) in a feed-forward (FF) and in a bi-parallel (BP) motif, the faster the overall conversion rate. Below are examples of comparison between different types of motifs with their IDs followed by average path length (L) (See Figure 10).

3-node FF motifs (not shown):

Motif 12 (L = 2) < Motif 38 (L = 1.5)

4-node FF motifs

536 (L=3) < 2076, 2118 (L=2.5) < 652, 2126, 2254, 2204 (L=2) < 2190 (L=5/3)

4-node bi-parallel motifs:

2140 (L = 7/3) < 2182 (L = 2)

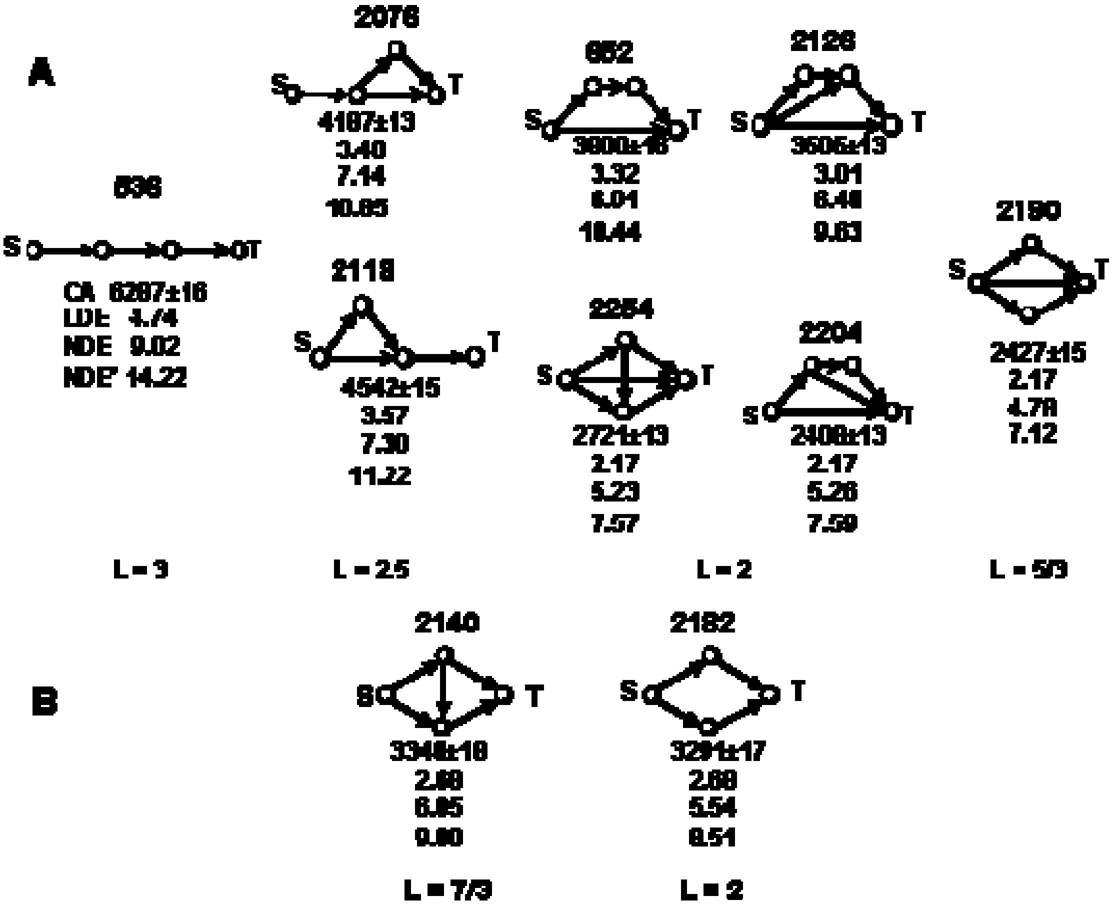


Figure 10. The average path length *L* between the source and the target node (calculated by counting each directed link as unit distance) imposes a partial order in the classes of (A) 4-node feed-forward and (B) 4-node bi-parallel network motifs: the smaller *L*, the faster the motif performance. Given for each motif from top to bottom are its ID [65], sub graph, the number of cellular automata (CA) iterations with the range of standard deviation, and the time in seconds as found from solving linear ODEs and two versions of nonlinear ODEs.

Figure 10 demonstrates the average path length pattern. The topological descriptor *L*, which measures the easiness of communication between the source and the target nodes, is shown to capture the trend of accelerating motif performance with the decrease in the average path length. The eight feed-forward motifs are partitioned into four groups, and the two bi-parallel motifs are ordered according to this topological pattern of accelerated dynamics. A further discrimination of the motifs having the same value of parameter *L* is possible proceeding from additional topological and entropy analysis.

2.3.3.2 The Ring Closure Pattern:

Any ring closure of a linear chain of biochemical reactions converting a source substrate *S* into a target product *T* accelerates the transformation. The acceleration of the process is the strongest when the feed-forward link directly transforms the substrate into the target and weakest when only one of the intermediate substrates directly converts into the target product [61].

Example of the same can be seen in Figure 7 which establishes following ordering of the motifs,

536 < 2118 < 2076 < 652

2.3.3.3 Adding a Second Feed-Forward Link.

Adding a second feed-forward link between any pair of nodes in the longer path of the FF loop, creating thus a second FF loop, speeds up the source substrate conversion into the target product [61] (Figure 11)

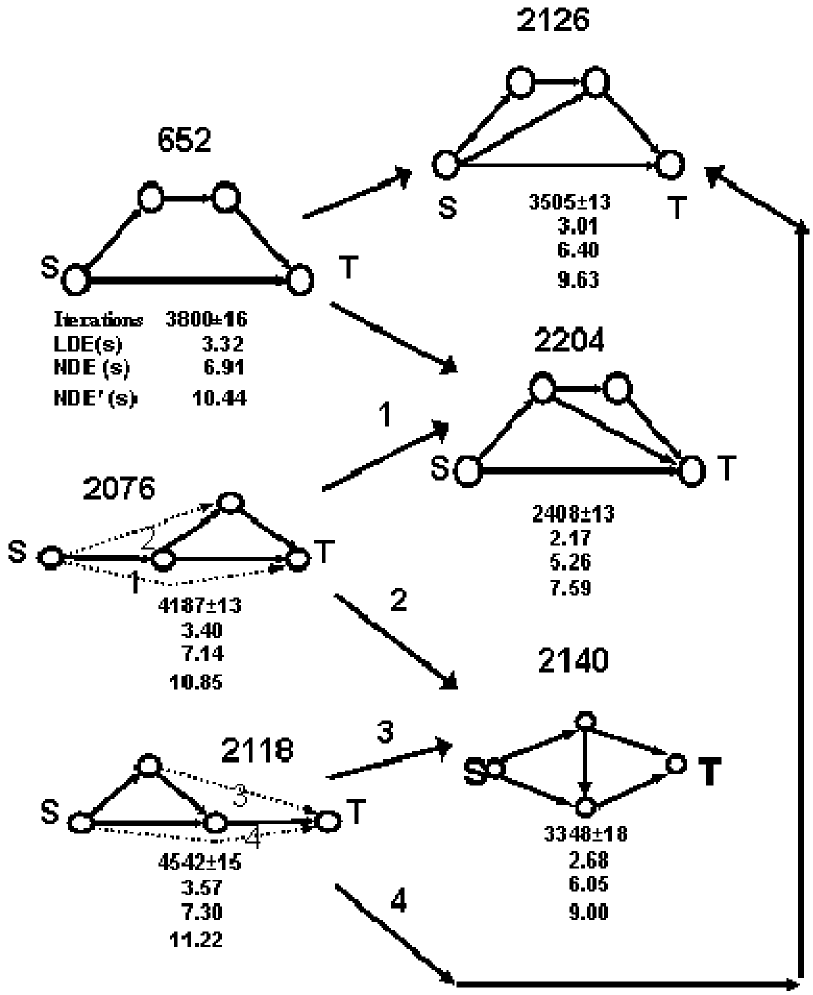


Figure 11. Formation of a double feed-forward loop by adding an extra forward link accelerates the process of converting the source substrate *S* into the target product *T*. The highest acceleration is achieved in motif 2204, which emerges after adding the additional link to the two faster initial motifs 652 and 2076.

2.3.3.4 Pattern of Feed-Forward Motif Transformation into Bi-Parallel or Tri-parallel Motif.

Specific to 4 node motifs, reversing the direction of a link that enters the target node in a feed-forward motif turns the motif into a bi-parallel or tri-parallel motif and increases its efficacy (fig 12).

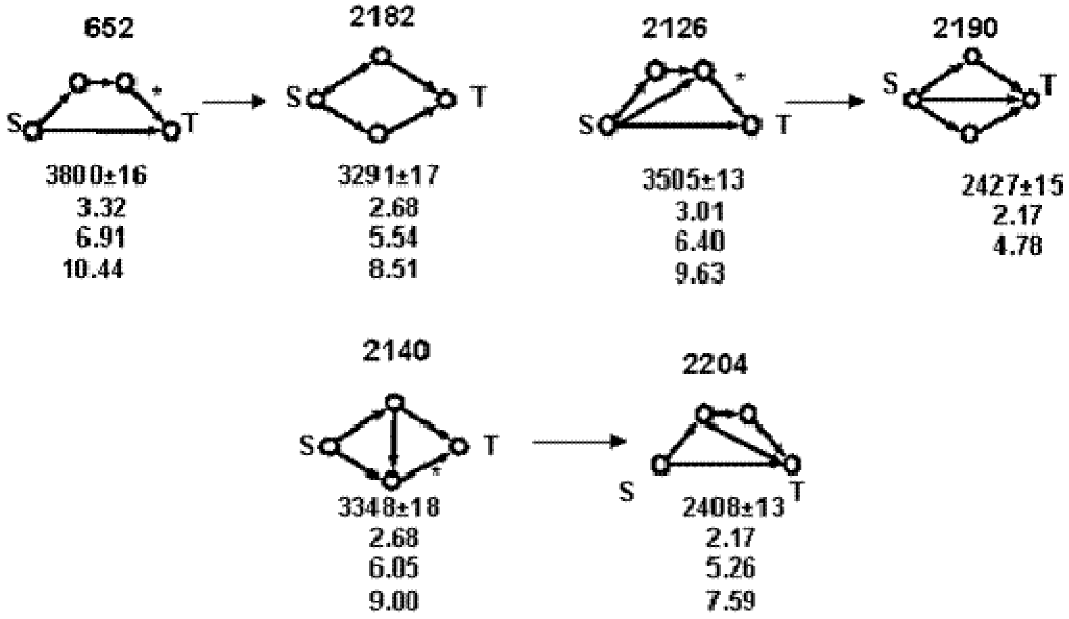


Figure 12. Transforming a feed-forward motif into a bi-parallel or tri-parallel motif by reversing a link direction accelerates the process of converting the source substrate into a target product.

2.3.3.5 Pattern of Entropy-Based Acceleration of Feed-Forward Performance.

The smaller the differences in the lengths of two directed chains connecting a single source to a single target in a feed-forward or bi-parallel motif, the faster the motif performance. This pattern can be interpreted in terms of the Shannon information theory as increasing the information entropy *H* in the sequence of structures having two chains of varying length when the same overall number of links *L* is distributed more and more evenly between the two chains having L_1 and L_2 links, respectively.

From $H = L\log_2 L - \sum L_i \log_2 L_i$, more even distribution of the L_i terms will result in smaller sum and the larger entropy H, as illustrated in Fig. 13.

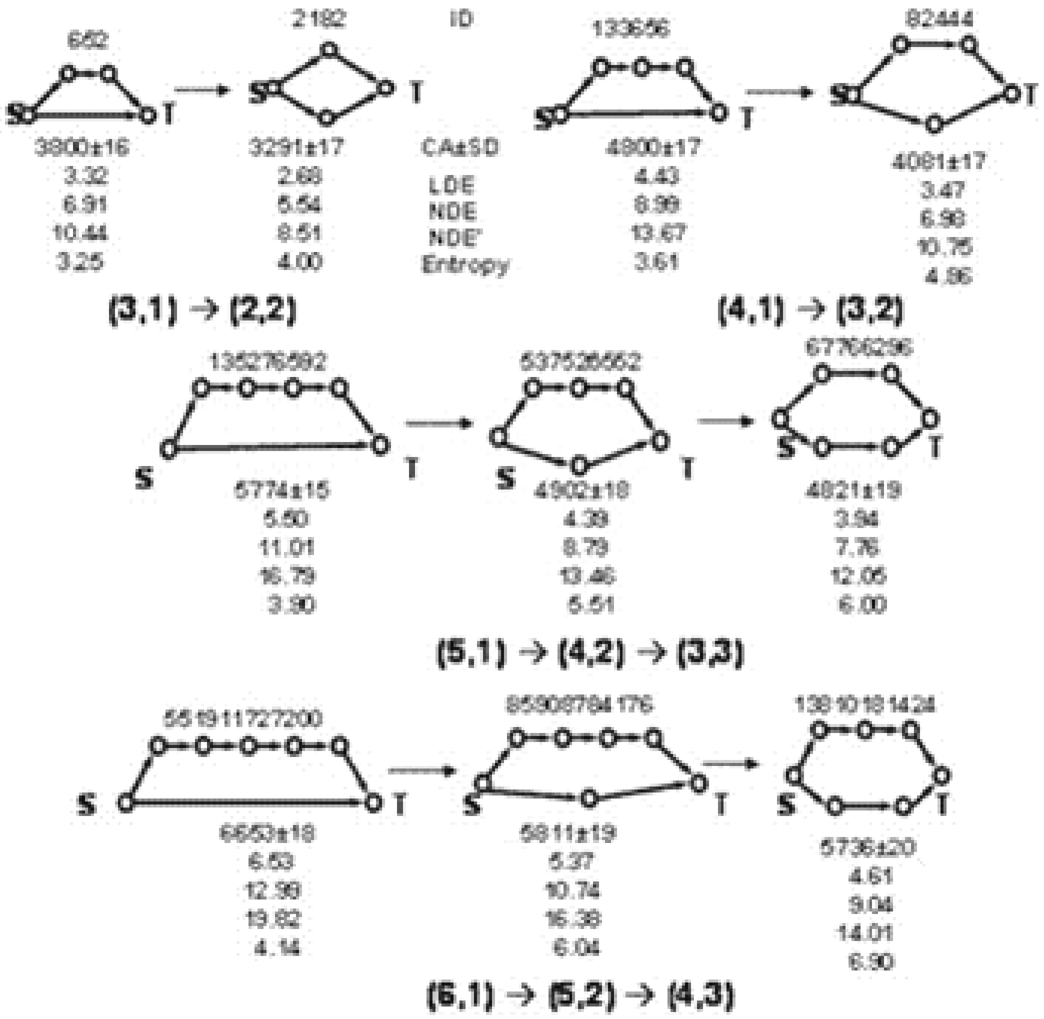


Figure 13. The more even the distribution of the lengths of two directed chains connecting the source node *S* to the target node *T*, the faster the transformation of the substrate *S* into the product *T*.

2.3.3.6 Isodynamicity Pattern

Some feed-forward motifs with different topology produce the same overall S → T conversion rate (termed herein *isodynamic*) by the linear ODE models and close values according to the nonlinear ODE models and the cellular automata (CA). (Figure 14)

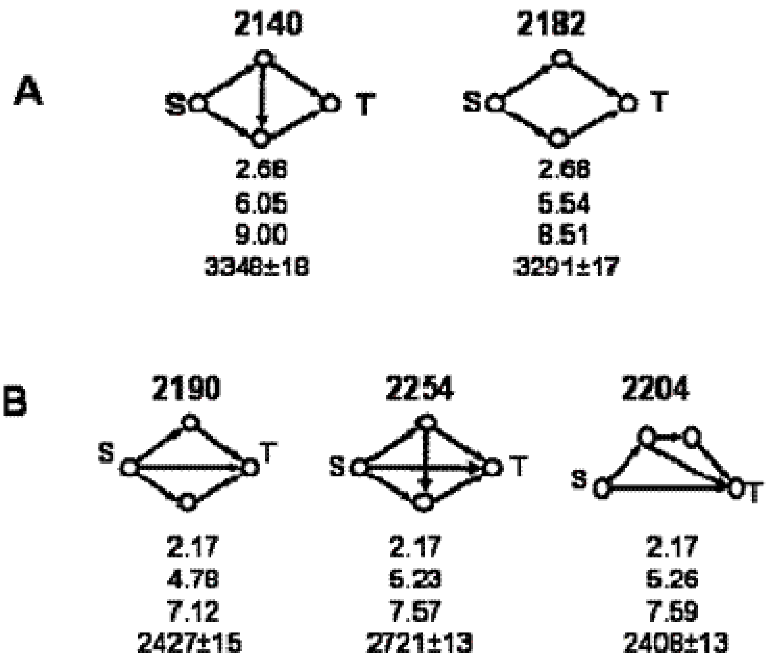


Figure 14. Two sets of isodynamic 4-node motifs characterized by constant overall time in *s* according to the linear ODE models, and close values of the corresponding nonlinear ODE models and the CA iterations, needed for 90% conversion of the source substrate *S* into the target product *T*.

Two theorems provide examples of classes of motifs that are pair-wise isodynamic according to linear ODE models [64]. The first one (Fig. 15A) extends this property to all feed-forward motifs, in which the target vertex has the maximal in-degree of V-1, where V is the total number of motif nodes. The second theorem deals with bi-parallel motifs having V vertices with alternating labeling (Fig. 15B). They were shown to be isodynamic to the derivative motifs having extra directed link(s) that connect vertices k and (k + 1) (regardless of orientation) if k has the same parity as V and 1 < k < (V − 1).

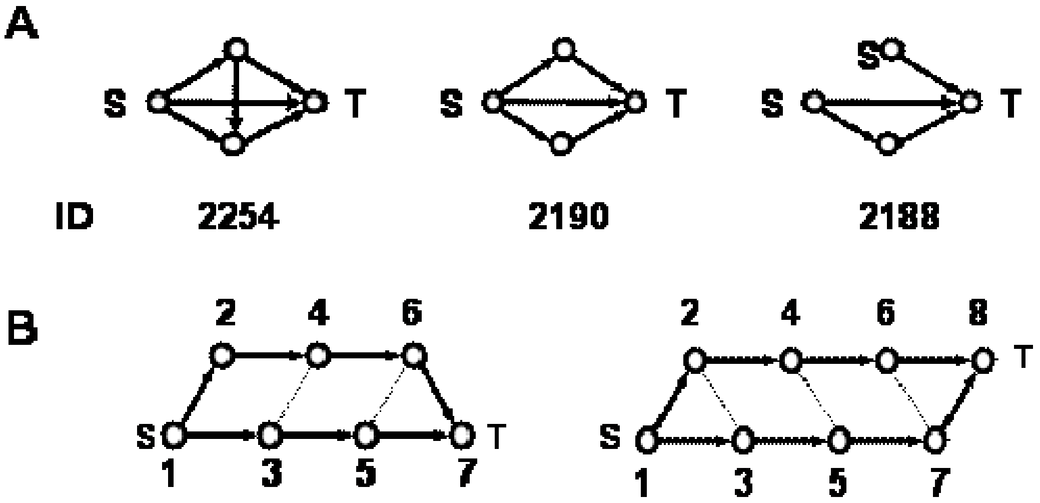


Figure 15. Illustration of the two theorems of isodynamicity of network motifs [64]. **A.** Deletion of an extra link in the row of motifs 2254 → 2190 → 2188, or **B.** Adding one or more extra directed link (both directions allowed) between two nodes labeled with consecutive numbers, does not change the time needed for transformation of the substrate *S* into the product *T* as calculated by linear ODEs.

