Modeling Protein Regulatory Networks that Control Mammalian Cell Cycle Progression and that Exhibit Near-Perfect Adaptive Responses

Rajat Singhania

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Genetics, Bioinformatics and Computational Biology

John J. Tyson, Chair
David R. Bevan
Yang Cao
Rahul V. Kulkarni
Jill C. Sible

April 22, 2011
Blacksburg, VA

Keywords: mathematical modeling, cell cycle regulation, motifs, adaptation

Modeling Protein Regulatory Networks that Control Mammalian Cell Cycle Progression and that Exhibit Near-Perfect Adaptive Responses

Rajat Singhania

ABSTRACT

Protein regulatory networks are the hallmark of many important biological functionalities. Two of these functionalities are mammalian cell cycle progression and near-perfect adaptive responses. Modeling and simulating these functionalities are crucial stages to understanding and predicting them as systems-level properties of cells.

In the context of the mammalian cell cycle, the timing of DNA synthesis, mitosis and cell division is regulated by a complex network of biochemical reactions that control the activities of a family of cyclin-dependent kinases. The temporal dynamics of this reaction network is typically modeled by nonlinear differential equations describing the rates of the component reactions. This approach provides exquisite details about molecular regulatory processes but is hampered by the need to estimate realistic values for the many kinetic constants that determine the reaction rates. To avoid this problem, modelers often resort to 'qualitative' modeling strategies, such as Boolean switching networks, but these models describe only the coarsest features of cell cycle regulation. In this work, we describe a hybrid approach that combines features of continuous and discrete networks. The model is evaluated in terms of flow cytometry measurements of cyclin proteins in asynchronous populations of human cell lines. Using our hybrid approach, modelers can quickly create quantitatively accurate, computational models of protein regulatory networks found in various contexts within cells.

Large-scale protein regulatory networks, such as the one that controls the progression of the mammalian cell cycle, also contain small-scale motifs or modules that carry out specific dynamical functions. Systematic characterization of smaller, interacting, network motifs whose individual behavior is well known under certain conditions is therefore of great interest to systems biologists. We model and simulate various 3-node network motifs to find near-perfect adaptation behavior. This behavior entails that a system responds to a change in its environmental cues, or signals, by coming back nearly to its pre-signal state even in the continued presence of the signal. We let various topologies evolve in their parameter space such that they eventually stumble upon a region where they score well under a pre-defined scoring metric. We find many such parameter sample sets across various classes of topologies.

DEDICATION

This dissertation is dedicated to my parents and brother. Without their constant encouragement and moral support, I would not have been able to reach this point. They were able to guide me wonderfully through the 5 or so years whenever the chips were down, and made me believe in myself. I am truly blessed to have such a wonderful cast having my back.

ACKNOWLEDGEMENTS

It has been a great journey of self-discovery through all these years of working with Dr. John Tyson. It is truly a great pleasure to work with someone of such high achievement, and yet who is also a humble, down-to-earth, and a very caring person. Without his constant patience, great understanding, exemplary guidance, and belief in me, I would have quit on myself a long time ago. I have learned a lot from him, and shall endeavor to make him proud and continue on improving myself wherever I go throughout my career.

I would also like to thank all the other members of my committee for their constant encouragement throughout the whole process. Each meeting with them has been very fruitful and warm.

Lastly, I would like to mention the other members of my lab, and all my wonderful friends at Virginia Tech past and present who have helped me to get through the long haul. My extended family and cousins in the US and India have played a vital and constructive role in shaping who I am as well.

I will always be grateful to each one of you.

EXTERNAL ATTRIBUTIONS

The experimental data in Chapter 2 was provided by Dr. James W. Jacobberger of the Case Comprehensive Cancer Center; Dr. Sramkoski, of the same lab, contributed to the write-up of the corresponding experimental protocols. I would also like to thank Kathy Chen and Tongli Zhang for helpful suggestions for my modeling work.

For Chapter 3, I thank Dr. Mark Paul and Alireza Karimi, of the Mechanical Engineering department at Virginia Tech, for letting me use their Beowulf cluster for most of my simulations. Other simulations were done through a web-service (SCIcluster.com). I employed the user-friendly R/parallel software to parallelize my code (Vera, 2008). In addition, Dr. Rahul Kulkarni, Dr. Pedro Mendes, Dr. T.M. Murali and Dr. Layne Watson were kind enough to help me when needed.

Table of Contents

List of Figures

List of Tables

Chapter 1. Overview of the Research

At the cellular level, many important biological functions are controlled by Protein Regulatory Networks (PRNs) that have evolved over time to help organisms survive and proliferate. Computational modeling can help to provide a better understanding of how the components of a PRN work together to carry out their collective function. In this dissertation, we are specifically interested in modeling PRNs that control mammalian cell cycle progression (Chapter 2), and that exhibit near-perfect adaptive responses (Chapter 3). While these contexts are quite different, they both help demonstrate the paradigm that there are underlying networks of interacting proteins that control many crucial physiological behaviors, and that these can indeed be modeled computationally.

In Chapter 2, we describe the design of a new framework to model the network of cyclin-dependent kinases that controls the timing of events in the cell cycle. Existing frameworks to describe this network are either continuous or discrete. In the continuous case, the Ordinary Differential Equations (ODEs) that model the cyclins and their regulators rely on estimations of numerous kinetic parameters that are hard to measure experimentally. In the discrete case, the species involved are modeled using Boolean variables only, which makes it hard to simulate the smooth changes found in cyclin levels, and other fine-grained features of cell cycle behavior such as cell size and cell age. Our 'hybrid' framework seeks to incorporate the best features from the two cases, while avoiding the problems inherent in both.

In our model, we first separate the protein species involved in cell cycle regulation into two classes: (1) the cyclins, the proteins that are the primary drivers of cell cycle progression, and (2) the cyclin regulators, such as transcription factors and cyclin degradation pathway initiators. The activity or inactivity of the cyclin regulators is represented by discrete (Boolean) variables that modulate the continuous ODEs used to model cyclin levels. Apart from the continuous-discrete sense, our model is also hybrid in the deterministic-stochastic sense. Progression through the ordered stages of the cell cycle is divided up into distinct "states". Each state comprises a specific combination of values of the Boolean-modeled cyclin regulators. While this sequence of Boolean states is deterministic, the residence time in each state is modeled stochastically, using an exponential distribution prescribed by the average residence time of each state, which is estimated from experimental data.

Apart from modeling the time spent in each state (and thus, in the whole cycle) in a more realistic manner than the Boolean framework, our framework also succeeds in incorporating measures of cell size. We assume exponential growth in mass.

We tested our hybrid model using flow cytometry data that characterizes changing cyclins and DNA levels across an asynchronous population of RKO (colon carcinoma) cells. The cyclins we model are Cyclin A, which mediates entry into S phase (DNA Synthesis), and Cyclin B, which mediates entry into M phase (Mitosis). We estimate the kinetic parameters for cyclin synthesis and degradation from the data itself.

To make our model more realistic, we also factor in sources of noise that affect the experimental data. Intrinsic noise in the regulatory system is modeled by the stochasticity of the times spent in each state. Instrumental measurement error, or extrinsic noise, is also taken into account in the simulations.

Apart from simulating snapshots of flow cytometry profiles showing cyclins and DNA, we are also able to simulate dynamic profiles that show data for up to 6 days. By accounting for contact inhibition among cells in a growing culture, our model also captures the varying daily distribution of the cell population across the phases of the cell cycle.

The primary significance of our work is the development of a new and easy-to-use paradigm for modeling PRNs that control mammalian cell cycle progression. Since the parameters come from the data itself, the models are relatively easy to build; yet, they are powerful and accurate in both a quantitative and qualitative sense. Used correctly, the hybrid modeling framework can be extended to other molecular control systems as well.

In Chapter 3, we seek to find 3-node motifs which exhibit the near-perfect adaptation behavior. A motif is a pattern of regulations, such as activations and inactivations, among the protein species, or nodes, of a small PRN. Near-perfect adaptation describes a system that responds to a stepwise change in an environmental cue (or signal) by an initial pulse and then a return (nearly) to its pre-signal state, even in the continued presence of the signal. This study is important as near-perfect adaptation is characteristic of a variety of biological responses, including chemotaxis in *E. coli*, and adenylate cyclase activation in *Dictyostelium*.

In a 3-node PRN, we specify that the signal comes into Node 1, the response is read from Node 3, while Node 2 adds complexity into the behavior of each motif through its interactions with the other two nodes. Since each of the 3 nodes can regulate the other two nodes, there are six possible interactions. Each interaction can take any of three forms: an activation, an inactivation, or no regulation. Therefore, the total number of possible topologies is $3^6 = 729$.

The change in level of each of the three nodes over time is modeled using a nonlinear ODE that incorporates several parameters describing (among other things) how a node is affected by the other nodes. There are a total of 12 parameters, whose values together constitute a set. Each set of parameter values is simulated on its own and assigned a score based on how well it shows near-perfect adaptation. Scores are based on both the sensitivity of the response to the signal, and also on the precision with which the response comes back to its pre-signal level.

We search for parameter sets with high scores by using an evolutionary algorithm. This algorithm goes from one generation of parameter sets to the next. Each generation consists of a certain number of parent sets that each spawn off a number of progeny parameter sets by introducing random changes in each of the parameter values. The scores of all progeny sets are then calculated, and a selection procedure is used to determine which sets survive to become the parent sets for the next generation, and so on. The evolutionary algorithm always tries to find a better scoring region in parameter space, and once such a region is found, it tends to remain there. This property enables us to collect a representative sample of high-scoring parameter sets.

Instead of simulating each of the 729 topologies one-by-one, we start with a sample of 40 different topologies, and let them mutate into other topologies using the evolutionary algorithm. We find that only two classes of motifs, or topologies, show average scores above a certain threshold. Both these topologies belong to the category of Incoherent Feed Forward Loops (IFFLs), and are called IFFL-1's and IFFL-4's. Another category called Negative Feedback Loops with Buffering (NFLBs) are also found, coupled with these IFFLs. These NFLBs are simulated separately on their own (not allowed to evolve into other topologies) and found to be not as high-scoring as the two classes of IFFLs. Among themselves, the 'upper' NFLBs - containing the negative feedback loop between Node 1 and Node 2 - score better than the 'lower' NFLBs, containing the negative feedback loop between Node 2 and Node 3. The high-scoring IFFL sets were found to be almost exclusively coupled to upper NFLBs. The contributions of the IFFLs and upper NFLBs in the various topologies are validated by examining the strengths of the relevant interaction coefficients.

Also, when allowed to evolve into other topologies, all members of the IFFL-1 and IFFL-4 classes stay within their own class, thus forming two distinct, high-scoring plateaus, or 'mesas' in topology space. Almost all NFLBs that are not coupled to IFFL's evolve onto the IFFL-1 mesa, which shows their evolutionary superiority. The two 'mesas' exist in contrast to other topology classes which score low and are therefore mostly in the 'desert' region. We are able to find the regions in parameter space where the mesas exist. We validate our findings by various analyses, and also characterize the robustness of an adaptive motif by calculating the volume of the multi-dimensional ellipsoid that approximates the size of the high-scoring region in the motif's parameter space.

Our work significantly extends previous work on near-perfect adaptation in the literature. We establish the superiority of the IFFLs to the NFLBs, show which combinations of the two classes score best, which classes are evolutionarily stable, and finally, where in parameter space these classes of interest are most likely to score high. We now have a much more nuanced idea of what kinds of PRNs exhibit the property of near-perfect adaptation, and we use a very efficient evolutionary algorithm to come to our conclusions. Our approach can be easily extended to study other biological behaviors of interest within small-scale PRNs.

The research summarized above occurs in two distinct contexts that still share some common themes. The central idea behind the two projects is that physiological behaviors crucial to the cell's survival and propagation are controlled by networks of interacting proteins that can be modeled mathematically. The signal dictating the behavior is read or sensed by proteins, and the cellular response is carried out primarily through proteins as well. The underlying network contains intermediary proteins that transmit the signal from the input component to the output component. The specific pattern of regulations within and among all these components is crucial to determining the actual response of the cell. Different network topologies can produce different behaviors. Importantly, the behavior of the overall network may be unpredictable or counter-intuitive due to the complex web of regulations present. This is where computational modeling approaches, such as the ones we use, can play a significant role in both checking our understanding of how a well-characterized physiological behavior (such as progression of the unperturbed mammalian cell cycle) is regulated, and in yielding new light on regulatory mechanisms that are relatively unknown (such as the PRNs controlling near-perfect adaptation).

Chapter 2. A Hybrid Model of Mammalian Cell Cycle Regulation*

2.1 Introduction

The cell division cycle is the fundamental physiological process by which cells grow, replicate, and divide into two daughter cells that receive all the information (genes) and machinery (proteins, organelles, etc.) necessary to repeat the process under suitable conditions (Mitchison, 1971). This cycle of growth and division underlies all biological expansion, development and reproduction. It is highly regulated to promote genetic fidelity and meet the demands of an organism for new cells. Altered systems of cell cycle control are root causes of many severe health problems, such as cancer and birth defects.

In eukaryotic cells, the processes of DNA replication and nuclear/cell division occur sequentially in distinct phases (S and M) separated by two gaps (G1 and G2). Mitosis (M phase) is further subdivided into stages: prophase (chromatin condensation, spindle formation, and nuclear envelope breakdown), prometaphase (chromosome attachment and congression), metaphase (chromosome residence at the mid-plane of the spindle), anaphase (sister chromatid separation and movement to opposite poles of the spindle), telophase (re-formation of the nuclear envelopes), and cytokinesis (cell division). G1 phase is subdivided into uncommitted and committed sub-phases, often referred to as G1-pm (postmitotic interval) and G1-ps (pre S phase interval), separated by the 'restriction point' (Zetterberg et al, 1995). In this paper, we shall refer to the sub-phases G1-pm and G1-ps as 'G1a' and 'G1b' respectively.

Progression through the correct sequence of cell-cycle events is governed by a set of cyclin-dependent kinases (Cdk's), whose activities rise and fall during the cell cycle as determined by a complex molecular regulatory network. For example, cyclin synthesis and degradation are controlled, respectively, by transcription factors and ubiquitin-ligating complexes whose activities are, in turn, regulated by cyclin/Cdk complexes.

Current models of the Cdk control system can be classified as either continuous or discrete. Continuous models track the changes of protein concentrations, $C_j(t)$ for $j = 1, 2, \ldots, N$, by solving a set of nonlinear ordinary differential equations (ODEs) of the form:

$$\frac{dC_j}{dt} = \sum_{r=1}^{R} \nu_{jr} \rho_r (C_1, C_2, ..., C_N) \qquad \text{[Eq.2.1]}$$

where ρ_r is the rate of the r^{th} reaction and ν_{jr} is the stoichiometric coefficient of species j in reaction r. To each rate term is associated one or more kinetic constants that determine exactly how fast the reaction proceeds under specific conditions. These kinetic constants must be estimated from experimental data, and often there is insufficient kinetic data to determine their values. Nonetheless, continuous models, based on rate equations, have been used successfully to account for the properties of cell proliferation in a variety of cell types: yeast (Chen et al, 2004;

*R. Singhania, R.M. Sramkoski, J.W. Jacobberger & J.J. Tyson, PLoS Comput. Biol. 7:e1001077 (2011).

Chen et al, 2000; Novak et al, 2001), fruit fly (Calzone et al, 2007), frog egg (Novak & Tyson, 1993; Pomerening et al, 2005), and cultured mammalian cells (Aguda & Tang, 1999; Novak & Tyson, 2004; Qu et al, 2003). They have also proved successful in predicting novel cell-cycle characteristics (Pomerening et al, 2003; Sha et al, 2003).

Discrete models, on the contrary, represent the state of each regulatory protein as $B_j(\tau) = 0$ or 1 (inactive or active), and the state variables update from one discrete time step to the next ($\tau = 0$, 1, 2, … = ticks of a metronome) according to the rule:

$$B_j(\tau+1) = B_j(B_1(\tau), B_2(\tau), \ldots, B_n(\tau)), \qquad \text{[Eq.2.2]}$$

where $B_j(\ldots)$ is a Boolean function (i.e., it equates to either 0 or 1) determined by the topology of the reaction network. For Boolean networks (BNs), there is no notion of reaction 'rate' and, hence, no need to estimate kinetic constants. BN models of the Cdk regulatory network have been proposed for yeast cells (Davidich & Bornholdt, 2008; Li et al, 2004) and for mammalian cells (Faure et al, 2006). They have been used to study notions of 'robustness' of the cell cycle, but they have not been compared in detail to quantitative properties of cell cycle progression, and they have not been used as predictive tools.

In this paper we propose to combine the strengths of both continuous and discrete modeling, while avoiding the weaknesses of each. Our 'hybrid' model is inspired by the work of Li et al. (2004), who proposed a BN for cell cycle controls. Their model employs 11 state variables that move around in a space of $2^{11} = 2048$ possible states. Quite remarkably they found that 1764 of these states converge quickly onto a 'super highway' of 13 consecutive states that represent a typical cell cycle trajectory (G1b—S—G2—M—G1a). The results of Li et al. indicate that the cell cycle control network is 'robustly designed' in the sense that even quite large perturbations away from the usual sequence of cell cycle states are quickly restored to the super highway. In the model of Li et al., G1a is a stable steady state; they do not address the signals that drive cells past the restriction point (the G1a-to-G1b transition).

Despite their intuitive appeal, Boolean models have severe limitations. First of all, metronomic time in BN's is unrelated to clock time in the laboratory, so Boolean models cannot be compared to even the most basic observations of time spent by cells in the four phases of the division cycle (Mitchison, 1971). Also, these models do not incorporate cell size, so they cannot address the evident importance of cell growth in driving events of the cell cycle (Fantes & Nurse, 1981; Tyson, 1985; Tyson, 1987). Lastly, cyclins are treated as either absent or present (0 or 1), so Boolean models cannot simulate the continuous accumulation and removal of cyclin molecules at different stages of the cell cycle (Darzynkiewicz et al, 2004).

Our goal is to retain the elegance of the Boolean representation of the switching network, while introducing continuous variables for cell size, cell age, and cyclin composition, in order to create a model that can be compared in quantitative detail to experimental measurements with a minimal number of kinetic parameters that must be estimated from the data. To this end, we keep the cyclin regulators as Boolean variables but model the cyclins themselves as continuous concentrations that increase and decrease due to synthesis and degradation. Next, we replace the Boolean model's metronome with real clock time to account for realistic rates of cyclin synthesis

and degradation, and for stochastic variability in the time spent in each Boolean state of the model. Finally, we introduced a cell size variable, $M(t)$, which affects progression through late G1 phase. $M(t)$ increases exponentially with time as the cell grows and decreases by a factor of ~2 when the cell divides. (The assumption of exponential growth is not crucial; similar results are obtained assuming linear growth between cell birth and division.)

Since the pioneering work of Leon Glass (Glass & Kauffman, 1973; Glass & Pasternack, 1978), hybrid (discrete-continuous) models have been employed by systems biologists in a variety of forms and contexts (Bosl, 2007; Li et al, 2009; Matsuno et al, 2006). Engineers have been modeling hybrid control systems for many years (Alur et al, 2001; Fishwick, 2007; Klee & Allen, 2011), and they have created powerful simulation packages for such systems (Mosterman, 1999): SHIFT (Deshpande et al, 1997), CHARON (Alur et al, 2000), SIMULINK (Klee & Allen, 2011), and UPPAAL (Bengtsson et al, 1996), to name a few. We have not used these simulation packages because our model can be solved analytically.

2.2 Results

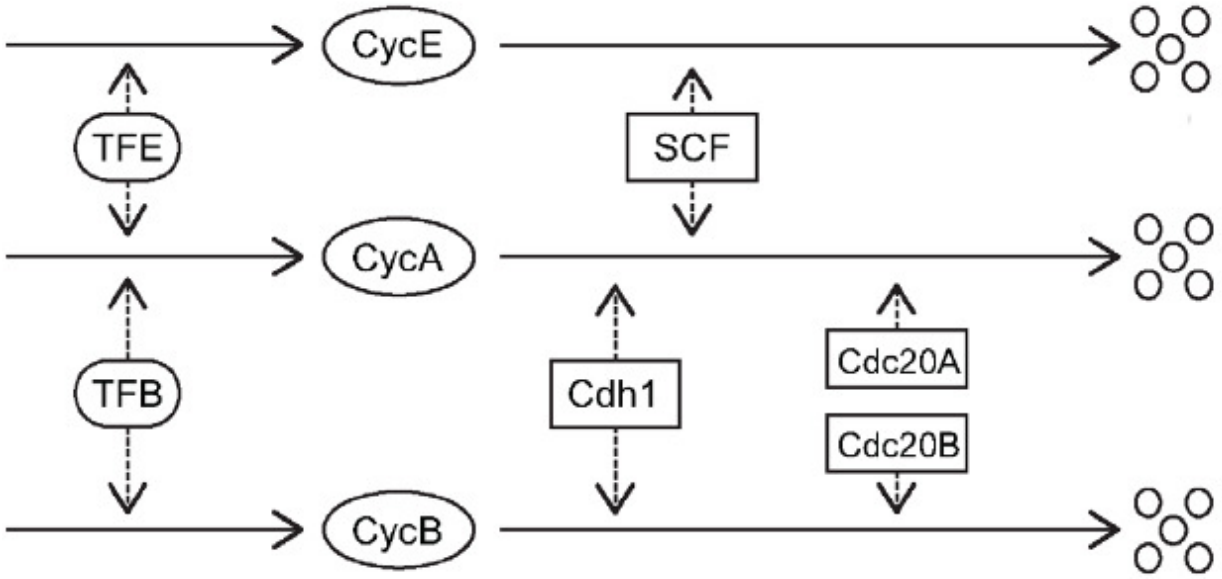
Hybrid modeling approach

The modeling approach we are proposing is hybrid in two senses. First, we employ both continuous and discrete variables, and second, we allow for both deterministic and stochastic processes. Concerning the components of the control system, we track cyclin levels as continuous concentration variables, but we use discrete Boolean variables to represent the activities ('on' or 'off') of the regulatory proteins (transcription factors and ubiquitinating enzymes) that control cyclin synthesis and degradation. This distinction is equivalent to a presumed 'separation of time scales': the activities of the regulatory proteins change rapidly between 0 and 1, while the concentrations of cyclins change more slowly due to synthesis and degradation. The Boolean variables, we assume, proceed from one state to the next according to a fixed sequence corresponding roughly to the super highway of Li et al. (2004). The time spent in each state, however, is not a 'tick' of the metronome but rather the sum of a deterministic execution time (which may be 0) plus a random, exponentially distributed waiting time. In this sense, the model combines deterministic and stochastic processes.

In its present version, our model is not fully autonomous. The discrete variables do not update according to Boolean functions of the current state of the network. Rather, they go through a fixed sequence of states predetermined by the Boolean network model of Li et al. [14]. The discrete variables determine the rates of synthesis and degradation of the continuous variables (the cyclins), and the cyclins feedback on the discrete variables by determining how much time is spent in some of the Boolean states. This strategy keeps the model simple and is appropriate for the cases, considered in this paper, of unperturbed cycling of 'wild type' cells, which travel serenely along the super highway of Li et al. To consider more complicated cases, of mutant cells that travel a different route through discrete state space or of cells that are perturbed by drugs or radiation, we will have to elaborate on this basic model with additional rules governing the interactions of the discrete and continuous variables. We are currently working on alternative strategies to adapt this basic modeling paradigm to more complex situations.

Our model (see Figure 2.1(a)) tracks three cyclin species (A, B and E), two transcription factors ('TFE' and 'TFB') and two different E3 ubiquitin-ligase complexes (APC-C and SCF). TFE drives the synthesis of cyclins E and A early in the cell cycle (comparable to the E2F family of transcription factors) (Trimarchi & Lees, 2002), and TFB drives the synthesis of cyclins B and A late in the cell cycle (comparable to FoxM1 and Myc) (Laoukili et al, 2005; Wierstra & Alves, 2007). The Anaphase Promoting Complex—Cyclosome (APC-C) is active during M phase and early G1, when it combines with Cdc20 and Cdh1 to label cyclins A and B for degradation by proteasomes. We make a further distinction between Cdc20 activity on cyclin A (Cdc20A, active throughout mitosis) from Cdc20 activity on cyclin B (Cdc20B, activated at anaphase). The SCF labels cyclin E for degradation via ubiquitination, but only when cyclin E is phosphorylated (Cardozo & Pagano, 2004), which we assume is correlated primarily with cyclin A/Cdk2 activity (Welcker et al, 2003).

(a)



(b)

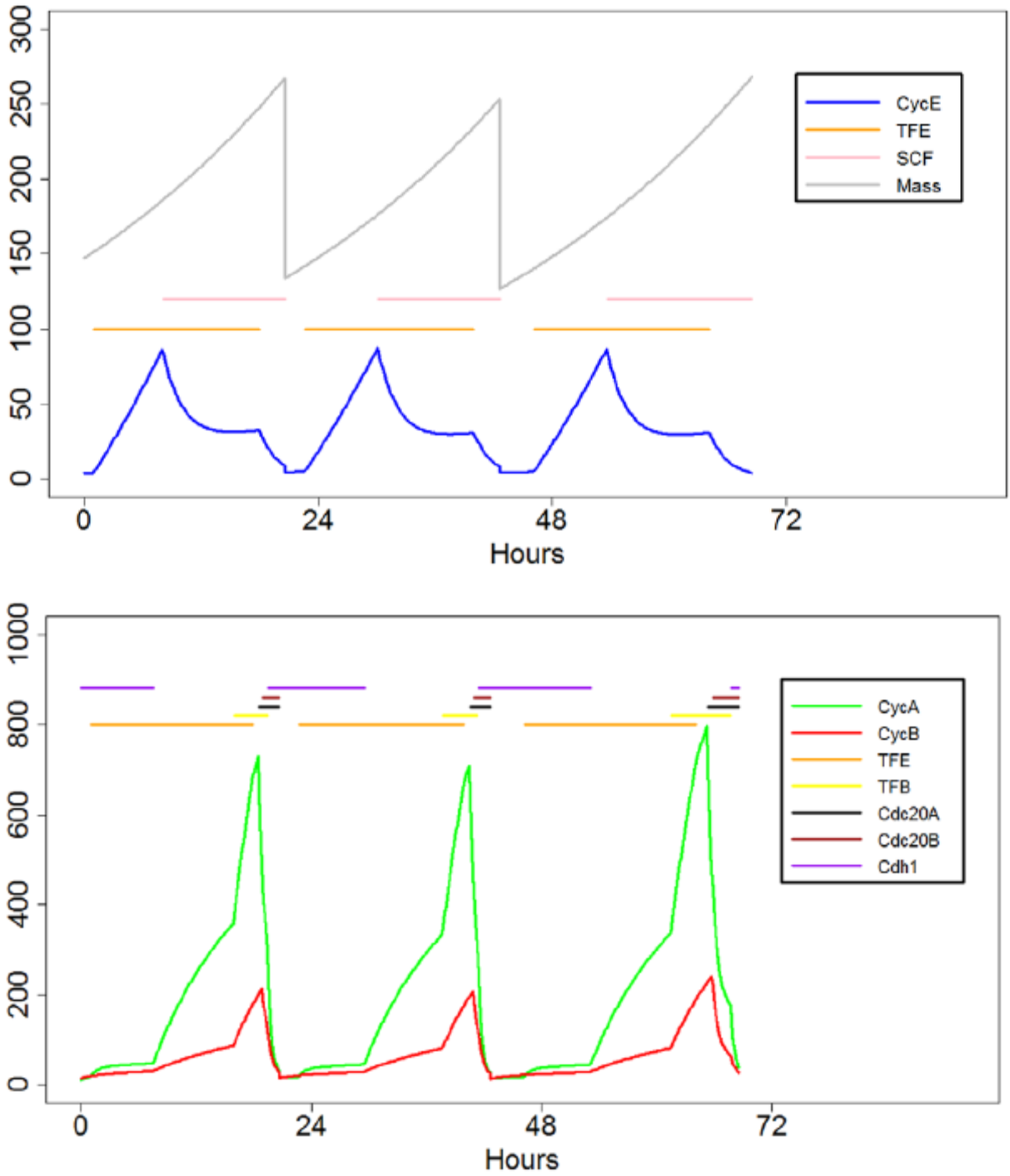


Figure 2.1. The model. (a) The synthesis and degradation of cyclin proteins is regulated by transcription factors (TFE and TFB) and by ubiquitination machinery (SCF, Cdc20 and Cdh1). (b) Three successive cell cycles are simulated as explained in the Methods. Upper panel: gray curve, $30 \cdot M(t)$; blue curve, $[\mathrm{CycE}] \cdot M(t)$; the gold line and the pink line indicate the time periods when TFE = 1 and SCF = 1, respectively. Lower panel: green curve, $[\mathrm{CycA}] \cdot M(t)$; red curve, $[\mathrm{CycB}] \cdot M(t)$; the colored bars indicate the time periods when the Boolean variables are active, according to the legend in the inset.