Although in the more detailed nonlinear ODE modeling these motifs are no longer isodynamic, yet their dynamic properties remain fairly close. For example, motif 2140 is performing slower than motif 2182 by about half a arbitrary time unit in both nonlinear ODE simulations and almost exactly the same is the slightly lower performance of motif 2254, as compared to motif 2190 (Figure 16).

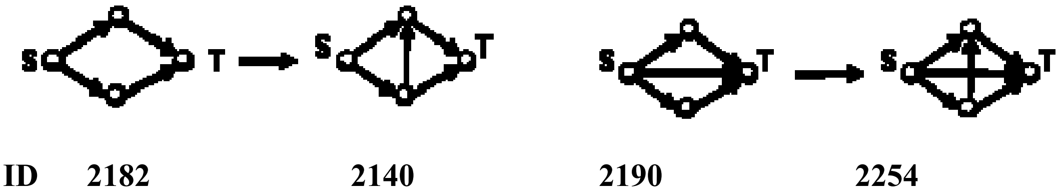


Figure 16. Pairs of isodynamic motifs having identical dynamic characteristics.

2.4. Conclusions

The technique of dynamic modeling with cellular automata shows great promise in modeling complex biological systems. Such systems can be broken down to subsystems of smaller scale (to ease computational time) and simulated independently so as to shed light on the processes on a larger scale. The essential element in such applications is the extraction of useful topological-dynamic (topodynamic) patterns, which identify specific effects of topological structure on the dynamics of network processes while keeping all kinetic parameters constant. The beauty of the topological approach in studies of dynamics is in the generality of the patterns found, which are independent of the nature of the processes, and may be applied to any process of chemical transformation, as well as to any process of mass, energy or information transfer down the forward direction of the motifs. The dynamics of the feed forward motifs observed in this study revealed important aspects of networks with such components. Not only does any feed-forward link added to a linear cascade of chemical/biochemical reactions accelerate the process, but the acceleration is further enhanced by adding a second forward link in the feed-forward loop. The topological hierarchy established in this study for four-node motifs predicts that the acceleration of the overall process in such motifs will continue increasing with the decrease in the distance (both along the shortest path and along all paths) between the input and output nodes, whereas at the same distance the cellular automata and differential equation simulations produce in a similar manner a further distinction between the motifs' dynamic efficacy. The intriguing property of isodynamicity was identified showing motifs with the same number of nodes and different topology to have the same overall rate of input-to-output transformation. If shown to be present in larger biological networks, the observed isodynamic property could indicate a level of biological robustness at a topological level. Further topology-dynamics studies involving construction of networks from combinations of such structural blocks will aid in increasing the understanding of complex biological networks.

This study was conducted in order to study different patterns of network motifs. Out of many patterns investigated, isodynamic patterns are of most interest due to their ability to provide extra tolerance against random mutations. Finding pairs of isodynamic (or near-isodynamic) motifs raises the question whether motif topologies with extra link(s), which do not contribute to a better efficiency, do really exist in biochemical networks and if so, what could be their biological meaning. Also it will be interesting to query networks on different organizational levels to check if there is evolutionary preference for motifs like with IDs 2190, 2254 and 2204 which perform faster than other four node motifs. To address such questions, following hypothesis were formulated, work on which is currently under way [66]:

Hypothesis #1: Network motif topologies enabling high overall rate of biochemical reactions converting a source substrate into a target product are evolutionary beneficial and could be conserved with higher frequency in some types of biochemical networks.

Hypothesis #2: Isodynamic motifs having extra "null-efficiency" link(s) provide higher resilience against attacks compared to the isodynamic motifs lacking such links and could be conserved with higher frequency in some types of biochemical reactions.

CHAPTER 3 STUDY OF FAS-L INDUCED APOPTOSIS

3.1 Introduction

Out of the many patterns shown in the previous chapter, isodynamicity was of particular interest in highly redundant networks. One good example of such robust network is the FAS-L induced apoptosis pathway. It is a signaling pathway, which is activated by an extracellular protein (FAS-ligand) and ends with the decomposition of DNA by DFF heterodimer and protein cleavage by caspase 6 (CASP6) (fig. 17). Study of FAS-L induced apoptosis started with building a CA model of the apoptosis process and searching for isodynamic motifs within the apoptotic network. The isodynamicity revealed in initial studies demanded further enhancements to the model to study FAS-L pathway in detail and apply it to model strategies to fight cancer. Different components of the apoptosis pathway were added systematically to assess the effects of pro and anti-apoptotic molecules. Two main strategies were considered while modeling the FAS-L induced apoptosis network. First, inhibitors of apoptosis were modulated to inhibit apoptosis in T-cells. Secondly, inhibitors of inhibitors of apoptosis were used to eliminate the effects of apoptosis inhibitors and allow cancer cells to proceed with programmed cell death. Before proceeding with the model, it is important to overview the FAS-L induced apoptosis process.

Apoptosis is a process of programmed cell death, which is the most common mechanism by which the body eliminates damaged or unneeded cells [67]. The abnormal functioning of apoptosis is associated with a multitude of diseases [68], and is critical aspect of cancer development and tumor cell survival [69]. Such health implications underscore the pharmacological potential of designing strategies to manipulate apoptosis [70-77]. Apoptosis is initiated via two types of signaling pathways, the components of which are potential anticancer therapeutic targets. The intrinsic pathways are activated by developmental signals or severe cell stress, while the extrinsic pathways are triggered by pro-apoptotic ligands, such as FAS ligand (FASL), which binds to the pro-apoptotic death receptor FAS (CD95). The FASL/FAS binding induces recruitment of the FAS-associated death domain protein (FADD) and the initiator caspases 8 or 10 as procaspases, forming a death-inducing signaling complex (DISC). Formation of the DISC brings procaspase molecules into close proximity of one another, enabling their autocatalytic activation and release into the cytoplasm where they activate effector caspases (CASP3, CASP6, and CASP7). The latter split the heterodimer DFF (DNA Fragmentation Factor), and the released DFF40 fragments DNA, while CASP6 cleaves proteins vital for cell survival. The caspase cascade is regulated by c-FLIP (FADD-like apoptosis regulator) proteins and the

IAP (Inhibitor of Apoptosis) protein family, XIAP being known as the most potent inhibitor (Fig. 17).

The FAS protein plays a critical role for the immune system by killing pathogen-infected cells [78] and preventing autoimmunity and tumors. Recent studies [79] have revealed important details of the role of T-cells in the immune response to fight cancerous and HIV-infected cells. The latter induce apoptosis in T-cells and kill them by overexpressing FASL, while preventing their own destruction by the same apoptotic mechanism. Based upon these recent findings, the present study developed a dynamic model simulating two strategies against cancer. The classical approach aims to kill cancerous cells, a goal that we simulated by strong suppression of apoptosis inhibitors FLIP and IAP by siRNA and SMAC, respectively, following the recent study of Wilson *et al*. [80] and Cheung *et al*. [81]. The opposite strategy of maximizing the inhibition of FLIP and IAP was applied to simulate blocking the apoptotic process in T-cells that is caused by a tumor counterattack [79]. Such a strategy to restore the potency of the immune system could also be of interest in the fight against HIV-infection.

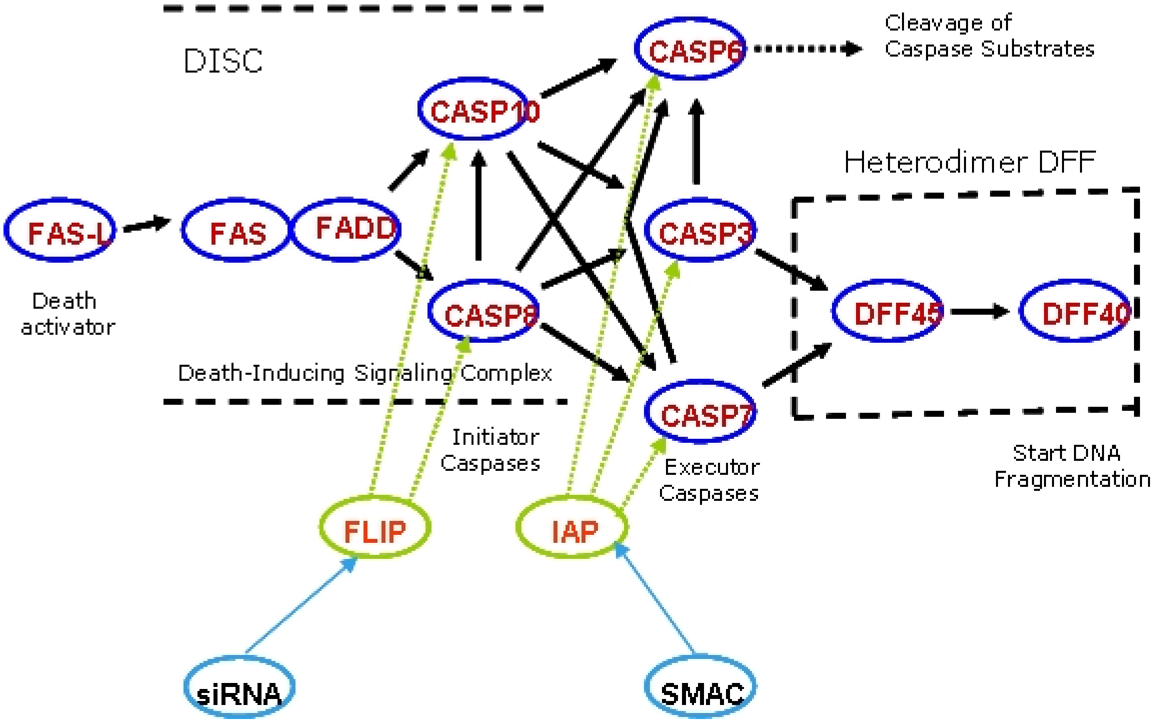


Figure 17. The FAS-ligand induced apoptosis and its fine regulation by the apoptosis inhibitors FLIP and IAP, and their suppressors siRNA and SMAC.

3.2 Methods

3.2.1 The Set of Apoptosis Interactions

The following system of equations was used to simulate FAS-L induced apoptosis as given in Figures 18 and 22 (Letter A is used as a prefix for protein name to denote the protein active form):

Attachment of Ligand

FAS + FASL → AFAS + FASL

Recruitment of DISC

AFAS + AFAS → FASFAS

FASFAS + FADD → FASFADD

Formation of DISC Complex

FASFADD + CASP8 → FASFADDACASP8

FASFADD + CASP10 → FASFADDACASP10

Activation of Caspases

FASFADDACASP8 + CASP10→ FASFADDACASP10 + ACASP8

FASFADDACASP8 + CASP3 → ACASP3 + FASFADDACASP8

FASFADDACASP8 + CASP6 → ACASP6 + FASFADDACASP8

FASFADDACASP8 + CASP7 → ACASP7 + FASFADDACASP8

FASFADDACASP10 + CASP3 → ACASP3 + FASFADDACASP10

FASFADDACASP10 + CASP6 → ACASP6 + FASFADDACASP10

FASFADDACASP10 + CASP7 → ACASP7 + FASFADDACASP10

ACASP3 + CASP6→ ACASP6 + ACASP3

ACASP3 + CASP7 → ACASP7 + ACASP3

ACASP7 + CASP6 → ACASP6 + ACASP7

Activation of Apoptosis

DFF45DFF40 + ACASP3 → DFF45ACASP3 + DFF40

DFF45DFF45 + ACASP7 → DFF45ACASP7 + DFF40

Inhibition of Apoptosis

FLIP + FASFADDACASP8 → FLIP FASFADDCASP8

FLIP + FASFADDACASP10 → FLIP FASFADDCASP10

AIAP + ACASP3 → IAPCASP3

AIAP + ACASP6 → IAPCASP6

AIAP + ACASP7 → IAPCASP7

Suppression of Inhibitors

siRNA + FLIP → siRNAFLIP

SMAC + IAP → IAPSMAC

Mitochondrial Apoptosis Pathway

ACASP8 + Bid → ABID + ACASP8

ABID + Mit → cytC + AMIT

ABID + Mit → SMAC + AMIT

cytC + CASP9 → ACASP9 + cytC

ACASP9 + CASP3 → ACASP3 + ACASP9

ACASP3 + CASP9 → ACASP9 + ACASP3

ACASP9 + AIAP → ACASP9 + IAP

3.2.2 Parameters Used

Expression profiles of the apoptotic proteins [91] were used as a basis for input concentrations (number of CA cells): FASL - 50, FAS - 200, FADD - 125, FLIP - 125, IAP - 300, DFF - 500, CASP3 - 150, CASP6 - 40, CASP7 - 400, CASP8 - 750, CASP9 - 375, CASP10 - 50, Mit - 1000, Bid - 75, SMAC - 400, siRNA - 200. (The DFF and siRNA expression levels were not found in the literature, and the values presented here were assumed within the same range with the measured expression data). The initial concentrations of the products of the above equations were assumed zero.

The modeling was performed by 100 runs for each specific case considered. The number of iterations and the cell count in each run was recorded after reaching a steady (or near-steady) state configuration, and averaged over all runs. In the rare cases when the attainment of steady state was computationally highly demanding we stopped the iterative process at a sufficiently large number of iterations (25,000). We used a variable square grid of minimum 100x100 cells size embedded on a torus. The variations in the size were related to the condition of having comparable cell density of 60%. The basic parameter varied was that of the transitional probability P(T) of apoptosis elementary steps. The preliminary parameter testing has shown that selecting P(T) = 0.5 for almost all pathway steps, except few which were varied within the 0 to 1 range, provides stable patterns of the pathway modeled.

3.2.3 Variations of essential probabilities

DISC consists of lattice of FAS dimer and FADD, to which CASP 8 or CASP10 or FLIP can be attached. To simulate a stable and functional DISC lattice, the breaking (br) and joining (j) probabilities for FAS-FAS and FAS-FADD interactions were varied, as shown in Table 4. A dependence between the joining and breaking probabilities of FAS and FADD was assumed as a condition for the formation of a stable lattice and approximately defined as P(j, 2FAS-FADD) ≈ 1-P(br, FAS-FAS). The probability values P(br) = 0.3 for FAS dimer dissociation, P(j) = 0.8 for joining FAS and FADD and P(tr) = 0.4 for FAS-FADD bond formation were selected as these probabilities enabled the fastest formation of a DISC lattice to make the model consistent with existing evidence. The stability of the DISC lattice was further ensured by assuming a low probability of breaking, P(br, FAS-FADD) = 0.1. This simulated the transient nature of DISC, allowing it to dissociate weakly rather than making it a permanent structure with P (br) = 0.

Table 4: Variation of Probabilities Relevant to the Formation of the DISC Apoptotic Complex

FAS-FAS P(br)	2FAS-FADD P(j)	2FAS-FADD P(tr)	Iterations	STD DEV
0.02	0.98	0.9	3334	15.00
0.1	0.9	0.7	1409	11.63
0.3	0.7	0.1	2961	11.35
0.3	0.7	0.4	1122	15.24
0.3	0.9	0.4	1127	12.48
0.3*	**0.8**	**0.4**	**1116**	**14.42**
0.5	0.5	0.1	1163	15.16
*Optimized values given in bold				

Different transition probabilities of inhibition ranging from 0.05 to 0.9 were tested to simulate the effects of progression from weaker inhibition to a stronger one. A linear trend was observed with the DFF40 concentration showing inverse relationship to the strength of inhibition. Hence, P (tr) = 0.5 was used for all inhibitory effects in order to optimize computational time.

3.3 Results and discussion

3.3.1 Study of isodynamicity in FAS-L apoptosis pathway

FAS-L apoptosis pathway presents, without including inhibitors, a 7 nodes and 14 edges network structure with high amount of cross linkage to preserve the ability of the network to function even in the case of excessive damage (Figure 17).

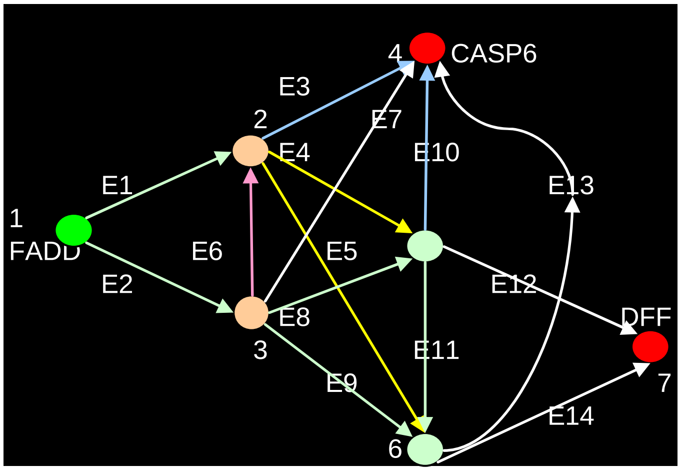


Figure 18: The FAS-L induced apoptosis network is shown, without inhibitors, represented as a network structure purely based on nodes and edges with source indicated by FADD, representing DISC for simplification, and targets indicated by CASP6 and DFF.

This network was subjected to single and double edge deletion and single node deletion to biologically correspond to the gene deletion or mutation event. The results are presented in tables 5, 6 and 7. In table 5 and 7, similar color is used to show deletions with isodynamic behavior. These colors also correspond to respective node and edge with same color in Figure 18.

Table 5: Deletion of single edges and compared for isodynamicity. Simulations performed for constant 20000 iterations. Final concentrations given in cellular automata cells with corresponding standard deviation for Caspase 6 and DFF40.

Single Deletion		CASP6		DFF	
Link removed	iteration	final conc	std dev	final conc	std dev
none	20000	665.91	15.2	433.99	15.22
E1	20000	662.98	16.99	433.96	16.79
E2	20000	664.07	18.18	432.54	18.08
E3	20000	593.13	19.57	506.65	19.57
E4	20000	703.34	17.75	396.55	17.78
E5	20000	702.38	16.04	397.55	16.05
E6	20000	666.49	17.14	433.43	17.15
E7	20000	625.04	17.38	474.85	17.38
E8	20000	686.48	18.56	413.45	18.56
E9	20000	685.93	15.28	413.95	15.3
E10	20000	579.69	17.07	520.16	17.03
E11	20000	666.67	16.32	433.27	16.34
E12	20000	752.11	14.11	347.71	14.08
E13	20000	445.35	16.84	651.88	16.74
E14	20000	886.63	11.03	210.4	10.94

Table 6: Deletion of two edges simultaneously and compared for isodynamicity. Simulations performed for constant 20000 iterations. Final concentrations given in cellular automata cells with corresponding standard deviation for Caspase 6 and DFF40.

Double Deletion		CASP6		DFF	
Links removed	iteration	final conc	std dev	final conc	std dev
E8/E9	20000	725.48	15.12	374.36	15.12
E8/E11	20000	684.07	16.39	415.83	16.39
E9/E11	20000	686.96	17.34	412.95	17.33
E5/E9	20000	723.53	16.37	376.3	16.32
E2/E8	20000	669.79	16.01	426.93	16.05
E1/E6	20000	664.56	16.66	432.05	16.65
E1/E4	20000	686.45	16.18	409.83	15.83
E1/E8	20000	699.58	15.48	396.98	15.19

Table 7: Deletion of single nodes and compared for isodynamicity. Simulations performed for constant 20000 iterations. Final concentrations given in cellular automata cells with corresponding standard deviation for Caspase 6 and DFF40.

Node Deletion		CASP6		DFF	
Node removed	**iteration**	**final conc**	**std dev**	**final conc**	**std dev**
2	**20000**	599.37	16.31	399.52	16.25
3	**20000**	597.15	16.03	401.74	15.97
5	**20000**	674.34	15.33	325.64	15.34
6	**20000**	675.95	15.88	324.04	15.9

Analyzing tables 5, 6 and 7, tremendous amount of isodynamicity was revealed in the FAS-L induced apoptosis network by CA simulations. Hence the motivation to study FAS-L network in detail using CA, was the complexity presented by this network and the ability of CA to elucidate this complex behavior. The regulatory control of this pathway which leads to successful apoptosis in normal circumstances can be altered in some conditions as cancer, leading to such cells evading apoptosis and proliferating uncontrolled. This regulatory mechanism offers two different opportunities to fight cancer. Both the strategies were modeled using CA

3.3.2 Modeling FAS-L apoptosis pathway using CA

The strategy of blocking the apoptosis in the immunological response of T-cells by maximizing the inhibitor action of FLIP (flagellar biosynthesis protein) [82-88] and IAP (inhibitor of apoptosis) [89, 90], either individually or jointly [80] was simulated. IAPs are characterized by the presence of between 1 and 3 specific domains called baculoviral repeats (BIRs), which are directly involved in their caspase-inhibitory activity. FLIP acts by attaching to DISC (death-inducing signaling complex), which blocks the activation of CASP8 and CASP10 (See Fig. 17). The DISC complex thus acts as a mechanistic switch to regulate apoptosis. The very process of DISC formation is a highly regulated and delicately balanced process. It includes formation of a FAS-FAS dimer and FADD association with the FAS dimer, followed by Caspase 8 and Caspase 10 recruitment and activation. The binding between FAS-FAS dimer, and that between FAS dimer and FADD, is shown to be weak [91-93] so as to prevent accidental activation of apoptosis, and to proceed to formation of DISC complex only in the case of strong stimulus. DISC consists of a lattice of FAS dimer and FADD. The sequence of elementary steps of this complex mechanism is described in detail in the Experimental Part section.

This simulation approach was first tested by comparing the apoptosis process in cancer and normal tissue (based upon microarray gene expression data taken from [94]). As shown in Fig. 19, the DFF40 protein, which starts the process of DNA decomposition, is considerably suppressed in cancer cells, relative to non-cancerous cells. The very broad range of transition probabilities for releasing DFF40 under the influence of effector caspases (CASP3 and CASP7), is consistent with the model.

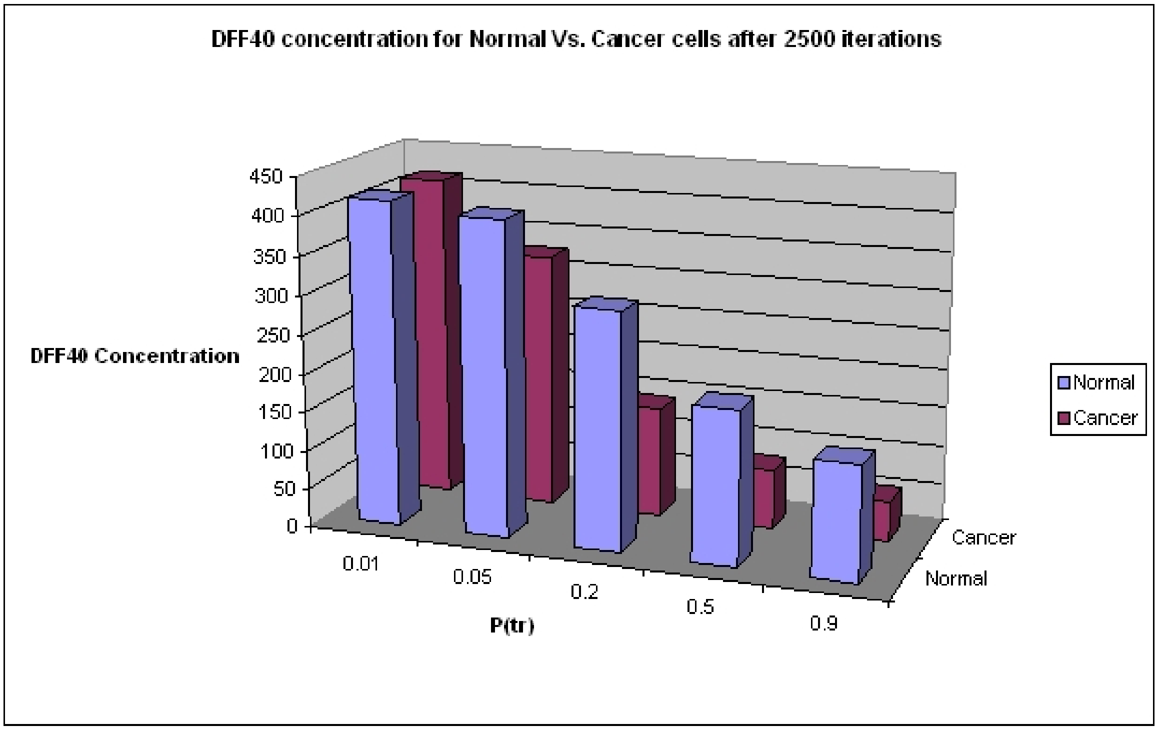


Fig. 19. The strong suppression of apoptosis in cancer cells as compared to normal cells (expression data for the FASL-induced apoptosis from [28]) is reproduced by the cellular automata simulation within a very broad range of values of the transitional probability for releasing the DNA "killer" DFF40.

The strategy for fighting cancer by helping the immune system to restore its response by blocking apoptosis in the immune system T-cells was simulated by varying the potency of FLIP and IAP inhibitors. This was modeled by varying the transition probability for each inhibitor, with an increase in the probability value correlating to a higher inhibitor potency. As seen in Fig. 20, when the two inhibitors were used jointly, the resulted concentration of DFF40 was considerably lower than the one expected from a summing up of the individual effects of each inhibitor. Thus our model predicts a considerable synergy in the joint action of the two inhibitors.

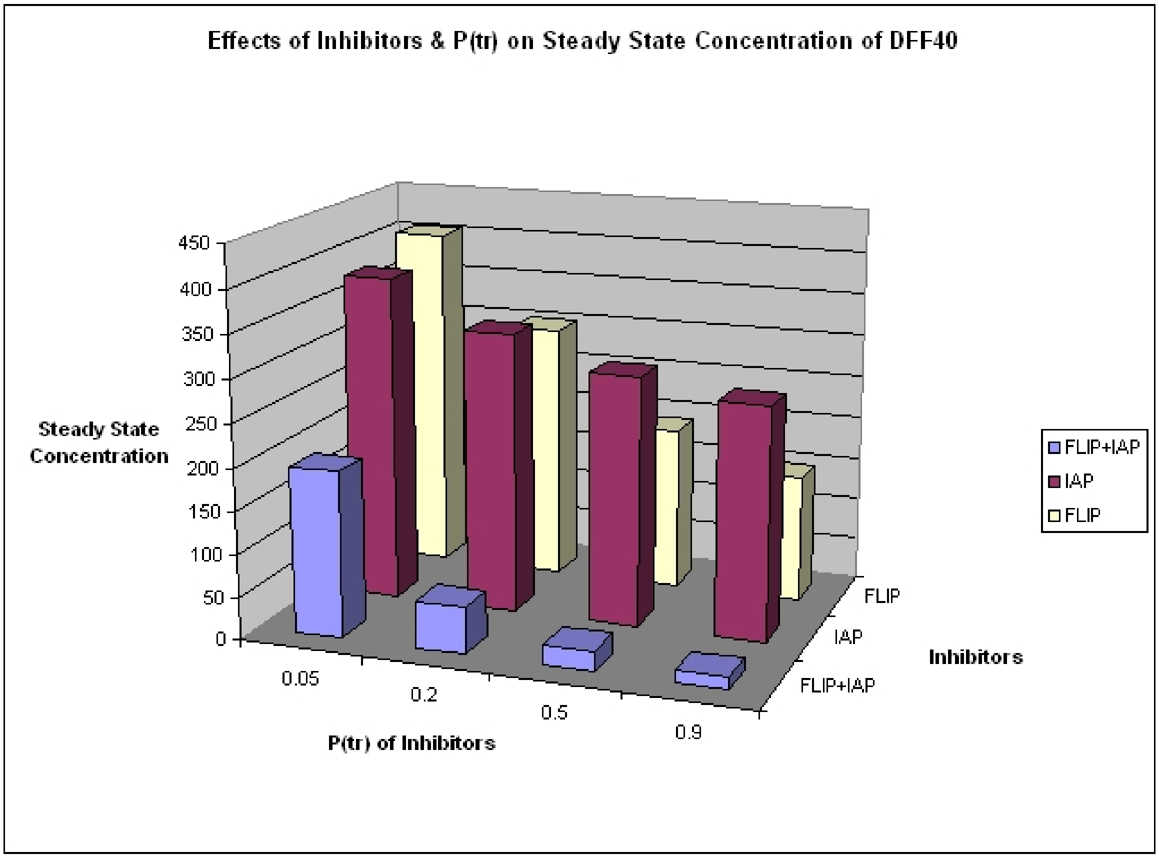


Fig. 20. Synergistic effect during joint suppression of the apoptotic process by the FLIP and IAP inhibitors. This finding might be of interest in developing clinical treatments preventing the killing of the immune system T-cells by cancer cells.

The classical strategy to fight cancer by inducing apoptosis in cancerous cells was simulated by maximizing the DFF40 expression level. This was achieved by modulation of the FLIP and IAP inhibitors of apoptotic process performed by varying the transitional probability for their suppressors siRNA and SMAC (Fig. 17). The benefits of simultaneous suppression of the two apoptosis inhibitors, FLIP and IAP, by siRNA and respectively SMAC, are demonstrated in Fig. 21 with the almost complete release of DFF40 from the synergy of the two models of apoptosis modulation. Although FLIP is a stronger inhibitor than IAP by acting on the first part of the signaling cascade, its silencing cannot enable full scale apoptosis. Only the joint suppression of both inhibitors (FLIP and IAP) was able to kill target cancer cell through apoptosis.

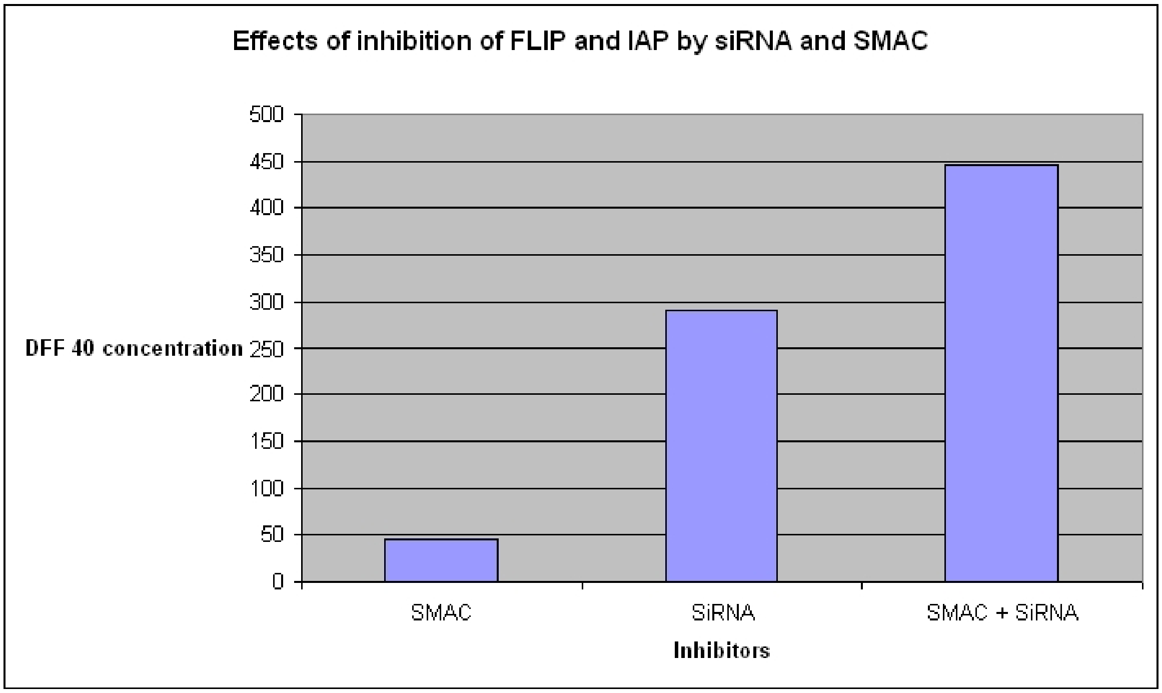


Fig. 21. Simulating the effect of FLIP and IAP inhibitors suppression by siRNA and SMAC, respectively. The values of the DFF40 steady state concentration after 25000 model iterations enable prediction that a full-scale FAS-ligand induced apoptosis is achievable only via joint synergistic suppression of FLIP and IAP inhibitors.

Building upon these simulation results, a more complete model of apoptosis was created by integrating the FAS-ligand-induced apoptosis pathways with the endogenous mitochondria-activated apoptosis pathways. Cells undergoing apoptosis by FASL mediated pathways are called type I, while those committing suicide by the mitochondria-mediated cascade are classified as type II [80, 85]. Apoptosis of type I cells requires high expression of caspase 8 (CASP8), while apoptosis of type II cells proceeds at low levels of caspase 8. In the case of type II cells, caspase 8 activates the Bid protein, which in turn activates the mitochondria mediated apoptosis pathway (Fig. 22). CASP8 thus acts as a switch between type I and type II cells. The mode of action for the mitochondria mediated apoptosis pathway is that mitochondria releases cytochrom C into the cytoplasm activating caspase 9, which then closes the apoptosis chain by activating caspase 3. In addition, a feedback loop from caspase 3 to caspase 9 to IAP has been hypothesized [95-97] that deactivates IAP.

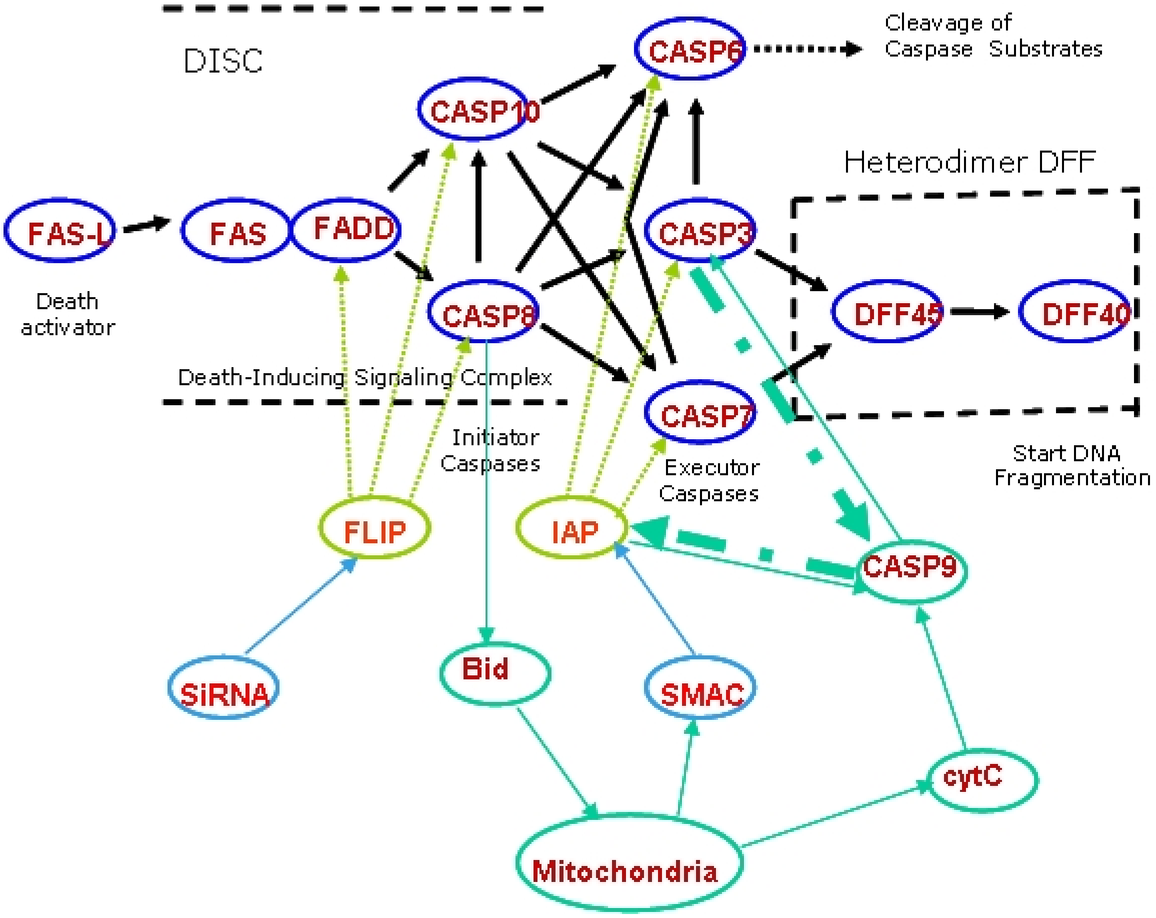


Figure 22. Combined action of the exogenous (FASL-induced) and the endogenous (mitochondria-induced) apoptosis pathways that is active or inactive depending upon the expression level of CASP8. The mitochondria releases cytochrome C in the cytoplasm, which activates CASP9, which in turn activates CASP3 merging thus to the FAS-cascade. A feedback from CASP3 to CASP 9 [95-97] increases the concentration of CASP9 active form, which accelerates apoptosis by suppressing the inhibition from IAP

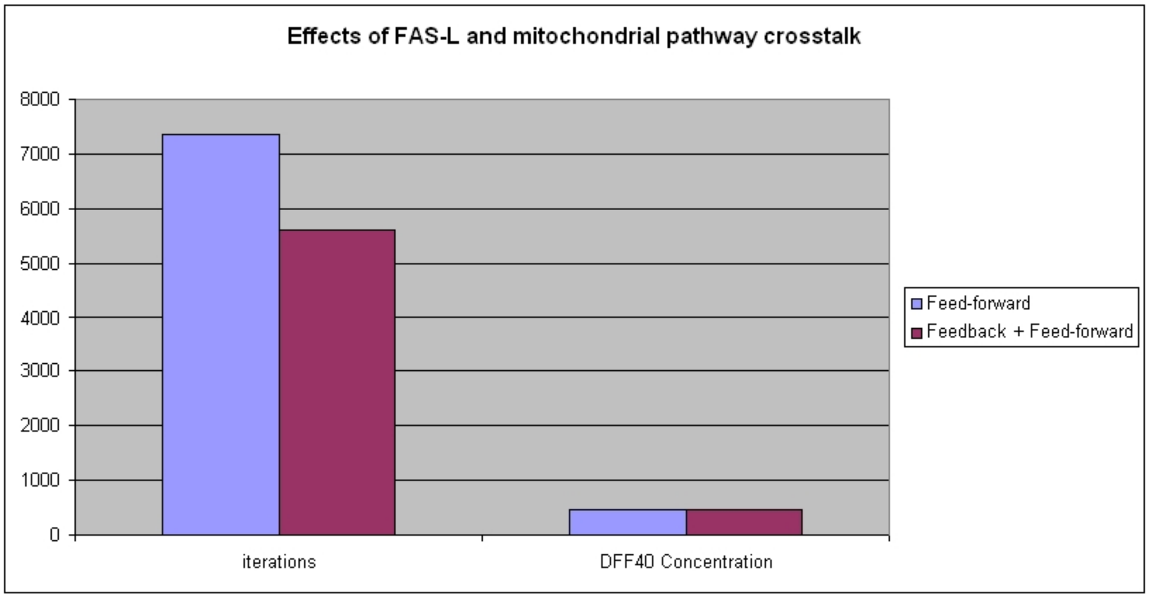


Figure 23. Adding a feedback loop to the mitochondria mediated apoptosis pathways does not increase the DFF40 concentration, which is close to the maximal one needed for DNA decomposition, but speeds up the apoptotic process.

As seen in Figure 23, adding the feedback loop CASP3 → CASP9 --| IAP to the mitochondria mediated apoptosis pathway did not produce major changes in the concentration of DFF40. DFF40 was still present at sufficiently high levels for a cell to undergo apoptosis. Our model predicts that the real benefit of this feedback is that the overall signaling process is in speeding up the process from 7300 to 5600 iterations, due to enhanced suppression of inhibitor IAP and extra activation of CASP9.