In our model, the two transcription factors and the four ubiquitination factors are each represented by a Boolean variable, B_{TFE}, etc. For each cyclin component we write an ordinary differential equation, $\mathrm{d[CycX]}/\mathrm{d}t = k_{\mathrm{sx}} - k_{\mathrm{dx}}[\mathrm{CycX}]$, where the rate 'constants' for synthesis and degradation, k_{sx} and k_{dx}, depend on the Boolean variables (see Table 2.1). Hence, each cyclin concentration is governed by a piecewise linear ODE. The parameters in the model (k'_{sx}, k''_{sx}, etc.) are assigned numerical values (Table 2.1), chosen to fit observations of how fast cyclins accumulate and disappear during different phases of the cell cycle.

Next, we must assign rules for updating the Boolean variables in the model. We assume that the Boolean variables follow a strict sequence of states (see Table 2.1) that corresponds roughly to the super highway discovered by Li et al. (2004). This sequence of states conforms to current ideas of how the mammalian cell cycle is regulated. Newborn cells are said to be in 'G1a' state, because they are not yet committed to a new round of DNA synthesis and mitosis. The transcription factors, TFE and TFB, are silent, and Cdh1/APC-C is active, so the levels of cyclins A, B and E are low in newborn cells. For a mammalian cell to leave the G1a state and commit to a new round of DNA replication and division, it must receive a specific set of extracellular signals (growth factors, matrix binding factors, etc.), which up-regulate the activity of TFE. We assume that these 'proliferation signals' are present and that our (simulated) cell spends only a few hours in G1a before transiting into G1b.

Table 2.1. Hybrid model of mammalian cell cycle control.

$$\frac{d[CycA]}{dt} = k_{sa} - k_{da}[CycA]$$

$$\frac{d[CycB]}{dt} = k_{sb} - k_{db}[CycB]$$

$$\frac{d[CycE]}{dt} = k_{se} - k_{de}[CycE]$$

$$\frac{dM}{dt} = \gamma \cdot M$$

$$k_{sa} = k'_{sa} + k''_{sa}\, B_{TFE} + k'''_{sa}\, B_{TFB}$$

$$k_{da} = k'_{da} + k''_{da}\, B_{Cdc20A} + k'''_{da}\, B_{Cdh1}$$

$$k_{sb} = k'_{sb} + k''_{sb}\, B_{TFB}$$

$$k_{db} = k'_{db} + k''_{db}\, B_{Cdc20B} + k'''_{db}\, B_{Cdh1}$$

$$k_{se} = k'_{se} + k''_{se}\, B_{TFE}$$

$$k_{de} = k'_{de} + k''_{de}\, B_{SCF}$$

$M = \delta \cdot M$ at division

$k'_{sa} = 5$ $\quad k''_{sa} = 6$ $\quad k'''_{sa} = 20$

$k'_{da} = 0.2$ $\quad k''_{da} = 1.2$ $\quad k'''_{da} = 1.2$

$k'_{sb} = 2.5$ $\quad k''_{sb} = 6$

$k'_{db} = 0.2$ $\quad k''_{db} = 1.2$ $\quad k'''_{db} = 0.3$

$k'_{se} = 0.02$ $\quad k''_{se} = 2$

$k'_{de} = 0.02$ $\quad k''_{de} = 0.5$

$\gamma = 0.029\ \text{hr}^{-1}$ $\quad \delta = 0.5 \cdot G$

$G: \mu = 1, \sigma = 3$

G is a Gaussian random variable with mean = 1, $\sigma = 3.3\%$

State	Phase	B_{TFE}	B_{SCF}	B_{TFB}	B_{Cdc20A}	B_{Cdc20B}	B_{Cdh1}	Condition for exit	λ (h)
1	G1a	0	0	0	0	0	1	none	2
2	Early G1b	1	0	0	0	0	1	[CycE]*M = θ_E	0
3	Late G1b	1	0	0	0	0	0	[CycA] > θ_A	0.01
4	S	1	1	0	0	0	0	$T_{\min}$ = 7 h	1
5	G2	1	1	1	0	0	0	[CycB] > θ'_B	0.5
6	Prophase	0	1	1	0	0	0	none	0.75
7	Metaphase	0	1	1	1	0	0	none	1.5
8	Anaphase	0	1	1	1	1	0	none	0.5
9	Telophase	0	1	0	1	1	1	[CycB] < θ''_B	0.025

$\theta_A = 12.5,\ \theta'_B = 21.25,\ \theta''_B = 3,\ \theta_E = 80$

In our model, the time spent in G1a is an exponentially distributed random variable with mean = 2 h. When the cell passes the 'restriction point' and enters G1b, TFE is activated and CycE begins to accumulate. Among other chores, Cdk2/CycE inactivates Cdh1/APC-C, allowing Cdk2/CycA dimers to accumulate. In our model, the transition from early G1b to late G1b is weakly size dependent, because the condition for this transition is that [CycE]*Mass exceeds a certain threshold (Θ_E). Because this transition depends on cell mass, those cells that are larger than average tend to make the transition sooner, and cells that are smaller than average tend to make the transition later. This effect allows the cell population to achieve a stable size distribution. In the late G1b state, CycA/Cdk2 level rises to a certain threshold (Θ_A), when it triggers entry into S phase. Cdk2/CycA also promotes the degradation of cyclin E by SCF during S phase. We assume that DNA synthesis requires at least 7 h.

Cyclin B begins to accumulate in late G1 and S, after Cdh1 is inactivated, but the major accumulation of cyclin B protein occurs in G2 phase, after DNA synthesis is completed and TFB is activated. The G2—M transition is delayed until enough Cdk1/CycB dimer accumulates ([CycB] > Θ_B') to promote entry into prophase and the appearance Cdc20A/APC-C, which begins the process of cyclin A degradation (Geley et al, 2001; Harper et al, 2002; Peters, 2002). Cdc20B/APC-C is activated at the metaphase—anaphase transition, where it promotes three crucial tasks: (1) separation of sister chromatids by the mitotic spindle, (2) partial degradation of cyclin B, and (3) re-activation of Cdh1. Cdh1/APC-C degrades Cdc20 (Pfleger & Kirschner, 2000), and then finishes the job of cyclin B degradation (telophase). When [CycB] drops below the threshold Θ_B", the cell finishes telophase and divides into two newborn daughter cells in G1 phase (unreplicated chromosomes) with low levels of cyclins A, B and E.

We assume that cell division is symmetric, with some variability; i.e., the mass of the two daughter cells at birth are δM_{div} and $(1-\delta)M_{div}$, where M_{div} = mass of mother cell at division, and δ is a Gaussian-distributed random variable with mean = 0.5 and standard deviation = 0.0167. In all simulations reported here we assume that cells grow exponentially between birth and division. However, we have also simulated linear growth, and the results are not significantly different.

We introduce stochastic effects into the model by assuming that the time spent in each state of the Boolean subsystem, as it moves along the super highway, has a random component (T_i^r) as well as a deterministic component (T_i^d): $T_i = T_i^d + T_i^r$. From Table 2.1, we see that $T_i^d = 0$ for i = 1, 6, 7, 8, and $T_4^d = 7$ h. For the remaining cases (i = 2, 3, 5, 9), T_i^d is however long it takes for the cyclin variable to reach its threshold. The stochastic component for each transition is a random number chosen from an exponential distribution with mean = λ_i. The random time delay is calculated from a uniform random deviate, r, by the formula $T_i^r = -\lambda_i \ln(r)$. The values chosen for the λ_i's are given in Table 2.1.

In the Methods section, we describe how we simulate the progression of a single cell through its DNA replication/division cycle. Because the model's differential equations are piecewise linear, they can be solved analytically, and an entire 'cell cycle trajectory' can be determined by computing a few random numbers and solving some algebraic equations. A typical result of such simulations, over three cell cycles, is illustrated in Figure 2.1(b). Not surprisingly, the accumulation and loss of the cyclins correlate with the activities of the cyclin regulators. At the

beginning of each cycle, the cell starts in State 1 (G1a phase in Table 2.1), with low levels of all cyclin because TFE and TFB are off and Cdh1 is on. When the cell leaves G1a, TFE turns on and cyclin E rises rapidly, but cyclin A increases only modestly, because Cdh1 is still active in early G1b. Cdh1 turns off when cyclin E level crosses Θ_E, allowing cyclin A to increase dramatically in late G1b and drive the cell into S phase (State 4). Cyclin B increases modestly in late G1 and S phase, because Cdh1 is off but TFB has not yet turned on. Cyclin E is degraded in S phase, because SCF is now active. When the cell finishes DNA synthesis, TFB turns on, causing further increase of cyclins A and B. When cyclin B level rises above its first threshold, Θ_B', the cell enters prophase (State 6) and then prometaphase-metaphase (State 7). During State 7, cyclin A level drops precipitously because Cdc20A is turned on. After the replicated chromosomes are fully aligned on the mitotic spindle, Cdc20B turns on (State 8) and cyclin B is partially degraded. Cdc20B activates Cdh1 (State 9) and cyclin B is degraded even faster. When cyclin B level drops below its second threshold, Θ_B", the cell divides and returns to G1a (State 1).

Cyclin distributions in an asynchronous culture

Our first test for the hybrid model is to simulate flow cytometry measurements of the DNA content and cyclin levels in an asynchronous population of RKO (colon carcinoma) cells (Yan et al, 2004). In the data set, a typical scatter plot has about 65000 data points, each point displaying the measurements of two observables in a single cell chosen at random from the cell cycle (see Figure 2.2). When the data are plotted in this way, they form a cloudy tube of points through a projection of the state space (say, cyclin B versus cyclin A). Because there will be some cells from every phase of the cell cycle, the tube closes on itself. If the system were completely deterministic and the measurements were absolutely precise, the data points would be a simple closed curve (a 'limit cycle') in the state space. The data actually present a fuzzy trajectory that snakes through state space before closing on itself. The indeterminacy of the points comes (presumably) from two sources: intrinsic noise in the molecular regulatory system (modeled by the random waiting times, T_i^r) and extrinsic measurement errors, which we shall introduce momentarily. Our strategy for simulating flow-cytometry data is explained in more detail in the Methods section.

In Figure 2.2 we compare our simulated flow-cytometry scatter plots with experimental results of Yan et al. (Yan et al, 2004). We color-code each cell in the simulated plot according to which Boolean State (Table 2.1) the cell is in at the time of fixation. In Figure 2.3 we plot cyclin E fluctuations, as predicted by our model, along with a projection of the cell cycle trajectory in a subspace spanned by the three cyclin variables ((a), (b) and (e)).

Contact inhibition of cultured cells

As a further test of the utility of this modeling approach, we have used our hybrid model to simulate an exponentially growing population of an immortalized Human Umbilical Vein Endothelial cell line (HUVEC). In the experiment (see Figure 2.4(a) and the subsection "Cells, culture, and fixation" in Methods), a culture is seeded with 5×10^4 cells on 'Day 0' and allowed to grow. At Day 6, it reaches confluence and cell number plateaued at a constant level.

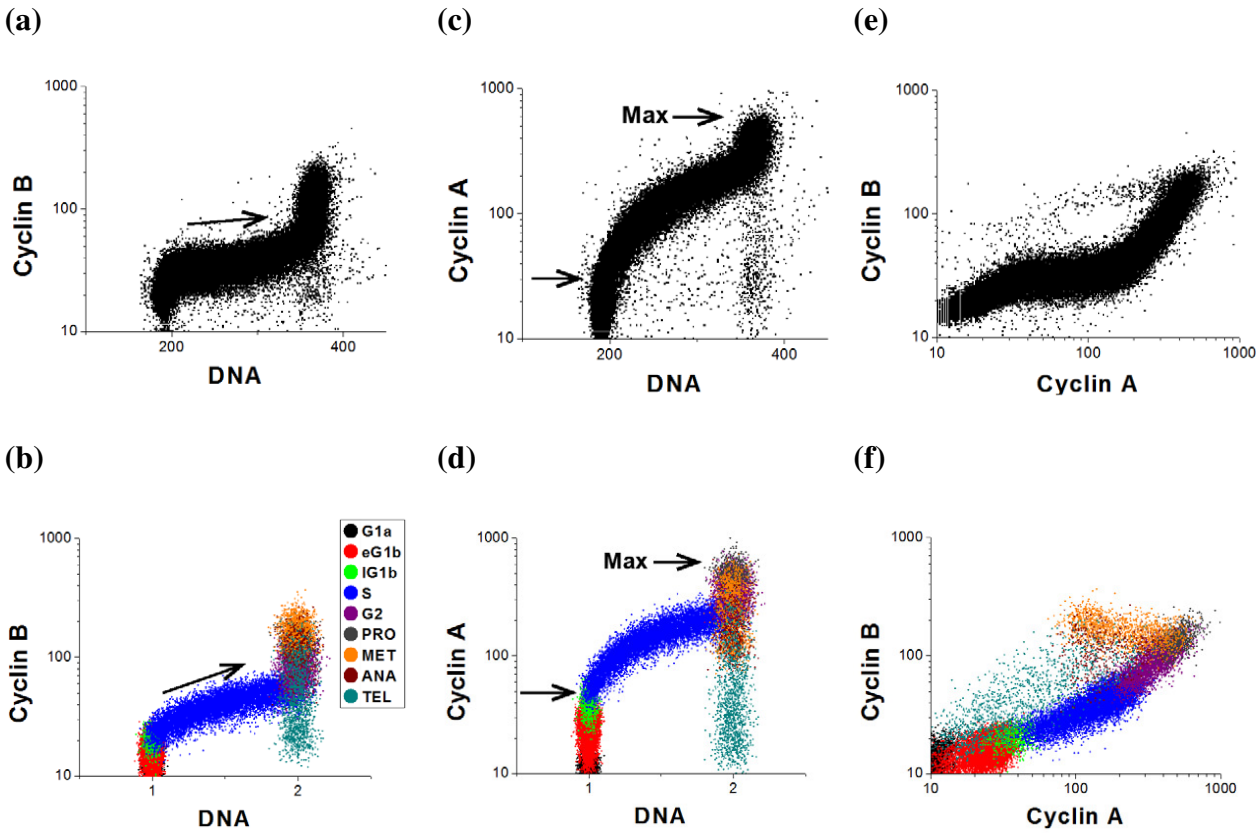


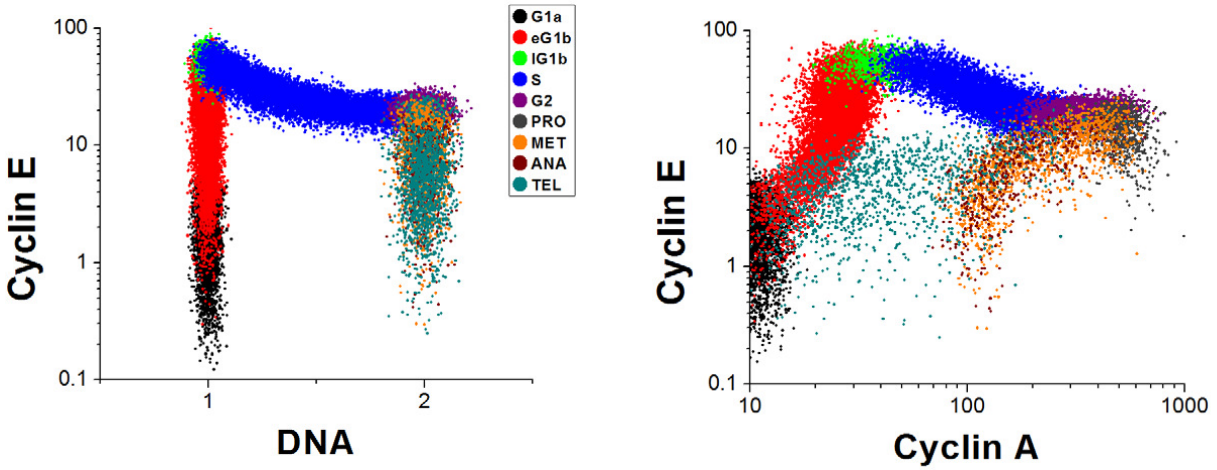
Figure 2.2. Scatter plots. (a,c,e) Flow cytometry data from Yan et al (Yan et al, 2004). Used with permission per email from Dr. James W. Jacobberger, Case Comprehensive Cancer Center, to Rajat Singhania April 12, 2011, attached. DNA = 190 corresponds to G1 and DNA = 380 corresponds to G2/M. (b,d,f) Our simulations. We are plotting the total amount of cyclin A and cyclin B per cell, i.e., [CycA]·$M(t)$ and [CycB]·$M(t)$. DNA = 1 in G0/G1 phase; = 2 in G2/M phase. Some 'instrumental noise' has been added to the calculated levels of cyclins and DNA, as described in the Methods. The arrows in (a, b) indicate the rate of cyclin B accumulation in S phase in the measurements and in the model. The arrows in (c, d) indicate the cyclin A level at the onset of DNA synthesis, compared to the maximum expression level of ~600 AU.

To apply the hybrid model to this data, we had to devise a way to model contact inhibition, which arrests cells in a stable quiescent state. To this end, we assume that the transition probability, p, for exiting State 1 is a function of the number of cells alive at that time, N:

$$p = \frac{p_0}{1 + \exp\left(\dfrac{N - N_0}{N_1}\right)}. \qquad \text{[Eq. 2.3]}$$

For $0 < N_1 << N_0$, p is a sigmoidal function of N that drops abruptly from p_0 to 0 for $N > N_0$. For each cell in this simulation, we set λ_1 (the mean for the random time spent in G1a) to $1/p$, and we choose $p_0 = 0.5\ \text{h}^{-1}$ to conform to the value of λ_1 in Table 2.1. As the population size N increases, the time spent in G1a phase increases until cells eventually arrest in State 1, and the growth curve, $N(t)$, levels off. In this case, State 1 in our model corresponds to a quiescent state (G0) in which cells are alive but not proliferating.

(a)



(b)

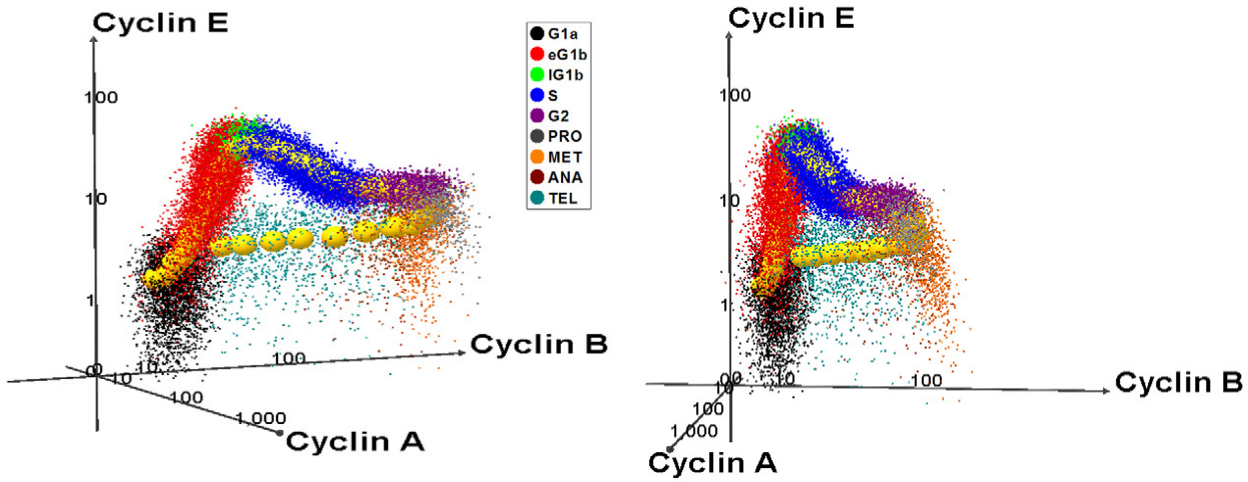


Figure 2.3. Model predictions of cyclin E dynamics. (a,b) Scatter plots. (c,d) Stochastic limit cycle in the state space of cyclins A, B and E. We provide two different perspectives of this three dimensional figure to help visualize how the cyclin levels go up and down. In addition, we have added golden-colored balls to help guide the eye along the cell cycle trajectory. Each ball represents the average of the cyclin levels of all the cells binned over a hundredth of the φ_i interval [0,1], where φ_i refers to the fraction of the cell cycle completed by cell i (as described in the Methods section). Finally, it may help to recognize that Fig. 2.2(e) is a projection of the data on the CycA-CycB plane, and Fig. 2.3(b) is a projection on the CycA-CycE plane.

To make the simulation more tractable, we start off with 500 cells (instead of 50,000 cells) and follow the lineage of each initial cell until Day 10. Every 24 hours, we compute the number of cells alive at that point of time and plot the results in Figure 2.4(a), along with the experimental data (scaled down by a factor of 100). The parameter values, N_0 =11,000 and N_1 = 500, are chosen to fit the simulation to the observed growth curve. From the model we can also compute the percentage of cells in G0/G1, S and G2/M phases on each day (see Figure 2.4(c)), and the results compare favorably with the experimental observations (see Figure 2.4(b)). Lastly, we

also simulate the patterns of cyclin A2 and cyclin B1 expression on each day for the growing population of HUVEC cells (simulations not shown).

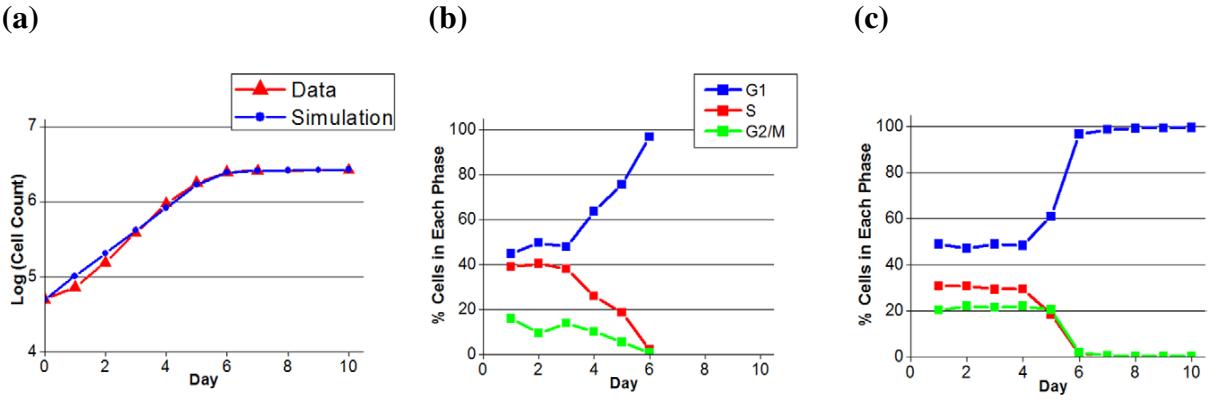


Figure 2.4. Contact inhibition of a culture of human umbilical vein endothelial cells. (a) Growth curve for the HUVEC population over 10 days, showing the base-10 logarithm of the cell count for both experimental data and our simulation (with $N_0 = 11000$ and $N_1 = 500$). (b) Daily distribution of cells across the phases of the cell cycle, from experimental data. (c) Model simulation of the phase distributions.

2.3 Discussion

We have constructed a simple, effective model of the cyclin-dependent kinase control system in mammalian cells and used the model to simulate faithfully the accumulation and degradation of cyclin proteins during asynchronous proliferation of RKO (colon carcinoma) cells. The model is inspired by the work of Li et al. (2004), who proposed a robust Boolean model of cell cycle regulation in budding yeast. Our goal was to retain the elegance of the Boolean representation of the switching network, while introducing continuous variables for cell size, cell age, and cyclin composition, in order to create a model that could be compared in quantitative detail to experimental measurements.

We have shown that this model can accurately simulate flow-cytometric measurements of cyclin abundances in asynchronous populations of growing-dividing mammalian cells. The parameters in the model that allow for a quantitative description of the experimental measurements are easily estimated from the data itself. Now that the model is parameterized and validated for wild-type cells, we are currently extending it to handle the behavior of cell populations perturbed by drugs and by genetic interference. In some cases, only modest extensions of the model are required; in other cases, a more thorough overhaul of the way the discrete and continuous variables interact with each other is necessary.

We have chosen parameter values in our model to capture the major features of cyclin fluctuations as measured by flow cytometry during the somatic division cycle of mammalian cells. We have used a human tumor cell line to calibrate our model. Between cell lines and normal human cultured cells, there are differences in the expressions of A and B cyclins (Gong et al, 1994); however, when the levels of cyclin B1 were rigorously compared for HeLa, K562, and RKO cells, both the patterns and magnitudes of expression are remarkably similar, apparently dependent to some degree on the rate of population growth (Frisa & Jacobberger,

2009). In addition, the patterns of expression of cyclins A2 and B1 are similar for these human tumor cell lines and stimulated normal human circulating lymphocytes (figures not shown). Overall, the simulation outputs have satisfying similarity both in pattern and magnitude to the real data for RKO cells, and our simulated expression patterns of cyclins A, B and E for the tumor cell line are quite similar to the simulated expression patterns in HUVEC cells (figures not shown).

However, there remain some inconsistencies between our mathematical simulations and our experimental observations that point out where future modifications to the model are needed. For example, in the model DNA synthesis starts when cyclin A has accumulated to ~8% of its maximum level (see arrow in Figure 2.2(d); 50/600 ≈ 8%), whereas in our measurements DNA synthesis starts when cyclin A is ~5% of its maximum level (see arrow in Figure 2.2(c)). This discrepancy is tempered by the fact that we are not confident of the quantitative accuracy of cyclin A expression levels below ~4% of its maximum level in Figure 2.2(c). Where we place the minimum expression level of cyclin A in see Figure 2.2(d) affects our estimate of the cyclin A level at onset of DNA synthesis (50 AU at present). By lowering the minimum expression level of cyclin A below 10 AU in Figure 2.2(a) (e.g., by lowering k'_{sa}), we could line up the two arrows in Figures 2.2(c) and 2.2(d). Nonetheless, we observe (figures not shown) that cyclin A expression correlates highly with BrdU incorporation, suggesting that significant accumulation of cyclin A begins simultaneously with the onset of DNA synthesis, whereas in our model cyclin A production begins in mid-G1 phase. This discrepancy could be minimized by lowering the cyclin A threshold (Θ_A) in the model.

The simulation (see Figure 2.2(b)) captures the observed accummulation of cyclin B in late G1 (when Cdh1 turns off), but the simulated rise in cyclin B during S phase appears to be faster than the observed rise (Jacobberger et al, 1999) (compare the arrows in Figures 2.2(a) and 2.2(b)). The simulation does capture the rapid accumulation of cyclin B observed in G2. Finally, while we did not calibrate the cyclin E expression parameters to any specific dataset, the pattern of expression in Figure 2.3(a) is quite similar to expected expression patterns for normal human somatic cells and some human tumor cell lines (Darzynkiewicz et al, 1996).

We believe that our hybrid approach will be generally useful for modeling macromolecular regulatory networks in cells, because it combines the qualitative appeal of Boolean models with the quantitative realism of reaction kinetic models.