Figure 23 presents the chance to compare the dynamics of the intrinsic mitochondria mediated apoptosis pathway (with a feed-back and without it) to that of the extracellular FASL-induced apoptosis. Our simulation shows that FAS-L is 32 % faster than mitochondrial feed forward and 12 % faster than mitochondrial feed-forward + feed-back (The number of iterations for the FASL apoptosis is 5012±11.7, vs. 5596±11.1 and

7368±13.0 for the mitochondrial apoptosis with and without feedback, respectively). This relates apoptosis in type I and type II cells apoptosis in the following manner. Depending on the urgency to carry out apoptosis, cell may control the expression of caspase 8 by using lower levels of CASP8 to activate the mitochondrial mediated pathway and higher levels to activate FASL pathway. In the event of a failure of the mitochondria mediated apoptosis pathway due to damage, caspase 8 could be highly expressed to respond to a FASL stimulus. On the other hand, failures in the FASL apoptosis pathway (such as no DISC formation or mutated membrane bound FAS, etc.) could be compensated by switching on the mitochondrial pathway with feedback, which is only 12% slower at about the same DFF40 concentration. Thus, there is built in biological redundancy to activate apoptosis where the different pathways have a similar (but not identical) signal transduction speed controlled by CASP8. This indicates the need of further modeling studies on controlling caspase 8 as toggle switch between intrinsic and extrinsic pathways.

3.4 Conclusions

This study of FAS-L induced apoptosis focused on two primary objectives; to activate apoptosis in cancer cells and to inhibit apoptosis in normal cells in order to fight cancer. The goal of activating apoptosis was achieved by simultaneously inhibiting the inhibitors of the apoptosis, FLIP and IAP by siRNA and SMAC respectively which produced significant amount of DFF40 required for proceeding to apoptosis. It was shown that individual inhibition of inhibitors is not sufficient and hence synergistic inhibition is proposed as most effective strategy to activate apoptosis mechanism [98, 99]. The problem of inhibiting apoptosis in immune system cells can be addressed by simultaneously inducing the expressions of inhibitors of apoptosis, FLIP and IAP, sufficient so as to block the caspase cascade mechanism from going forward. We propose this as the most effective strategy for sensitizing cancer cells for chemotherapy. It is evident that lower expression of caspase 8 is enough to trigger the mitochondrial pathway while FAS-L induced apoptosis requires higher levels of caspase 8. In the case of higher caspase 8 concentration the mitochondrial pathway does not produce significant change in the concentrations of DFF40 but in case of failure on the part of *intrinsic* apoptosis mechanism due to damaged mitochondria, the cells can be induced to express higher levels of caspase 8 to activate the *extrinsic* pathway as a rescue mechanism. These findings can prove pivotal in identifying drug design targets and required expression levels of specific proteins to suggest drug design strategies to fight cancer and HIV infection.

CHAPTER 4 MODELING CELLULOSE HYDROLYSIS

4.1 Introduction

In all the models studied so far, the strength of CA modeling was evident in modeling complex network structure with easy and accurately predicts the dynamics of such networks. Another strength of CA, its ability to account for physical considerations, can be applied to model cellulose hydrolysis of cellulytic bacteria. In cellulose hydrolysis, there are many stochastic events such as release of cellulase enzymes, binding of cellulase and bacteria to cellulose and random movements of bacteria and cellulose. These processes take place on different temporal scales. CA is well suited to capture evolution of such a system with multiple temporal scales due to its ability to confer individuality to every component of the system. This allows different reactions with different rates to occur simultaneously contributing to the overall system evolution. Hence choice of CA was natural to model cellulose hydrolysis.

4.2 Background

Cellulolytic bacteria express cellulases, enzymes that hydrolyze cellulose, for breaking down cellulose into glucose and cellobiose to be used as energy source in the absence of usual glucose sources [100-103]. The cellulosic biomass is hydrolyzed by enzymes in a coordinated manner involving synergistic actions of different types of cellulases such as exogluconases, endoglucanases and beta-glucosidases along with hemicellulases and accessory proteins. Different cellulytic bacteria have been shown to possess two different enzyme systems of cellulose hydrolysis viz. complexed and non-complexed [104, 105]. Organisms with complexed systems form a complex structure on their cell walls called cellulosome. All the required cellulases are expressed in the cellulosome instead of secreting them in the outer cellular matrix. These organisms then attach to cellulose with the help of dockerin proteins and hydrolyze cellulose. On the other hand, organisms with non-complexed enzyme system, secret cellulases freely in the outer cellular space to degrade cellulose. Cellulose is a long polymer made up of cellobiose which in turn is composed of two glucose molecules. These cellobiose molecules form the building blocks of cellulose. Cellulose chains can vary between few hundred to few thousand cellobiose units depending on the biomass. Structurally cellulose consists of an amorphous core with crystalline cellulose forming the reducing and non reducing ends. Endoglucanases are shown to hydrolyze the amorphous center while exogluconases degrade the reducing and non reducing ends. Beta-glucosidases usually hydrolyze smaller chains of cellulose made up of six or fewer cellobiose units [100].

It is believed that organisms having complexed cellulose hydrolysis enzyme systems degrade cellulose more efficiently that non-complexed organisms. This is made possible due to the fact that all enzymes required for cellulose degradation are available in the cellulosome. This close physicality thus enables complexed organisms to achieve better efficiency at degrading cellulose. Hence a model of cellulose hydrolysis by complexed and non-complexed enzyme systems to evaluate the efficiency of the two systems irrespective of the underlying characteristics of a particular organism was built. Some organisms are shown to possess both the systems together where they express almost all of the cellulases in cellulosome and also secrete few cellulases in outer cellular matrix [106]. This fact was also taken into consideration while comparing enzyme systems and effects of having both enzyme systems was evaluated.

4.3 Methods

The agent based modeling (ABM) method was used to study cellulose hydrolysis. The model was built using Netlogo version 4.1RC5, *October 13, 2009* [107]. Following assumptions were made while constructing the model.

Initially, the idea for modeling the two different enzyme systems can be seen in figure 24.

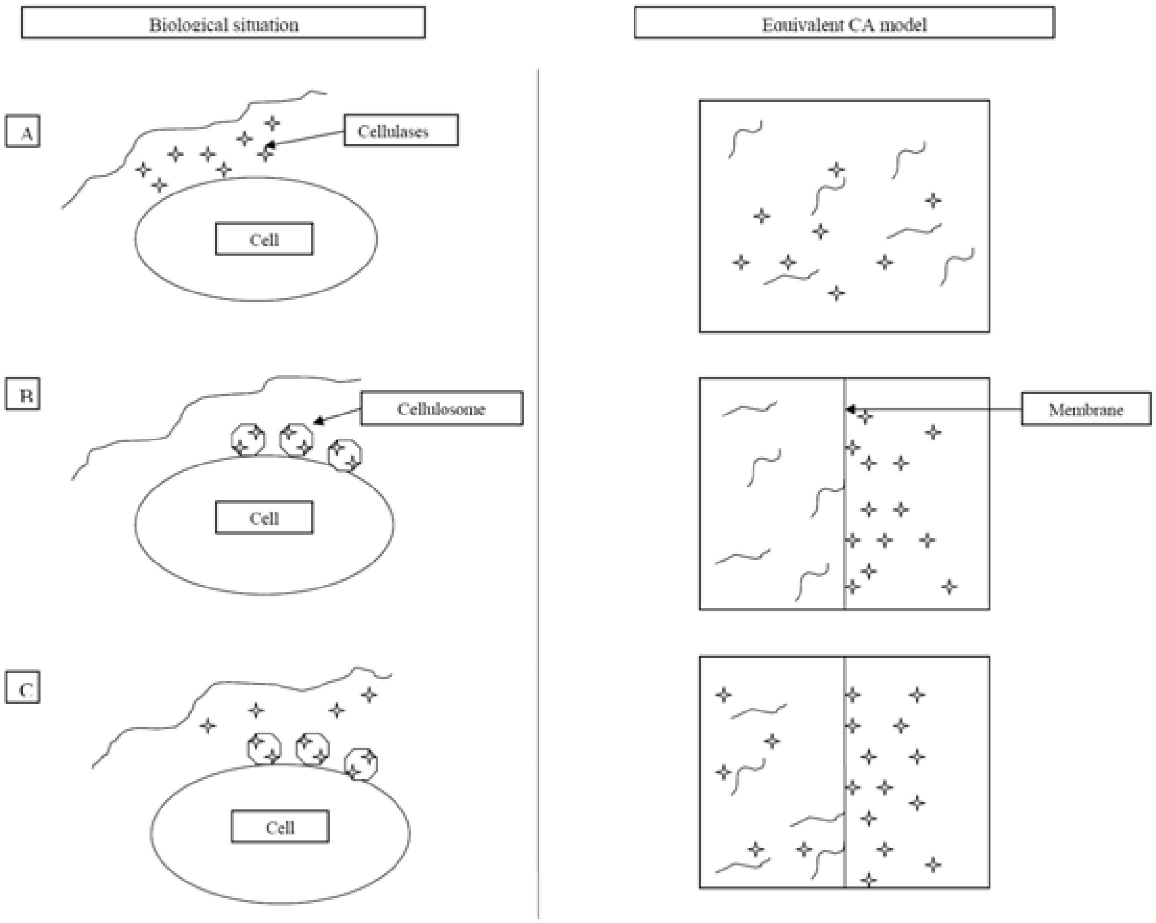


Figure 24: The initial scheme for cellulose utilization model. A represents noncomplex, B is complexed and C is a hybrid enzyme system for cellulose utilization.

Here, the complexed and hybrid systems were thought to be modeled as a two compartment system while non-complexed system as a single compartment. But the compartmentalization proved to be very difficult to model using CA, as it required some editing of the CASim simulator. ABM offered the possibility to model complexed system with two compartments but only one such bacterial cell was possible to be shown making the model not comparable with non-complexed system.

Another attempt was made to model the complexed system assuming the metric sides as cell walls and having the cellulases fixed on the walls forming enzyme clusters. This resolved the issue of having single bacterial cell in complexed system but raised other issues. This scheme was not biologically realistic as cellulases have no movement in this model while cellulose with much bigger mass was the only moving component in the system. In reality cellulose moves much slower than the cellulase and bacteria as its volume is an order of magnitude greater than cellulases. This again makes the two models for complexed and non-complexed systems incomparable.

These issues are addressed in the current version of the model where in case of complexed system the cellulytic bacteria are assumed to have already expressed the enzyme clusters on their cell walls and move faster than the cellulose. These bacteria can then attach to cellulose and start hydrolysis. To make the non-complexed system comparable, it is built with a “enzyme release rate” parameter which compensates the advantage of complex system by having cellulases already expressed on the cell wall.

4.3.1 Model Assumptions:

1) Hydrolysis is the rate limiting step in cellulose break down [108, 109]. Hence activity of cellulases is comparable in cases of complexed and non-complexed enzyme systems.

2) All movement and interactions are random.

3) Cellulase enzymes have a “half-life”, currently equal to 100 time steps.

4) Presence of cellulose has no effect on movement of bacteria.

5) There is no preferential reattachment of complexed bacteria.

6) Rate of cellulose degradation of cellulases is equal regardless of enzyme system.

7) Order of release of particular type of enzyme in non-complexed system is random and enzyme concentrations throughout the simulation are based on initial concentration ratios.

4.3.2 Parameters

Many parameters affecting cellulose hydrolysis were taken into consideration.

1) Cellulose Related Parameters

Number of cellulose chains and their lengths can be adjusted in the model. Considering the fact that larger chains of cellulose will be less mobile in a solution compared to smaller chains, cellulose movement was set inversely proportional to its size to mimic the chain length effect.

2) Enzyme Related Parameters

Enzyme activity for non-complexed and hybrid system can be controlled by setting the rate at which cellulases are released from cells. For the same cases, ratios for enzyme compositions can also be set which can mimic the actual physiological expression levels of organisms. A “half-life” can also be set for enzymes to mimic their biological behavior in outer cellular matrix. The activity of enzymes is controlled by their interaction probability which defines how probable it is for an enzyme to hydrolyze cellulose if interaction is possible by physical closeness.

3) Organism Related Parameters

Behavior of the organisms can be controlled by setting their population and probability of intake for particular type of sugar molecules.

4.4 Results and Discussion

4.4.1 Effects of Initial conditions

Initial conditions of any ABM can greatly affect the steady state of the model making it difficult to predict outcome. To assess the effects of bacterial concentration on model predictions, number of cells was varied between 10 and 100 in steps of 10. As seen in figures 25A, 26A and 27A, time steps required for complete break down of cellulose varies greatly with low starting concentration of bacteria. Standard deviation was measured for these simulations and it is evident from figures 25B, 26B and 27B for standard deviations that the randomness induced in the system due to low bacterial concentrations, decreases with increasing bacterial concentration, and standard deviations are steadily decreasing after increasing bacterial concentration beyond 60 cells. Hence a bacterial concentration of 60 was chosen for further simulations.

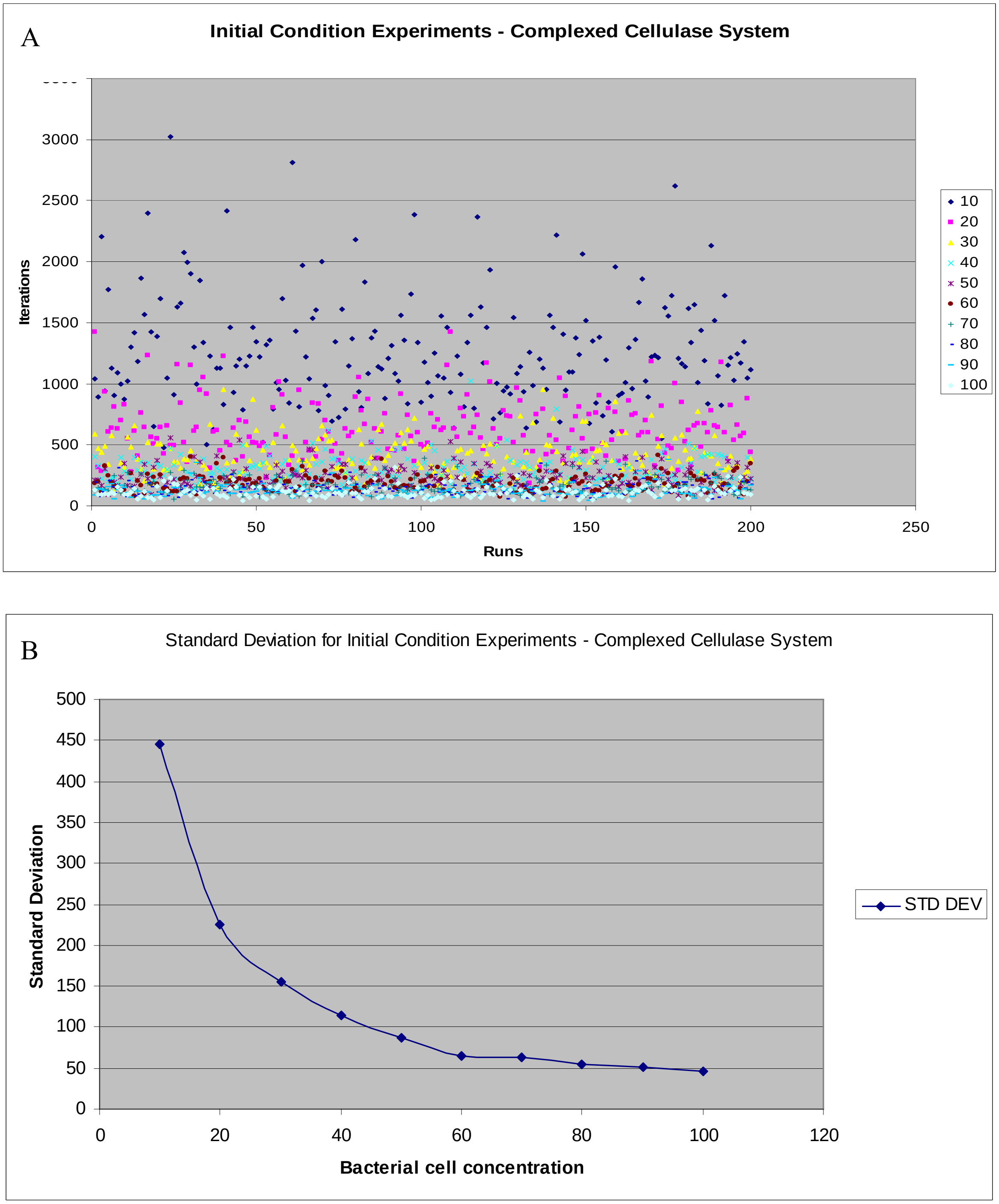


Figure 25: A) Effects of initial conditions on complexed cellulase enzyme system. Varying bacterial cell concentration from 10 to 100 in steps of 10. B) Standard deviation

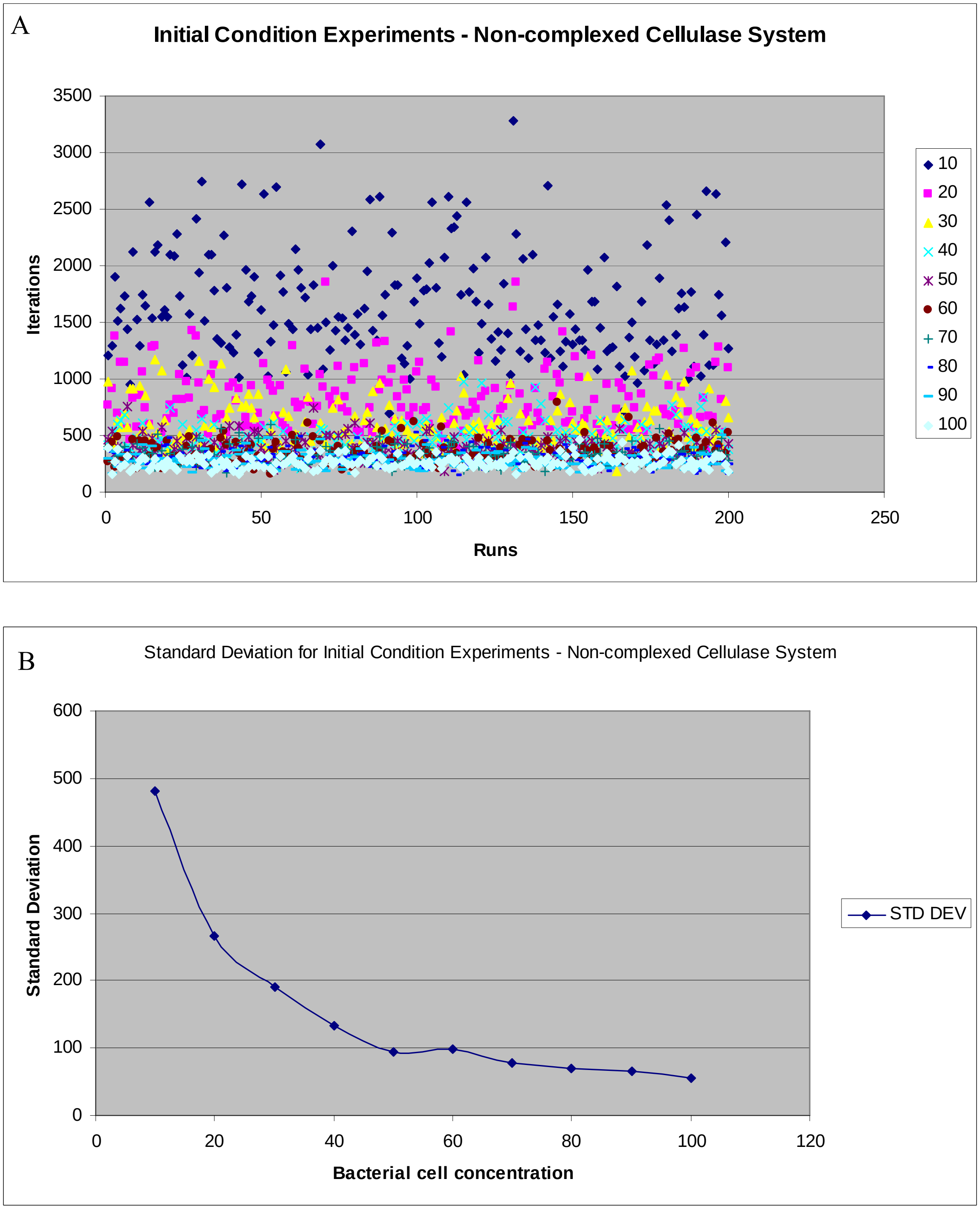


Figure 26: A) Effects of initial conditions on non-complexed cellulase enzyme system. Varying bacterial cell concentration from 10 to 100 in steps of 10. B) Standard deviation

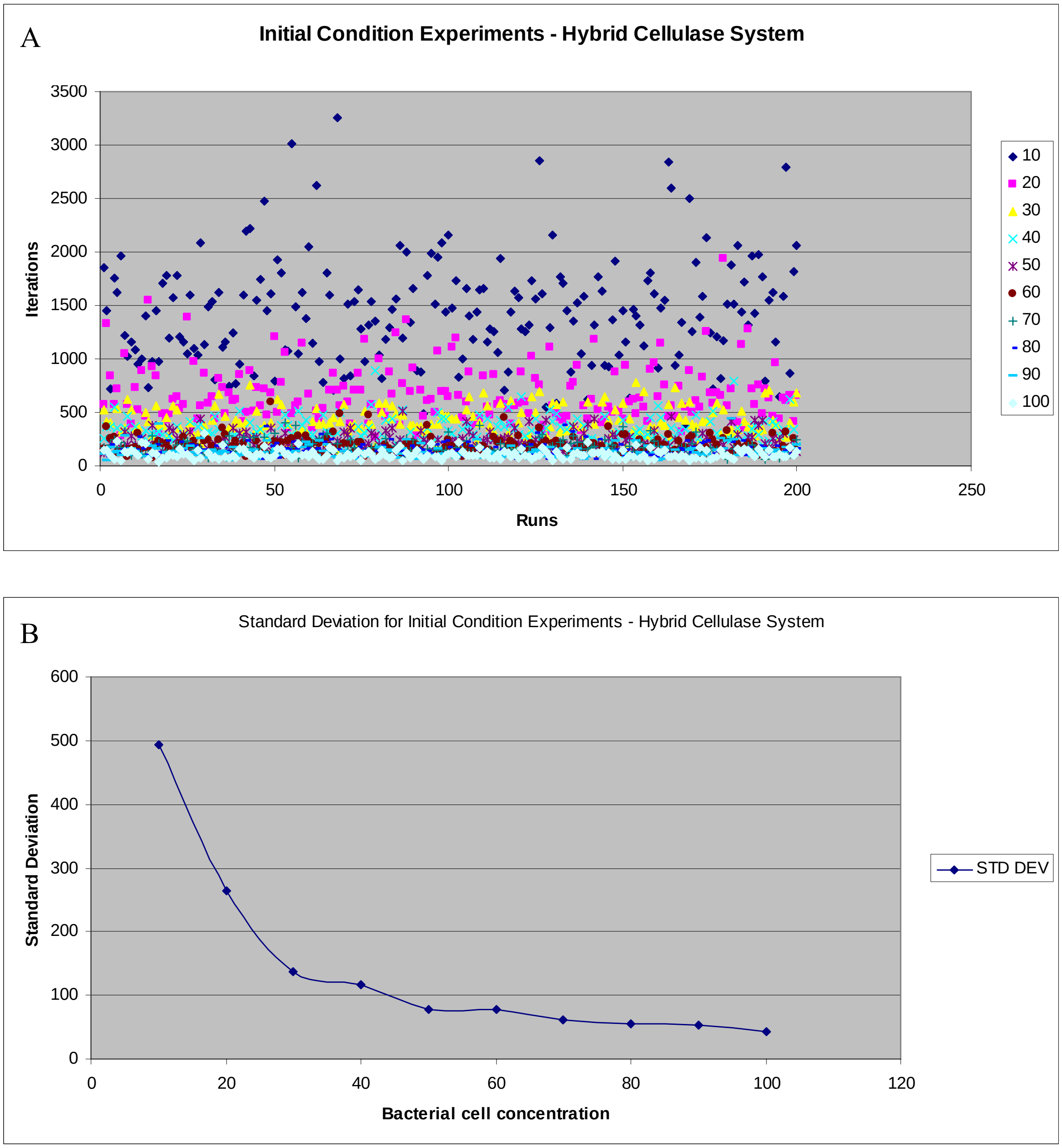


Figure 27: A) Effects of initial conditions on hybrid cellulase enzyme system. Varying bacterial cell concentration from 10 to 100 in steps of 10. B) Standard deviation

4.4.2 Effects of rate of enzyme release

To make the expression of enzymes in non-complexed system comparable with the complexed system, rate of enzyme release was assessed by increasing the rate from 10 to 100 in steps of 10 followed by statistical analysis to set the enzyme release rate for non-complexed system such that both the systems perform equally while hydrolyzing cellulose. The comparable rate was found to be 60.

4.4.3 Effects of simulating real bacterial enzyme ratio

Published data [110-113] shown in table 8 for the ratios of cellulases in Cellulytic organisms was used to assess whether having a particular system is beneficial for the organism to hydrolyze cellulose. Interaction probability of an enzyme to hydrolyze cellulose upon encounter is kept 0.5 for all enzymes to make the enzyme action comparable within all system. In the case of C. thermocellum with hybrid system, an enzyme production rate of 10% (6 compared to 60 for other organisms) was used based on the fact that only 2 putative non-cellulosome bound cellulases are expressed in C. thermocellum as compared to expression of about 20 different cellulases in other organisms, which is about 10 % [106, 111-113].

Table 8. Cellulase enzyme ratios used for simulating bacterial enzyme systems

	T. fusca	***T. reesei***	***C. thermocellum-hybrid***	***C. thermocellum-complexed***
Exoglucanases	70	80	40	0
Endoglucanases	20	15	60	0
Beta-glucosidases	10	5	0	0
enzyme interaction probability	0.5	0.5	0.5	0
enzyme production rate	60	60	6	0

An F-test was conducted to test the null hypothesis stating that “There is no difference hydrolyzing cellulose due to particular enzyme system” The results for F-test are presented in table 9.

Table 9: Performance of bacterial cellulase systems hydrolyzing cellulose

Organism	*Runs*	*Average**	*Variance*	*Deviation*
T. fusca	200	302.93	7247.985	85.135
T. reesei	200	280.045	5622.636	74.984
*C. thermocellum Hybrid***	200	190.345	6650.066	81.547
C. thermocellum Complexed	200	188.015	4246.658	65.166

* Average number of iterations required to completely hydrolyze cellulose over 200 repetitions
** *C. thermocellum* treated as hybrid by expressing non-cellulosome bound cellulases

From table 9 it is evident that *T.fusca* and *T.reesei*, organisms that possess non-complexed enzyme system require almost 100 iterations (nearly 50 %) more than *C.thermocellum* which as a complexed enzyme system. Allowing *C.thermocellum* to secrete cellulases in the outer cellular matrix in addition to its complexed enzyme system takes almost equal amount of iterations for cellulose hydrolysis, which is also evident from the F statistic. But the statistical analysis of these four simulations does not support the averages strongly. In spite of having very close average performance, the F statistic shows *T.fusca* and *T.reesei* to be different followed by differentiating *C.thermocellum* treated as complexed and hybrid. On the other hand *T.fusca* and *C.thermocellum* hybrid systems are shown as performing similar by the F statistic.

Tables 10 and 11 summarize the statistical simulation results for the bacteria *T.fusca, T.reesei* and *C.thermocellum*.

Table 10: F-test summary of bacterial systems with each other

Organism	*T. fusca*	*T. reesei*	*C. thermocellum Hybrid*	*C. thermocellum Complexed*
T. fusca		1.26334* 1.2890**	1.26334* 1.089912**	1.26334* 1.70675**
T. reesei			0.791552* 0.845501**	1.26334* 1.324014**
C. thermocellum H				1.26334* 1.565953**
C. thermocellum C				

* F critical one tail
** F value

Table 11: Comparison summary of bacterial systems with each other

Organism	*T. fusca*	*T. reesei*	*C. thermocellum H*	*C. thermocellum C*
T. fusca		**different**	**same**	**different**
T. reesei			**different**	**different**
C. thermocellum H				**different**
C. thermocellum C				

4.4.4 Effects of cellulase ratio

After simulating real bacterial enzyme ratios, simulations were performed to query the best performing ratio of cellulases. It is known from published data that concentration of beta-glucosidase does not go beyond 20% [110-113]. Hence three different series of simulations were performed reducing concentration of beta-glucosidase in steps of 10 from 20 to 0 and holding it constant for each series. The remaining percentage of concentration was divided among endoglucanases and exoglucanases increasing and decreasing the concentrations simultaneously. As seen in figure 28, efficiency of the enzyme system increases with increasing endoglucanase and decreasing exogluconase at constant beta-glucosidase and vice versa. All the three series show high correlations for the trend established for iterations required to hydrolyze cellulose.

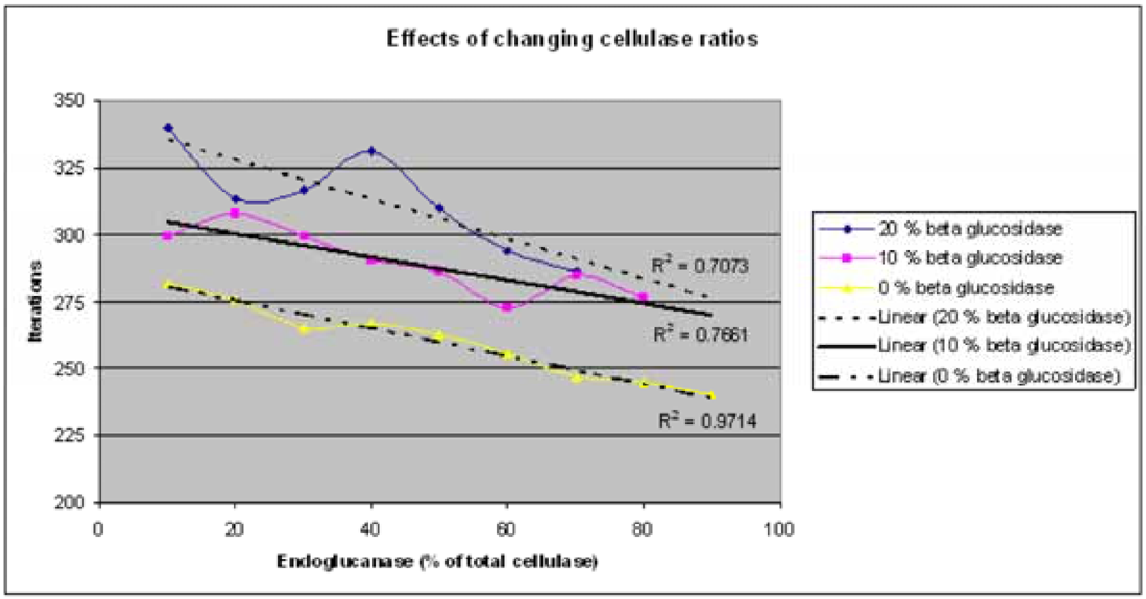


Figure 28: Iterations required for changing ratios of exogluconase and endoglucanase at constant beta-glucosidase.

4.5 Conclusions

The focus of this study was to build model of different cellulase enzyme systems employed by Cellulytic bacteria to hydrolyze cellulose and evaluate them on the basis of efficiency hydrolyzing cellulose. To make the model realistic, effects of initial conditions were assessed, and it can be concluded that the initial conditions greatly affect the model outcome. As the cells are placed randomly in the grid while starting the simulation, physical closeness to begin with can greatly decrease the number of iterations required to achieve complete hydrolysis. A high enough initial concentration of cells can make the model predict the performance of a system more accurately. Though the average number of iterations for cellulose hydrolysis for complexed and non-complexed systems shows significant difference, it is not substantiated by statistical tests which give mixed results while differentiating the two enzyme systems. Hence it can not be concluded based on this data alone that there exists a significant difference in the efficiency of these enzyme systems. The efficiency of hydrolysis could be more dependent on particular organism, its expression capabilities and specific characteristic of its cellulases rather than depending alone on the type of cellulase enzyme system. On the other hand, the test statistics agree with the average number of iterations showing no significant different between complexed cellulase systems vs. having both complexed and non-complexed together. Hence it can be concluded that there is virtually no difference between these systems with regards to efficiency at hydrolyzing cellulose.

From the results of testing different enzyme ratios it can be said that in the case of no beta-glucosidase in the system, endoglucanase concentration is directly proportional to efficiency while exoglucanase concentration is inversely proportional (R^2 = 0.97). Examining the three series, beta-glucosidase concentrations of 10 and 20 show reasonable trend (R^2 = 0.71 and R^2 = 0.76) for above mentioned relationship. This can be explained by the fact that by randomly cutting the amorphous sites, endoglucanases are producing shorter cellulose chains and exposing more reducing and non reducing cellulose ends in the process. It is then easy to determine the optimum cellulase ratio which could be 99% endoglucanase followed by 1% exogluconase and 0% beta-glucosidase if integer values are used for enzyme proportions. As exogluconases are essential for hydrolyzing reducing and non-reducing ends, 100 % endoglucanase is not a feasible solution.

These results, especially the suggested optimal enzyme proportions, can be useful while designing stains of cellulytic organisms for industrial purposes. Experiments can be designed to precisely control the expression of cellulase enzyme genes in cellulytic bacteria to verify these findings and suggest a biologically feasible optimal cellulase composition.

CHAPTER 5 DESIGN AND DEVELOPMENT OF ALGORITHMS TO DETECT MUTATIONS IN BACTERIAL GENOME

5.1 Introduction

The aim of the algorithm development work was to detect and study mutations in bacteria grown over several generations and differentiate them from randomly occurring mutations. Two different organisms namely *E.coli* and *T.fusca* were sequenced using next generation sequencing technology, Solexa Genome Analyzer [114]. Two *E.coli* strain designs grown in replicate were optimized for maximum lactate production and maximum growth using flux balance analysis [115-117]. These were then grown on minimal media for 60 days. Two *T.fusca* strains, one grown on minimal media with cellobiose and the second strain was switched to glucose from cellobiose and both grown for 40 days [117]. These enzymes, termed cellulases, are not expressed when glucose is available as energy source.

The above mentioned strains were run on 8 lanes Solexa Genome Analyzer and the output files generated were analyzed for mutations by designing algorithms. The software employed to achieve this task includes mom-0.2 [118], Python and Matlab.

5.2 Algorithms

Figure 29 shows the flow of logic for the design and development procedure of the algorithm development process. Output from solexa genome analyzer was used an input to extract the raw sequences using sequence extraction algorithm designed in Matlab. These raw sequences were then used by MOM-0.2 re-sequencing software producing the output shown in table 12. MOM-0.2 output was then used to generate the mutations and coverage tables which led to mutation and coverage analysis.

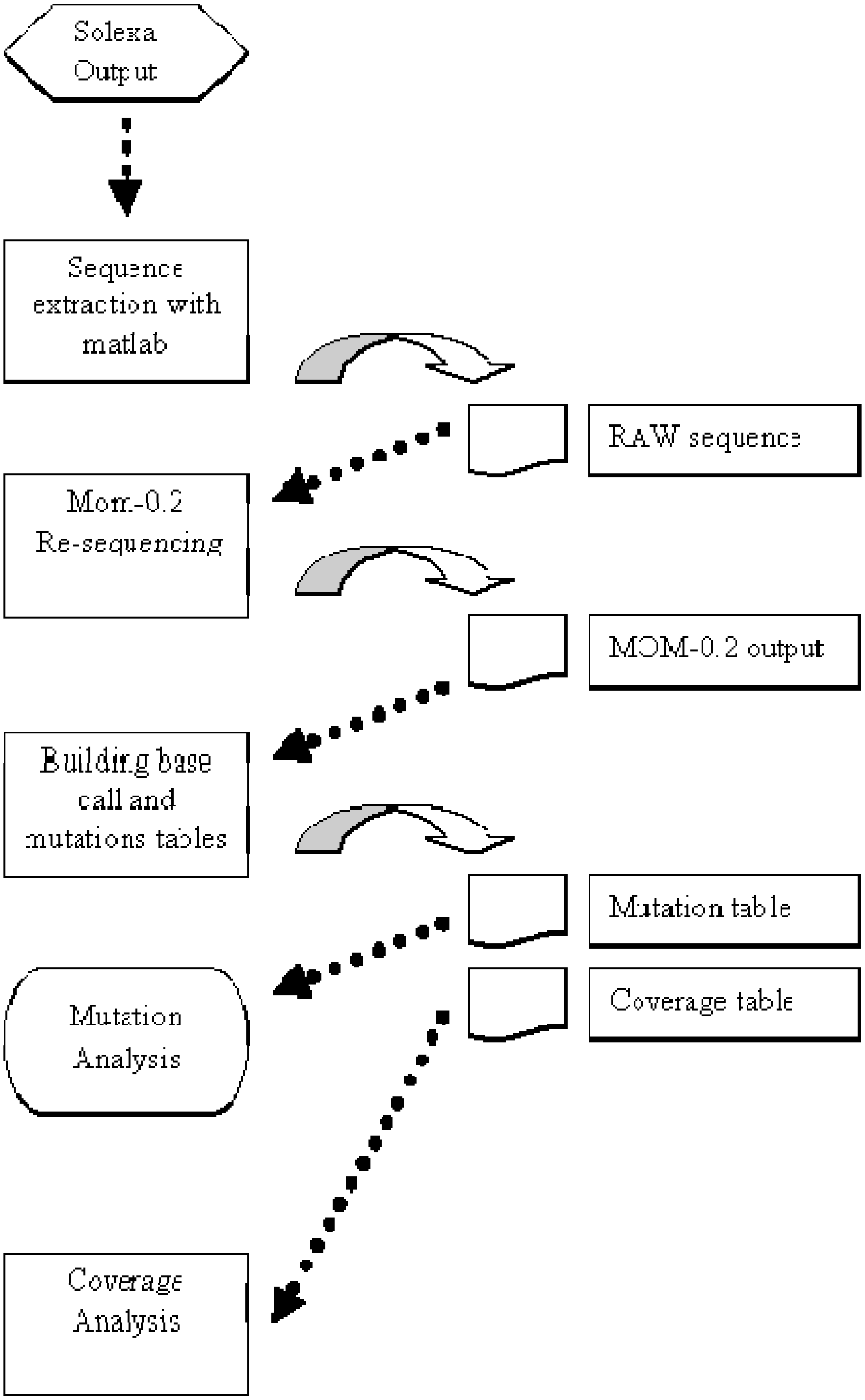


Figure 30. Logic flow for the algorithms to detect mutations

5.2.1 Extraction of RAW sequences

As MOM-0.2 can accept sequences only in “fasta” or “RAW” format, the output from Solexa Illumina was converted into RAW sequences by removing the quality scores and retaining only the sequence reads. The Matlab code for this procedure is included in the appendix B. A snapshot of output generated by Solexa Illumina can be seen below,

```
@HWI-EAS102:2:1:0:361#0/1
NAGCGCGTGCAGTGGGCATTGACCAAAGCCGCATTGTG
+HWI-EAS102:2:1:0:361#0/1
DLSTUSOSUTRTSMJOSTUQKQPSTSRBBBBBBBBBBB
@HWI-EAS102:2:1:1:156#0/1
NATAACCCCAGATCGGCGTACCCCCAATACATCCCACT
+HWI-EAS102:2:1:1:156#0/1
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWI-EAS102:2:1:1:784#0/1
NCATAGCAACGCGCCAGGTTGTCCTGCAGCGGGCGGAC
+HWI-EAS102:2:1:1:784#0/1
DNTUSSSSUTUSSTSPKQSSSSSRPRBBBBBBBBBBBB
```

After applying the algorithm, Solexa output will be converted into raw sequence reads and look like following,

```
NAGCGCGTGCAGTGGGCATTGACCAAAGCCGCATTGTG
NATAACCCCAGATCGGCGTACCCCCAATACATCCCACT
NCATAGCAACGCGCCAGGTTGTCCTGCAGCGGGCGGAC
```

5.2.2 Re-sequencing with MOM-0.2

The mom-0.2 software compares the query sequence with reference genome and produces output file containing mutation information for each read. A sample MOM-0.2 output can be seen in table 12,

Table12. Sample mom-0.2 output.