2.4 Methods

Simulations

We simulate a flow cytometry experiment with our hybrid model in two steps.

Step 1: Creating complete 'life histories' for thousands of cells. At the start of the simulation, we specify initial conditions at the beginning of the cycle (State 1) for a progenitor cell. We used the following initial values of the state variables: [CycA] = [CycB] = [CycE] = 1 and M = 3. Our strategy is to follow this cell through its cycle until it divides into two daughters. We then choose one of the two daughters at random and repeat the process, continuing for 32500 iterations. We

discard the first 500 cells, and keep a sample of 32000 cells that have completed a replication-division cycle according to our model. In the second step, we create a simulated sample of 32000 cells chosen at random phases of the cell cycle, to represent the cells that were assayed by the flow cytometer.

Let us consider cell i ($1 < i < 32500$) at the time of its birth, t_{i0}. By definition, this cell is in State 1, and we assume that we know its birth mass, $M(t_{i0})$, and its starting concentrations of cyclins A, B and E. Denote the starting concentrations as [CycA(t_{i0})], [CycB(t_{i0})], [CycE(t_{i0})]. In the ensuing discussion, unless it is necessary for clarity, we drop the i subscript, it being understood that we are talking about a representative cell in the population. We will follow this cell until it divides to produce a daughter cell with known concentrations of cyclins.

According to Table 2.1, a cell in State 1 has no special conditions to satisfy before moving to State 2. Hence the residence time in State 1 is a random number T_1^r chosen from an exponential distribution with mean $\lambda_1 = 2$ h. The cell enters State 2 at $t_1 = t_0 + T_1^r$. Assuming exponential growth, its size at this time is $M(t_1) = M(t_0)\exp\{\gamma(t_1 - t_0)\} = M(t_0)\exp\{\gamma A_1\}$, where γ is the specific growth rate of the culture and $A_1 = t_1 - t_0$ is the age of the cell when it exits State 1. To compute the cyclin concentrations at $t = t_1$, we use cyclin A as an example. During the interval $t_0 < t < t_1$, [CycA] satisfies a linear ODE with effective rate constants $k_{sa1} = k'_{sa} = 5$ and $k_{da1} = k'_{da} + k'''_{da} = 1.4$, because $B_{TFE} = B_{TFB} = B_{Cdc20A} = 0$ and $B_{Cdh1} = 1$ for a cell in State 1. We can compute the concentration of cyclin A at any time during this interval from

$$[CycA(t)] = \frac{k_{sa1}}{k_{da1}} + \left([CycA(t_0)] - \frac{k_{sa1}}{k_{da1}} \right) e^{-k_{da1}(t-t_0)}, t_o \le t \le t_1 \qquad \text{[Eq. 2.4]}$$

Setting $t = t_1$ in this equation gives the number we seek. In this fashion, we start tabulating the following information for each simulated cell:

Time	t_0	t_1	t_2	…
Enter State	1	2	3	…
Age	0	$A_1 = t_1 - t_0$	$A_2 = t_2 - t_0$	
Size	$M(t_0)$	$M(t_1)$	$M(t_2)$	…
Cyclin A	[CycA(t_0)]	[CycA(t_1)]	[CycA(t_2)]	…
Cyclin B	[CycB(t_0)]	[CycB(t_1)]	[CycB(t_2)]	…
Cyclin E	[CycE(t_0)]	[CycE(t_1)]	[CycE(t_2)]	…

Notice that, at $t = t_1$ when the cell enters State 2, the transcription factor (TFE) for cyclins E and A turns on, and these cyclins start to accumulate. The cell cannot leave State 2 until cyclin E accumulates to a sufficiently high level: $[CycE](t)\cdot M(t) = \Theta_E$, according to Table 2.1. When this condition is satisfied, the cell leaves State 2 and enters State 3. The size dependence on this transition is a way to couple cell growth to the DNA replication-division cycle. According to the parameter settings in Table 2.1, there is no stochastic component to the transition out of State 2.

We continue in this fashion until the cell leaves State 9 and returns to State 1, when cyclin B is degraded at the end of mitosis. This is the signal for cell division. The age of the cell at division

is $A_9 = t_9 - t_0$, and the mass of the cell at division is $M(t_9) = M(t_0)\exp(\gamma \cdot A_9)$. The mass of the daughter cell at the beginning of her life history is $M_{\text{daughter}}(t_0) = \delta \cdot M_{\text{mother}}(t_9)$, where δ is a random number sampled from a normal distribution of mean 0.5 and standard deviation 0.0167 to allow for asymmetries of cell division.

Notice that simulating the life history of a single cell only requires generating about a dozen random numbers and performing a handful of algebraic calculations. At no point do we need to solve differential equations numerically. Hence we can quickly calculate the life histories of tens of thousands of cells.

Step 2: Finding the DNA and cyclin levels of each cell in an asynchronous sample. In the flow cytometry experiments of Yan et al. (Yan et al, 2004), a random sample of cells is taken from an asynchronous population, the cells are fixed and stained, and then run one-by-one through laser beams where fluorescence measurements are made. So each data point consists of measurements of light scatter (related to cell size) and fluorescence proportional to DNA and cyclin content for a single cell taken at some random point in the cell cycle. To simulate this experiment, we must assign to each of our 32000 simulated cells a number φ_i selected randomly from the interval [0,1], where φ_i refers to the fraction of the cell cycle completed by cell i when it was fixed and stained for measurement. Because each mother cell divides into two daughter cells, the density of cells at birth, $\varphi = 0$, is twice the density of cells at division, $\varphi = 1$. The 'ideal' probability density for an asynchronous population of cells expanding exponentially in number is

$$f(\varphi) = (\ln 2) \cdot 2^{1-\varphi} \qquad \text{[Eq. 2.5]}$$

According to the 'transformation method' (Press et al, 1992, Chapter 7.2), we compute φ as

$$\varphi = \log_2\left(\frac{2}{2-r}\right) \qquad \text{[Eq. 2.6]}$$

where r is a random number chosen from a uniform distribution on [0,1]. In this way, we generate 32000 fractions, φ_i.

If φ_i is the cell-cycle location of the i^{th} cell when it is selected for the flow cytometry measurements, then its age at the time of selection is $a_i = \varphi_i \cdot A_{i9}$, where A_{i9} is the age of the i^{th} cell at division. Given a value for a_i, we then find the state n (= 1, 2, … or 9) of the i^{th} cell at the time of its selection:

$$t_{i,n-1} \le t_{i0} + a_i < t_{i,n} \qquad \text{[Eq. 2.7]}$$

where $t_{i,n}$ (as defined above) is the time at which the i^{th} cell left state n to enter state n+1.

Once we know the state n of the cell, we can compute the concentration of each cyclin in the cell at its exact age a_i by analogy to Eq. [2.4]:

$$[CycA(a_i)] = \frac{k_{sa,n}}{k_{da,n}} + \left([CycA(t_{i,n-1})] - \frac{k_{sa,n}}{k_{da,n}} \right) e^{-k_{da,n}(t_{i0}+a_i-t_{i,n-1})} \qquad \text{[Eq. 2.8]}$$

where $k_{sa,n}$ and $k_{da,n}$ are the synthesis and degradation rate constants for cyclin A in state n. This is a straightforward calculation because in Step 1 we stored the values of t_n and [CycA(t_n)] for every state of each cell. We can also calculate the mass of cell i at the time of its selection:

$$M(a_i) = M(t_{i0}) \cdot \exp(\gamma \cdot a_i) \qquad \text{[Eq. 2.9]}$$

where $M(t_{i0})$ is the mass at birth of cell i and γ is the specific growth rate of the culture. Because the flow cytometer measures the total amount of fluorescence proportional to all cyclin A molecules in the i^{th} cell, we take as our measurable the product of [CycA(a_i)] times $M(a_i)$.

Lastly we determine the DNA content of cell i at age a_i according to:

DNA = 1 for $t_{i0} \le t_{i0} + a_i < t_{i3}$ = entry of i^{th} cell into S phase
DNA = 1 + $(t_{i0} + a_i - t_{i3})/(t_{i4} - t_{i3})$ for $t_{i3} \le t_{i0} + a_i < t_{i4}$ = exit of i^{th} cell from S phase
DNA = 2 for $t_{i4} \le t_{i0} + a_i < t_{i9}$

Now we have simulated values for the measurable quantities of each cell at the time point in the cell cycle when it was selected for analysis. Before plotting these numbers, we should take into account experimental errors, such as probe quality, fixation, staining and measurement. We do so by multiplying each measurable quantity (DNA content and cyclin levels) by a random number chosen from a Gaussian distribution with mean 1 and standard deviation = 0.03 for DNA measurements and 0.15 for cyclin measurements. These choices give scatter to the simulated data that is comparable to the scatter in the experimental data.

Cells, culture, and fixation

Culture and fixation of RKO cells have been described (Yan et al, 2004). The immortalized HUVEC cells (Freedman & Folkman, 2005) at passage 93 were seeded at 2.5×10^3 cells/cm^2 in 10 ml EGM-2 media with 2% fetal bovine serum (Lonza, Basel). Duplicate plates were prepared for each time point at days 1, 2, 3, 4, 5, 6, 7, 10, and 15. Cells were fed every other day by replacing half the volume of used media. At the indicated times, cells were trypsinized, washed, and cell counts performed with a Guava Personal Cytometer (Millipore, Billerica, MA). Fixation was as previously described (Schimenti & Jacobberger, 1992); briefly, cells were treated with 0.125% formaldehyde (Polysciences, Warrington, PA) for 10 min at 37°C, washed, then dehydrated with 90% Methanol. Cells were fixed in aliquots of 1×10^6 cells (days 1 – 3) or 2×10^6 (days 4 – 15). Fixed cell samples were stored at -20°C until staining for cytometry.

Immunofluorescence staining, antibodies, flow cytometry

Staining and cytometry for RKO cells have been described (Yan et al, 2004). Briefly, cells were trypsinized, fixed with 90% MeOH, washed with phosphate buffered saline, then stained with monoclonal antibodies reactive with cyclin B1, cyclin A, hosphor-S10-histone H3, and with 4',6-diamidino-2-phenylindole (DAPI). For a detailed, updated version of antibodies, staining,

and cytometry for cyclins A2 and B1, Phosphor-S10-histone H3, and DNA content, see Jacobberger et al. [38].

Data pre-processing

Data pre-processing was performed with WinList (Verity Software House, Topsham, ME). Doublet discrimination (peak versus area DAPI plot) was used to limit the analysis to singlet cells; non-specific binding was used to remove background fluorescence from the total fluorescence related to cyclin A2 and B1 staining. The phycoerythrin channel (cyclin A2) was compensated for spectral overlap from FITC or Alexa Fluor 488. For simplification, very large 2C G1 HUVEC cells and any cells cycling at 4C → 8C were removed from the analysis. These were present at low frequency. Data were written as text files then transferred to Microsoft Excel.

2.5 References

1. Mitchison JM (1971) The Biology of the Cell Cycle. Cambridge UK: Cambridge Univ. Press. 320 p.
2. Zetterberg A, Larsson O, Wiman KG (1995) What is the restriction point? Curr Opin Cell Biol 7: 835-842.
3. Chen KC, Csikasz-Nagy A, Gyorffy B, Val J, Novak B, et al. (2000) Kinetic analysis of a molecular model of the budding yeast cell cycle. Mol Biol Cell 11: 369-391.
4. Novak B, Pataki Z, Ciliberto A, Tyson JJ (2001) Mathematical model of the cell division cycle of fission yeast. Chaos 11: 277-286.
5. Chen KC, Calzone L, Csikasz-Nagy A, Cross FR, Novak B, et al. (2004) Integrative analysis of cell cycle control in budding yeast. Mol Biol Cell 15: 3841-3862.
6. Calzone L, Thieffry D, Tyson JJ, Novak B (2007) Dynamical modeling of syncytial mitotic cycles in Drosophila embryos. Mol Syst Biol 3: 131-141.
7. Novak B, Tyson JJ (1993) Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos. J Cell Sci 106: 1153-1168.
8. Pomerening JR, Kim SY, Ferrell Jr. JE (2005) Systems-level dissection of the cell-cycle oscillator: bypassing positive feedback produces damped oscillations. Cell 122: 565-578.
9. Aguda BD, Tang Y (1999) The kinetic origins of the restriction point in the mammalian cell cycle. Cell Prolif 32: 321-335.
10. Qu Z, Weiss JN, MacLellan WR (2003) Regulation of the mammalian cell cycle: a model of the G1-to-S transition. Am J Physiol Cell Physiol 284: C349-C364.
11. Novak B, Tyson JJ (2004) A model for restriction point control of the mammalian cell cycle. J Theor Biol 230: 563-579.
12. Sha W, Moore J, Chen K, Lassaletta AD, Yi C-S, et al. (2003) Hysteresis drives cell-cycle transitions in *Xenopus laevis* egg extracts. Proc Natl Acad Sci USA 100: 975-980.
13. Pomerening JR, Sontag ED, Ferrell Jr. JE (2003) Building a cell cycle oscillator: hysteresis and bistability in the activation of Cdc2. Nature Cell Biol 5: 346-351.
14. Li F, Long T, Lu Y, Ouyang Q, Tang C (2004) The yeast cell-cycle network is robustly designed. Proc Natl Acad Sci USA 101: 4781-4786.
15. Davidich MI, Bornholdt S (2008) Boolean network model predicts cell cycle sequence of fission yeast. PLoS One 3: e1672.
16. Faure A, Naldi A, Chaouiya C, Thieffry D (2006) Dynamical analysis of a generic Boolean model for the control of mammalian cell cycle. Bioinformatics 22: e124-131.
17. Fantes PA, Nurse P (1981) Division timing: controls, models and mechanisms. In: John PCL, editor. The Cell Cycle. Cambridge UK: Cambridge Univ. Press. pp. 11-33.
18. Tyson JJ (1985) The coordination of cell growth and division -- intentional or incidental? Bioessays 2: 72-77.
19. Tyson JJ (1987) Size control of cell division. J Theor Biol 126: 381-391.
20. Darzynkiewicz Z, Crissman H, Jacobberger JW (2004) Cytometry of the cell cycle: cycling through history. Cytometry A 58: 21-32.
21. Glass L, Kauffman SA (1973) The logical analysis of continuous, non-linear biochemical control networks. J Theor Biol 39: 103-129.
22. Glass L, Pasternack J (1978) Stable oscillations in mathematical models of biological control systems. Journal of Mathematical Biology 6: 207-223.
23. Matsuno H, Inouye ST, Okitsu Y, Fujii Y, Miyano S (2006) A new regulatory interaction suggested by simulations for circadian genetic control mechanism in mammals. J Bioinform Comput Biol 4: 139-153.

24. Bosl WJ (2007) Systems biology by the rules: hybrid intelligent systems for pathway modeling and discovery. BMC Syst Biol 1: 13.
25. Li C, Nagasaki M, Ueno K, Miyano S (2009) Simulation-based model checking approach to cell fate specification during Caenorhabditis elegans vulval development by hybrid functional Petri net with extension. BMC Syst Biol 3: 42.
26. Alur R, Dang T, Esposito JM, Fierro RB, Hur Y, et al. (2001) Hierarchical Hybrid Modeling of Embedded Systems. Proceedings of the First International Workshop on Embedded Software: Springer-Verlag. pp. 14-31.
27. Fishwick PA (2007) Handbook of dynamic system modeling. Boca Raton: Chapman & Hall/CRC. 760 p.
28. Klee H, Allen R (2011) Simulation of dynamic systems with MATLAB and Simulink. Boca Raton, FL: CRC Press. 840 p.
29. Mosterman P (1999) An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages. In: Vaandrager F, van Schuppen J, editors. Hybrid Systems: Computation and Control: Springer Berlin / Heidelberg. pp. 165-177.
30. Deshpande A, Gollu A, Varaiya P (1997) SHIFT: A Formalism and a Programming Language for Dynamic Networks of Hybrid Automata. Hybrid Systems IV: Springer-Verlag. pp. 113-133.
31. Alur R, Grosu R, Hur Y, Kumar V, Lee I (2000) Modular Specification of Hybrid Systems in CHARON. Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control: Springer-Verlag. pp. 6-19.
32. Bengtsson J, Larsen K, Larsson F, Pettersson P, Yi W (1996) UPPAAL-a tool suite for automatic verification of real-time systems. Proceedings of the DIMACS/SYCON workshop on Hybrid systems III : verification and control: verification and control. New Brunswick, New Jersey, United States: Springer-Verlag New York, Inc. pp. 232-243.
33. Trimarchi JM, Lees JA (2002) Sibling viralry in the E2F family. Nat Rev Mol Cell Biol 3: 11-20.
34. Laoukili J, Kooistra MR, Bras A, Kauw J, Kerkhoven RM, et al. (2005) FoxM1 is required for execution of the mitotic programme and chromosome stability. Nat Cell Biol 7: 126-136.
35. Wierstra I, Alves J (2007) FOXM1, a typical proliferation-associated transcription factor. Biol Chem 388: 1257-1274.
36. Cardozo T, Pagano M (2004) The SCF ubiquitin ligase: insights into a molecular machine. Nat Rev Mol Cell Biol 5: 739-751.
37. Welcker M, Singer J, Loeb KR, Grim J, Bloecher A, et al. (2003) Multisite phosphorylation by Cdk2 and GSK3 controls cyclin E degradation. Mol Cell 12: 381-392.
38. Harper JW, Burton JL, Solomon MJ (2002) The anaphase-promoting complex: it is not just for mitosis any more. Genes Dev 16: 2179-2206.
39. Peters JM (2002) The anaphase-promoting complex proteolysis in mitosis and beyond. Mol Cell 9: 931-943.
40. Geley S, Kramer E, Gieffers C, Gannon J, Peters J-M, et al. (2001) Anaphase-promoting complex/cyclosome-dependent proteolysis of human cyclin A starts at the beginning of mitosis and is not subject to the spindle assembly checkpoint. J Cell Biol 153: 137-148.
41. Pfleger CM, Kirschner MW (2000) The KEN box: an APC recognition signal distinct from the D box targeted by Cdh1. Genes Dev 14: 655-665.
42. Yan T, Desai AB, Jacobberger JW, Sramkoski RM, Loh T, et al. (2004) CHK1 and CHK2 are differentially involved in mismatch repair-mediated 6-thioguanine-induced cell cycle checkpoint responses. Mol Cancer Ther 3: 1147-1157.
43. Gong J, Ardelt B, Traganos F, Darzynkiewicz Z (1994) Unscheduled expression of cyclin B1 and cyclin E in several leukemic and solid tumor cell lines. Cancer Res 54: 4285-4288.
44. Frisa PS, Jacobberger JW (2009) Cell cycle-related cyclin b1 quantification. PLoS One 4: e7064.

45. Jacobberger JW, Sramkoski RM, Wormsley SB, Bolton WE (1999) Estimation of kinetic cell-cycle-related gene expression in G1 and G2 phases from immunofluorescence flow cytometry data. Cytometry 35: 284-289.
46. Darzynkiewicz Z, Gong J, Juan G, Ardelt B, Traganos F (1996) Cytometry of cyclin proteins. Cytometry 25: 1-13.
47. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C. The art of scientific computing. Cambridge: Cambridge University Press. 707-752 p.
48. Freedman DA, Folkman J (2005) CDK2 translational down-regulation during endothelial senescence. Exp Cell Res 307: 118-130.
49. Schimenti KJ, Jacobberger JW (1992) Fixation of mammalian cells for flow cytometric evaluation of DNA content and nuclear immunofluorescence. Cytometry 13: 48-59.

Chapter 3. Finding Protein Regulatory Networks that Exhibit Near-Perfect Adaptive Responses

3.1 Introduction

Living cells must adapt to environmental conditions in ways that promote their own survival and reproduction (for unicellular organisms), or the fitness of the multicellular organism to which they belong. Cells have evolved sensory systems that detect environmental cues and signal processing networks that interpret these cues and determine the appropriate response of the cell. In many cases the appropriate response is to detect an abrupt change in the external signal and then to 'adapt' (i.e., return to the stable resting state) in the presence of constant stimulus. For example, our sense of smell exhibits this sort of adaptive response. A change in odors in a room will be first picked up, but eventually we will be desensitized to the odor. In other words, we go back to the 'resting state' even though the signal (the odor) that triggered the response is still present. It is always the change in the level of the signal that determines the adaptive response, not the absolute value of the signal.

We might define 'perfect' adaptation as the case when the signal processing network always returns to the same steady state regardless of the final, constant level of stimulus (Figure 3.1, green line). By this definition, perfect adaptation may be impossible (or extremely rare), but near-perfect adaptation (Figure 3.1, blue line) might be good enough to serve the purpose of a living, responding cell.

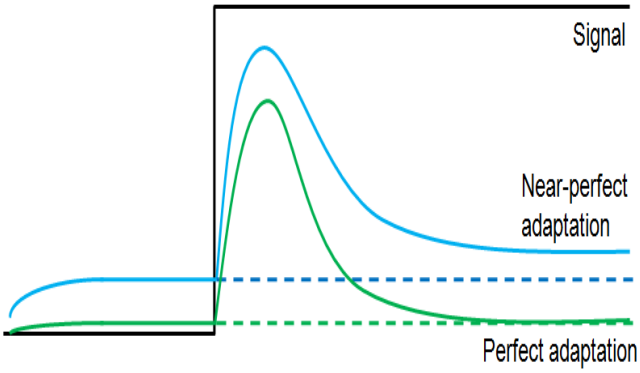


Figure 3.1. Perfect (green) and near-perfect adaptation (blue) in response to a persistent signal (black).

The adaptation behavior is crucial in biological systems in contexts as varied as chemotaxis in *Escherichia coli* (Berg & Brown, 1972; Macnab & Koshland, 1972), adenylate cyclase activation in *Dictyostelium* (Dinauer et al, 1980) and osmo-response in yeast (Mettetal et al, 2008). Various types of theoretical models have been proposed to account for perfect or near-perfect adaptation in these contexts (Levchenko & Iglesias, 2002; Mello & Tu, 2003; Parent & Devreotes, 1999; Yi et al, 2000). Initial models (Hauri & Ross, 1995; Knox et al, 1986) achieved perfect adaptation of receptor activity through fine-tuning of the biochemical parameters (reaction rate constants and enzyme concentrations). In an alternative model for adaptation, put forward by Barkai & Leibler (Barkai & Leibler, 1997), the steady state receptor

activity is independent of the ligand level, and the biochemical parameters that produce near-perfect adaptation can vary freely over orders of magnitude.

The goal of our theoretical study is to identify the topologies of 3-node motifs that show near-perfect adaptive responses. A motif is a simple pattern of activation and inhibition among a small number of interacting molecular species (Tyson & Novak, 2010). The idea that different motifs can carry out specific information-processing functions has been systematically investigated previously (Alon, 2007; Tyson et al, 2003).

Each motif in our analysis consists of three nodes, representing three interacting molecular species (see Figure 3.2). The signal goes into node 1 while the response is read from node 3. Node 2 adds complexity to the behavior of each motif through its interactions with the other two nodes. There are six possible interactions within each of these three-node motifs, since each node can regulate the other two nodes. Each regulation can be an activation, an inactivation, or just be absent. Therefore, there are a total of 3^6, or 729, possible 3-node topologies. We exclude self-activation of nodes as we think that they are unlikely to occur frequently in protein regulatory networks.

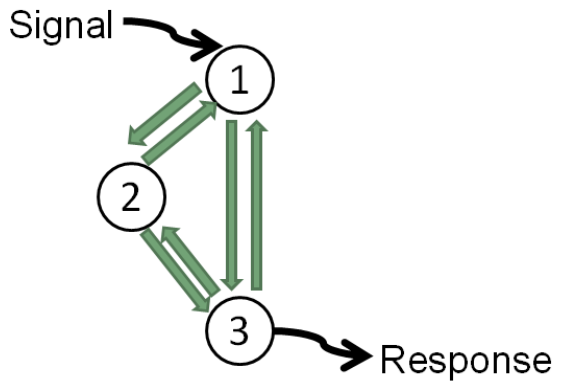


Figure 3.2. The three nodes of a motif; the six regulations permitted in our motifs are shown by the green arrows.

Quantifying the degree of near-perfect adaptation, using a scoring metric (see section 3.4 Methods: 'Generating a Single Score'), is essential to our process of identifying the 3-node topologies that show this biological behavior. Interestingly, we find that the same topology can score high or low depending on the choice of the model parameters. Therefore, another goal of our study is to identify the regions of parameter space in which each high-scoring motif does well.

A study in the journal *Cell* is a welcome first step in finding near-perfect adaptation motifs (Ma et al, 2009). They find that either Incoherent Feed Forward Loops (IFFLs) or Negative Feedback Loops with Buffering (NFLBs) are needed to get near-perfect adaptation. We are able to validate these results, as well as significantly extend them in multiple ways by our more systematic, evolutionary approach to exploring the topology and parameter space in which near-perfect adaptation occurs. A detailed comparison of ours and Ma's methodologies and results is presented in the Methods and Discussion sections, respectively.

3.2 Results

Identifying the topologies of 3-node motifs that show near-perfect adaptive responses

Initial exploration of the entire topology space

As noted earlier, a total of 729 possible topologies can be formed by 3-node motifs without self-activations. We decided to explore the topology space by picking forty initial topologies, and letting them macromutate into other topologies. Twenty of these topologies were generated by a process in which each of the six digits of a particular topology code were chosen randomly to be one of the three possible values of 1, 2, or 3 (see section 3.4 Methods: 'Topology Representation and Parameters'). This ensured that a wide variety of activation/inactivation patterns were given a chance to exhibit adaptation. The other twenty topologies were fixed in advance. Five categories were chosen, and each category was represented by four different topologies. These five categories were: 2-edged topologies, Incoherent Feed Forward Loops (IFFLs), Coherent Feed Forward Loops (CFFLs), Negative Feedback Loops with Buffering Nodes (NFLBs) and Positive Feedback Loops with Buffering Nodes (PFLBs).

Collectively, more than 500 unique topologies were generated across the forty runs. A particular topology may be present in more than one run, so the outputs from all forty runs were collated before calculating all topologies' average scores. We find that the scores range from nearly 0 to 14.18. The remarkable pattern we noticed is that all topologies that scored more than 6 belonged to two specific categories of Incoherent Feed Forward Loops (IFFLs), as shown in Table 3.1. According to Uri Alon's notation (Alon, 2007), the first category is called IFFL-1, and the second category IFFL-4 (see Figure 3.3).

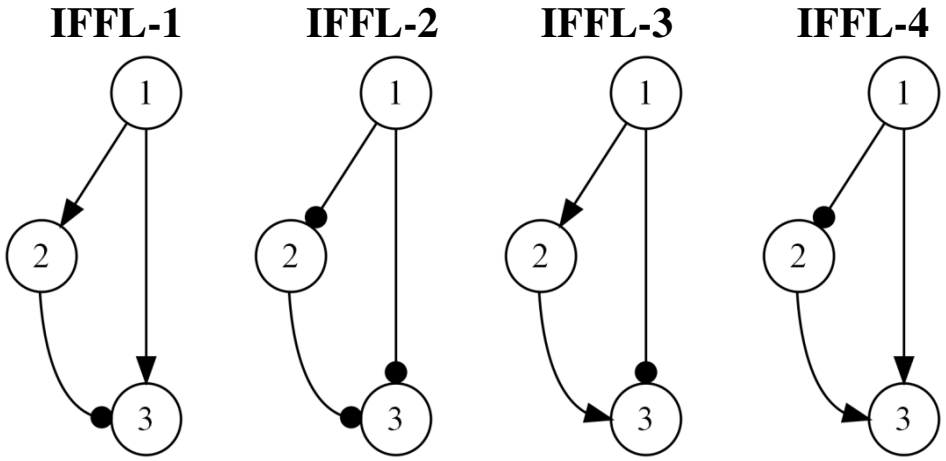


Figure 3.3. The four types of basic Incoherent Feed Forward Loops. Arrow heads represent activation, and circle heads inactivation.

IFFL-1 topologies have the encoding XX3X31 in our notation, and IFFL-4 topologies XX1X33. "X" means that the digit can represent any of the three possible types of regulations.