MOM-0.2 output fragments	**Explanation**
Fong_sample2_s_2_sequence_04_24_09_output.txt:1	Read ID
NAGCGCGTGCAGTGGGCATTGACCAAAGCCGCATTGTG	Matching Read Sequence
U1	Match Type
0	Number of 0 mismatches
1	Number of 1 mismatches
0	Number of two mismatches
gi|48994873|gb|U00096.2| Escherichia coli str. K-12 substr. MG1655 complete genome	The FASTA id of the matching reference sequence
3905971	Match Position
R	Reading Direction
1G	Mismatch base and position within read

5.2.3 Building base call and mutation table

Mutations for the whole genome from the mom-0.2 output file need to be summarized before analysis. This is achieved by reading mom-0.2 output file line by line and extracting information for base calls. It is then compared with the call for each base in the reference genome. This information is then converted into base call table which will consist of total number of calls for each nucleotide for each base followed by the base with maximum call and the reference call for that particular base position. Using this base call table, a mutations table is built only for those positions which include a mutation flag (U1, U2….) in the mom-0.2 output. Different types of mutation flags are used in mom-0.2. A mutation tag starting with "U" means a unique match for the read sequence. It is then followed by a number, 0 for a 100 % match and higher numbers suggesting that many mismatching bases. In case of multiple matches with the reference genome subsequent fields containing number of matching sequences with number of mismatches below the allowable mismatches are populated. Mutations table is similar to base call table except that it only consists of bases with a different maximum nucleotide call from reference genome. A third table built with this algorithm will consist of only entries exceeding 75 % of mismatch criteria with the reference genome along with the coverage for that nucleotide in the Solexa output [119]. The Matlab and Python codes for this procedure are included in the appendix B.

5.2.4 Gene search

As the mom-0.2 output contains reference to the *fasta* ID of reference genome, names of genes with mutations can be extracted by querying the reference genome with base position of mutation or by using the *fasta* ID from mom-0.2 output. The Python code for these procedures is included in the appendix B.

5.2.5 Coverage analysis

After building the base call and mutations tables it becomes easy to perform whole genome coverage analysis. It is important to calculate the coverage for each position in genome as higher coverage indicates a higher probability that the mutations detected are true. The algorithm for calculating coverage simply iterates through the entire base call table and counts the total number of times a base has a nucleotide call, either matching the reference or not. A table for coverage between 1 and 100 was built. Coverage analysis also shows what proportion of the total genome is not covered in the re-sequencing runs. The quality of re-sequencing can also be judged from average coverage of the entire genome. Hence two more procedures for calculating average and maximum coverage were added to the algorithm. The Python codes for these procedures are included in the appendix B.

5.3 Results and discussion

5.3.1 Coverage

Table 13 shows the amount of whole genome coverage. As an *E.coli* genome is about 4.6 million bases, about 400,000 bases are not at all present in the re-sequencing. Also, only about 60 % of the genome re-sequenced is above acceptable coverage threshold assuming that a threshold of 30 fold coverage is sufficient for the results to be treated authentic.

Table 13. Genome coverage for sample 1***.

Coverage*	Bases**
1	4229283
5	3972341
10	3667977
15	3373049
20	3091257
25	2822475
30	2573707
35	2340549
40	2126054
45	1926874
50	1745895
55	1581229
60	1429366
80	951782
100	632796

* Fold coverage for each base, ** Bases with specified and above coverage
*** *E. coli* re-sequenced whole genome

5.3.2 Mutations

Table 14 shows the mutations detected by the algorithms in sample 1. Results are produced for combinations of coverage and percentage of total calls which favor mutation in the base. A threshold of at least 75 % was set for a mutation to be called.

Table 14. Detected mutations for sample 1.

Coverage*	%**	Mutations***
1	0.75	284
1	0.8	261
1	0.85	240
1	0.9	220
1	0.95	202
1	1	200
5	0.75	139
5	0.8	119
5	0.85	98
5	0.9	78
5	0.95	60
5	1	58
10	0.75	103
10	0.8	87
10	0.85	73
10	0.9	57
10	0.95	39
10	1	37
15	0.75	75
15	0.8	59
15	0.85	50
15	0.9	35
15	0.95	24
15	1	22
20	0.75	47
20	0.8	37
20	0.85	31
20	0.9	20
20	0.95	14
20	1	12
25	0.75	27
25	0.8	20
25	0.85	18
25	0.9	9
25	0.95	5
25	1	4
30	0.75	17
30	0.8	13
30	0.85	11
30	0.9	5
30	0.95	2
30	1	2

* Fold coverage for each base, ** Percentage of base calls in favor of mutations
*** Number of mutations detected at specified coverage and percentage

5.4 Conclusions

This re-sequencing study was conducted to detect mutations in the bacterial genome and identify them differently with high confidence from the re-sequencing errors. This goal was achieved by using threshold criteria for coverage and percentage of mismatch calls. It is difficult to set fixed criteria with regards to both the criteria parameters mentioned above as it can vary from genome to genome and depends more on the methods ands precautions used while running the re-sequencing experiments. But in general, a coverage of 30 or above and percentage mismatch of 85 or above seem reasonable criteria to have high confidence in making a mutation call for the eight different re-sequenced samples studied in this section.

CHAPTER 6 CONCLUSIONS

Most of the work contained in this dissertation is centered on development of a computational method to capture the essence of dynamics of biological systems which are otherwise difficult to capture due to greater amount of complexity encountered while modeling these systems. Many modeling methods currently exist to simulate biological systems. Sometime or other while using these methods one has to compromise on some aspects of the system such as level of details to be included in the model in order to reduce the complexity involved. Approximations are often used to deal with these complex biological systems which often miss out, in my opinion, on the important micro-dynamics of the system which eventually decide the system's temporal evolution. Many biological systems are stochastic in nature with different scales of temporal dynamics which further limits the ability to build a model which can accurately predict the emergent behavior of biological systems. Modeling methods which can actually account for molecular level details are limited into their application to model system level dynamics due to the extraordinary number of components that make living organisms.

After working for over four years in systems biology, my goal while modeling a biological system is to understand the holistic nature of the system under question rather than characterizing each individual component. This "big picture" provides general understanding about the functioning of the system and allows capturing the essential and interesting features of the system. In these regards, I have found the method of Cellular automata or agent based modeling more appealing than any other modeling methods. A basic knowledge about the working of a biological system, or any system for that matter, is sufficient on many occasions to model it using CA/ABM. Its strength lays in its simplicity to represent any kind of stochastic system with few basic rules or principles governing the dynamics of the system without overly emphasizing on spurious details. I agree with Stephen Wolfram on the idea that many complex behaviors originate from very simple and basic underlying rules which decide the temporal evolution of the system. Answers to many questions in biology can be found by following these simple governing principles in nature and cellular automata has shown its strength in successfully incorporating these basic design principles into system level models.

Still, many shortcomings of CA modeling can be pointed out. Though the basic notion is to simulate a biological system as it appears in its natural setting, currently there exists hardly any simulation environment which can sufficiently account for the three dimensional biological environment. Hence mostly two-dimensional spaces are used to mimic three dimensional biological environments. Though two dimensional matrices appear to be sufficient at this point of time, the simulation can not be called 100 % realistic while using one less dimension. As modeling biological systems with CA is a relatively new field, further interest in these type of modeling and advances in computer processor technology can greatly improve the chances of having a truly three dimensional modeling environment. A second major disadvantage is that CA modeling can not account for structural details of the components in a system making it relatively useless for modeling structural events like binding of proteins and other biological molecules.

Different avenues exit to take the current research further with CA modeling. I would like to suggest a few directions in which I would like to advance the knowledge of biological systems, using cellular automata. Firstly, synthetic biology will benefit greatly with the concept of isodynamicity. Industrial organisms are used generation after generation and, therefore, evolution must be taken into account when designing biological circuits. Isodynamicity can contribute by endorsing desired characteristics to synthetically constructed strains. Strains can be designed to specifications by optimizing cost of gene expression, reaction rates, signaling responses, regulatory controls and robustness against random mutations by incorporating isodynamic networks in the biological circuit design process. Cellular automata simulations can be brought in along with traditional experimental research to evaluate the different synthetic circuit designs before implementation. A higher cost associated with testing all possible biological circuits via experiments can be significantly reduced by selecting best performing designs as suggested by cellular automata simulations.

Secondly, the pivotal role of caspase 8 was shown as a regulatory switch between type I and type II cells. Lower concentrations of caspase 8 is sufficient to trigger mitochondria mediated apoptosis while higher concentrations of caspase 8 are required to respond to the FAS-L induced apoptosis signal. Hence further simulations of CA will be designed with the goal to turn a type II cell into Type I cell by increasing the expression of caspase 8. This will allow cells with failed mitochondrial apoptosis machinery to carry out apoptosis via FAS-L induced apoptosis, establishing FAS-L as a alternative route for self induced death. Also to characterize fully the mitochondrial pathway, inhibitors of the BCL-2 family will be modeled in the simulations to test the synergy between intrinsic and extrinsic pathways. This will enable selective switching between different apoptosis mechanisms suggesting the role of other proteins of these pathways as new drug targets. The strength of CA/ABM modeling can be utilized best in conjunction with other types of modeling methods to create hybrid simulation platforms. Whole cell models then can be constructed by applying appropriate modeling methodology to candidate processes depending on the inherent nature of the process and capturing every possible spatial and temporal aspect of the system dynamics.

CHAPTER 7 REFERENCES

1. Elaine R. Mardis, The impact of next-generation sequencing technology on genetics, *Trends in Genetics* Volume 24, Issue 3, March 2008, Pages 133-141

2. Stephan C Schuster, Next-generation sequencing transforms today's biology, *Nature Methods*, Vol.5 No.1 January 2008

3. Elaine R. Mardis, Next-Generation DNA Sequencing Methods, Vol. 9: 387-402, June 24, 2008

4. Olena Morozova[a] and Marco A. Marra, Applications of next-generation sequencing technologies in functional genomics, *Genomics* Volume 92, Issue 5, November 2008, Pages 255-264

5. Andreas von Bubnoff, Next-Generation Sequencing: The Race Is On, *Cell*, Volume 132, Issue 5, 721-723, 7 March 2008

6. Douglas R. Smith et al, Rapid whole-genome mutational profiling using next-generation sequencing technologies, *Genome Res*. 2008. 18: 1638-1642

7. Hiroaki Kitano, Systems Biology: A Brief Overview, *Science* 1 March 2002: Vol. 295. no. 5560, pp. 1662 – 1664

8. Wu J, Voit E., Hybrid modeling in biochemical systems theory by means of functional petri nets, *J Bioinform Comput Biol*. 2009 Feb;7(1):107-34

9. Hardy S, Robillard PN., Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches, *J Bioinform Comput Biol*. 2004 Dec;2(4):595-613

10. Zaiyi Guo a, PeterM.A.Sloot b, JocCingTay, A hybrid agent-based approach for modeling microbiological systems, *Journal of Theoretical Biology* 255 (2008) 163–175

11. Oliver Kotte1,2 and Matthias Heinemann, A divide-and-conquer approach to analyze underdetermined biochemical models, *Bioinformatics, Vol. 25 no. 4 2009, pages 519–525*

12. M. Hucka et al, The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models, *Bioinformatics* Vol. 19 no. 4 2003 Pages 524-531

13. Casey et al., 2006 R. Casey, H. de Jong and J.-L. Gouzé, Piecewise-linear models of genetic regulatory networks: equilibria and their stability, *J. Math. Biol.* 52 (1) (2006), pp. 27–56

14. Coutinho et al., 2006 R. Coutinho, B. Fernandez, R. Lima and A. Meyroneinc, Discrete time piecewise affine models of genetic regulatory networks, *J. Math. Biol.* 52 (2006), pp. 524–570.

15. Glass and Kauffman, 1973 L. Glass and S.A. Kauffman, The logical analysis of continuous non-linear biochemical control networks, *J. Theor. Biol.* 39 (1973), pp. 103–129

16. Advait A Apte, John W Cain, Danail G Bonchev and Stephen S Fong. Cellular automata simulation of topological effects on the dynamics of feed-forward motifs. *Journal of Biological Engineering* 2008, 2:2.

17. Dewey T. Taylor, John W. Cain, Danail G. Bonchev, Stephen S. Fong, Advait A. Apte, and Lauren E. Pace, Toward a Classification of Isodynamic Feed-Forward Motifs, *Journal of Biological Dynamics*, 99999:1, 28 July 2009

18. Tornøe CW, Overgaard RV, Agersø H, Nielsen HA, Madsen H, Jonsson EN., Stochastic differential equations in NONMEM: implementation, application, and comparison with ordinary differential equations, *Pharm Res.* 2005 Aug; 22(8):1247-58

19. Klim S, Mortensen SB, Kristensen NR, Overgaard RV, Madsen H., Population stochastic modelling (PSM)--an R package for mixed-effects models based on stochastic differential equations, *Comput Methods Programs Biomed*. 2009 Jun;94(3):279-89

20. Goutsias J, Classical versus stochastic kinetics modeling of biochemical reaction systems, *Biophys J.* 2007 Apr 1;92(7):2350-65

21. Fong SS *et al.*, In silico design and adaptive evolution of Escherichia coli for production of lactic acid, *Biotechnol Bioeng* 91:643–8, 2005.

22. Fong SS, Palsson BO, Metabolic gene-deletion strains of Escherichia coli evolve to computationally predicted growth phenotypes, *Nat Genet* 36:1056–8, 2004.

23. Ibarra RU *et al.*, Quantitative analysis of Escherichia coli metabolic phenotypes within the context of phenotypic phase planes, *J Mol Microbiol Biotechnol* 6: 101–8, 2003.

24 . Wiback SJ, Mahadevan R, Palsson BO, Using metabolic flux data to further constrain the metabolic solution space and predict internal flux patterns: the Escherichia coli spectrum, *Biotechnol Bioeng* 86:317–31, 2004.

25. Famili I, Palsson BO, Systemic metabolic reactions are obtained by singular value decomposition of genome-scale stoichiometric matrices, *J Theor Biol* 224:87–96, 2004.

26. Covert MW *et al.*, Integrating high-throughput and computational data elucidates bacterial networks, *Nature* 429:92–6, 2004.

27. Bonarius HPJ, Schmid G, Tramper J, Flux 26. Mahadevan R, Palsson BO, Properties of metabolic networks: structure versus function, *Biophys J* 88:L07–9, 2005.

28. Drozdov-Tikhomirov LN, Scurida GI, Davidov AV, Alexandrov AA, Zvyagilskaya RA, Mathematical modeling of living cell metabolism using the method of steady-state stoichiometric flux balance, *J Bioinform Comput Biol*. 2006 Aug;4(4):865-85.

29. L. N. Drozdov-Tikhomirov, G. I. Scurida, A. V. Davidov and A. A. Alexandrov, Mathematical modeling of living cell metabolism using the method of steady-state stoichiometric flux balance, *Journal of Bioinformatics and Computational Biology*, Vol. 4, No. 4 (2006) 865–885

30. Reed JL, Palsson BO, Genome-scale in silico models of E. coli have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states, *Genome Res* 14:1797–805, 2004.

31. Allen TE, Genome-scale analysis of the uses of the Escherichia coli genome: modeldriven analysis of heterogeneous data sets, *J Bacteriol* 185:6392–9, 2003.

32. Reed JL, *et al.*, An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR), *Genome Biol* 4:R54, 2004.

33. Foster J *et al.*, Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network, *Genome Res* 13:244–253, 2003.

34. Duarte NC, Herrhard MJ, Palsson BO, Reconstruction and validation of Saccharomyces iND750, a fully compartmetalized genom-scale metabolic model, *Genome Res* 14:1298–1309, 2004.

35. Famili I, *et al.*, Saccharomyces cerevisiae phenotypes can be predicted by using constraint-based analysis of a genome-scale reconstructed metabolic network, *Proc Natl Acad Sci USA* 100:13134–9, 2003.

36. Duarte NC, Palsson BO, Fu P, Integrated analysis of metabolic phenotypes in Saccharomyces cerevisiae, *BMC Genomics* 5:63, 2004.

37. Forster J *et al.* Large-scale evaluation of in silico gene deletions in Saccharomyces cerevisiae, *OMICS* 7:193–202, 2003.

38. Num. Natalie C. Duarte, Scott A. Becker, Neema Jamshidi, Ines Thiele, Monica L. Mo, Thuy D. Vo, Rohith Srivas, and Bernhard Ø. Palsson ,Global reconstruction of the human metabolic network based on genomic and bibliomic data, *PNAS* February 6, 2007 vol. 104 no. 6 1777-1782

39. Faeder JR, Blinov ML, Hlavacek WS., Rule-based modeling of biochemical systems with BioNetGen, *Methods Mol Biol.* 2009;500:113-67

40. Wolfram S: *A New Kind of Science* Champaign, IL: Wolfram Media; 2002.

41. Lemont B. Kier, Paul G. Seybold, Chao-Kun Cheng, *Modeling Chemical systems using cellular automata*, 2005, Springer,.

42. Weimar JR: *Cellular Automata* Edited by: Bandini S, Tomasini M, Chopard B. Berlin: Springer; 2002:294-303.

43. Alber M S, Kiskowski M A, Glazier J A and Jiang Y 2002 On cellular automaton approaches to modeling biological cells *Mathematical Systems Theory in Biology, Communication, and Finance* ed J Rosenthal and D S Gilliam (New York: Springer) pp 1-40 IMA 142

44. Deutsch A, Dormann S: *Cellular Automaton Modeling of Biological Pattern Formation* Boston, MA: Birkhauser; 2005.

45. G. Bard Ermentrout and Leah Edelstein-Keshet, Cellular Automata approaches to biological Modeling, *J. theor. Biol.* (1993) 160, 97-133

46. Frederik Graw*, Roland R. Regoes, Investigating CTL Mediated Killing with a 3D Cellular Automaton, *PLoS Computational Biology* August 2009 Volume 5 Issue 8 e1000466

47. Kier LB, Cheng C-K, Testa B, Carrupt P-A: A cellular automata model of enzyme kinetics. *J Molec Graphics* 1996, 14:227-231.

48. Kier LB, Bonchev DG, Buck GA: Modeling biochemical networks: A cellular automata approach. *Chem Biodiversity* 2005, 2:233-243.

49. Neuforth A, Seybold PG, Kier LB, Cheng CK: Cellular automata models of kinetically and thermodynamically controlled reactions. *Int J Chem Kinet* 2000, 32:529-534.

50. Jijrg R. Weimar* and Jean-Pierre, Class of cellular automata for reaction-diffusion systems, *Physical Review* Volume 49, Number 2 February 1994

51. Bonchev DG, Kier LB, Cheng CK: Cellular automata (CA) as a basic method for studying network dynamics. *Lecture Series on Computer and Computational Sciences* 2006, 6:581-591.

52. Kier LB, Cheng CK: A cellular automata model of an anticipatory system. *J Molec Graphics* 2000, 18:29-32.

53. Wilkinson D: *Stochastic Modeling for Systems Biology*, Boca Raton FL: Chapman & Hall/CRC, 2006.

54. Gillespie DT, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* 1976, 22, 403-434.

55. Rathinam M, Petzold LR, Cao Y, Gillespie DT, Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. Chem. Phys. 119*, Issue 24, pp. 12784-12794 2003.

56. Bremaud P: *Gibbs Fields, Monte Carlo Simulation, and Queues*, Berlin, New York: Springer, 2nd printing, 2001.

57. Wolfram S. in *A New Kind of Science* Champaign, IL: Wolfram Media; 2002.

58. Deutsch A & Dormann S: *Cellular Automaton Modeling of Biological Pattern Formation* Boston, MA: Birkhauser; 2005

59. Weimar JR. in *Cellular Automata* Edited by Bandini S, Tomasini M, Chopard B Berlin: Springer; 2002: 294-303.

60. Neuforth A, Seybold PG, Kier LB & Cheng CK. Cellular automata models of kinetically and thermodynamically controlled reactions. *Int. J. Chem. Kinet.* 2000, 32: 529-534.

61. Advait A Apte, John W Cain, Danail G Bonchev and Stephen S Fong. Cellular automata simulation of topological effects on the dynamics of feed-forward motifs. Journal of Biological Engineering 2008, 2:2.

62. Deutsch A, Dormann S: *Cellular Automaton Modeling of Biological Pattern Formation* Boston, MA: Birkhauser; 2005.

63. Wolfram S: *A New Kind of Science* Champaign, IL: Wolfram Media; 2002.

64. D. T. Taylor, J. W. Cain, D. G. Bonchev, S. S. Fong, A. A. Apte, L. E. Pace, Toward a Classification of Isodynamic Feed-Forward Motifs, *J Biol Dynamics*, submitted.

65. Alon U: Network motifs: theory and experimental approaches. *Nature Rev Genet* 2007, 8:450-461.

66. D.G. Bonchev, M.R. Woodcock, A. Mazurie, D.T. Taylor, J.W. Cain, A.A. Apte, and S.S.Fong, Networks Motif Topology and Evolution, in preparation 2009

67. H. Wajant, The Fas signaling pathway: more than a paradigm. *Science* 2002, *296,* 1635.

68. I. N. Lavrik, R. Eils, N. Fricker, C. Pforra, P. H. Krammer, *Molecular BioSystems* 2009, Advance online publication 3rd August 2009, DOI:10.1039/b905129p.

69. B. Fadeel, S. Orrenius, Apoptosis: a basic biological phenomenon with wide-ranging implications in human disease. *J. Intern. Med.* 2005, *258*, 479.[4] D. Hanahan, R. A.Weinberg, The hallmarks of cancer. *Cell* 2000, *100*, 57.

70. D. Hanahan, R. A.Weinberg, The Hallmarks of Cancer, *Cell* 2000, *100*, 57.

71. L. Marek, C. J. Burek, C. Stroh, K. Benedyk, H. Hug, A. Mackiewicz, Anticancer drugs of tomorrow: apoptotic pathways as targets for drug design. *Drug Discov. Today* 2003, *8*, 67-77.

72. S. W. Lowe, E. Cepero, G. Evan, Intrinsic tumor suppression. *Nature* 2004, *432*, 307.

73. S. Fulda, K. M.Debatin, Apoptosis signaling in tumor therapy. *Ann. N. Y. Acad. Sci.* 2004, *1028*, 150.

74. Z. Jin, W. S. El-Deiry, Overview of cell death signaling pathways. *Cancer Biol. Ther.* 2005, *4*, 139.

75. I. M. Ghobrial, T. E Witzig, A. A. Adjei, Targeting apoptosis pathways in cancer therapy. *CA Cancer J. Clin.* 2005, *55*, 178.

76. S. W. Fesik. Promoting apoptosis as a strategy for cancer drug discovery. *Nat. Rev. Cancer* 2005, *5*, 876.

77. J. C. Reed. Drug insight: cancer therapy strategies based on restoration of endogenous cell death mechanisms. *Nat. Clin. Pract. Oncol.* 2006, *3*, 38

78. T. A. Ferguson, T. S. Griffith, A vision of cell death: Fas ligand and immune privilege 10 years later, *Immunol. Revs.* 2006, *213*, 228 - 238.

79. F.H. Igney, P.H. Krammer, Tumor counterattack: fact or fiction? *Cancer Immunol. Immunother.* 2005, *54*, 1127–1136.

80. T. R. Wilson, M. McEwan, K. McLaughlin, C. Le Clorennec, W.L. Allen, D.A. Fennell, P. G.Johnston, D.B. Longley, Combined inhibition of FLIP and XIAP induces Bax-independent apoptosis in type II colorectal cancer cells, *Oncogene* 2009, *28*, 63.

81. H. H. Cheung, D. J. Mahoney, E. C. LaCasse, R. G. Korneluk, A Small Molecule Smac Mimic Potentiates TRAIL- and TNFα-Mediated Cell Death, *Cancer Res*. 2009, *69*, 7729.

82. M. Irmler, M. Thome, M. Hahne, P. Schneider, K. Hofmann, V. Steiner, J. L. Bodmer, M. Schroter, K. Burns, C. Mattmann, D. Rimoldi, L.E. French, J. Tschopp, Inhibition of death receptor signals by cellular Flip. *Nature* 1997, *388*, 190–195.

83. M. Irisarri, J. Plumas, T. Bonnefoix, M. C. Jacob, C. Roucard, M. A. Pasquier, J. J. Sotto, A. Lajmanovich, Resistance to CD95-mediated apoptosis through constitutive c-Flip expression in a non-Hodgkin's lymphoma B cell line. *Leukemia* 2000, *14*, 2149–2158.

84. M. Thome, J. Tschopp, *Regulation of lymphocyte proliferation and death by FLIP*, *Nature Rev. Immunol.* 2001, *1*, 50.

85. D. W. Chang, Z. Xing, Y. Pan, A. Algeciras-Schimnich, B.C. Barnhart, S. Yaish-Ohad, M.E. Peter, X. Yang, *c-FLIP*$_L$ *is a dual function regulator for caspase-8 activation and CD95-mediated apoptosis*. *EMBO J.* 2002, *21*, 3704.

86. L. S. Hyung, K.H. Sug, K.S. Young, L.Y-Sil, P.W. Sang, K.S. Ho, L.J. Young, Y.N. Jin, Increased expression of FLIP, an inhibitor of FAS-mediated apoptosis, in stomach cancer. *APMIS* 2003, *111*, 309.

87. H Wajant, Targeting the FLICE Inhibitory Protein (FLIP) in Cancer Therapy, *Molec. Intervent.* 2003, *3*, 124-127.

88. M. Flahaut, A. Muhlethaler-Mottet, K. Auderset, K. Balmas Bourloud, R. Meier, M. B. Popovic, J. M. Joseph, N. Gross, Persistent inhibition of FLIP$_L$ expression by lentiviral small hairpin RNA delivery restores death-receptor-induced apoptosis in neuroblastoma cells, *Apoptosis* 2006, *11*, 255.

89. Y. L. Yangand and X.M. Li, *The IAP family: endogenous caspase inhibitors with multiple biological activities*, *Cell Res.* 2000, *10*, 169.

90. A. M. Verhagen, E.J. Coulson, D.L. Vaux, Inhibitor of apoptosis proteins and their relatives: IAPs and other BIRPs, *Genome Biol*. 2001, *2*, 1.

91. A. M. Chinnaiyan, K. O'Rourke, M. Tewari, V. M. Dixit, FADD, a novel death domain-containing protein, interacts with the death domain of Fas and initiates apoptosis, *Cell*, 1995, *81*, 505.

92. F. L. Scott, B. Stec, C. Pop, M. K. Dobaczewska, J. E. J. Lee, E. Monosov, H. Robinson, G. S. Salvesen, R. Schwarzenbacher, S. J. Riedl*,* The Fas–FADD death domain complex structure unravels signalling by receptor clustering, *Nature* 2009, *457*, 1019.

93. G. S. Salvesen , S. J. Riedl, Structure of the Fas/FADD complex: A conditional death domain complex mediating signaling by receptor clustering, *Cell Cycle* 2009 Sep 30; 8(17). [Epub ahead of print]

94. M. Jourdan, T. Reme, H. Goldschmidt, G. Fiol, V. Pantesco, J. De Vos, J.-F. Rossi, D. Hose, B. Klein, Gene expression of anti- and pro-apoptotic proteins in malignant and normal plasma cells, Brit. J. Haematol. 2009, *145*, 45–5

95. C. Adrain, E. M. Creagh, J. Seamus, Apoptosis-associated release of Smac/DIABLO from mitochondria requires active caspases and is blocked by Bcl-2, *EMBO J.* 2001, *20*, 6627.

96. L. L. Zhou, L. Y. Zhou, K. Q. Luo, D. C. Chan, Smac/DIABLO and cytochrome c are released from mitochondria through a similar mechanism during UV-induced apoptosis, *Apoptosis*, 2005, *10*, 289.

97. N. Okazaki, R. Asano, T. Kinoshita, H. Chuman, Simple computational models of type I/type II cells in Fas signaling-induced apoptosis, *J. Theor. Biol.* 2008, *250*, 621

98. Danail Bonchev, Sterling Thomas, Advait Apte, Lemont Kier, Cellular Automata (CA) Modeling of Biomolecular Networks Dynamics, SAR & QSAR in Environmental Research, in press 2009

99. Advait Apte, Danail Bonchev, Stephen Fong, Cellular Automata Modeling of FASL-Initiated Apoptosis, Chemistry and Biodiversity, submitted 2009

100. Lee R. Lynd, Paul J. Weimer, Willem H. van Zyl, and Isak S. Pretorius. Microbial Cellulose Utilization: Fundamentals and Biotechnology. *Microbiology and molecular biology reviews*. Sept. 2002, p. 506–577.

101. Arnold L. Demain,1* Michael Newcomb,2 and J. H. David Wu. Cellulase, Clostridia, and Ethanol, *Microbiology and molecular biology reviews*. Mar. 2005, p. 124–154.

102. W. H. Schwarz, The cellulosome and cellulose degradation by anaerobic bacteria, *Appl Microbiol Biotechnol* (2001) 56:634–649

103. Edward A. Bayer, Rina Kenig and Raphael Lamed, Adherence of Clostridium thermocellum to Cellulose, *Journal of Biotechnology*, Nov. 1983, p. 818-827 Vol. 156, No.2

104. Yuval Shoham, Raphael Lamed and Edward A. Bayer, The cellulosome concept as an efficient microbial strategy for the degradation of insoluble polysaccharides, *Trends in Microbiology* VOL. 7 NO. 7 JULY 1999

105. B. S. Dien, M. A. Cotta, T. W. Jeffries. Bacteria engineered for fuel ethanol production: current status. *Appl Microbiol Biotechnol* (2003) 63: 258–266.

106. Emanuel Berger1, Dong Zhang1, Vladimir V. Zverlov1 & Wolfgang H. Schwarz, Two noncellulosomal cellulases of Clostridium thermocellum, Cel9I andCel48Y, hydrolyse crystalline cellulose synergistically. *FEMS Microbiol Lett* 268 (2007) 194–201.

107. Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

108. Tatsuya Noike, Ginro Endo, Juu-En Chang, Jun-Ichi Yaguchi, Jun-Ichiro Matsumoto, Characteristics of carbohydrate degradation and the rate-limiting step in anaerobic digestion, 18 Feb 2004, Volume 27, Issue 10 , Pages 1482 – 1489

109. Anantharam P. Dadi, Constance A. Schall and Sasidhar Varanasi, Mitigation of cellulose recalcitrance to enzymatic hydrolysis by ionic liquid pretreatment, Volume 137-140, Numbers 1-12 / April, 2007 pp. 407-421

110. Shaolin Chen and David B. Wilson, Proteomic and Transcriptomic Analysis of Extracellular Proteins and mRNA Levels in Thermobifida fusca Grown on Cellobiose and Glucose, *Journal of Biotechnology*, Sept. 2007, p. 6260–6265 Vol. 189, No. 17

111. Babu Raman1,4, Chongle Pan2,4, Gregory B. Hurst3,4, Miguel Rodriguez, Jr.1,4, Catherine K. McKeown1, Patricia K. Lankford1, Nagiza F. Samatova2,5, Jonathan R. Mielenz, Impact of Pretreated Switchgrass and Biomass Carbohydrates on Clostridium thermocellum ATCC 27405 Cellulosome Composition: A Quantitative Proteomic Analysis, *PLoS ONE,* 1 April 2009 Volume 4 Issue 4 e527

112. Diego Martinez Et al, Genome sequencing and analysis of the biomass-degrading fungus Trichoderma reesei (syn. Hypocrea jecorina), *Nature Biotechnology* VOLUME 26 NUMBER 5 MAY 2008

113. Athanasios Lykidis, Konstantinos Mavromatis, Natalia Ivanova, Iain Anderson, Miriam Land, Genevieve DiBartolo, Michele Martinez, Alla Lapidus, Susan Lucas, Alex Copeland, Paul Richardson, David B. Wilson, and Nikos Kyrpides, Genome Sequence and Analysis of the Soil Cellulolytic Actinomycete Thermobifida fusca YX, *Journal of Bacteriology*, Mar. 2007, p. 2477–2486 Vol. 189, No. 6

114. Solexa Genome Analyzer by Illumina Inc.

115. Fong, S.S., Burgard, A.P., Herring, C.D., Knight, E.M., Blattner, F.R., Maranas, C.D., and Palsson, B.Ø., "In Silico Design and Adaptive Evolution of "Escherichia coli" for Production of Lactic Acid", *Biotechnology and Bioengineering*, 91(5):643-648 (2005)

116. Hua Q, Joyce AR, Fong SS, Palsson BØ, Metabolic analysis of adaptive evolution for in silico-designed lactate-producing strains, *Biotechnol Bioeng*. 2006 Dec 5; 95(5): 992-1002