Table 3.1. The highest scoring topologies from the initial analysis.

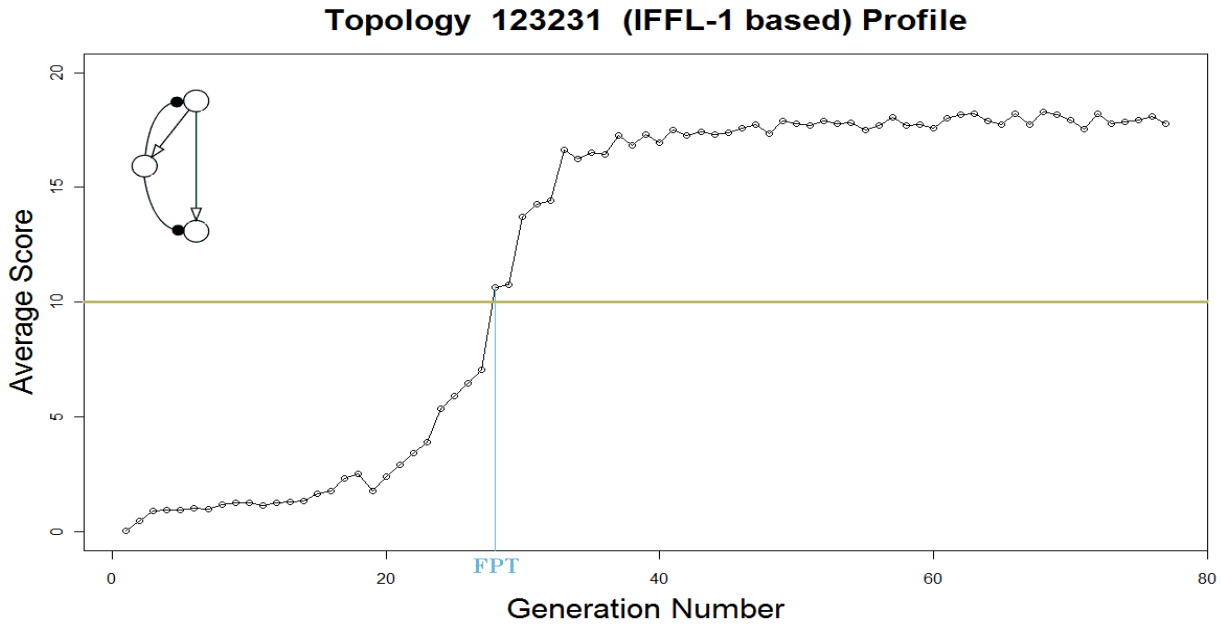
Code	< Z >	Code	< Z >	Code	< Z >
123331	14.18	113331	10.40	113231	7.81
133231	13.84	233231	10.24	233331	7.40
123231	13.46	121333	10.07	121233	7.26
133331	13.35	333331	9.19	213331	7.04
233131	13.31	333231	9.12	321233	6.83
321133	12.10	131233	9.06	323231	6.50
223131	11.42	223331	8.79	221333	6.41
123131	11.14	133131	8.78	333131	6.07
131133	11.11	223231	8.39		
111233	10.50	113131	8.08		

Topologies in red: IFFL-1's (XX3X31); in green: IFFL-4's (XX1X33).
< Z > represents average scores.

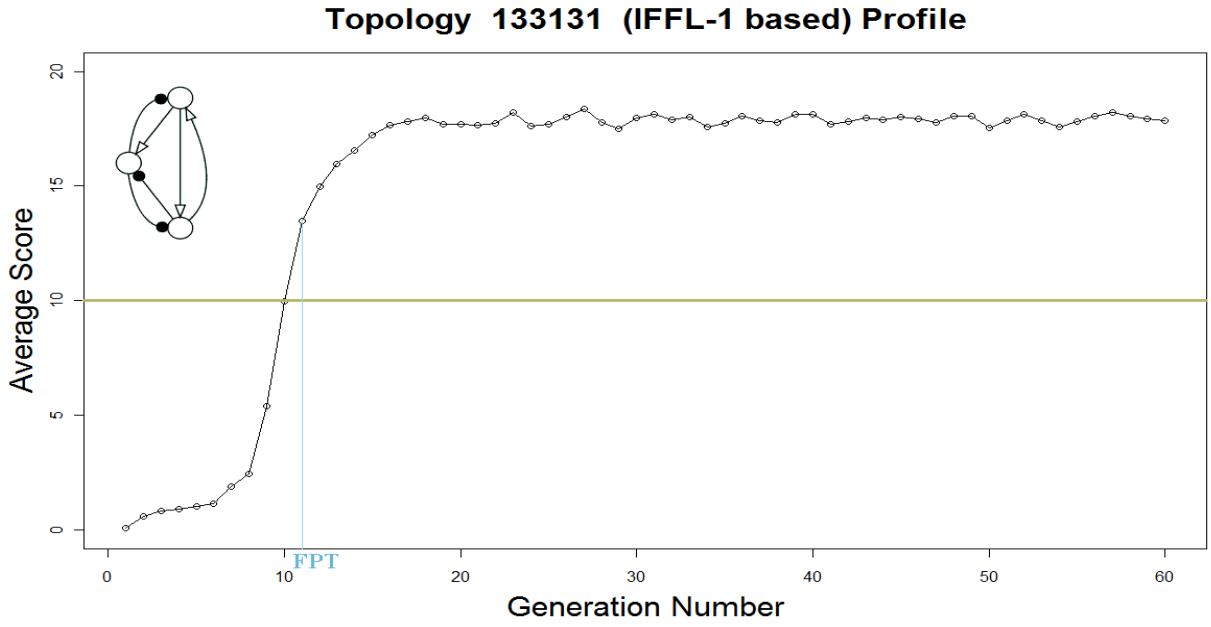
Examining IFFL-1 and IFFL-4 topologies

When these two categories of IFFLs were identified from the initial analysis as showing the greatest propensity to show near-perfect adaptive responses, the next logical step was to examine the topologies belonging to both these categories. Note that not all 27 members belonging to each category are present in this initial list. All 54 topologies are simulated on their own, i.e., only with micromutations to their parameters, and without macromutations (see the subsection "Evolutionary Algorithm" in Methods). Each simulation starts with randomly-chosen parameters that yield low scores. We average the scores of the parental parameter sets in a generation, and keep track of these average scores across the simulation. Four example IFFL-1 and IFFL-4 runs are shown in Figure 3.4.

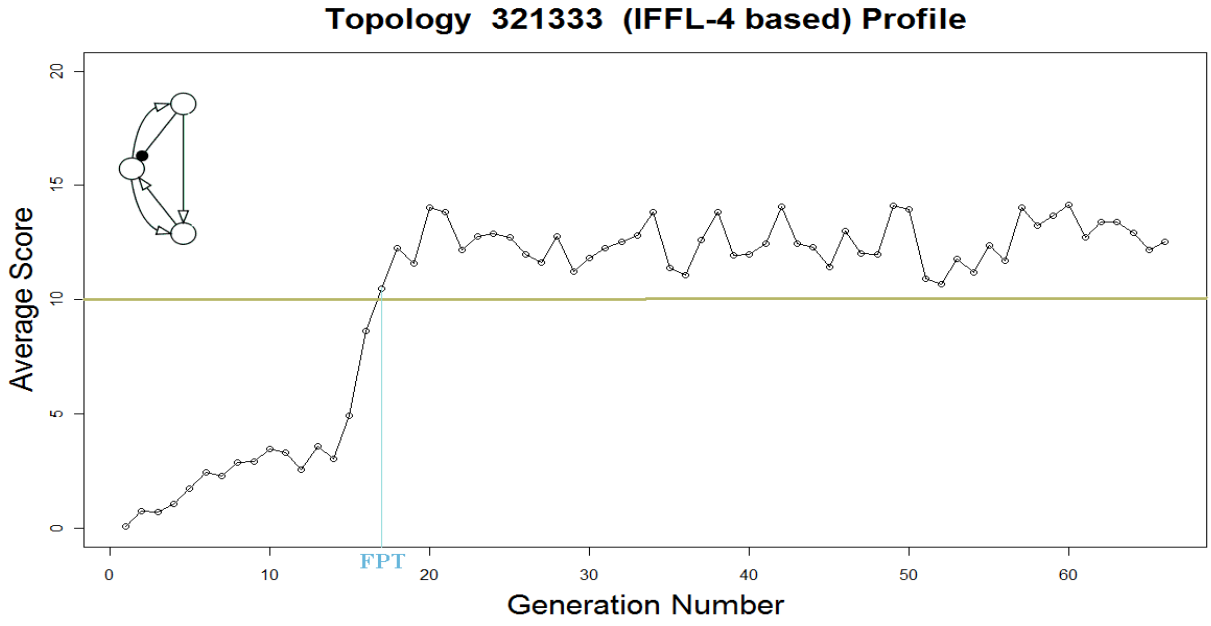
(a)



(b)


Topology 133131 (IFFL-1 based) Profile
Average Score
Generation Number
FPT
0
5
10
15
20
0
10
20
30
40
50
60

(c)


Topology 321333 (IFFL-4 based) Profile
Average Score
Generation Number
FPT
0
5
10
15
20
0
10
20
30
40
50
60

(d)

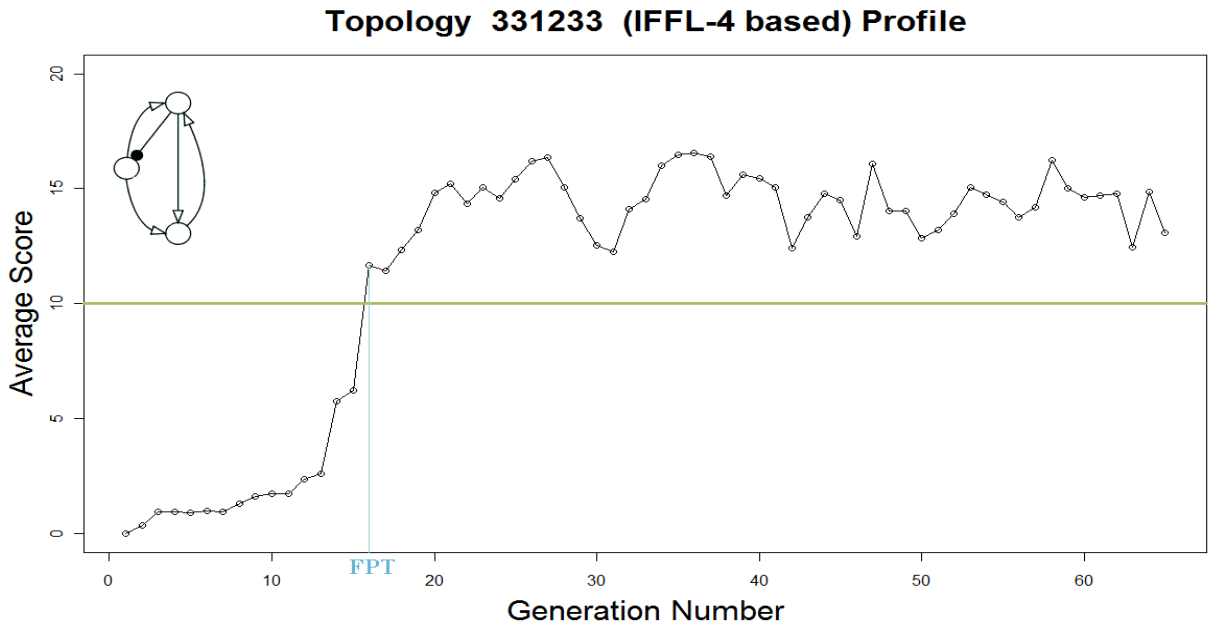


Figure 3.4. Examples of evolutionary simulation runs with N=20 and R=20. Each depicted topology stays in a low-scoring region, or 'desert', for the first few generations, but then quickly climbs onto a high-scoring region, or 'mesa', showing 'punctuated equilibrium'. (a) Topology 123231; (b) 133131; (c) 321333; (d) 331233.

For each IFFL topology that is simulated, we are looking for a First Passage Time (FPT), which is the number of generations it takes for the average score to cross 10. We stop the simulation 50 generations after its FPT, and also calculate the topology's overall average score using these last 50 generations. Table 3.2 shows the overall average scores and FPTs for all IFFL-1 and IFFL-4 topologies.

Table 3.2. The average scores and First Passage Times of all IFFL-1 and IFFL-4 topologies.

IFFL-1 Topologies			IFFL-4 Topologies		
Code	**< *Z* >**	**FPT**	**Code**	**< *Z* >**	**FPT**
113131	16.87	118	111133	12.48	15
113231	16.26	27	111233	11.45	70
113331	16.10	20	111333	14.04	25
123131	16.92	17	121133	11.96	31
123231	17.15	28	121233	12.02	100
123331	16.76	16	121333	13.47	26
133131	17.68	11	131133	12.57	33
133231	17.03	24	131233	13.64	23
133331	16.77	22	131333	13.55	42
213131	11.43	34	211133	11.41	18
213231	13.19	49	211233	11.89	26
213331	10.93	38	211333	13.25	12
223131	15.18	7	221133	13.67	66
223231	15.25	26	221233	11.96	30
223331	9.34	17	221333	12.15	17
233131	13.88	53	231133	11.27	53
233231	14.38	69	231233	12.19	32
233331	14.99	10	231333	12.61	80
313131	13.14	28	311133	13.64	24
313231	14.73	6	311233	12.73	31
313331	10.33	134	311333	11.81	32
323131	10.80	48	321133	12.94	10
323231	15.62	5	321233	12.90	31
323331	14.91	25	321333	12.53	17
333131	15.49	21	331133	12.55	22
333231	14.40	8	331233	14.36	16
333331	9.19	41	331333	14.34	41

Topologies in red: IFFL-1's (XX3X31); in green: IFFL-4's (XX1X33). < *Z* > represents average score. FPT represents First Passage Time. These micromutations-only simulations were done with *N*=20 and *R*=20.

All 27 IFFL-1's and 27 IFFL-4's find high-scoring regions, even from a poor start, showing that these two classes of topologies can exhibit near-perfect adaptation.

We performed a similar search with the 27 IFFL-2's and 27 IFFL-3's, but found that they have much lower average scores (see Appendix A).

These simulations are done with *N*=20 parents per generation and *R*=20 offspring per parent. If we use fewer than *R*=20 progeny, the evolutionary algorithm often does not find a high-scoring region (see Appendix B).

Following these simulations that only permit micromutations, we wanted to investigate what happens when we allow the initial IFFL-1 or IFFL-4 topologies to macromutate. We start off each run from the highest-scoring parameter set obtained from the corresponding micromutations-only run. We find that the IFFL-1 topologies drift among themselves, keeping their high scores, and likewise for the IFFL-4 topologies. An example run with macromutations is shown in Figure 3.5.

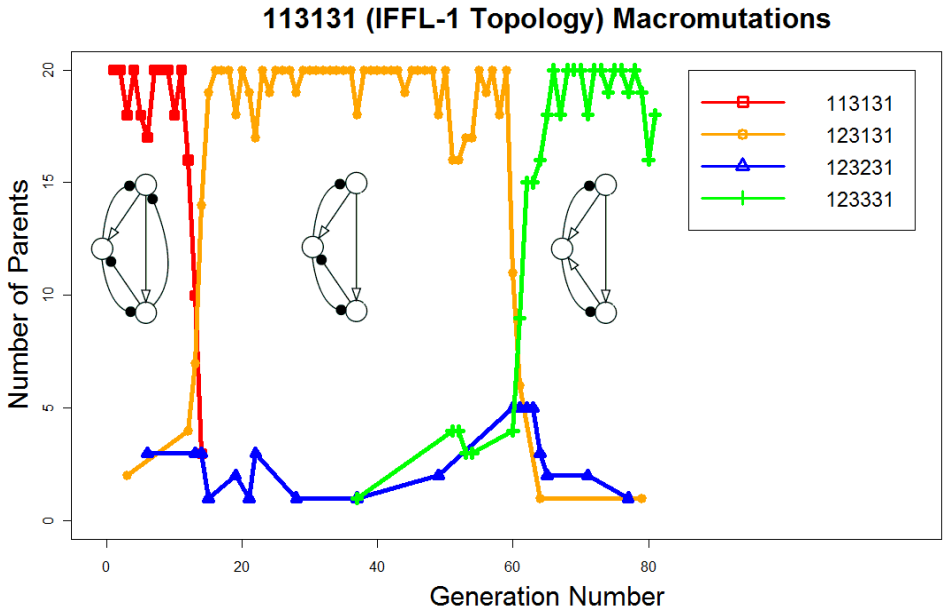


Figure 3.5. A sample run with macromutations, starting from an IFFL-1 topology. Each colored line depicts a unique topology. The total number of parents in each generation is 20. Each point shows how many of the 20 parents a topology occupies in a generation. The three dominant topologies remain based on the basic IFFL-1.

We now propose that these two specific categories, IFFL-1's and IFFL-4's, form two separate 'mesas' in topology space in the sense that (1) they both have very high average scores when examined on their own, and (2) they are evolutionarily stable, i.e., they stay within themselves when allowed to macromutate. In fact, as we will see in due course, these are the only two mesas that exist (as represented in Figure 3.6), i.e., these are the only two classes that satisfy the criteria of scoring well on their own and also being evolutionarily stable.

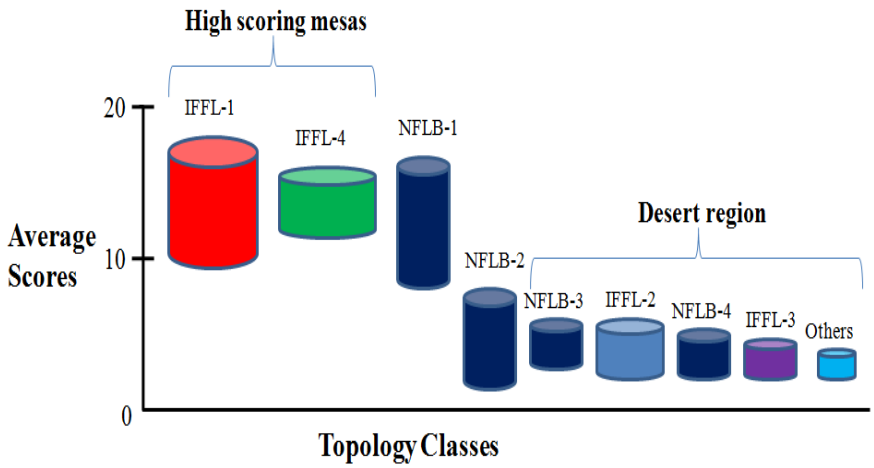


Figure 3.6. The topology scores landscape. The range of average scores for each topology class from micromutations-only runs. The widths of the IFFL-1 and IFFL-4 cylinders correspond to their robustness, as calculated by hyper-ellipsoid volumes. Other topologies include Classsic Negative Feedback Loops (see Appendix C).

Examining all four NFLB classes that are not coupled with IFFLs

After IFFLs, the next candidate category of topologies that merited investigation was the Negative Feedback Loops with Buffering, or NFLBs. We specify four separate classes of NFLBs, as shown in Figure 3.7.

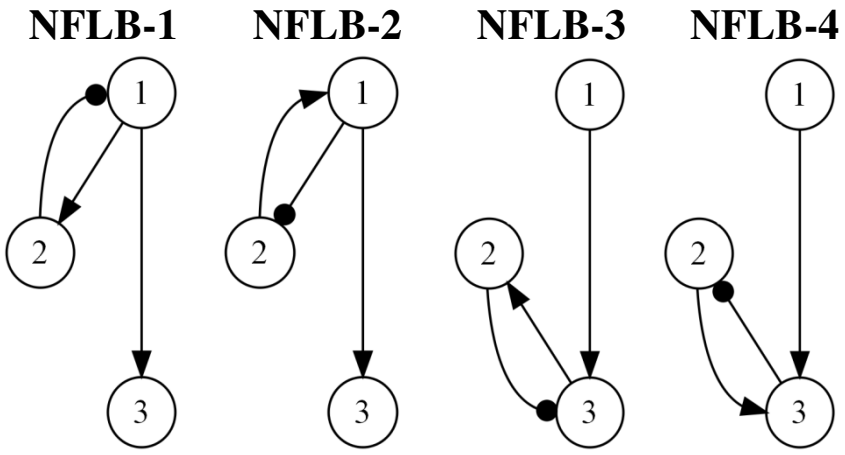


Figure 3.7. The four types of basic Negative Feedback Loops with Buffering (NFLBs). NFLB-1 and NFLB-2 are called the "upper NFLBs" as the negative feedback is between node1 and node 2. NFLB-3 and NFLB-4 are, by extension, called the "lower NFLBs".

A few of the topologies in the two IFFL-based mesas indeed contain NFLBs as well. In the next section, we shall carefully examine the relative contributions of IFFLs and NFLBs to these topologies, but first we need to investigate NFLBs that are not coupled to IFFLs. These NFLBs shall henceforth be referred to as "uncoupled NFLBs".

Table 3.3. The average scores of all topologies belonging to the four NFLB classes.

NFLB-1		NFLB-2		NFLB-3		NFLB-4	
Code	**< Z >**	**Code**	**< Z >**	**Code**	**< Z >**	**Code**	**< Z >**
133131	17.68	331233	14.36	123331	16.77	221133	13.67
123231	17.15	331333	14.34	133331	16.69	311133	13.64
133231	17.03	311133	13.64	113331	16.11	133133	13.20
123131	16.92	321133	12.94	233331	14.99	321133	12.94
113131	16.87	321233	12.90	323331	14.91	131133	12.57
123331	16.76	331133	12.55	213331	10.93	331133	12.55
133331	16.77	321333	12.53	313331	10.33	111133	12.48
113231	16.26	311333	11.81	223331	9.34	121133	11.96
113331	16.10	311233	10.40	333331	9.19	211133	11.41
133132	15.40	331332	8.20	122331	5.67	231133	11.27
133232	15.00	331132	7.58	132331	5.67	123133	10.46
133332	14.29	321132	7.39	112331	5.05	113133	8.63
133133	13.20	*321232*	*7.10*	321331	4.55	322133	3.25
133233	12.48	311132	6.23	121331	4.19	332133	3.22
123332	12.29	321231	5.09	131331	3.91	312133	2.90
123132	12.06	331232	4.78	212331	3.39	112133	2.74
133333	11.99	331231	4.76	322331	3.36	122133	2.68
123333	10.71	311232	4.74	312331	3.35	*222133*	*2.64*
113332	10.69	331131	4.73	*222331*	*3.13*	212133	2.64
123233	10.55	311231	4.55	332331	3.11	323133	2.63
123133	10.46	321331	4.55	232331	2.96	313133	2.60
113232	10.32	321332	4.42	311331	2.80	132133	2.59
123232	*9.80*	311332	4.30	331331	2.80	233133	2.43
113132	9.35	331331	2.80	231331	2.77	223133	2.27
113333	8.97	311331	2.79	221331	2.49	213133	2.24
113233	8.73	311131	1.64	211331	2.40	232133	2.12
113133	8.63	321131	1.33	111331	2.13	333133	1.81

The basic topologies are in *italics*. Red topologies are NFLB's coupled with IFFL-1's; green with IFFL-4's. High-scoring Coherent Feed Forward Loops coupled with NFLB-1's are underlined. < Z > represents average score.

Starting with the four basic NFLBs shown in Figure 3.7, we find that only basic NFLB-1 and NFLB-2, or the basic "upper NFLBs" score decently on their own, i.e., in runs with micromutations-only. This is in contrast to the basic "lower NFLBs", which score poorly on their own. Their scores are italicized in Table 3.3, which also shows the average scores of all extended NFLBs in all four classes that are obtained by adding links to the basic NFLBs.

Following the cue of the basic NFLBs, the extended uncoupled upper NFLBs also score better as a group than the extended uncoupled lower NFLBs (all uncoupled NFLBs are in black).

Surprisingly, we find that certain Coherent Feed Forward Loops (CFFLs) coupled with NFLB-1's, underlined in Table 3.3, score well too. This is not true of topologies having CFFLs coupled with NFLBs belonging to the other classes. We examine the interaction coefficients in these topologies, and from the high-scoring ($Z \geq 10$) subset, find that only the regulations which are part of the NFLB-1 are strong, i.e., they are close to their highest possible absolute value of 1. On the other hand, ω_{32}, the regulation that completes the CFFL, is termed weak as it is close to its lowest possible value, 0.1. (see Table 3.4). Clearly, it is the NFLB-1 that is contributing to the good scores in these cases, and not the CFFL. An example topology, 123233, that has the NFLB-1 and CFFL coupled together, is shown in Figure 3.8.

Table 3.4. The mean weights of all interaction coefficients from every NFLB-1 + CFFL topology's high-scoring sample.

NFLB-1 + CFFL Code	Interaction Coefficients					
	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
113133	-0.99	-0.11	0.95	-0.23	0.99	0.12
113233	-0.99	-0.11	0.95	0	0.99	0.12
113333	-0.99	-0.11	0.95	0.47	0.99	0.12
123133	-0.99	0	0.93	-0.2	0.99	0.12
123233	-0.98	0	0.93	0	0.99	0.12
123333	-0.99	0	0.92	0.28	0.99	0.12
133133	-0.96	0.38	0.86	-0.2	0.96	0.13
133233	-0.96	0.37	0.84	0	0.96	0.14
133333	-0.97	0.37	0.84	0.23	0.97	0.13

Indicated in blue are the three ω_{ij}'s that form the NFLB-1, and in purple is ω_{32}, the positive regulation on node 3 from node 2, that completes the CFFL. Note that ω_{32} is much weaker than the NFLB-1 ω_{ij}'s.

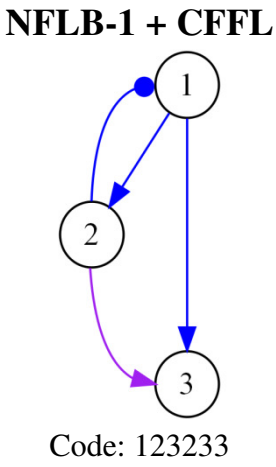


Figure 3.8. A sample NFLB-1 + CFFL topology, encoded 123233. Note that the three ω_{ij}'s that form the NFLB-1 are in blue, like in Table 3.4, and ω_{32}, which completes the CFFL, is shown in purple again also.

Examining the effects of adding IFFLs to the four NFLB classes

Now that we have established the scoring patterns among uncoupled NFLBs, we can compare their scores to those of NFLBs combined with IFFLs. Shown in Figure 3.9 are the fundamental, 4 links, topologies that result from those couplings. The rest of the NFLB + IFFL topologies are based on these fundamental topologies.

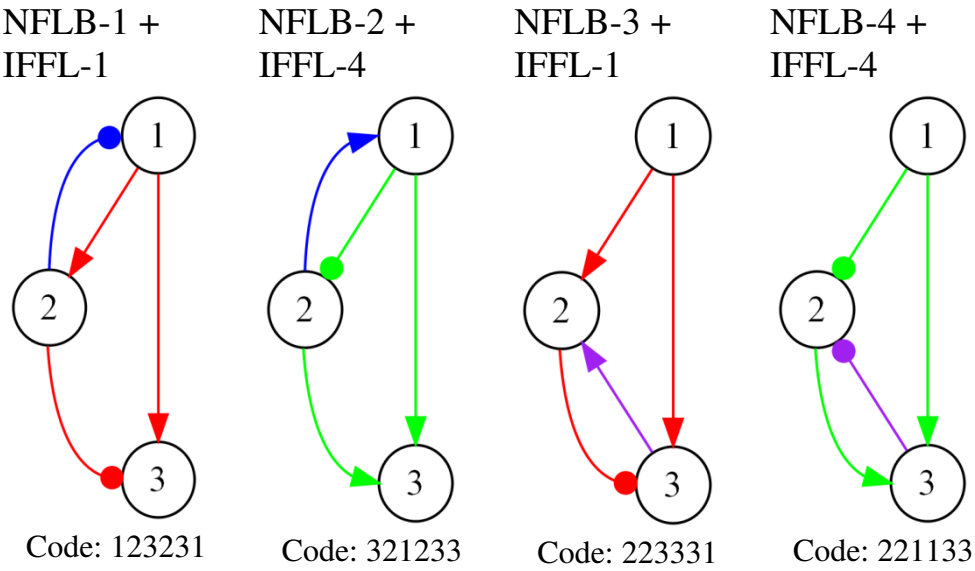


Figure 3.9. The fundamental topologies that result from coupling the four NFLBs and the two dominant IFFLs. IFFL-1 links are shown in red, and IFFL-4 links in green. NFLB-1 and NFLB-2, the upper NFLBs, are shown in blue, and NFLB-3 and NFLB-4, the lower NFLBs, are shown in purple.

Table 3.3 shows the averages scores of the extended NFLBs that are coupled with IFFLs (in color), and we see that in almost all cases, these topologies have higher average scores than the uncoupled NFLBs (in black). To quantify the overall increase in scores of a certain class of

NFLBs from the addition of IFFLs, we build tables that show the exact percentage change in score per NFLB-1 topology. For example, Table 3.5 shows the score of the coupled topology which corresponds to each of the NFLB-1 only topology. The change in score is averaged over all NFLB-1 only topologies to get an overall change percentage. These overall percentages are shown in Figure 3.10.

Table 3.5. The percentage changes in scores going from each of the uncoupled NFLB-1 topologies to the NFLB-1 topologies coupled with IFFL-1's.

NFLB-1 only		**NFLB-1 + IFFL-1**		**Percentage**
Code	**< Z >**	**Code**	**< Z >**	**Change**
113132	9.35	113131	16.87	80
113133	8.63	113131	16.87	95
113232	10.32	113231	16.26	58
113233	8.73	113231	16.26	86
113332	10.69	113331	16.10	51
113333	8.97	113331	16.10	79
123132	12.06	123131	16.92	40
123133	10.46	123131	16.92	62
123232	9.80	123231	17.15	75
123233	10.55	123231	17.15	63
123332	12.29	123331	16.76	36
123333	10.71	123331	16.76	56
133132	15.40	133131	17.68	15
133133	13.20	133131	17.68	34
133232	15.00	133231	17.03	14
133233	12.48	133231	17.03	36
133332	14.29	133331	16.77	17
133333	11.99	133331	16.77	40

The encoding for NFLB-1 + IFFL-1 topologies is 1X3X31. The last digit changes to 1 in each case. $< Z >$ represents average score.

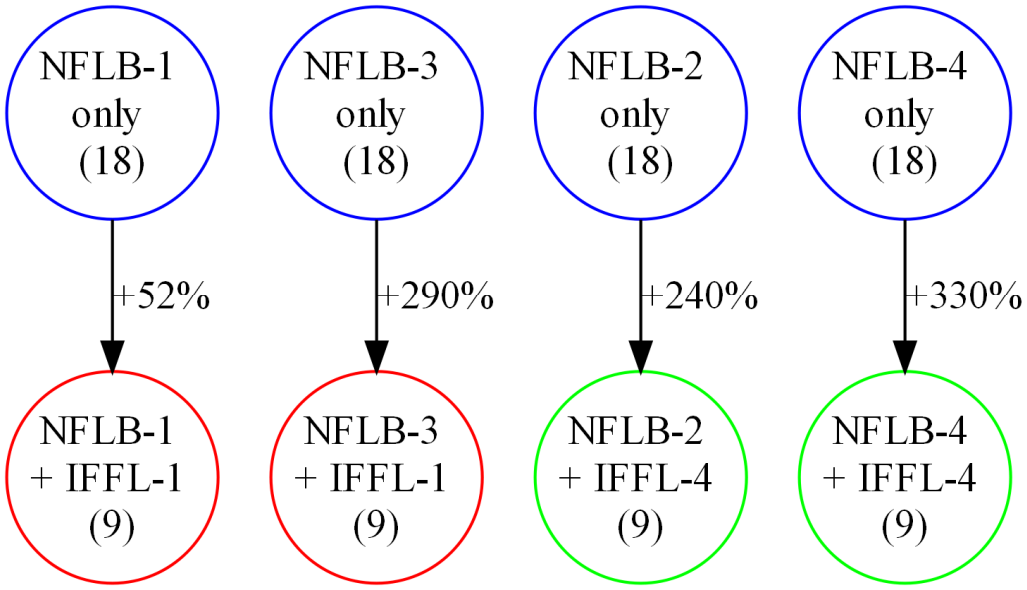


Figure 3.10. The overall percentage changes in NFLB scores when IFFLs are added. The numbers in brackets indicate the number of topologies having that particular combination.

This is clear evidence that adding IFFLs increases NFLB scores. We still need to establish, though, if this combination of IFFLs and NFLBs relies more on IFFLs or on NFLBs.

Examining the effects of adding the four NFLB classes to IFFL-1's and IFFL-4's

We have seen so far that IFFLs combined with NFLBs score really well. It is possible that it is the NFLBs that are contributing more significantly to this combination than the IFFLs. We check this by constructing tables similar to Table 3.5 for checking the effect of adding NFLBs to IFFLs. An example is shown in Table 3.6, in which we record the percentage change in scores when adding NFLB-1 to each uncoupled IFFL-1 topology. We record the average change in score across all uncoupled IFFL-1 topologies.

Table 3.6. The percentage changes in scores going from each of the uncoupled IFFL-1 topologies to the IFFL-1 topologies coupled with NFLB-1's.

IFFL-1 only		NFLB-1 + IFFL-1		Percentage
Code	**< Z >**	**Code**	**< Z >**	**Change**
213131	11.43	113131	16.87	48
313131	13.14	113131	16.87	28
213231	13.19	113231	16.26	23
313231	14.73	113231	16.26	10
223131	15.18	123131	16.92	11
323131	10.80	123131	16.92	57
223231	15.25	123231	17.15	12
323231	15.62	123231	17.15	10
233131	13.88	133131	17.68	27
333131	15.49	133131	17.68	14
233231	14.38	133231	17.03	18
333231	14.40	133231	17.03	18

The encoding for IFFL-1 + NFLB-1 topologies is 1X3X31. The first digit changes to 1 in each case. $< Z >$ represents average score.

We calculate the average changes to IFFL-1 scores through the addition of only NFLB-1 and, separately, of only NFLB-3. Since NFLB-1 and NFLB-3 topologies can both be added to IFFL-1's at the same time (see Figure 3.11 for an example), we also calculate (1) the change to IFFL-1 scores when both the NFLBs are added at the same time, (2) the change to IFFL-1 + NFLB-1 scores upon addition of NFLB-3's, and (3) the change to IFFL-1 + NFLB-3 scores upon addition of NFLB-1's. These changes are summarized in Figure 3.12. Similar calculations are made for the addition of NFLB-2 and/or NFLB-4 to IFFL-4's (see Figure 3.13).

We see in all these cases that adding either NFLB separately, or both the NFLBs together, increases the IFFL score by a much lower percentage than the cases in which the IFFLs are added to the NFLBs (see Figure 3.10 for comparison). Therefore, we can say that IFFLs contribute much more significantly to the coupled topologies' scores than the NFLBs.

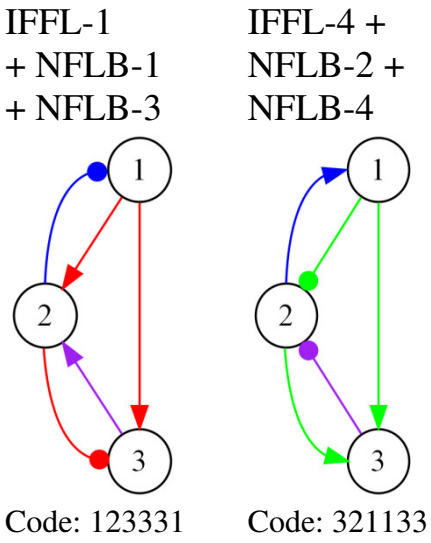


Figure 3.11. The topologies produced when the two high-scoring IFFLs combine with multiple NFLBs.

A critical observation from Figure 3.12 is that adding NFLB-1, the upper NFLB, always increases the IFFL-1 scores much more than adding NFLB-3, the lower NFLB. In fact, in these particular cases, adding the lower NFLB decreases the scores.

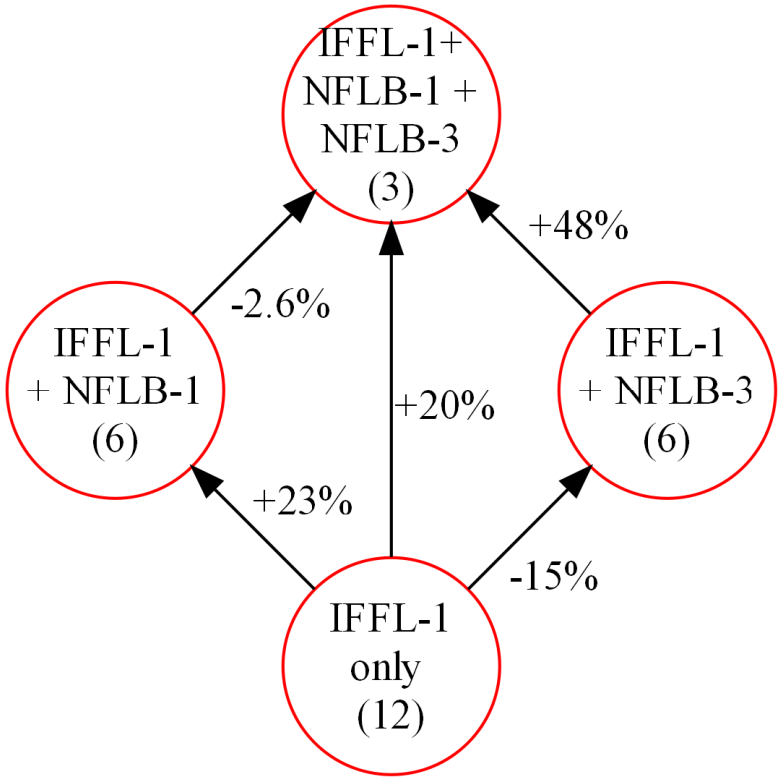


Figure 3.12. The overall percentage changes in scores when adding NFLB-1 and/or NFLB-3 to IFFL-1's. The numbers in brackets indicate the number of topologies having that particular combination.

We see this pattern repeated in Figure 3.13 when adding NFLB-2, the upper NFLB, always increases the IFFL-4 scores much more than adding NFLB-4, the lower NFLB.

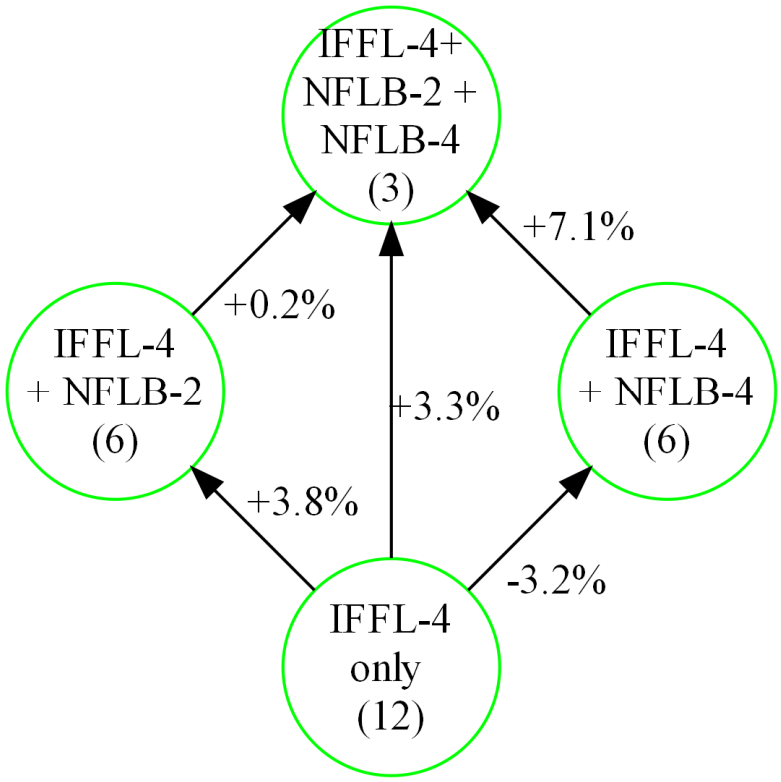


Figure 3.13. The overall percentage changes in scores when adding NFLB-2 and/or NFLB-4 to IFFL-4's. The numbers in brackets indicate the number of topologies having that particular combination.

These findings are consistent with our earlier observation that the uncoupled upper NFLBs tend to score better on their own than the uncoupled lower NFLBs. In the next section, we shall validate the contributions of upper NFLBs to high-scoring IFFLs, as compared to lower NFLBs.

Validating the evidence that upper NFLBs contribute much more significantly than lower NFLBs to high-scoring IFFL sets

An appropriate way to check that the upper NFLBs contribute more significantly to high-scoring IFFLs than lower NFLBs is to compare the means of all relevant ω_{ij}'s (the interaction coefficients). From each of the micromutations-only simulations presented earlier (see Table 3.2), we separate the high-scoring sample, i.e., all sets having a score ≥ 10, and calculate the means of all six ω_{ij}'s. Their values in IFFL-1's are recorded in Table 3.7.

Table 3.7. The means of the six interaction coefficients from all IFFL-1 topologies' high-scoring samples.

IFFL-1 Code	Interaction Coefficients					
	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
113131	-0.98	-0.14	0.89	-0.20	0.97	-0.96
113231	-1.00	-0.13	0.93	0.00	0.95	-0.98
113331	-0.97	-0.13	0.85	0.26	0.97	-0.88
123131	-0.99	0.00	0.88	-0.17	0.97	-0.95
123231	-0.99	0.00	0.89	0.00	0.97	-0.96
123331	-1.00	0.00	0.95	0.16	0.97	-0.99
133131	-0.92	0.52	0.82	-0.16	0.97	-0.63
133231	-0.85	0.62	0.54	0.00	0.97	-0.90
133331	-0.84	0.50	0.85	0.30	0.97	-0.64
213131	0.00	-0.13	0.93	-0.29	0.95	-0.97
213231	0.00	-0.16	0.92	0.00	0.96	-0.98
213331	0.00	-0.19	0.56	0.17	0.66	-0.96
223131	0.00	0.00	0.90	-0.34	0.96	-0.98
223231	0.00	0.00	0.89	0.00	0.96	-0.98
223331	0.00	0.00	0.52	0.79	0.97	-0.95
233131	0.00	0.27	0.90	-0.28	0.95	-0.97
233231	0.00	0.20	0.92	0.00	0.97	-0.98
233331	0.00	0.29	0.91	0.34	0.97	-0.98
313131	0.20	-0.15	0.91	-0.33	0.96	-0.97
313231	0.27	-0.23	0.92	0.00	0.97	-0.98
313331	0.15	-0.21	0.58	0.24	0.68	-0.96
323131	0.12	0.00	0.48	-0.20	0.64	-0.95
323231	0.33	0.00	0.92	0.00	0.97	-0.98
323331	0.27	0.00	0.91	0.19	0.97	-0.98
333131	0.18	0.38	0.90	-0.25	0.96	-0.98
333231	0.27	0.36	0.92	0.00	0.97	-0.98
333331	0.23	0.30	0.62	0.37	0.68	-0.96

The ω_{ij}'s shown in red represent the three links of the underlying IFFL-1. ω_{12}'s in blue show the NFLB-1 cases; ω_{23}'s in purple show the NFLB-3 cases.

The first observation that can be made from the table is that for the three ω_{ij}'s corresponding to the underlying IFFL-1 class, most of the means have strong weights (marked in red in Table 3.7). In other words, in the case of positive regulations, they are close to their maximum possible value of +1, and in the case of negative regulations, they are close to -1. The interaction strengths are also strong in the case of negative ω_{12}'s (marked in blue), which accounts for the negative feedback loop of NFLB-1's, the upper NFLB class (see Figure 3.11, left panel). In the case of positive ω_{23}'s (marked in purple), which accounts for the negative feedback loop of NFLB-3's, the lower NFLB class, we see that the interaction coefficients are weak (close to their minimum possible value of 0.1). This is clear evidence that it is the IFFL-1's along with the NFLB-1's, the upper NFLBs, that are driving the near-perfect adaptive responses. In contrast,

the NFLB-3's, the lower NFLBs, are playing a negligible role in such responses. A similar narrative emerges for IFFL-4's. All six ω_{ij}'s mean values in high-scoring IFFL-4's are recorded in Table 3.8.

Table 3.8. The means of the six interaction coefficients from all IFFL-4 topologies' high-scoring samples.

IFFL-4 Code	Interaction Coefficients					
	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
111133	-0.30	-0.22	-0.87	-0.21	0.95	0.96
111233	-0.39	-0.16	-0.86	0.00	0.95	0.96
111333	-0.43	-0.14	-0.86	0.25	0.94	0.96
121133	-0.29	0.00	-0.88	-0.26	0.95	0.97
121233	-0.27	0.00	-0.88	0.00	0.96	0.97
121333	-0.48	0.00	-0.83	0.22	0.93	0.96
131133	-0.16	0.30	-0.90	-0.24	0.95	0.97
131233	-0.31	0.47	-0.89	0.00	0.95	0.97
131333	-0.50	0.38	-0.89	0.14	0.94	0.97
211133	0.00	-0.21	-0.88	-0.25	0.96	0.97
211233	0.00	-0.16	-0.87	0.00	0.96	0.96
211333	0.00	-0.21	-0.88	0.25	0.95	0.96
221133	0.00	0.00	-0.90	-0.17	0.96	0.97
221233	0.00	0.00	-0.88	0.00	0.95	0.97
221333	0.00	0.00	-0.85	0.21	0.94	0.96
231133	0.00	0.30	-0.87	-0.15	0.95	0.97
231233	0.00	0.46	-0.82	0.00	0.94	0.96
231333	0.00	0.37	-0.87	0.20	0.95	0.96
311133	0.88	-0.13	-0.81	-0.18	0.95	0.66
311233	0.23	-0.15	-0.83	0.00	0.95	0.95
311333	0.15	-0.27	-0.84	0.14	0.95	0.96
321133	0.83	0.00	-0.77	-0.49	0.94	0.70
321233	0.90	0.00	-0.82	0.00	0.95	0.67
321333	0.85	0.00	-0.84	0.18	0.94	0.61
331133	0.61	0.34	-0.82	-0.31	0.95	0.71
331233	0.83	0.29	-0.83	0.00	0.94	0.57
331333	0.77	0.45	-0.82	0.15	0.94	0.59

The ω_{ij}'s shown in green represent the three links of the underlying IFFL-4. ω_{12}'s in blue show the NFLB-2 cases; ω_{23}'s in purple show the NFLB-4 cases.

Again, in addition to the ω_{ij}'s for the underlying IFFL-4 links (marked in red), the interaction strengths are strong in almost all cases of positive ω_{12}'s (marked in blue), which accounts for the negative feedback loop of NFLB-2's, the upper NFLB class (see Figure 3.11, right panel). In the case of negative ω_{23}'s (marked in purple), which accounts for the negative feedback loop of NFLB-4's, the lower NFLB class, we see that the interaction weights are weak (close to their minimum possible absolute value of 0.1). This shows that it is the IFFL-4's along with the

NFLB-2's, the upper NFLBs, that are driving the near-perfect adaptive responses, and not the NFLB-4's, the lower NFLBs.

Taken together, the last few results show that IFFLs combined with upper NFLBs score best among all classes, and that these regulations also tend to be most strongly present in the high-scoring sets.

Almost all NFLB topologies not coupled with IFFLs climb onto the IFFL-1 mesa

While some of the NFLB-1's that are not coupled with IFFL-1s score higher on the average than some IFFL-1's and IFFL-4's (see Table 3.3), we still cannot frame the uncoupled NFLB-1's into a high-scoring mesa of their own as we find that they are not evolutionarily stable. In fact, almost all of them evolve onto the IFFL-1 mesa (as shown in Table 3.9). Furthermore, almost all of the uncoupled NFLB-2's, NFLB-3's and NFLB-4's macromutate onto the IFFL-1 mesa as well (see Tables 3.10, 3.11 & 3.12).

As noted in the previous sections, NFLB-1, the upper NFLB, contributes more significantly to the IFFL-1 mesa than NFLB-3, the lower NFLB. Therefore, as a test of that property, we list all nine IFFL-1 + NFLB-1 topologies as columns in Table 3.9, and show that most of the uncoupled NFLB's indeed evolve into at least one of them. We take all high-scoring ($Z \geq 10$) parameter sets from a particular uncoupled NFLB's evolutionary run with macromutations, and record the percentage of those sets that belong to one of the nine IFFL-1 + NFLB-1 topologies. Shown also are the cases in which other classes of high-scoring topologies are found, or in which no high-scoring topologies are found at all.

The pre-dominance of IFFL-1 + NFLB-1 topologies can also be seen from Table 3.2 which shows the scores of all IFFL-1 topologies without macromutations. The topologies encoded 1X3X31 are the IFFL-1 + NFLB-1 topologies (see Figure 3.9, far left panel), and they score better than the rest of the IFFL-1 topologies, and even better than all IFFL-4 topologies.

The evolutionary superiority of the IFFL-1 + NFLB-1 topologies, together with the evidence presented in the previous section that the weights of the IFFL-1 and NFLB-1 regulations are strongest in high-scoring sets, gives us a clear picture of the specific regulatory patterns that tend to exhibit near-perfect adaptive responses.

Table 3.9. NFLB-1 topologies (1X3X3X) macromutate predominantly into high-scoring IFFL-1 + NFLB-1 topologies.

	IFFL-1 + NFLB topologies									
	113131	**113231**	**113331**	**123131**	**123231**	**123331**	**133131**	**133231**	**133331**	**Others**
NFLB-1 Topologies										
113132	0	0	0.001	0	**0.954**	0.024	0.001	0.009	0.004	0.006
113133	0	0.004	0.001	0	**0.957**	0.008	0	0.022	0.001	0.007
113232	0	0	0	0	0.006	**0.385**	0	**0.329**	**0.277**	0.004
113233	0	0.001	0.002	0.001	**0.651**	**0.331**	0	0.007	0	0.006
113332	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
113333	0	0	0	0.010	0.110	0.108	0.000	0.018	**0.754**	0
123132	0	0	0	0	0.055	0.010	0.001	0.015	**0.917**	0.001
123133	0.001	0.007	0	**0.459**	**0.214**	0.011	0.043	**0.161**	0.097	0.007
123232	0	0	0	0.005	**0.189**	0.008	0.001	**0.788**	0.006	0.005
123233	0	0	0	0.001	**0.321**	0.019	0.001	**0.652**	0.007	0
123332	0	0	0.000	0.001	**0.210**	**0.352**	0.001	0.004	**0.431**	0
123333	0	0	0	0.001	**0.976**	0.015	0	0.007	0.002	0
133132	0	0	0.002	0.005	0.020	**0.359**	0	0.106	**0.505**	0.002
133133	0.001	0	0	**0.471**	0.003	0	**0.476**	0.030	0.014	0.005
133232	0	0	0	0	0.003	0.004	0.002	**0.543**	**0.447**	0
133233	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
133332	0	0.000	0.000	0.001	**0.320**	**0.280**	0.000	**0.393**	0.004	0.001
133333	0	0	0	0.000	**0.278**	**0.346**	0	**0.206**	**0.169**	0

The basic NFLB-1 (123232) is marked in *italics*. Bold percentages indicate cases where an IFFL-1 + NFLB-1 topology occupies more than 15% of the high-scoring sample in a macromutation run. Percentages bordered in red show cases where a topology occupies more than 50% of the high-scoring sample. NA means no high-scoring topologies were found.

Table 3.10. NFLB-2 topologies (3X1X3X) macromutate predominantly into high-scoring IFFL-1 + NFLB-1 topologies.

	IFFL-1 + NFLB-1 Topologies									
	113131	**113231**	**113331**	**123131**	**123231**	**123331**	**133131**	**133231**	**133331**	**Others**
NFLB-2 Topologies										
311131	0	0.005	**0.317**	0	**0.620**	0.045	0	0.003	0	0.011
311132	0.001	0	0	**0.838**	0.015	0.002	0.117	0.005	0.004	0.016
311231	0.001	0	0	0.128	0.015	0.002	0	0.002	0	**0.853**[a]
311232	0	0	0	0	0.017	**0.326**	0	0.010	**0.646**	0.002
311331	0	0.001	0	0	0.006	0	0.003	**0.946**	0.011	0.033
311332	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
321131	0.001	0.006	0.004	0	**0.399**	**0.578**	0	0.001	0.004	0.008
321132	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
321231	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
321232	0	0	0	0	0	0	0	0	0	**1.000**[b]
321331	0	0.001	0.002	0.013	**0.293**	**0.682**	0	0.001	0.001	0.007
321332	0.005	0.003	0	0.016	**0.635**	**0.328**	0	0.003	0.003	0.006
331131	**0.527**	0.003	0.005	**0.448**	0.005	0.002	0.003	0.002	0	0.005
331132	0	0.004	**0.501**	0.001	0.116	**0.331**	0	0	0.037	0.009
331231	0.002	**0.220**	0.002	0	0.005	0.008	0.001	**0.230**	**0.520**	0.011
331232	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
331331	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
331332	0	0.001	0	0	**0.973**	0.004	0	0.018	0	0.005

[a]The other dominant topologies in this case are NFLB-1's not coupled with IFFL-1's, viz. 133232, 123232 (basic NFLB-1), and 123132.
[b]The other dominant topologies in this case are IFFL-4 + NFLB-2's, viz. 331333, 321233, and 321133.

The basic NFLB-2 (321232) is marked in *italics*. Bold: More than 15% of cases in an IFFL-1 + NFLB-1 topology; red border: more than 50%. NA means no high-scoring topologies were found.

Table 3.11. NFLB-3 topologies (XXX331) macromutate predominantly into high-scoring IFFL-1 + NFLB-1 topologies.

	IFFL-1 + NFLB-1 Topologies									
	113131	**113231**	**113331**	**123131**	**123231**	**123331**	**133131**	**133231**	**133331**	**Others**
NFLB-3 Topologies										
111331	0.001	0.004	**0.564**	0	**0.372**	0.047	0	0.007	0	0.005
112331	0	0	0.009	0	0.015	**0.956**	0.001	0.007	0.005	0.006
121331	0	0.001	0	0.005	**0.542**	**0.439**	0	0.006	0	0.007
122331	0	0	0.002	0	0.038	**0.949**	0	0	0.006	0.006
131331	0	0	0	0.001	0.031	**0.957**	0	0	0.006	0.005
132331	0	0.001	0.002	0.001	0.031	**0.691**	0.001	0.001	**0.247**	0.025
211331	0	0	0	0	0	0.003	0	**0.409**	**0.588**	0
212331	0	0.003	0	0.001	0.001	0.002	0.001	0	0	**0.992**[a]
221331	0	0.003	0	0.001	0.109	**0.876**	0	0.001	0.005	0.005
222331	0	0	0	0.002	0.004	**0.987**	0	0.001	0.005	0.001
231331	0	0.002	0	0	**0.957**	0.029	0	0.006	0.002	0.002
232331	0	0.003	0.001	0	0.012	0.008	0	0.021	**0.951**	0.003
312331	0	0	0	0	0.006	0.006	0.002	**0.541**	**0.444**	0.002
322331	0	0.001	0.003	0.007	0.028	**0.911**	0	0.042	0.005	0.002
332331	0.001	0.004	0	0	**0.543**	0.007	0	**0.429**	0.003	0.013

[a]The other dominant topologies in this case are IFFL-1's not coupled with NFLB-1's, viz. 213231, 323331, 323231, 333331, and 223231 (basic IFFL-1).

The basic NFLB-3 (222331) is marked in *italics*. Bold: More than 15% of cases in an IFFL-1 + NFLB-1 topology; red border: more than 50%. NA means no high-scoring topologies were found.