117. Christopher M. Gowen, Robert B. Seth, Advait A. Apte, Stephen S. Fong, *work in progress*, 2009

118. Yu Deng, Robert B. Seth, Advait A. Apte, Stephen S. Fong, *work in progress*, 2009

119. Eaves, H. and Gao, Y. (2009) MOM: Maximal Oligo Mapping. *Bioinformatics* 2009

APPENDIX A

Computer code for cellulose hydrolysis model written in Netlogo.

```
globals [numberofchains wholist randiffusionexo randiffusionendo randiffusiongluco
enzymeconstant decideenzyme celluloseidexo celluloseidendo celluloseidgluco
celluloseate brokendown? brokendownflag number]

breed [ celluloses cellulose ]
breed [ bacteria bacterium ]
breed [ bacteriaC bacteriumC ]
breed [ bacteriaH bacteriumH ]
breed [ endoglucanases endoglucanase ]
breed [ exoglucanases exoglucanase ]
breed [ B-glucosidases B-glucosidase ]

celluloses-own [chain leader neighbour amileader mychainlength parent-xcor parent-ycor
taken?]
bacteriaC-own [stuck?]
bacteriaH-own [stuck?]
endoglucanases-own [birthTicks]
exoglucanases-own [birthTicks]
B-glucosidases-own [birthTicks]

to setup
  ca
  ca
  ca
  setup-cellulose
  setup-grid
  setup-bacteria
end

to setup-cellulose
  let j 1
  set numberofchains CelluloseChains
  repeat CelluloseChains
  [
    create-celluloses 1
    [
      set taken? false
      set size 0.5
      set color brown
      set shape "box"
      setxy random-pxcor random-pycor
      set chain j
      set j j + 1
      set leader [who] of self
      set parent-ycor pycor
      set parent-xcor pxcor
      let i 1
      ifelse RandomChain = false
      [
        repeat CelluloseLength - 1
        [
          hatch 1 [setxy parent-xcor + i parent-ycor decide-neighbour]
          set i i + 1
        ]
      ]
      [
        let ranChain random CelluloseLength
        repeat ranChain
        [
          hatch 1 [setxy parent-xcor + i parent-ycor set color brown decide-neighbour]
          set i i + 1
        ]
      ]
    ]
  ]
  ask celluloses
  [
    check-ifiamtheleader
    decide-neighbour
    decide-chainlength
    check-leader
  ]
  set celluloseate 0
  set brokendownflag true
end

to setup-grid ;
  ask patches
  [
    set pcolor green
  ]
```

```
end

to setup-bacteria
  if EnzymeSystem = "NonComplexed"
  [
    create-bacteria BacteriaConcentration [set color red set size 1 set shape "bug" setxy
random-pxcor random-pycor]
  ]
  if EnzymeSystem = "Complexed"
  [
    create-bacteriaC BacteriaConcentration [set color red set size 1 set shape "bug" setxy
random-pxcor random-pycor]
  ]
  if EnzymeSystem = "Hybrid"
  [
    create-bacteriaH BacteriaConcentration [set color red set size 1 set shape "bug" setxy
random-pxcor random-pycor]
  ]

end

to go
  if not any? celluloses with [mychainlength > 1]
  [
    set brokendown? true
  ]
  if brokendown? = true and brokendownflag = true
  [
    output-print (word "All cellulose broken down at:" ticks)
    set brokendown? false
    set brokendownflag false
  ]
  ifelse GlucoseIntake = true
  [
    if celluloseate * 100 / (CelluloseLength * CelluloseChains) >= StopAtIntake
    [
      output-print (word "cellulose intake complete at:" ticks)
      stop
    ]
  ]
  [
    if not any? celluloses with [mychainlength > 1]
    [
  stop
 ]
]
ask endoglucanases
[
 if Ticks - birthTicks > halflife
 [
  die
 ]
]
ask exoglucanases
[
 if Ticks - birthTicks > halflife
 [
  die
 ]
]
ask B-glucosidases
[
 if Ticks - birthTicks > halflife
 [
  die
 ]
]
if EnzymeSystem = "Complexed"
[
 move-bacteria-complexed
]
if EnzymeSystem = "NonComplexed"
[
 set randiffusionexo random 100
 set randiffusionendo random 100
 set randiffusiongluco random 100
 set enzymeconstant random 100
 set decideenzyme random 100
 move-bacteria-noncomplexed
 move-exoglucanases
 move-endoglucanases
 move-B-glucosidases
]
if EnzymeSystem = "Hybrid"
[
 set randiffusionexo random 100
```

```
  set randiffusionendo random 100
  set randiffusiongluco random 100
  set enzymeconstant random 100
  set decideenzyme random 100
  move-bacteria-hybrid
  move-exoglucanases
  move-endoglucanases
  move-B-glucosidases
 ]
 do-sugar-intake
 move-celluloses
 do-plots
 ask celluloses
 [
  check-ifiamtheleader
  decide-neighbour
  decide-chainlength
 ]
end

to do-sugar-intake
 if GlucoseIntake = true
 [
  let rangluco random 100
  if rangluco < PrbGluIntake
  [
   ask celluloses with [mychainlength = 1]
   [
    if any? bacteria-here or any? bacteriac-here or any? bacteriah-here
    [
     output-print (word "Intake of Cellulose no." [who] of self ".It is a glucose")
     set celluloseate celluloseate + 1
     output-print (word "total cellulose intake:" celluloseate)
     output-print (word "percent cellulose intake:" (celluloseate * 100 / (CelluloseLength
* CelluloseChains)))
     die
    ]
   ]
  ]
 ]
 if CellobioseIntake = true
 [
  let ranbio random 100
```

```
    if ranbio < PrbBioIntake
    [
      ask celluloses with [mychainlength = 2]
      [
        if any? bacteria-here or any? bacteriac-here or any? bacteriah-here
        [
          output-print (word "Intake of Cellulose chain no." [chain] of self ".It is a cellobiose")
          set celluloseate celluloseate + 2
          output-print (word "total cellulose intake:" celluloseate)
          output-print (word "percent cellulose intake:" (celluloseate * 100 / (CelluloseLength
* CelluloseChains)))
          ask celluloses with [chain = [chain] of myself]
          [
            die
          ]
        ]
      ]
    ]
  ]
  if CellotrioseIntake = true
  [
    ask celluloses with [mychainlength = 3]
    [
      let rantri random 100
      if rantri < PrbTriIntake
      [
        if any? bacteria-here or any? bacteriac-here or any? bacteriah-here
        [
          output-print (word "Intake of Cellulose chain no." [chain] of self ".It is a
cellotriose")
          set celluloseate celluloseate + 3
          output-print ( word "total cellulose intake:" celluloseate)
          output-print (word "percent cellulose intake:" (celluloseate * 100 / (CelluloseLength
* CelluloseChains)))
          ask celluloses with [chain = [chain] of myself]
          [
            die
          ]
        ]
      ]
    ]
  ]
end
```

```
to move-celluloses
  ask celluloses with [amileader = 1]
    [
      set heading random 360
    ]
  ask celluloses with [amileader = 0]
    [
      ifelse cellulose leader = nobody
      [
        check-leader
      ]
      [
        set heading [heading] of cellulose leader
      ]
    ]
  ask celluloses
    [
      ifelse ChainLengthEffect = true
      [
        ifelse cellulosemovement = false
        [
        ]
        [
          fd (1 / (cellulosemovementconstant * mychainlength))
        ]
      ]
      [
        ifelse cellulosemovement = false
        [
        ]
        [
          fd 1
        ]
      ]
    ]
  tick
end

to move-bacteria-noncomplexed
  ask bacteria
  [
    set heading random 360
    fd 1
  ]
  if count celluloses with [mychainlength > 1] > 0
  [
    if EnzymeProductionRate > enzymeconstant
    [
      ifelse decideenzyme < CompositionExo
      [
        ask one-of bacteria [hatch-exoglucanases 1 [set color blue set size 0.01 set shape
"arrow" set birthTicks ticks]]
      ]
      [
        ifelse decideenzyme < 100 - CompositionGluco and decideenzyme >
CompositionExo
        [
          ask one-of bacteria [hatch-endoglucanases 1 [set color violet set size 0.01 set shape
"arrow" set birthTicks ticks]]
        ]
        ;if decideenzyme > CompositionExo + CompositionEndo
        [
          ask one-of bacteria [hatch-B-glucosidases 1 [set color yellow set size 0.01 set shape
"arrow" set birthTicks ticks]]
        ]
      ]
    ]
  ]
end

to move-bacteria-complexed
  ask bacteriac
  [
    set heading random 360
    fd 1
    if any? celluloses-here with [mychainlength > 3]
    [
      break-chain
    ]
  ]
end

to break-chain
  print "yo!"
  ask one-of celluloses-here with [mychainlength > 1]
  [
   let downstreamC count celluloses with [who < [who] of myself and chain = [chain] of
myself]
   let upstreamC count celluloses with [who > [who] of myself and chain = [chain] of
myself]
   ifelse downstreamC < upstreamC
    [
     set numberofchains numberofchains + 1
     ask celluloses with [who > [who] of myself and chain = [chain] of myself]
     [
      set chain numberofchains
      check-leader
     ]
    ]
    [
     ifelse upstreamC < downstreamC
      [
       set numberofchains numberofchains + 1
       ask celluloses with [who > [who] of myself and chain = [chain] of myself]
       [
        set chain numberofchains
        check-leader
       ]
      ]
      [
       set numberofchains numberofchains + 1
       ask celluloses with [who > [who] of myself and chain = [chain] of myself]
       [
        set chain numberofchains
        check-leader
       ]
      ]
    ]
  ]
end

to move-bacteria-hybrid
 ask bacteriaH
 [
  set heading random 360
  fd 1
  if any? celluloses-here with [mychainlength > 3]
  [
   break-chain
  ]
 ]
 if count celluloses with [mychainlength > 1] > 0
 [
  if EnzymeProductionRateH > enzymeconstant
  [
   ifelse decideenzyme < CompositionExo
   [
    print decideenzyme
    ask one-of bacteriaH [hatch-exoglucanases 1 [set color blue set size 0.5 set shape
"arrow"]]
   ]
   [
    ifelse decideenzyme < 100 - CompositionGluco and decideenzyme >
CompositionExo
    [
     ask one-of bacteriaH [hatch-endoglucanases 1 [set color violet set size 0.5 set shape
"arrow"]]
    ]
    [
     ask one-of bacteriaH [hatch-B-glucosidases 1 [set color yellow set size 0.5 set shape
"arrow"]]
    ]
   ]
  ]
 ]
end

to move-exoglucanases
 ask exoglucanases
 [
  set heading random 360
  fd 1
  if any? celluloses-here with [mychainlength > 2]
  [
   ifelse (count endoglucanases-here + count exoglucanases-here +  count B-glucosidases-
here) < 2
   [
    let ranexo random 100
    if PrbExo > ranexo / 100
    [
     ask one-of celluloses-here with [mychainlength > 2]
     [
      let downstreamexo count celluloses with [who < [who] of myself and chain =
[chain] of myself]
      let upstreamexo count celluloses with [who > [who] of myself and chain = [chain]
of myself]
      if downstreamexo = 2
      [
       set numberofchains numberofchains + 1
       cut-exo-down
       output-print (word "Number of Cellulose chains is" numberofchains)
       set celluloseidexo [who] of myself
      ]
      if upstreamexo = 2
      [
       set numberofchains numberofchains + 1
       cut-exo-up
       output-print (word "Number of Cellulose chains is" numberofchains)
       set celluloseidexo [who] of myself
      ]
     ]
    ]
   ]
   [
    output-print "Too many Cellulases here"
   ]
  ]
 ]
end

to move-endoglucanases
 ask endoglucanases
 [
  set heading random 360
  fd 1
  if any? celluloses-here with [neighbour = 2 and mychainlength > 3]
  [
   ifelse (count endoglucanases-here + count exoglucanases-here +  count B-glucosidases-
here) < 2
   [
    let ranendo random 100
    if PrbEndo > ranendo / 100
    [
     ask one-of celluloses-here with [neighbour = 2 and mychainlength > 3]
     [
      let downstreamendo count celluloses with [who < [who] of myself and chain =
[chain] of myself]
      let upstreamendo count celluloses with [who > [who] of myself and chain = [chain]
of myself]
      ifelse downstreamendo > upstreamendo
      [
       set numberofchains numberofchains + 1
       cut-endo-down
       output-print (word "Number of Cellulose chains is" numberofchains)
       set celluloseidendo [who] of myself
      ]
      [
       set numberofchains numberofchains + 1
       cut-endo-up
       output-print (word "Number of Cellulose chains is" numberofchains)
       set celluloseidendo [who] of myself
      ]
     ]
    ]
   ]
   [
    output-print "Too many Cellulases here"
   ]
  ]
 ]
end

to move-B-glucosidases
 ask B-glucosidases
 [
  set heading random 360
  fd 1
  if any? celluloses-here with [mychainlength = 2]
  [
   ifelse (count endoglucanases-here + count exoglucanases-here +  count B-glucosidases-
here) < 2
   [
    let rangluco random 100
    if PrbGluco > rangluco / 100
    [
     ask one-of celluloses-here with [mychainlength = 2]
     [
            let downstreamgluco count celluloses with [who < [who] of myself and chain = [chain] of myself]
            let upstreamgluco count celluloses with [who > [who] of myself and chain = [chain] of myself]
            ifelse downstreamgluco > upstreamgluco
            [
              set numberofchains numberofchains + 1
              cut-gluco-down
              output-print (word "Number of Cellulose chains is" numberofchains)
              set celluloseidgluco [who] of myself
            ]
            [
              set numberofchains numberofchains + 1
              cut-gluco-up
              output-print (word "Number of Cellulose chains is" numberofchains)
              set celluloseidgluco [who] of myself
            ]
          ]
        ]
      ]
      [
        output-print "Too many Cellulases here"
      ]
    ]
  ]
end

to check-leader
  let k 1
  repeat numberofchains
  [
    ifelse one-of celluloses with [chain = k] = nobody
    [
    ]
    [
      ask one-of celluloses with [chain = k]
      [
        set wholist [who] of celluloses with [chain = k]
      ]
      ask celluloses with [chain = k]
      [
        set leader min wholist
      ]
    ]
    set k k + 1
  ]
end

to decide-neighbour
  ask celluloses
  [
    set neighbour count (celluloses-on neighbors) with [chain = [chain] of myself]
  ]
end

to decide-chainlength
  ask celluloses
  [
    set mychainlength count celluloses with [chain = [chain] of myself]
  ]
end

to check-ifiamtheleader
  ask celluloses
  [
    ifelse leader = [who] of self [set amileader 1] [set amileader 0]
  ]
end

to cut-exo-up
  output-print (word "Cellulose cut by exogluconase" celluloseidexo)
  ask celluloses with [who > [who] of myself and chain = [chain] of myself]
    [
      set chain numberofchains
      check-leader
    ]
end

to cut-exo-down
  output-print (word "Cellulose cut by exogluconase" celluloseidexo)
  ask celluloses with [who < [who] of myself and chain = [chain] of myself]
    [
      set chain numberofchains
      check-leader
    ]
end
```

```
to cut-endo-up
  output-print (word "Cellulose cut by endogluconase" celluloseidendo)
  ask celluloses with [who > [who] of myself and chain = [chain] of myself]
    [
      set chain numberofchains
      check-leader
    ]
end

to cut-endo-down
  output-print (word "Cellulose cut by endogluconase" celluloseidendo)
  ask celluloses with [who < [who] of myself and chain = [chain] of myself]
    [
      set chain numberofchains
      check-leader
    ]
end

to cut-gluco-up
  output-print (word "Cellulose cut by B-glucosidase" celluloseidgluco)
  ask celluloses with [who > [who] of myself and chain = [chain] of myself]
    [
      set chain numberofchains
      check-leader
    ]
end

to cut-gluco-down
  output-print (word "Cellulose cut by B-glucosidase" celluloseidgluco)
  ask celluloses with [who < [who] of myself and chain = [chain] of myself]
    [
      set chain numberofchains
      check-leader
    ]
end

to do-plots
  set-current-plot "Glucose Concentration"
  set-current-plot-pen "Glucose"
  plot count celluloses with [mychainlength = 1]
  set-current-plot-pen "Cellobiose"
  plot count celluloses with [mychainlength = 2]
  set-current-plot "Enzyme Expression"
  set-current-plot-pen "ExoExpression"
  plot count exoglucanases
  set-current-plot-pen "EndoExpression"
  plot count endoglucanases
  set-current-plot-pen "GlucoExpression"
  plot count B-glucosidases
end
```

APPENDIX B

Matlab code for Sequence extraction algorithm

```
clear %clear screen.
clc %clear variables.

fidw = fopen('/data1/Solexa/Fong_sample1_s_1_sequence_03_22_09_output.txt','wt');
%open specified file for writing.
fidr = fopen('/data1/Solexa/Fong_sample1_s_1_sequence_03_22_09.txt','r'); %open
specifiend file for reading.

[seq,pos] = textscan(fidr,'%s'); %read the file with each line as seperate string.

for i=2:4:length(seq{1,1}) %sequences are 2,6,10,14...
    current_string = char(seq{1,1}(i)); %extract the current sequence
    fprintf(fidw, '%s\n', current_string); %write current sequence to specified file as output
end

fclose(fidw); %close files
fclose(fidr);
```

Python code for building base call and coverage tables

```
#script purpose:

import sys

#decide whether to consider only U0, U1, etc...
keep = {'U0': 1, 'U1': 1}

#read mom file
inputfilename = sys.argv[1]

#bases, together with two functions, allows creation of reverse complement of reads (if
read direction == 'R')
bases = {'A':'T', 'C':'G', 'G':'C', 'T':'A', 'N':'N'}
```

```
def complement(s):
    letters = list(s)
    letters = [bases[base] for base in letters]
    return ''.join(letters)
def reverse_complement(s):
    s = list(s)
    s.reverse()
    s = ''.join(s)
    s = complement(s)
    return s

#make a dictionary: key is position, value is reference genome base at that position
refgenome = {}
pos = 1
file = open('tfusca.genome.fasta.txt')
while True:
    line = file.readline()
    if line == '': break
    line = line.rstrip()
    if line == '': continue
    #skip header line
    if line[0] == '>': continue
    for base in line:
        refgenome[pos] = base
        pos += 1

#::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:::
#read results...
pos2results = {}
file = open(inputfilename)
#file = open('short')
while True:
    line = file.readline()
    if line == '': break
    line = line.rstrip()
    if line == '': continue
    if line[0] == '#': continue
    col = line.split('\t')
    #this is to make sure read has at most x mismatches, maps to a unique position on
the genome, and is from e coli...
    if col[2] in keep and 'Thermobifida fusca YX' in line:
        [readid, readseq, matchtype, numzero, numone, numtwo, refid, start,
direction, unused] = line.split('\t')[:10]

        #ensure no gaps found, if so stop and deal with these
        if len(col) > 10:
        #   assert not 'X' in col[10], 'gap found:\n%s' % (line)
            if 'X' in col[10]:
                continue

        #if direction of read is 'R', get reverse complement & use this in results
        if direction == 'R':
            readseq = reverse_complement(readseq)

        #walk thru each base in the read...
        for i, base in enumerate(readseq):
            # i is the ith base in the read -> the position of this in the genome is
the start pos of this read + i
            pos = int(start) + i

            #a check only for the U0 case
            #assert base == refgenome[pos], 'Error: in U0 match, %s, pos %s,
%s != %s (ref)' % (readseq, str(pos), base, refgenome[pos])

            #if no results for this position up to now, initialize counts for this
position
            if not pos in pos2results:
                pos2results[pos] = {'A':0, 'C':0, 'G':0, 'T':0, 'N':0, 'Tot':0}

            #ensure that the base at this position is one of: A, C, G, T, or N
            assert base in pos2results[pos], 'Unknown base: %s, in: %s, seq: %s'
% (base, readid, readseq)

            #add to counts for this base
            pos2results[pos][base] += 1

            #add to coverage at this position
            pos2results[pos]['Tot'] += 1

#get all positions and sort
positions = pos2results.keys()
positions.sort()

#these are coverages and percents for which we will keep results
```

```
coverages = [1, 5, 10, 15, 20, 25, 30]
percents = [0.75, 0.80, 0.85, 0.90, 0.95, 1]

table, detailedtable = {}, {}
covfile = open(inputfilename + '.coverage.tfusca.out', 'w')
for pos in positions:

    #the base for this position in the ref genome sequence...
    refbase = refgenome[pos]

    #initialize the base call and the max counts for this position from the sequencing
run
    #call from sequencing at this position is simply the base with the greatest number
of counts
    call, max = '', 0
    for base in ['A', 'C', 'G', 'T']:
        count_at_this_pos = pos2results[pos][base]
        if count_at_this_pos > max:
            max = count_at_this_pos
            call = base
    #get coverage at this position
    thiscoverage = pos2results[pos]['Tot']

    #calculate agreement across all reads for this position (what % of total coverage is
the number of reads with 'called' base?)
    thispercent = float(max)/thiscoverage

    #ensure that 'thispercent' is <= 1 and >= 0 (should be a percent)
    assert thispercent <= 1, 'Error in max agreement percent...max: %s, cov: %s, perc:
%s' % (str(max), str(thiscoverage), str(thispercent))
    assert thispercent >= 0, 'Error in max agreement percent...max: %s, cov: %s, perc:
%s' % (str(max), str(thiscoverage), str(thispercent))

    for c in coverages:
        for p in percents:
            if not (c, p) in table:
                table[(c, p)] = 0
                detailedtable[(c, p)] = {}
            #increment counts if cutoff criteria satisfied and called base is
different from reference base
            if (thispercent >= p and thiscoverage >= c and call != refbase):
                table[(c, p)] += 1
```

```
                                        detailedtable[(c, p)][('\t').join(   [ str(pos),
str(pos2results[pos]['A']), str(pos2results[pos]['C']), str(pos2results[pos]['G']),
str(pos2results[pos]['T']), str(pos2results[pos]['Tot']), call, refbase ]   )] = 1

        print >>covfile, '%s' % (    ('\t').join(   [ str(pos), str(pos2results[pos]['A']),
str(pos2results[pos]['C']), str(pos2results[pos]['G']), str(pos2results[pos]['T']),
str(pos2results[pos]['Tot']), call, refbase ]   )   )


coords = table.keys()
coords.sort()
tablefile = open(inputfilename + '.table.tfusca.out', 'w')
for c, p in coords:
        print >>tablefile, c, p, table[(c, p)]

detailed = open(inputfilename + '.detailedres.tfusca.out', 'w')
for c, p in coords:
        print >>detailed, c, p
        for case in detailedtable[(c, p)]:
                print >>detailed, '    ' + case
```

Python code for printing gene names with mutations

```
import sys

#read mom output file
momfile = sys.argv[1]

file = open('mut.txt')
line = file.readline()
line = line.rstrip()
basepos = line.split('\t')
for k in range (len(basepos)):
        print basepos[k]
for j in range (len(basepos)):
        file1 = open(momfile,'r')
        while True:
                line1 = file1.readline()
                if line1 == '': break
                line1 = line1.rstrip()
                if line1 == '': continue
                if line1[0] == '#': continue
                col = line1.split('\t')[:10]
                if len(col) > 10: continue
                if len(col) < 7: continue
                if basepos[j] == col[7]:
#           print col[7]
            print line1
            print basepos[j]
############## write code to check each basepos against the mom output file if found split the line and get the gene name and save it in a file
```

Python code for mutation analysis

```
#script purpose: interpret results where sequences differ compared to reference

import sys

#read coverage file
coveragefile = sys.argv[1]

counts = {0:0, 1:0, 5:0, 10:0, 15:0, 20:0, 25:0, 30:0, 35:0, 40:0, 45:0, 50:0, 55:0, 60:0, 80:0, 100:0}
file = open(coveragefile)
while True:
    line = file.readline()
    if line == '': break
    line = line.rstrip()
    if line == '': continue
    if line[0] == '#': continue
    col = line.split('\t')

    cov = int(col[5])

    for c in counts:
        if cov >= c:
            counts[c] += 1
```

```
for c in counts:
        print c, counts[c]
```

Python code for coverage analysis

```
import sys

total = 0
countlines = 0
max = 0
#read base call table
inputfilename = sys.argv[1]
file1 = open(inputfilename,'r')
while True:
        line1 = file1.readline()
        if line1 == '': break
        line1 = line1.rstrip()
        if line1 == '': continue
        countlines += 1
        col = line1.split('\t')[:8]
        total = total + int(col[5])
        if int(col[5]) > max:
                max = int(col[5])
                line = line1
print max
print line
print "If ecoli"
averageecoli = total / 4600000
print averageecoli
print "If tfusca"
averagetfusca = total / 3642249
print averagetfusca
```

VITA

Advait Apte was born on August 31, 1982 in Mumbai, India. He is Indian by birth and attended University of Mumbai, India from August 2000 to May 2004 earning the degree of Bachelor of Engineering in Biomedical Engineering. Following his undergraduate degree, he attended Virginia Commonwealth University between August 2005 and December 2009 earning Master of Science in Engineering, majoring in Chemical engineering in December 2006, followed by Doctor of Philosophy in Engineering, majoring in Chemical and Life Science Engineering in December 2009. He is an inducted member of Phi Kappa Phi Honor Society since 2007

In addition to these, he has presented his work in many conferences including, *BMES 2007 fall meeting* in Hollywood, CA. Three of his publications are published in *Journal of Biological Engineering*, *Journal of Biological Dynamics* and *SAR & QSAR in Environmental Research* each, with fourth one submitted to *Chemistry and Biodiversity*. Work on two more publications is currently under way. He has also worked as a teaching assistant for many courses in Chemical and Life Science Engineering.