Table 3.12. NFLB-4 topologies (XXX133) macromutate predominantly into high-scoring IFFL-1 + NFLB-1 topologies.

	IFFL-1 + NFLB-1 Topologies									
	113131	**113231**	**113331**	**123131**	**123231**	**123331**	**133131**	**133231**	**133331**	**Others**
NFLB-4 Topologies										
112133	0	0	0	0	0	0.003	0	0.008	**0.989**	0
122133	0	0	0	0	0.015	0.001	0	**0.974**	0.009	0.001
132133	0	0	0.001	0.004	**0.878**	0.006	0	0.007	0.004	0.101
212133	0	0	0	0.003	0.096	0.004	0	0.001	0	**0.896**[a]
213133	0.005	0	0	**0.644**	**0.329**	0.005	0.011	0.003	0.002	0.003
223133	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**	**NA**
222133	0	0	0	0	0	0	0	0	0	**1.000**[b]
232133	0	0.003	0.051	0	**0.436**	**0.503**	0	0.001	0.006	0
233133	0	0	0	0.063	0	0.003	**0.905**	0.018	0.003	0.010
312133	0	0	0	0	0.006	0.014	0	**0.250**	**0.729**	0.001
313133	0	0	0	0	0.007	0.003	**0.254**	**0.707**	0.026	0.002
322133	0.001	0.004	0.001	0	**0.476**	0.001	0.001	**0.198**	**0.313**	0.005
323133	0	0	0	0	0	0.013	0	0.008	**0.975**	0.004
332133	0	0	0	0.001	0.013	0.119	0	**0.533**	**0.331**	0.003
333133	0	0	0.001	0.002	**0.606**	**0.377**	0	0.005	0.007	0.002

[a]The other dominant topologies in this case are NFLB-1's uncoupled with IFFL-1's, viz. 123332 and 123232 (basic NFLB-1).
[b]The other dominant topologies in this case are IFFL-4 + NFLB-2's, viz. 331133 and 321133.

The basic NFLB-4 (222133) is marked in *italics*. Bold: More than 15% of cases in an IFFL-1 + NFLB-1 topology; red border: more than 50%. NA means no high-scoring topologies were found.

Identifying the regions in parameter space in which high-scoring motifs score well

A major goal of our study is to find and characterize regions of parameter space where a particular motif generates high scores. We do this by analyzing the high-scoring parameter sets. For the present, we identify the high-scoring region for each parameter independently and ignore the cross-correlations between the interaction coefficients. (A study that takes cross-correlations into account shall be done in due course).

Extracting and validating parameter distributions from high-scoring samples

We observe from earlier simulations that when given random initial conditions, all IFFL-1 and IFFL-4 topologies start with very low scores in a 'desert' region, and quickly climb onto the high-scoring 'mesa', but only after some generations in which they wander around in the desert looking for the mesa. Using the high-scoring sample from these random start simulations, we want to derive 'conducive' parameter distributions for each topology, and see if (1) they lead to higher average scores and lower First Passage Times when used as starting conditions for simulations with only micromutations, and (2) they give enriched histograms when used to generate a large number of new scores *de novo*, as compared to scores generated *de novo* from random parameters.

In order to be able to extract the conducive parameter distributions, for every topology, we simply calculate the mean and standard deviation of each parameter from the high-scoring ($Z \geq 10$) sample obtained from the corresponding random-start simulations run with only micromutations. These means and standard deviations are shown in Appendix D.

For every IFFL-1 and IFFL-4 topology, we use these conducive parameter distributions to start a new set of simulations. Just like in the random-start case, a simulation stops 50 generations after finding its FPT. Again, a topology's overall average score is calculated using these last 50 generations.

From Table 3.13, we see that for most of the topologies (26 out of 27 for both IFFL-1 and IFFL-4 runs), the average scores are indeed higher for the conducive-start runs, as compared to the random-start runs. The mean percentage change in average scores of IFFL-1 topologies is 13.5 %; for IFFL-4 topologies, it is 15.6 %. Also, as expected, conducive-start runs have decreased First Passage Times in most of the topologies when compared to the random-start runs (21 out of 27 for IFFL-1, and 17 out of 27 for IFFL-4 runs). Collectively, these results show that in most of the cases, it is indeed possible to define a conducive parameter set that can serve as a reliable starting point to quickly find a higher-scoring sample set.

Table 3.13. Statistics for IFFL-1 and IFFL-4 topologies with conducive-start runs.

IFFL-1 Topologies					IFFL-4 Topologies				
Code	**< Z >**	**% Δ < Z >**	**FPT**	**Δ FPT**	**Code**	**< Z >**	**% Δ < Z >**	**FPT**	**Δ FPT**
113131	18.15	7.6	3	-115	111133	14.17	13.5	27	12
113231	17.10	5.2	9	-18	111233	14.49	26.6	6	-64
113331	17.35	7.8	11	-9	111333	13.95	-0.6	36	11
123131	16.97	0.3	10	-7	121133	14.79	23.7	12	-19
123231	17.21	0.3	15	-13	121233	14.42	20.0	26	-74
123331	17.46	4.2	5	-11	121333	15.41	14.4	4	-22
133131	17.79	0.6	9	-2	131133	14.72	17.1	23	-10
133231	17.87	4.9	9	-15	131233	14.88	9.1	24	1
133331	17.89	6.7	14	-8	131333	14.63	8.0	5	-37
213131	15.82	38.4	12	-22	211133	13.98	22.5	34	16
213231	16.91	28.2	3	-46	211233	14.71	23.7	25	-1
213331	12.08	10.5	5	-33	211333	14.37	8.5	16	4
223131	15.79	4.0	11	4	221133	14.33	4.8	18	-48
223231	16.77	10.0	5	-21	221233	14.63	22.3	12	-18
223331	16.47	76.3	31	14	221333	14.88	22.5	28	11
233131	16.33	17.7	5	-48	231133	13.97	24.0	8	-45
233231	17.04	18.5	4	-65	231233	13.91	14.1	27	-5
233331	16.31	8.8	19	9	231333	15.70	24.5	3	-77
313131	16.82	28.0	2	-26	311133	14.28	4.7	21	-3
313231	14.80	0.5	13	7	311233	14.02	10.1	18	-13
313331	11.91	15.3	19	-115	311333	13.69	15.9	29	-3
323131	12.07	11.8	50	2	321133	14.09	8.9	21	11
323231	15.82	1.3	24	19	321233	14.88	15.3	26	-5
323331	16.57	11.1	5	-20	321333	14.32	14.3	24	7
333131	14.83	-4.3	11	-10	331133	16.62	32.4	31	9
333231	17.42	21.0	3	-5	331233	15.91	10.8	21	5
333331	12.04	31.0	20	-21	331333	15.73	9.7	32	-9

< Z > represents average score. FPT stands for First Passage Time. Also shown is the percentage change in average scores from the random-start runs (Table 3.2), and Δ FPT = FPT (conducive-start) – FPT (random-start).

We can visualize what the parameter distributions look like with these conducive-start runs by constructing histograms. A sample collection of parameter histograms is shown in Figure 3.14.

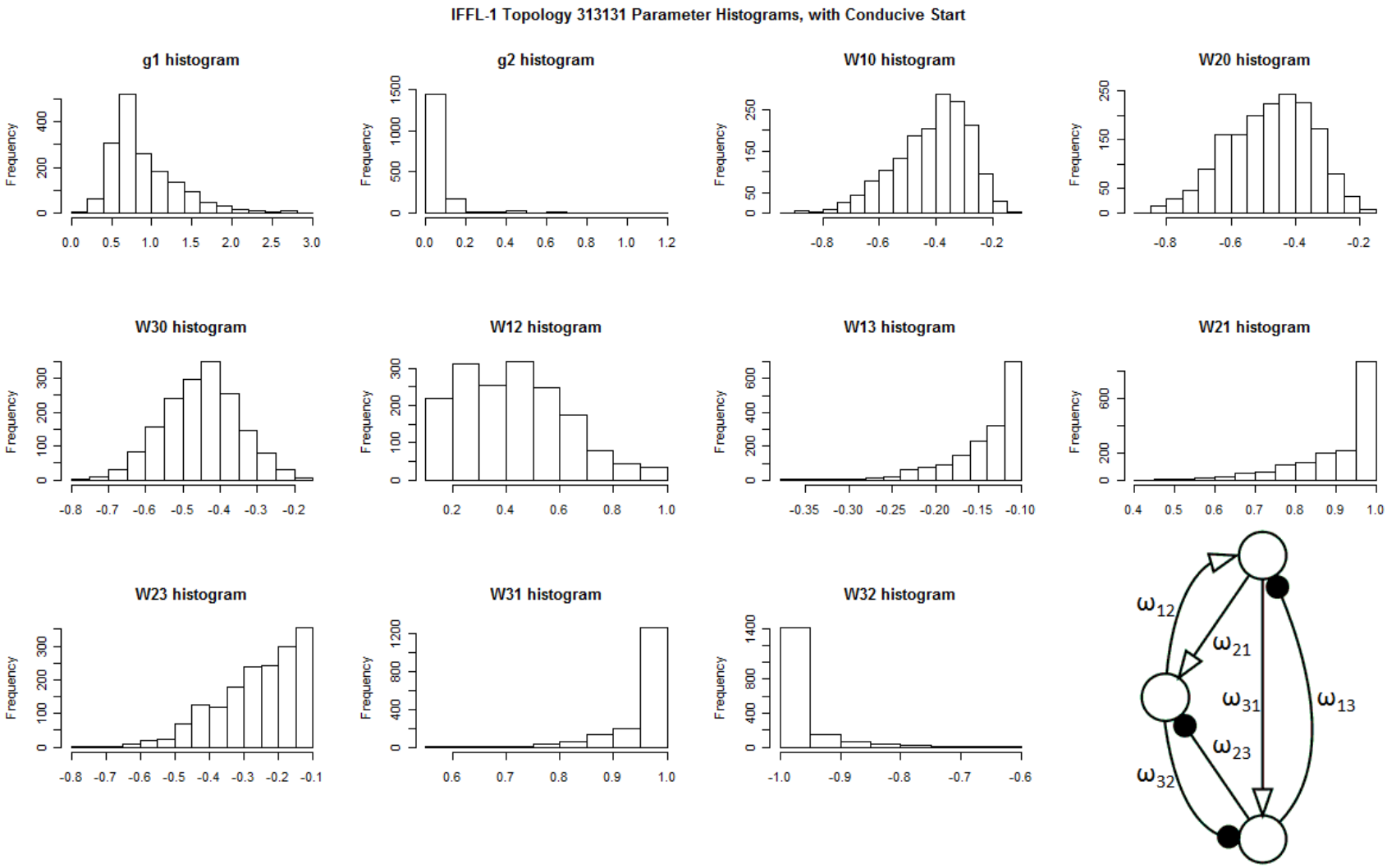


Figure 3.14. Parameter histograms from an IFFL-1 topology's conducive-start run. The topology's six interaction coefficients are labeled.

We are able to further validate the conducive parameter sets' propensity to yield high scores by constructing enrichment histograms. For every topology, we generate 1,000 scores *de novo* using parameters derived from the conducive means and standard deviations (while also making sure to enforce the parameter bounds, as described in "Topology Representation and Parameters" in Methods, and shown in Table 3.16), and compare those scores to 1,000 random scores also generated *de novo*. An example enrichment histogram is shown in Figure 3.15. The white bars represent the scores generated using the conducive parameters, and the red bars are for the scores obtained using parameters randomly selected from the uniform distributions within the pre-specified bounds.

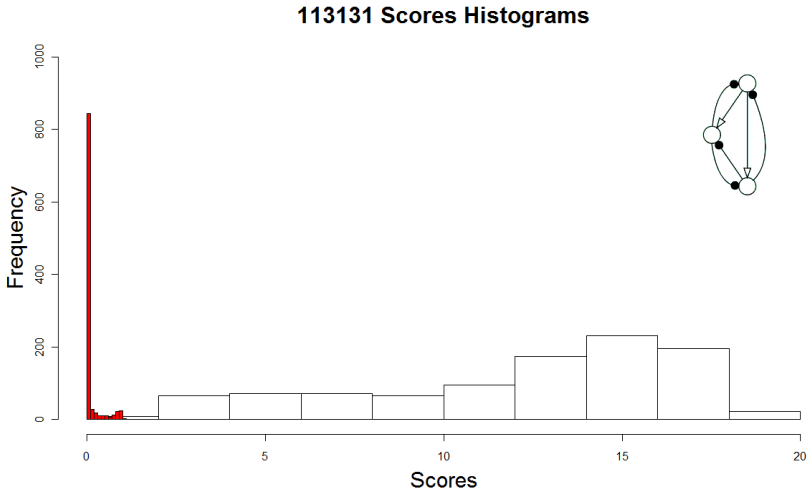


Figure 3.15. An example 'enrichment' histogram. White bars: distribution of the 1,000 scores generated using the conducive parameters. Red bars: distribution of the 1,000 scores generated using the random parameters.

As evident from Figure 3.15, the scores are greatly 'enriched' in the conducive parameters case when compared to the random parameters case. In fact, all scores generated using random parameters are in the 'desert' region. Table 3.14 shows, for both IFFL-1 and IFFL-4 topologies, the percentage of random scores above 0.5, and the percentage of conducive scores above 10. Clearly, most of the topologies find a significant number of good scores using the conducive parameters.

Overall, we see that even favorable topologies need to be in a certain region in parameter space to score high; if the parameters are chosen at random, the topologies score poorly.

Table 3.14. Enrichment statistics for IFFL-1 and IFFL-4 topologies.

IFFL-1 Topologies			IFFL-4 Topologies		
Code	% Random $Z > 0.5$	% Conducive $Z > 10$	Code	% Random $Z > 0.5$	% Conducive $Z > 10$
113131	8	72	111133	10	63
113231	9	100	111233	9	43
113331	10	59	111333	8	58
123131	9	89	121133	9	56
123231	8	81	121233	8	49
123331	10	100	121333	8	63
133131	7	53	131133	7	64
133231	8	55	131233	6	71
133331	8	45	131333	8	56
213131	8	78	211133	11	61
213231	9	48	211233	8	64
213331	10	24	211333	9	61
223131	9	54	221133	9	64
223231	8	72	221233	8	57
223331	7	19	221333	8	59
233131	7	46	231133	8	51
233231	7	50	231233	5	57
233331	8	81	231333	6	44
313131	9	47	311133	9	46
313231	8	77	311233	10	47
313331	9	17	311333	9	64
323131	9	28	321133	11	46
323231	9	85	321233	7	52
323331	8	76	321333	6	44
333131	6	73	331133	9	21
333231	7	73	331233	7	44
333331	6	72	331333	6	33

The percentages of scores above a certain cutoff are shown from 1,000 sets generated *de novo* using either random or conducive parameter distributions.

Characterizing the robustness of high-scoring regions in parameter space

Now that we have a handle on the regions of parameter space in which we can expect to see near-perfect adaptive responses, we want to get an idea of how large this region is, and to compare such measurements across the various topology classes. The approach we take is to calculate the volume of the hyper-ellipsoid containing the high-scoring sets within the *n*-dimensional parameter space. More robust topologies will have larger volumes.
For each topology, the high-scoring sets are within an 8 to 11-dimensional parameter space, depending on the number of links in the topology. The first 2 parameters are γ_1 and γ_2, the next 3

are the offsets ω_{10}, ω_{20}, and ω_{30}, and the final 3 to 6 are the interaction coefficients that are non-zero. γ_3 does not add an extra dimension to the parameter space since we keep its value at 1. See Table 3.16 for the fixed range assigned to each parameter.

Since we are interested in robustness to parameter variation, we use the technique of Principal Component Analysis (Jolliffe, 1986) to define orthogonal axes that are oriented in a way such that the first axis accounts for as much variability in the data as possible, the second axis for the next highest variability, and so on. The eigenvectors of the covariance matrix of the original sample specify the directions of these orthogonal axes, and the corresponding eigenvalues are proportional to the lengths of the semi-axes that are used to calculate the hyper-ellipsoid volumes.

The length of a semi-axis is the square root of its eigenvalue (Delforge et al, 1989), multiplied by a factor from Snedecor's *F* distribution that is set according to the confidence intervals over which the volume of the hyper-ellipsoid is to be calculated. We use a 99% confidence interval.

The calculation of a hyper-ellipsoid volume is done by taking the product of all semi-axes lengths, multiplied by a pi-related factor which depends on the number of dimensions involved. The formula for an *n*-dimensional hyper-ellipsoid volume is:

$$V_n = \left(\prod_{k=1}^{n} a_k \right) \frac{2\pi^{n/2}}{n\Gamma(n/2)} . \qquad \text{[Eq. 3.1]}$$

Here, a_k is the length of the *k*-th semi-axis; Γ is the gamma function*.

Since the semi-axes lengths are proportional to the eigenvalues, and the eigenvalues tend to be less than 1, multiplying all semi-axes lengths gives volumes which are much lower than 1.

The hyper-ellipsoid volumes for all IFFL-1's and IFFL-4's, the two topology classes which form high-scoring mesas, are recorded in Table 3.15. From the IFFL-1 and IFFL-4 random start simulations, the parameters sets that score more than 10 are used to calculate these volumes. At the bottom of Table 3.15 we record, for each dimension *n*, the geometric mean of the volumes over all topologies with an *n*-dimensional parameter space. Comparing the mean volumes of IFFL-1 and IFFL-4 topologies (at each value of *n*), we see that IFFL-4 topologies are more robust (i.e., have significantly larger volumes) than IFFL-1 topologies. This is depicted graphically in Figure 3.6 by the broader width of the IFFL-4 mesa compared to the IFFL-1 mesa.

* $\Gamma(x) = (x\text{-}1)!$ For odd values of *n*, $\Gamma(n/2) = \Gamma(0.5 + m)$, where $m = (n\text{-}1)/2$; $\Gamma(0.5 + m) = \Pi^{0.5}\ (2m)!\ /\ (4^m m!)$.

Table 3.15. The hyper-ellipsoid volumes from the high-scoring sets of every IFFL-1 and IFFL-4 topology.

IFFL-1 Topologies			IFFL-4 Topologies		
Code	***n***	**Volume**	**Code**	***n***	**Volume**
223231	8	0.006023	221233	8	0.045432
123231	9	1.81E-05	121233	9	0.03279
213231	9	0.007867	211233	9	0.015247
223131	9	0.007039	221133	9	0.016123
223331	9	4.05E-05	221333	9	0.009664
233231	9	0.03449	231233	9	0.031321
323231	9	0.003137	321233	9	0.024289
113231	10	7.50E-10	111233	10	0.000881
123131	10	3.29E-06	121133	10	0.013688
123331	10	5.07E-09	121333	10	0.002011
133231	10	0.000187	131233	10	0.041824
213131	10	0.001461	211133	10	0.021005
213331	10	0.000114	211333	10	0.002865
233131	10	0.025292	231133	10	0.009422
233331	10	0.02663	231333	10	0.14719
313231	10	0.00507	311233	10	0.009128
323131	10	1.37E-06	321133	10	0.097554
323331	10	0.004951	321333	10	0.009356
333231	10	0.003072	331233	10	0.029844
113131	11	2.09E-08	111133	11	0.003432
113331	11	0.000334	111333	11	0.000168
133131	11	0.001513	131133	11	0.019135
133331	11	0.056351	131333	11	0.001859
313131	11	0.00125	311133	11	0.00346
313331	11	5.06E-08	311333	11	0.002358
333131	11	0.006195	331133	11	0.33748
333331	11	0.001948	331333	11	0.033626

n	**<Volume>**	***n***	**<Volume>**
8	6.02E-03	8	4.54E-02
9	1.28E-03	9	1.97E-02
10	7.42E-05	10	1.29E-02
11	1.60E-04	11	6.10E-03

The IFFL-1 and IFFL-4 topologies are sorted according to number of dimensions *n*. Shown in the lower panel are the geometric means of volumes across *n*.

A related analysis that collects statistics on FPT and hyper-ellipsoid volumes over 100 iterations for each topology is presented in Appendix E.

Visualizing high-scoring regions in Principal Component space

We can visualize a topology's high-scoring parameter sets using the first three principal components only. Sample plots are shown in Figure 3.16. As evident, only one cluster, or distinct region, is found in parameter space. The sharp edges on the scores are due to bounds we put on parameter values (see Table 3.16).

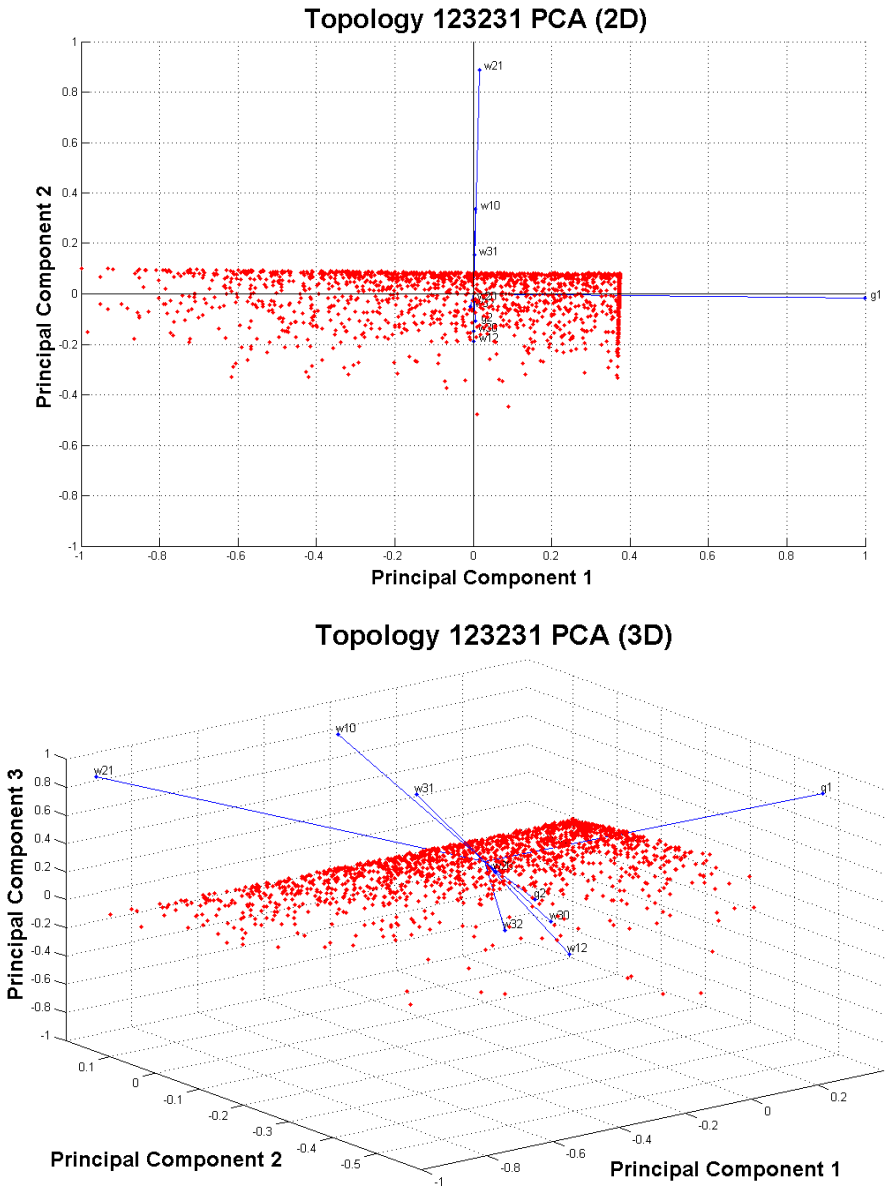


Figure 3.16. Sample Principal Component Analysis plot. The high-scoring parameter sets belonging to topology 123231 (IFFL-1 + NFLB-1) are plotted in the Principal Components space both in 2D and 3D. Each red dot represents the 'score' assigned to a particular parameter set. The blue arrows show the 'loadings' – the contributions of the annotated parameters to the Principal Components.

3.3 Discussion

It is evident from our analyses that both Incoherent Feed Forward Loops (IFFLs) and Negative Feedback Loops with Buffering (NFLBs) favored near-perfect adaptive responses. Within these categories, we narrowed down the classes which performed better using our scoring metric. Specifically, IFFL-1 and IFFL-4 (see Figure 3.3) along with the upper NFLBs, NFLB-1 and NFLB-2 (see Figure 3.7) scored higher than the other classes, as pictured in Figure 3.6.

To give a fair chance to a wide variety of topology categories, we started off our exploration of the topology space with 40 unique topologies that were allowed to macromutate into other topologies. This procedure covered more than 500 topologies, and identified IFFL-1's and IFFL-4's as the best classes showing the desired response (see Table 3.1). Next, we explored all 54 member topologies of these two classes in greater detail by running our evolutionary algorithm with the criterion that only the values of the parameters changed, but not the topology itself. We find that they all yielded high average scores (see Table 3.2), confirming the results of the initial analysis in which these two classes were filtered as the best scoring. Moreover, even upon allowing these topologies to change to other topologies, we found that they remained within their respective categories (as exemplified in Figure 3.5). This evolutionary stability, along with their propensity to score high, let us say that IFFL-1's and IFFL-4's formed two distinct 'mesas' in topology space. We also found that NFLB-1's and NFLB-2's had high average scores when examined on their own (see Table 3.3), but these topologies were not evolutionarily stable. Therefore, we called them 'shoulders' instead of 'mesas'. The rest of the landscape was composed of topologies in the 'desert' (see Figure 3.6).

We noticed that the IFFL-1's and IFFL-4's were coupled with NFLBs as well. We therefore first tested NFLBs separately to see how they did on their own, i.e., without macromutations. Looking at NFLBs that were not coupled with IFFLs, or uncoupled NFLBs, we saw that it was the upper NFLBs, NFLB-1's and NFLB-2's, that scored better than their lower NFLB counterparts, NFLB-3's and NFLB-4's.

Keeping this evidence in mind, we next examined the relative contributions of the IFFLs and the NFLBs to the topologies which had both IFFLs and NFLBs combined. Going from the uncoupled NFLBs to these combinations, or adding IFFLs (see Figure 3.10), produced much higher increases in scores than reaching these combinations from uncoupled IFFLs (see Figure 3.12 and Figure 3.13). Therefore, we could say that it was the IFFLs which played a more significant role than the NFLBs in the coupled topologies. A related finding was that adding the lower NFLBs in fact decreased the score of uncoupled IFFLs.

To validate the idea that it was the upper NFLBs that contributed much more significantly than the lower NFLBs to the high-scoring IFFL sets, we examined the interaction coefficients that corresponded to each of these classes. The IFFL and upper NFLB interaction coefficients had strong weights, whereas the lower NFLB interaction coefficients were mostly weak (see Table 3.7 and Table 3.8). Furthermore, the results from macromutating the uncoupled topologies from all four NFLB classes allowed us to establish that IFFLs are in a sense "superior" to NFLBs in showing near-perfect adaptive responses. Nearly all of these macromutations climbed to the highest peaks of the IFFL-1 mesa, which were composed of the 9 combinations of the IFFL-1

topologies with NFLB-1, the upper NFLB (see Tables 3.9 – 3.12). These 9 topologies also scored better than the rest of the IFFL-1 topologies, and also all of the IFFL-4 topologies, when simulated without macromutations (see Table 3.2). The IFFL-1 + NFLB-1 combination may therefore be the regulatory network most conducive to showing near-perfect adaptation.

The binning of the IFFL-1 and IFFL-4 classes into mesas came with the observation that even these topologies needed to be in a certain region of parameter space to be able to climb from the desert region on to the mesa. Whenever these topologies started with 'random' initial conditions, they scored poorly and it took many generations of the evolutionary algorithm for these topologies to stumble into their high-scoring region in parameter space. We were able to extract the parameter values belonging to these regions from the high-scoring samples obtained from the random-start runs. Using these new 'conducive' initial conditions, we showed that it was indeed possible to climb to the top of the mesa quicker, for most of the topologies, while getting higher average scores (see Table 3.13). Scores generated *de novo* using both random and conducive parameters confirmed that the same topology could do much better in the latter case (see Figure 3.15 and Table 3.14). Lastly, using Principal Components Analysis (PCA), we found evidence that the high-scoring parameter space for IFFL-4's was more robust than for IFFL-1's (see Table 3.15).

The summary of results presented so far confirms that we have extended the work published in *Cell* (Ma et al, 2009) in multiple directions. Not only have we quantified in more precise terms the dominance of the IFFLs over the NFLBs, we have also shown which particular classes belonging to these two categories scored better. In fact, one of these classes, IFFL-4, has been incorrectly identified by Ma et al as not showing adaptation at all. Specifically, in Figure 2B of their paper, they provide evidence in the right-hand side motif of the lower panel that adaptation cannot be achieved when both node A (node 1 in our setup) and node B (node 2) exert the same regulation on node C (node 3). In other words, they contend that the regulations on node 3 from node 2 and node 1 must have the opposite signs, as supported by the adaptation classification of the minimal network shown in the right-hand side motif of the upper panel in the same figure, which is IFFL-1 in our analyses. We showed clearly that both of these minimal motifs could exhibit adaptation to a very high degree, and that they both formed the basis of two separate mesas in topology space.

An important mathematical difference between our and Ma et al's models needs to be highlighted. They are able to linearize the underlying Michaelis-Menten ODEs as they only consider a small change in the input to the system (from 0.5 to 0.6). From their linearized equations, the condition for NFLBs to show perfect adaptation, or for adaptation error to be zero, is satisfied in their mathematical setup only when $J_{22}^{0} \cong 0$. J_{22}^{0} represents the rate of change of the value of Node 2 with respect to itself, and is a diagonal element of the Jacobian matrix of the system at steady state.

$J_{22}^{0} \cong 0$ is satisfied when the enzymes acting on Node 2 are in saturation, or when the ODE for Node 2 has Michaelis constants much smaller than substrate concentrations. Under this condition, Node 2 implements integral feedback control in NFLBs by integrating the difference between the activity of response Node 3 and Node 3's signal-independent steady state value. See equations 2-4 in Ma et al (Ma et al, 2009) for the mathematical details.

In our model however, there is no requirement for $J_{22}^0 \cong 0$ to achieve adaptation in NFLBs, or for integral feedback control. Upon differentiating Eq. 3.2, we find $J_{22}^0 = -\gamma_2$, which is small but not zero. This difference may be due to the fact that we are looking for near-perfect, not perfect, adaptation. We also introduce a much higher change in the input signal level (from 0 to 1).

In summary, we were able to extend Ma et al's work by narrowing down the specific combination of IFFLs and NFLBs that favored near-perfect adaptation the most. We were able to further validate our results by directly testing the evolutionary stability of the concerned topology classes, a procedure which was beyond the scope of the methodology used by the other investigators. Finally, we were able to give a more concrete picture of the region in parameter space where the high-scoring topologies showed the desired response.

Examples of IFFLs and NFLBs from Biological Networks

IFFLs and NFLBs are common motifs in large-scale regulatory networks. In the transcription network of *Escherichia coli*, for example, of the 138 known feed-forward loops, 25-30% are known to be IFFL-1, whereas 5% or less are known to be the other three types of IFFLs (Mangan et al, 2006). Similarly, in yeast, between 35 and 40% of the 56 known FFLs are IFFL-1. The investigators go on to show how a typical IFFL-1 in *E. coli* helps to accelerate the response time of galactose utilization genes upon glucose starvation. After a rapid increase in synthesis rate of these genes via the direct arm of the IFFL-1, there is a net decrease in their levels as the inactivating indirect arm kicks in. This case, however, is not illustrative of near-perfect adaptation because the system ends up at a much higher steady state.
In *Dictyostelium discoideum* and neutrophils, the proposed mechanism for perfect adaptation in chemotaxis, in response to a chemoattractant gradient, is based on an IFFL-1 (Levchenko & Iglesias, 2002). The G-protein-associated chemokine receptors convey opposite signals to PIP3, which is a phosphoinositide phosphate that is an important upstream node in a cascade whose downstream nodes are the signaling components. The G-protein upregulates PI3-kinase, a PIP3 activator, while at the same time also upregulating PTEN, a phosphatase that inactivates PIP3. It has to be noted though that as per the investigators, the role of PTEN has not been experimentally verified yet. Still, an IFFL motif remains one of the few plausible explanations for the perfect adaptation observed in the levels of PIP3.

Yet another major role for IFFLs has been identified in the events of the cell cycle in budding yeast (Csikasz-Nagy et al, 2009). Both mitotic exit and DNA replication require transient activation of the appropriate 'executor' proteins, which are found to be both directly and indirectly regulated by cyclin-dependent kinases (Cdk1). For example, an IFFL-3 plays a major role in initiating mitotic exit via opposite regulation of the 'executor' protein Dfb2. Cdk1 inactivates Dbf2 directly (by phosphorylation) and regulates it indirectly by activating the transcription factor (Fkh2) that upregulates the production of Dbf2.

IFFL-1 also plays a major part in maintaining phenotypic robustness during animal development in a process called canalization (Hornstein & Shomron, 2006). It is microRNAs that play the role of the inactivating intermediary node in this case. Transcription factors activate both the target gene and a miRNA that down-regulates translation from the target gene. An example from

cell cycle regulation is upregulation of gene E2F1 by the transcription factor c-myc, which also upregulates miR-17-5p and miR-20a, microRNAs that reduce translational efficiency of the mRNA for E2F1 (O'Donnell et al, 2005).

An IFFL-1 has been studied as a stand-alone motif in a synthetic biology context as well (Basu et al, 2004). The pulse-like response shown to an inducer (LuxR) and repressor (C1 of phage lambda) working simultaneously is also akin to the near-perfect adaptive response. The investigators go on to show that the amplitude and timing of the pulse differs according to the concentration of the signal, but that the GFP expression level finishes at around the same (slightly higher) steady-state for each signal concentration.

The major example of NFLBs in adaptive motifs is in the network that controls chemotaxis of *E. coli* in response to a chemoattractant gradient (Alon et al, 1999). The output of the system is measured by activity of the enzyme CheY (Node 3) which controls the tumbling frequency of the bacteria. CheY is directly activated by CheA, which is in complex with the receptor (Node 1). Upon binding to ligand, the autophosphorylation rate of CheA decreases, which causes a transient decrease in CheY. CheY can now only come back up when CheA comes back, which happens when the CheA/receptor complex becomes methylated. In essence, the decreased activity of the receptor due to addition of ligand is offset by the methylation process, which occurs via a negative feedback loop from Node 1 to the 'buffering' Node 2 which represents the enzyme CheB. Decreased production of CheA leads to reduced phosphorylation of CheB, which in turn leads to decrease demethylation of CheA. In our context, this motif is NFLB-2, an upper NFLB.

These examples illustrate the important role of IFFLs and NFLBs in carrying out physiological functions that are akin to (or actually are cases of) near-perfect adaptation.

In addition to near-perfect adaptation, our evolutionary approach can also be applied to finding motifs and parameter sets displaying other behaviors, such as cock-and-fire, bistability, oscillations, and even chaos. Once the appropriate scoring function has been designed in each case, the rest of the approach can be the same as the one taken in this study. By confirming that our approach works for the case of near-perfect adaptation, we have taken the first step in creating a topological structure-function map that would be a very useful tool for systems biologists.

3.4 Methods

Modeling Regulatory Networks with Wilson-Cowan Equations

We model our 3-node regulatory networks using equations first proposed by Wilson & Cowan (Wilson & Cowan, 1972) in the context of modeling excitatory and inhibitory interactions in neural networks. Our ODEs are of the following form:

$$\frac{dC_i}{dt} = \gamma_i[F(W_i) - C_i],$$
$$W_i = S(t)\delta_{i1} + \omega_{i0} + \sum_{j \neq i} \omega_{ij} C_j, \; i = 1,..., N. \qquad \text{[Eq. 3.2]}$$

Here, C_i is the i-th continuous variable, representing the concentration (or activity) of species i in the system, γ_i is the timescale on which the variable changes value, and $F(W_i)$ is the production rate of C_i. The function W_i represents the net regulation on variable i by all other variables j. Each interaction coefficient ω_{ij} represents the weight of the regulation on variable i by variable j (ω_{ij} is positive for an activation, negative for an inhibition, and zero if the interaction is absent). The offset ω_{i0} determines W_i in the absence of any regulation on node i from the other nodes j in the network. We do not allow for self-regulations in our setup. Also, for node 1, a signal term $S(t)$ is added to the net regulation. Details on how $S(t)$ varies with time are described below in the section "Generating a single score".

We desire that the function $F(W_i)$ be sigmoidal in shape (see Figure 3.17). A convenient choice for $F(W_i)$ is the hyperbolic tangent-based function or soft Heaviside function

$$F_\sigma(W_i) = \frac{1}{2}\left[1 + \tanh\left(\frac{\sigma W_i}{2}\right)\right] = \frac{1}{1+e^{-\sigma W_i}} \qquad \text{[Eq. 3.3]}$$

where, for $\sigma > 0$, $F_\sigma(W_i) \cong 0$ when $W_i << -1/\sigma$, and $F_\sigma(W_i) \cong 1$ when $W_i >> 1/\sigma$. Hence, the steepness of the sigmoidal curve is controlled by the parameter σ. If $W_i = 0$, i.e., the net regulation on variable i is zero, then the function simply evaluates to 0.5, signifying that variable i is neither activated nor inhibited. The form of this hyperbolic tangent-based function also ensures that, if $0 \le C_i(0) \le 1$, then $0 \le C_i(t) \le 1$ for all $t > 0$. In particular, the steady state value, $C_i^{ss} = F(W_i^{ss})$, must lie between 0 and 1.

In all our calculations, we choose $\sigma = 10$, so we suppress the σ subscript and simply use the notation $F(W_i)$ for the function in Equation 3.3.

Reinitz and colleagues (Mjolsness et al, 1991) have modeled gene regulatory networks using the same mathematical formulations, albeit with a Hill function instead of a hyperbolic tangent-based function.

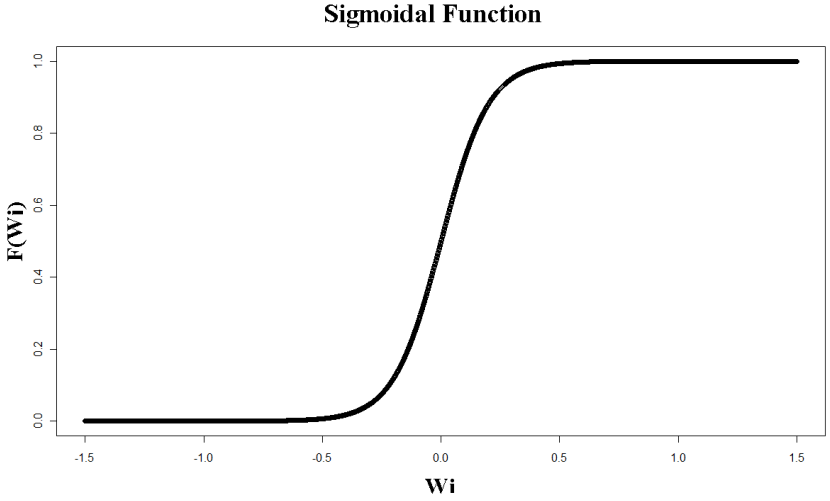


Figure 3.17. The sigmoidal function $F_\sigma(W_i) = [1 + \tanh(\sigma W_i/2)] / 2$, with $\sigma = 10$.

Topology Representation and Parameters

A network topology is collectively represented by the signs (+, -, 0) of the six interaction coefficients (ω_{12}, ω_{13}, ω_{21}, ω_{23}, ω_{31}, ω_{32}). Hence, we can encode a topology by six digits: $d_1d_2d_3d_4d_5d_6$ where d_k = 1, 2 or 3 for each k. Specifically, d_k = 1 for an inactivation ($\omega_{ij} < 0$), 3 for an activation ($\omega_{ij} > 0$), and 2 for an absent regulation ($\omega_{ij} = 0$). The 1st digit refers to ω_{12}; the 2nd digit to ω_{13}; the 3rd digit to ω_{21}; the 4th digit to ω_{23}; the 5th digit to ω_{31}; and finally, the 6th digit to ω_{32}. For example, the code 223231 represents the Type 1 Incoherent Feed-Forward Loop (IFFL-1) topology, depicted in Figure 3.3's far left panel. The third digit from left, 3, encodes the activation of node 2 by 1; the fifth digit, 3, the activation of node 3 by 1; and the sixth digit, 1, the inactivation of node 3 by node 2.

Every 3-node motif can be described by Equation 3.1 with 6 interaction coefficients (ω_{ij}'s), 3 offsets (ω_{i0}'s), and 3 timescale parameters (γ_i's). We assign a finite, continuous, range to each of these parameters. We keep the interaction coefficients between 0.1 and 1 for positive regulations, between -1 and -0.1 for negative regulations, and, obviously, to 0 for absent regulations. The offsets can assume any value between -2 and 2. They determine whether a node turns on or off in the absence of any external regulation on it. The timescale parameters, which determine how slow or fast the level of a node changes, can be any value between 0.1 and 3, except for γ_3, which we fix at 1 so that the rate of change of node 3 sets the characteristic timescale of the model. This is appropriate as node 3 is the response-measuring node of the topology. Table 3.16 summarizes the role and range of each parameter used in our models.

Table 3.16. The role and range of each parameter used in our models.

Parameter	Role	Range
γ_i	Timescale	[0.1, 3] γ_3 =1
σ	Sigmoidicity	10
ω_{i0}	Offset	[-2, 2]
ω_{ij}	Interaction Coefficient	[0.1, 1] [-1, -0.1] 0

Index i and j = 1, …, N where N = 3; $j \neq i$.

Generating a Single Score

For any parameter set, we start off looking for adaptation by setting the initial values for all three continuous variables to 0. Initially, the signal S is at 0 as well. We give the system sufficient time - 250 time units - to find its steady state levels of C_1, C_2, and C_3 for S = 0. If a steady state is found within that time, the signal is switched abruptly from 0 to 1; otherwise, a score of 0 is recorded for that parameter set. Keeping the signal at 1, we follow the time courses of the three variables until a new steady state is reached (or until another 250 time units have passed and no steady state is found, in which case also the score is recorded as 0). Since we are mostly interested in the time course of node 3 (the response variable), we record its value at the time point when the signal is applied (O_1); its maximum value in the presence of the signal (O_{max});

and its value when the simulation ends (O_2) in order to determine the adaptation score (see Figure 3.18).

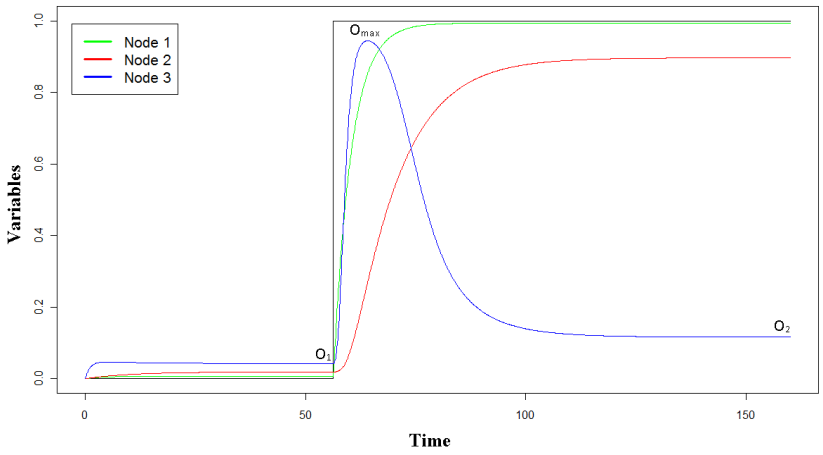


Figure 3.18. The response (blue), measured by node 3, to the signal (black). Score $Z \cong 7.30$.

The expression we use to get the score Z is:

$$Z = \frac{|O_{\max} - O_1|}{0.05 + |O_2 - O_1|} \qquad \text{[Eq. 3.4]}$$

This scoring function strongly favors a high peak response with a concomitant return close to the initial steady state value. We add 0.05 to the denominator so as not to give undue significance to cases for which $|O_2 - O_1|$ is close to 0. The scores can range from as low as 0 (no adaptation) to as high as 20 (perfect adaptation), since $|O_{\max} - O_1| \leq 1$. A decent score is $Z \geq 5$; a high score is $Z \geq 10$. The score that a motif exhibits depends on its topology (its code) and on the specific values assigned to its parameter set $Q = \{\gamma_1, \gamma_2, \omega_{10}, \omega_{20}, \omega_{30}, \omega_{12}, \omega_{13}, \omega_{21}, \omega_{23}, \omega_{31}, \omega_{32}\}$.

Evolutionary Algorithm: Generating scores over many generations

The scoring function described above is only for one parameter set. The aim is not to optimize the scoring function, but to find a sample of parameter sets that all exhibit high adaptation scores. In order to do so, we developed an evolutionary algorithm that systematically explores the parameter space. As the name suggests, the evolutionary algorithm we use operates in generation k with a set of N_k 'parental' parameter sets that each spawn off R_k 'progeny' parameter sets; these $N_k * R_k = M_k$ total progeny compete against each other to yield the N_{k+1} parental parameter sets of generation k+1.

As shown later in this section, it is not necessary that $N_{k+1} = N_k$. Each simulation has a 'characteristic' value of N, R, and therefore, $M = N*R$. For generation k+1:

$$R_{k+1} = \left\lceil \frac{M}{N_{k+1}} \right\rceil, \qquad \text{[Eq. 3.5]}$$

where R_{k+1} is the number of progeny parameter sets spawned by each of the N_{k+1} parents. The total number of progeny spawned in generation k+1 is $M_{k+1} = N_{k+1}*R_{k+1}$. The M_{k+1} total progeny compete against each other to yield the N_{k+2} parental parameter sets, and so on.

Each parameter of a progeny set $Q_{\text{Offspring}}$ is derived by mutating the corresponding parameter of the parental set Q_{Parent}. There are two types of mutations that can occur. The first is called a macromutation. Macromutations involve changing the sign of one of the six interaction coefficients ω_{ij}, resulting in the progeny having a different topology than its parent. If $\omega_{ij} < -0.2$, the macromutation converts ω_{ij} to zero or a positive value; if $\omega_{ij} > 0.2$, ω_{ij} becomes zero or negative; and, if $\omega_{ij} = 0$, ω_{ij} becomes positive or negative. Also, if $-0.2 < \omega_{ij} < 0.2$, but $\omega_{ij} \neq 0$, ω_{ij} is reset to 0. The newly positive ω_{ij} is always chosen from a Gaussian distribution with a mean of 1 and standard deviation of 0.1, and a newly negative ω_{ij} is chosen with a mean of -1 and standard deviation of 0.1. The particular interaction coefficient which is chosen to macromutate is selected at random, and so is the direction of the mutation.

One important feature of the macromutation step is that not all N parent parameter sets are macromutated. The percentage G_{macro} of the N sets that is macromutated is defined as follows:

$$G_{\text{macro}} = \frac{0.5}{1 + \frac{Z_{\max}}{4}}. \qquad \text{[Eq. 3.6]}$$

Here, $Z_{\max}$ is the highest score within the N parent parameter sets. This function ensures that the higher the maximum score, the fewer the number of sets that are macromutated. For example, if $Z_{\max} = 16$, the number of sets that will undergo a macromutation out of the total N, say 20, parents, would be $G_{\text{macro}} * N = 0.1 * 20 = 2$.

As we will see later, in certain evolutionary algorithms, we choose not to introduce macromutations in order to examine the parameter space of only one topology at a time. Therefore, the use of macromutations is optional.

The second type of mutation, which happens on all parameters of the parental set, is called a micromutation. Micromutations always occur after the optional macromutation has occurred. These set of mutations involve introducing random fluctuations by multiplying each of the parameters by 1 + r, where r is a Gaussian random number with mean = 0 and with standard deviation = 0.1 (for γ_1 and γ_2), = 0.25 (for the ω_{i0}'s), and = 0.15 (for the ω_{ij}'s). Note that γ_3 is never micromutated – it stays at 1. Unlike macromutations, micromutations do not change the sign of any of the interaction coefficients.

After all mutations have been applied, the values of the parameters are checked to make sure that they remain within their pre-specified ranges. For example, any $\omega_{ij} > 1$ is set to 1; any $\omega_{ij} < -1$ is set to -1. Also, any positive $\omega_{ij} < 0.1$ is set to 0.1, and any negative $\omega_{ij} > -0.1$ is set to -0.1. Similarly, the offsets ω_{i0}'s are constrained to be between -2 and 2, and the γ's between 0.1 and 3. Once the mutations have occurred and the total M progeny parameter sets have been derived, they are all scored one-by-one using the procedure described in the previous section. Only the M', out of M, progeny which score above 0 are considered. The selection criteria for choosing which N of these total M' progeny will survive to become the parental population for the next

generation is described in the next section. We will use p to index the progeny; $1 \le p \le M'$. Q_p is the parameter set of this offspring, and Z_p is its score.

Evolutionary Algorithm: Parent Selection Criteria

The "Beta" Criterion

We assign a survival probability q for each progeny p that determines the likelihood of that parameter set getting selected as a parent for the next generation. This probability q is a function of the score of that progeny relative to the highest and lowest score of all M' progeny in that generation. Formally, the relative score $Z_{p,\mathrm{rel}}$ is:

$$Z_{p,\mathrm{rel}} = \frac{Z_p - Z_{\min}}{Z_{\max} - Z_{\min}}, \qquad \text{[Eq. 3.7]}$$

where Z_p is the actual score of the progeny parameter set, as calculated by Equation 3.3; Z_{max} is the maximum score of all progeny in the current generation, and Z_{min} is the minimum score of all progeny in that generation.

The survival probability q_p for progeny p is calculated by:

$$q_p = e^{-\beta(1-Z_{p,\mathrm{rel}})}, \qquad \text{[Eq. 3.8]}$$

where β is a parameter controlling how many of the M' progeny are likely to be selected. We want to target selecting $2*N$ parents out of the M' progeny so that we are more or less guaranteed of getting at least N parents from this selection process. This is done by setting the average survival probability $<q_p>$ to $2*N/M'$. As a first guess for β, we compute the survival probability when $Z_{p,\mathrm{rel}} = 0.5$, which evaluates to:

$$< q_p >= \frac{2N}{M'} \cong e^{-\beta(1-0.5)}. \qquad \text{[Eq. 3.9]}$$

Hence, we get:

$$\beta = 2\ln\left(\frac{M'}{2N}\right), \qquad \text{[Eq. 3.10]}$$

which is our initial guess for β in Equation 3.7.

The survival probability q_p for a progeny is compared to a uniform random number r between 0 and 1 generated *de novo* for each progeny. If $q_p \ge r$, then the progeny is selected as a parent for the next generation. Otherwise, the progeny parameter set is discarded. Note that this process gives even low-scoring progeny a chance to survive. For example, with $\beta = 2$, and $Z_{min} = 0$, q ≈ 0.15, a higher than negligible survival probability.

If, by chance, we get more than $2*N$ survivors, we increase β by a factor of 1.33 and repeat the selection procedure (with the same progeny set) in order to decrease the number of survivors. Likewise, if we get less than N survivors, we decrease β by a factor of 0.5 to increase the number

of survivors. In the end, between N and $2*N$ survivors are chosen for the next generation. The stochastic nature of the selection method makes it unlikely that $N_{k+1} = N_k$. However, in the case of macromutations, we select exactly N parents per generation by randomly choosing that many sets from all survivors.

In some sets of simulations, we set a maximum value for β, and in others we don't. Also, within a simulation, setting $\beta_{k+1} = \beta_k$ as the initial guess did not speed up the process of getting the correct value of β. So the initial guess for β is always the value calculated from Equation 3.10.

Our selection criterion is not perfect, however. In the process of getting to the high-scoring region, β usually increases to and remains at a large value. This is because at lower values of β, even very-low scoring sets have a fair chance to survive (as illustrated with the $\beta = 2$ example above), and therefore there is a good chance that the selection step ends up with more than $2*N$ survivors. At this point, we keep increasing β by a factor of 1.33 until we get the desired number of survivors. With β high, there is a bias towards selecting only the best-scoring parameter sets. Even though this helps the search algorithm remain in the high-scoring region once such a region is found, ideally, the evolutionary algorithm should give low-scoring parameter sets the same chance to be selected at any stage in the search process.

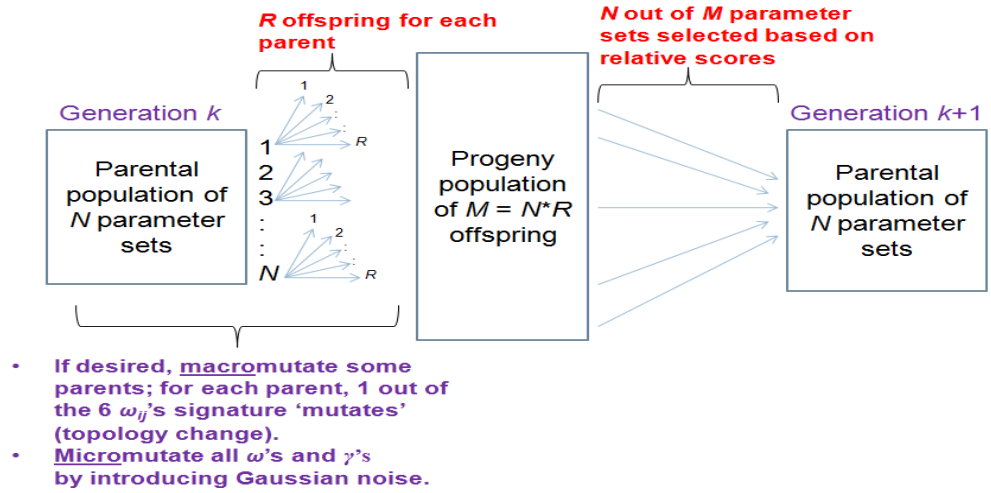


Figure 3.19. The evolutionary algorithm pipeline.

An alternative criterion for selecting parents, tournament selection, is presented in Appendix F.

Comparison of our methodology with Ma et al's

In the work done by Ma et al (Ma et al, 2009), scoring for adaptation is based on two separate quantities: sensitivity and precision. Sensitivity is defined as "the height of output response relative to the initial steady-state value", whereas precision is calculated by the inverse of "the difference between the pre- and post-stimulus steady states". The system is said to be adaptive when the response is both highly sensitive and highly precise, i.e., the response shows a high peak when the signal is applied along with a return close to the initial steady state. Both the sensitivity and precision calculations are scaled relative to $| S_{post} - S_{pre} |$, where the post-stimulus

signal value $S_{post} = 0.6$ and the pre-stimulus signal value $S_{pre} = 0.5$. In our case, $S_{pre} = 0$ and $S_{post} = 1$.

In our work, we combine sensitivity and precision in a single scoring function, Eq. 3.3. Sensitivity is in the numerator of the scoring function and the inverse of precision is in the denominator. In this way, a highly sensitive and highly precise system will give a high score. Ma et al use a different approach from ours to characterize the near-perfect adaptation property of a topology. They use a precision vs. sensitivity grid in which the upper-right quadrant has both high precision and high sensitivity. For a given topology, they simulate 10,000 parameters sets and require that 10 of them fall within this pre-defined quadrant for that topology to be classified as exhibiting near-perfect adaptation. Therefore, their notion of score is not associated with the degree of near-perfect adaptation of a single parameter set, like in our case; instead, their score is associated with how many parameter sets out of 10,000 show near-perfect adaptation. We classify a topology as exhibiting near-perfect adaptation if its average score is found to be high over a large number of parameter sets that are generated from an evolutionary search strategy.

Our evolutionary search strategy is pretty efficient in the sense that we are always looking for better scoring parameter sets than the ones the simulation has already found. We test a variety of parameter sets every generation, and favor the selection of those that increase the overall score. So not only are we exploring the parameters in a broad sense, we are also going deeper at each iteration towards the region in parameter space that consistently shows very high scores, assuming that such a region exists at all. Crucially, the simulation is able to stay in the high-scoring region once such a region is found. Even in the cases in which we are not able to find high scores, our broad search over a very large number of generations provides evidence that a high-scoring region does not exist. The size of the parameter space, with 11 dimensions, presents a problem for our methodology though. It may take many generations to stumble into a high-scoring region of parameter space. This process naturally takes plenty of computing time and resources.

The dynamical equations used to model regulatory motifs in both our work and Ma et al's work (Ma et al, 2009) are phenomenological in nature. In Ma et al, all three nodes are assumed to be proteins present in either an active or inactive form. The regulation of each protein (node i) by the other proteins (node $j \neq i$) is described by Ordinary Differential Equations (ODEs) with Michaelis-Menten kinetics. For example, if node 3 is activated by node 1 and inactivated by node 2, then:

$$\frac{dX_3}{dt} = X_1 k_{13} \frac{(1 - X_3)}{(1 - X_3) + J_{13}} - X_2 k'_{23} \frac{X_3}{X_3 + J'_{23}}. \qquad \text{[Eq. 3.11]}$$

In this case, X_3 is considered as a 'substrate' that is modified by the 'enzymes' X_1 and X_2, and the use of Michaelis-Menten rate law requires that total enzyme concentration << total substrate concentration. But in the ODE for $X_1(t)$ or $X_2(t)$, X_3 may appear as an 'enzyme' modifying the 'substrate' X_1 or X_2 according to the Michaelis-Menten rate law. Because the nodes change their roles as 'substrate' and 'enzyme', the use of Michaelis-Menten rate laws is internally inconsistent. Therefore, the dynamical system used by Ma et al, as the authors acknowledge (in their supplementary material) is not mechanistic but phenomenological.

In our case also, a topology's activations and inactivations, and the nonlinearity of the reactions, is captured in a phenomenological way using a limited number of parameters.

3.5 References

Alon U (2007) Network motifs: theory and experimental approaches. *Nat Rev Genet* **8:** 450-461

Alon U, Surette MG, Barkai N, Leibler S (1999) Robustness in bacterial chemotaxis. *Nature* **397:** 168-171

Barkai N, Leibler S (1997) Robustness in simple biochemical networks. *Nature* **387:** 913-917

Basu S, Mehreja R, Thiberge S, Chen MT, Weiss R (2004) Spatiotemporal control of gene expression with pulse-generating networks. *Proc Natl Acad Sci U S A* **101:** 6355-6360

Berg HC, Brown DA (1972) Chemotaxis in Escherichia coli analysed by three-dimensional tracking. *Nature* **239:** 500-504

Csikasz-Nagy A, Kapuy O, Toth A, Pal C, Jensen LJ, Uhlmann F, Tyson JJ, Novak B (2009) Cell cycle regulation by feed-forward loops coupling transcription and phosphorylation. *Mol Syst Biol* **5:** 236

Delforge J, Syrota A, Mazoyer BM (1989) Experimental design optimisation: theory and application to estimation of receptor model parameters using dynamic positron emission tomography. *Phys Med Biol* **34:** 419-435

Dinauer MC, Steck TL, Devreotes PN (1980) Cyclic 3',5'-AMP relay in Dictyostelium discoideum V. Adaptation of the cAMP signaling response during cAMP stimulation. *J Cell Biol* **86:** 554-561

Hauri DC, Ross J (1995) A model of excitation and adaptation in bacterial chemotaxis. *Biophys J* **68:** 708-722

Hornstein E, Shomron N (2006) Canalization of development by microRNAs. *Nat Genet* **38 Suppl:** S20-24

Jolliffe I (1986) *Principal component analysis*, 2nd edn. New York, NY, USA: Springer.

Knox BE, Devreotes PN, Goldbeter A, Segel LA (1986) A Molecular Mechanism for Sensory Adaptation Based on Ligand-Induced Receptor Modification. *P Natl Acad Sci USA* **83:** 2345-2349

Levchenko A, Iglesias PA (2002) Models of eukaryotic gradient sensing: application to chemotaxis of amoebae and neutrophils. *Biophys J* **82:** 50-63

Ma WZ, Trusina A, El-Samad H, Lim WA, Tang C (2009) Defining Network Topologies that Can Achieve Biochemical Adaptation. *Cell* **138:** 760-773

Macnab RM, Koshland DE, Jr. (1972) The gradient-sensing mechanism in bacterial chemotaxis. *Proc Natl Acad Sci U S A* **69:** 2509-2512

Mangan S, Itzkovitz S, Zaslaver A, Alon U (2006) The incoherent feed-forward loop accelerates the response-time of the gal system of Escherichia coli. *J Mol Biol* **356:** 1073-1081

Mello BA, Tu Y (2003) Quantitative modeling of sensitivity in bacterial chemotaxis: the role of coupling among different chemoreceptor species. *Proc Natl Acad Sci U S A* **100:** 8223-8228

Mettetal JT, Muzzey D, Gomez-Uribe C, van Oudenaarden A (2008) The frequency dependence of osmo-adaptation in Saccharomyces cerevisiae. *Science* **319:** 482-484

Mjolsness E, Sharp DH, Reinitz J (1991) A Connectionist Model of Development. *Journal of Theoretical Biology* **152:** 429-453

O'Donnell KA, Wentzel EA, Zeller KI, Dang CV, Mendell JT (2005) c-Myc-regulated microRNAs modulate E2F1 expression. *Nature* **435:** 839-843

Parent CA, Devreotes PN (1999) A cell's sense of direction. *Science* **284:** 765-770

Tyson JJ, Chen KC, Novak B (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr Opin Cell Biol* **15:** 221-231

Tyson JJ, Novak B (2010) Functional Motifs in Biochemical Reaction Networks. *Annu Rev Phys Chem* **61:** 219-240

Wilson HR, Cowan JD (1972) Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys J* **12:** 1-24

Yi TM, Huang Y, Simon MI, Doyle J (2000) Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proc Natl Acad Sci U S A* **97:** 4649-4653

APPENDICES

Appendix A: Examining IFFL-2 and IFFL-3 topologies.

We find that IFFL-2 and IFFL-3 topologies (Alon, 2007) do not score as well as IFFL-1's and IFFL-4's. The distinguishing factor between these two classes and the IFFL-1 and IFFL-4 classes is the regulation from node 1 to node 3. While it is positive in the latter, the IFFL-2 and IFFL-3 topologies always have a negative regulation from node 1 to node 3 (see Figure 3.3).

Table A.1. IFFL-2 and IFFL-3 average scores.

IFFL-2 Topologies		**IFFL-3 Topologies**	
Code	**< *Z* >**	**Code**	**< *Z* >**
111111	3.52	113113	2.33
111211	4.07	113213	2.25
111311	3.38	113313	1.54
121111	4.31	123113	2.27
121211	3.74	123213	2.28
121311	3.42	123313	1.84
131111	4.52	133113	2.18
131211	4.09	133213	2.38
131311	3.85	133313	1.62
211111	2.48	213113	2.38
211211	2.60	213213	2.52
211311	2.29	213313	1.63
311111	2.35	223113	2.24
311211	2.55	223213	2.20
311311	2.19	223313	1.67
221111	3.94	233113	2.38
221211	3.38	233213	1.87
221311	3.19	233313	1.65
231111	4.22	313113	2.41
231211	3.50	313213	2.22
231311	3.27	313313	1.91
321111	3.03	323113	2.20
321211	3.31	323213	2.39
321311	3.17	323313	1.44
331111	3.50	333113	2.21
331211	2.95	333213	1.98
331311	3.17	333313	1.51

< *Z* > represents average scores.

When we simulate the IFFL-2 and IFFL-3 topologies on their own, i.e., without macromutations, we find that they all have low average scores. For its 27 members, the IFFL-2 topologies' average scores range from 2.19 to 4.5. The IFFL-3 topologies' average scores range from 1.43 to 2.40. Table A.1 shows the average scores for the IFFL-2 and IFFL-3 sets – these were run with N=20 and R=20.

Table A.2. The hyper-ellipsoid volumes from the high-scoring sets of every IFFL-2 and IFFL-3 topology.

IFFL-2 Topologies			IFFL-3 Topologies		
Code	***n***	**Volume**	**Code**	***n***	**Volume**
221211	8	0.000524	223213	8	0.000118
121211	9	0.000113	123213	9	1.14E-05
211211	9	6.16E-05	213213	9	3.77E-05
221111	9	0.000521	223113	9	0.000141
221311	9	0.000158	223313	9	3.90E-06
231211	9	0.001529	233213	9	0.000165
321211	9	0.000615	323213	9	8.25E-05
111211	10	0.000217	113213	10	1.69E-05
121111	10	7.03E-05	123113	10	5.33E-05
121311	10	2.06E-05	123313	10	5.43E-06
131211	10	4.65E-05	133213	10	2.26E-05
211111	10	1.79E-05	213113	10	3.22E-05
211311	10	1.66E-05	213313	10	3.43E-06
231111	10	0.000934	233113	10	5.38E-05
231311	10	0.000201	233313	10	6.62E-07
311211	10	1.51E-06	313213	10	1.74E-05
321111	10	0.000595	323113	10	0.000279
321311	10	0.000186	323313	10	6.45E-05
331211	10	0.001559	333213	10	1.71E-05
111111	11	0.000514	113113	11	1.64E-05
111311	11	4.38E-05	113313	11	4.40E-07
131111	11	3.51E-05	133113	11	1.26E-05
131311	11	1.62E-05	133313	11	1.66E-07
311111	11	2.66E-06	313113	11	6.37E-05
311311	11	1.95E-07	313313	11	2.76E-06
331111	11	0.000535	333113	11	1.60E-05
331311	11	7.68E-05	333313	11	1.10E-07
	n	**<Volume>**		***n***	**<Volume>**
	8	5.24E-04		8	1.18E-04
	9	2.85E-04		9	3.842E-05
	10	8.789E-05		10	1.908E-05
	11	2.688E-05		11	2.875E-06

The IFFL-2 and IFFL-3 topologies are sorted according to number of dimensions n. Shown in the lower panel are the geometric means of volumes across n.

An in-depth analysis of these runs shows that most of them have a population of high-scoring sets along with a population of very low-scoring sets. We also find that the hyper-ellipsoid volumes of their high-scoring sets are at least one or two orders of magnitude smaller than those for IFFL-1's and IFFL-4's, making them less robust (compare the average ellipsoid volumes for each number of dimension in Table 3.15 to Table A.2). This leads us to believe that the Gaussian noise we introduce while micromutating the parameters, with a standard deviation of 0.15, may be too high for the IFFL-2 and IFFL-3 simulations to remain in the much narrower high-scoring region of parameter space, even after finding such a region.

Appendix B: Fewer progeny runs.

We have already established that given a random start in parameter space, all IFFL-1 and IFFL-4 topologies eventually evolve to a high-scoring region when allowed to evolve without macromutations, in the case of N=20 and R=20. We tested if these topologies would still evolve to a high-scoring region when R was decreased to 10 and 5, while keeping N at 20. We did this analysis only for IFFL-1 topologies with random starts. The results are shown in Table B.1.

Table B.1. IFFL-1 random start simulation results with N=20 and R=10 and 5.

IFFL-1	***N*=20 X *R*=10**		***N*=20 X *R*=5**	
Code	**< *Z* >**	**FPT**	**< *Z* >**	**FPT**
113131	14.21	46	9.52	92
113231	13.86	34	9.62	60
113331	13.25	56	9.04	91
123131	15.13	26	12.78	23
123231	14.64	29	12.5	37
123331	14.66	22	13.55	127
133131	15.81	31	14.23	127
133231	15.19	23	13.92	39
133331	15.21	38	12.51	74
213131	1.55	(500)	1.19	(1000)
213231	11.05	103	11.08	354
213331	5.5	(500)	6.44	(1000)
223131	9.57	146	10.15	29
223231	10.54	40	6.64	260
223331	7.58	(500)	5.72	(1000)
233131	11.94	83	5.13	(1000)
233231	1.43	(500)	1.39	(1000)
233331	11.67	22	6.93	(1000)
313131	3.94	(500)	7.92	129
313231	11.75	11	8.26	43
313331	5.51	(500)	4.69	(1000)
323131	6.02	(500)	8.61	134
323231	12.29	10	7.81	41
323331	12.03	28	6.08	(1000)
333131	5.31	(500)	7.96	64
333231	12.41	8	7.87	219
333331	7.57	104	6.17	(1000)

< *Z* > represents average scores. FPT stands for First Passage Time. Cases where the FPT is not found are indicated by the maximum number of generations in brackets.

Other runs were done via tournament selection with R decreased to 5 and 2, as shown in Table F.2.

Again, the First Passage Time (FPT) records the generation at which the mean score of the topology crosses 10, and the average score is calculated from the last 50 generations. The maximum number of generations, T_{max}, for which we ran the simulation with R=20 was 250. To make a fair comparison, we increase T_{max} by the same factor with which R is decreased. Therefore, with R=10, T_{max} = 500, and with R=5, T_{max} = 1000. The topologies which do not have a FPT are recorded with their T_{max} in brackets.

Note that these topologies get exactly the same initial conditions as the corresponding N=20 and R=20 runs. While the R=10 runs have 19 out of the 27 topologies finding a high-scoring region, the R=5 runs have 18 such topologies. Also, a comparison across topologies between the R=20 (Table 3.2) and R=10 and R=5 runs shows that the average score is always higher in the former case. This is despite a comparable swathe of parameter space being explored in each case.

Appendix C: Examining Classic Negative Feedback Loops.

For the sake of a more rigorous examination of negative feedback loops, we examined eight "classic" 3-edge networks that have negative feedback with all 3 nodes involved. Four of these topologies had higher than negligible scores – they are shown in Figure C.1.

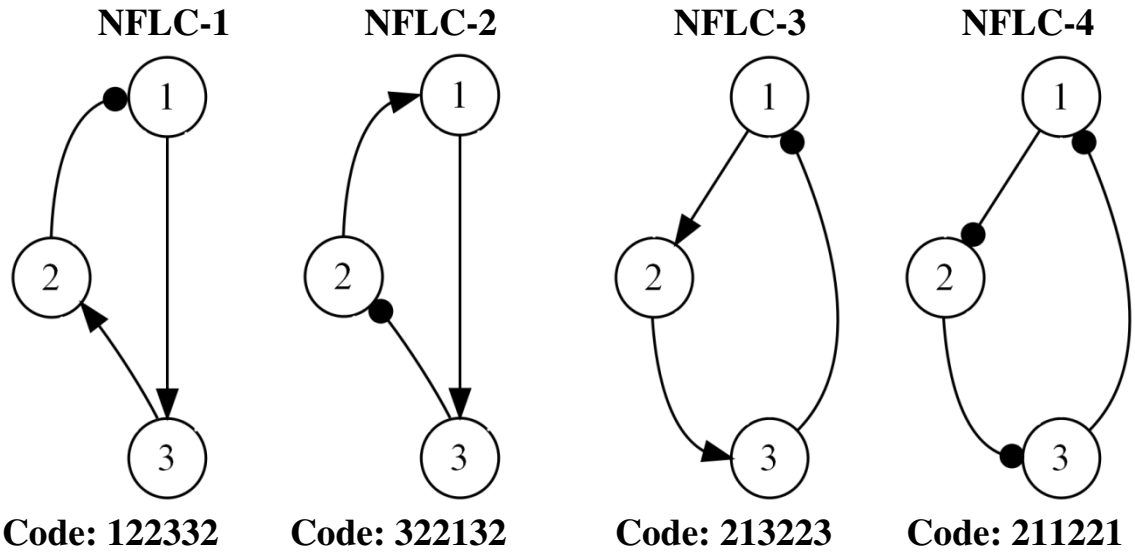


Figure C.1. The four Classic Negative Feedback Loops (NFLCs) that have a higher than negligible score.

All topologies were given low-scoring, random-starts in parameter space and given 250 generations to find a high-scoring region. The average scores, as computed from the last 50 generations, were approximately 4.5 for NFLC-1, 3 for NFLC-2, and 1.5 for both NFLC-3 and NFLC-4. Also, when allowed to macromutate, one of these topologies, NFLC-2, went to an IFFL-4 motif, and the other three went to IFFL-1 motifs, specifically 133231, 133331 and 123331. These three NFLB-1 coupled IFFL-1 motifs were also among the 9 dominant motifs found from the uncoupled extended NFLBs' macromutations analysis described earlier.

Appendix D: Listing of conducive parameters sets for the 27 IFFL-1 and 27 IFFL-4 topologies.

Mean and standard deviation of each parameter, for each of the 27 IFFL-1 and 27 IFFL-4 topologies. For each topology, the statistics were calculated over a sample of high-scoring parameter sets ($Z \geq 10$) from a random start run with N=20 and R=20.

Table D.1. IFFL-1 topologies' conducive-start parameters.

113131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.45	0.11	-0.02	-0.02	0.00	-0.98	-0.14	0.89	-0.20	0.97	-0.96
S.dev.	0.63	0.01	0.03	0.03	0.00	0.11	0.10	0.14	0.11	0.06	0.07

113231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.18	0.13	0.00	0.00	0.04	-1.00	-0.13	0.93	0.00	0.95	-0.98
S.dev.	0.58	0.06	0.01	0.00	0.04	0.01	0.04	0.11	0.00	0.07	0.04

113331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.03	0.14	-0.07	-0.06	-0.10	-0.97	-0.13	0.85	0.26	0.97	-0.88
S.dev.	0.58	0.08	0.09	0.07	0.11	0.07	0.04	0.16	0.23	0.06	0.16

123131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.05	0.16	-0.05	0.00	0.04	-0.99	0.00	0.88	-0.17	0.97	-0.95
S.dev.	0.53	0.09	0.05	0.00	0.04	0.03	0.00	0.13	0.08	0.06	0.08

123231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.43	0.12	-0.06	0.01	0.06	-0.99	0.00	0.89	0.00	0.97	-0.96
S.dev.	0.52	0.03	0.07	0.01	0.05	0.06	0.00	0.13	0.00	0.06	0.07

123331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.18	0.12	0.00	0.00	0.02	-1.00	0.00	0.95	0.16	0.97	-0.99
S.dev.	0.66	0.04	0.00	0.01	0.02	0.01	0.00	0.09	0.09	0.06	0.03

133131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.45	0.11	-0.38	-0.06	-0.26	-0.92	0.52	0.82	-0.16	0.97	-0.63
S.dev.	0.44	0.02	0.18	0.06	0.15	0.12	0.24	0.19	0.09	0.06	0.21

133231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.48	0.19	-0.46	-0.01	-0.01	-0.85	0.62	0.54	0.00	0.97	-0.90
S.dev.	0.46	0.14	0.16	0.02	0.01	0.16	0.22	0.12	0.00	0.06	0.12

133331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.35	0.18	-0.38	-0.10	-0.26	-0.84	0.50	0.85	0.30	0.97	-0.64
S.dev.	0.50	0.11	0.17	0.13	0.12	0.26	0.22	0.16	0.22	0.06	0.20

213131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.96	0.13	-0.43	-0.49	-0.41	0.00	-0.14	0.94	-0.32	0.98	-0.99
S.dev.	0.72	0.10	0.12	0.12	0.09	0.00	0.06	0.09	0.14	0.04	0.03

213231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	0.65	0.14	-0.42	-0.50	-0.33	0.00	-0.16	0.92	0.00	0.96	-0.98
S.dev.	0.47	0.12	0.14	0.14	0.11	0.00	0.09	0.11	0.00	0.08	0.05

213331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.51	0.12	-0.44	-0.11	0.01	0.00	-0.17	0.64	0.18	0.68	-0.96
S.dev.	0.53	0.02	0.15	0.04	0.02	0.00	0.11	0.22	0.16	0.08	0.08

223131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.24	0.25	-0.47	-0.49	-0.39	0.00	0.00	0.93	-0.24	0.97	-0.99
S.dev.	0.34	0.37	0.14	0.14	0.10	0.00	0.00	0.11	0.14	0.06	0.04

223231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.60	0.11	-0.43	-0.44	-0.40	0.00	0.00	0.89	0.00	0.96	-0.98
S.dev.	0.67	0.02	0.16	0.14	0.12	0.00	0.00	0.14	0.00	0.08	0.05

223331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.53	0.11	0.00	-0.46	-0.33	0.00	0.00	0.55	0.78	0.98	-0.97
S.dev.	0.43	0.01	0.01	0.09	0.11	0.00	0.00	0.13	0.21	0.04	0.06

233131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}

Mean	0.75	0.21	-0.51	-0.48	-0.38	0.00	0.27	0.90	-0.28	0.95	-0.97
S.dev.	0.32	0.22	0.17	0.14	0.11	0.00	0.19	0.13	0.14	0.08	0.06

233231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	0.91	0.19	-0.50	-0.49	-0.37	0.00	0.20	0.92	0.00	0.97	-0.98
S.dev.	0.45	0.27	0.17	0.15	0.10	0.00	0.16	0.11	0.00	0.07	0.05

233331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.27	0.14	-0.52	-0.43	-0.40	0.00	0.29	0.91	0.34	0.97	-0.98
S.dev.	0.49	0.08	0.18	0.13	0.10	0.00	0.24	0.13	0.29	0.06	0.06

313131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	0.48	0.19	-0.42	-0.52	-0.40	0.21	-0.15	0.94	-0.37	0.97	-0.99
S.dev.	0.27	0.18	0.13	0.12	0.10	0.16	0.07	0.10	0.13	0.06	0.04

313231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.30	0.13	-0.40	-0.45	-0.41	0.27	-0.23	0.92	0.00	0.97	-0.98
S.dev.	0.49	0.04	0.14	0.13	0.11	0.22	0.18	0.11	0.00	0.06	0.05

313331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.76	0.11	-0.44	-0.13	0.00	0.15	-0.21	0.58	0.24	0.68	-0.96
S.dev.	0.85	0.01	0.15	0.05	0.00	0.05	0.12	0.21	0.12	0.07	0.07

323131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.14	0.13	-0.53	-0.08	0.00	0.12	0.00	0.48	-0.20	0.64	-0.95
S.dev.	0.66	0.04	0.17	0.02	0.00	0.03	0.00	0.15	0.08	0.07	0.07

323231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.33	0.11	-0.45	-0.45	-0.42	0.33	0.00	0.92	0.00	0.97	-0.98
S.dev.	0.51	0.01	0.16	0.13	0.10	0.26	0.00	0.11	0.00	0.06	0.05

323331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.36	0.13	-0.45	-0.46	-0.39	0.27	0.00	0.91	0.19	0.97	-0.98
S.dev.	0.42	0.06	0.16	0.13	0.10	0.21	0.00	0.12	0.12	0.06	0.05

333131	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.01	0.13	-0.53	-0.43	-0.42	0.18	0.38	0.90	-0.25	0.96	-0.98
S.dev.	0.54	0.04	0.18	0.14	0.12	0.10	0.27	0.14	0.14	0.08	0.06

333231	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.04	0.11	-0.52	-0.47	-0.40	0.27	0.36	0.92	0.00	0.97	-0.98
S.dev.	0.66	0.01	0.19	0.14	0.10	0.18	0.20	0.11	0.00	0.07	0.05

333331	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.13	0.13	-0.66	-0.12	0.01	0.23	0.30	0.62	0.37	0.68	-0.96
S.dev.	0.54	0.04	0.24	0.05	0.02	0.16	0.23	0.23	0.27	0.09	0.07

Table D.2. IFFL-4 topologies' conducive-start parameters.

111133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.22	0.12	-0.11	0.42	-1.39	-0.30	-0.22	-0.87	-0.21	0.95	0.96
S.dev.	0.57	0.02	0.10	0.14	0.15	0.18	0.14	0.15	0.17	0.08	0.07

111233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.19	0.13	0.00	0.47	-1.39	-0.39	-0.16	-0.86	0.00	0.95	0.96
S.dev.	0.59	0.14	0.01	0.16	0.15	0.17	0.08	0.15	0.00	0.08	0.07

111333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.19	0.12	0.02	0.42	-1.41	-0.43	-0.14	-0.86	0.25	0.94	0.96
S.dev.	0.59	0.03	0.02	0.14	0.15	0.14	0.04	0.15	0.13	0.09	0.07

121133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.61	0.13	-0.16	0.43	-1.38	-0.29	0.00	-0.88	-0.26	0.95	0.97
S.dev.	0.58	0.04	0.16	0.15	0.15	0.17	0.00	0.15	0.19	0.08	0.07

121233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.29	0.16	-0.15	0.45	-1.38	-0.27	0.00	-0.88	0.00	0.96	0.97
S.dev.	0.42	0.20	0.12	0.15	0.14	0.16	0.00	0.14	0.00	0.08	0.06

121333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.72	0.12	-0.13	0.45	-1.42	-0.30	0.00	-0.89	0.23	0.96	0.98
S.dev.	0.51	0.03	0.12	0.15	0.14	0.15	0.00	0.14	0.14	0.08	0.06

131133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.29	0.14	-0.33	0.44	-1.39	-0.16	0.30	-0.90	-0.24	0.95	0.97
S.dev.	0.59	0.06	0.18	0.15	0.15	0.07	0.26	0.13	0.21	0.08	0.07

131233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.34	0.17	-0.38	0.47	-1.41	-0.16	0.54	-0.91	0.00	0.97	0.98
S.dev.	0.51	0.12	0.17	0.13	0.14	0.09	0.29	0.12	0.00	0.07	0.06

131333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.48	0.17	0.02	0.50	-1.40	-0.50	0.38	-0.89	0.14	0.94	0.97
S.dev.	0.70	0.15	0.04	0.15	0.13	0.18	0.20	0.12	0.05	0.07	0.06

211133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.21	0.15	-0.36	0.45	-1.38	0.00	-0.21	-0.88	-0.25	0.96	0.97
S.dev.	0.62	0.08	0.14	0.14	0.16	0.00	0.12	0.14	0.19	0.08	0.07

211233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.46	0.15	-0.36	0.41	-1.39	0.00	-0.16	-0.87	0.00	0.96	0.96
S.dev.	0.44	0.08	0.14	0.13	0.15	0.00	0.09	0.14	0.00	0.07	0.07

211333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.44	0.12	-0.39	0.44	-1.40	0.00	-0.21	-0.88	0.25	0.95	0.96
S.dev.	0.55	0.02	0.14	0.15	0.14	0.00	0.12	0.13	0.10	0.07	0.07

221133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.62	0.14	-0.41	0.45	-1.39	0.00	0.00	-0.89	-0.19	0.97	0.98
S.dev.	0.67	0.10	0.14	0.14	0.13	0.00	0.00	0.14	0.13	0.06	0.05

221233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.60	0.14	-0.43	0.43	-1.38	0.00	0.00	-0.88	0.00	0.95	0.97
S.dev.	0.68	0.06	0.16	0.16	0.15	0.00	0.00	0.15	0.00	0.08	0.06

221333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.15	0.12	-0.46	0.39	-1.40	0.00	0.00	-0.85	0.21	0.94	0.96
S.dev.	0.49	0.02	0.17	0.15	0.16	0.00	0.00	0.17	0.12	0.09	0.08

231133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	0.98	0.12	-0.51	0.43	-1.36	0.00	0.30	-0.87	-0.15	0.95	0.97
S.dev.	0.45	0.05	0.19	0.15	0.15	0.00	0.21	0.15	0.06	0.08	0.07

231233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.06	0.11	-0.56	0.34	-1.36	0.00	0.46	-0.82	0.00	0.94	0.96
S.dev.	0.56	0.02	0.22	0.14	0.17	0.00	0.28	0.18	0.00	0.10	0.08

231333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.70	0.18	-0.50	0.43	-1.39	0.00	0.37	-0.87	0.20	0.95	0.96
S.dev.	0.80	0.20	0.20	0.16	0.16	0.00	0.28	0.15	0.11	0.08	0.08

311133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.36	0.14	-1.13	0.14	-0.97	0.88	-0.13	-0.84	-0.17	0.96	0.67
S.dev.	0.48	0.08	0.15	0.08	0.20	0.15	0.04	0.18	0.11	0.07	0.20

311233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.94	0.13	-0.61	0.36	-1.35	0.23	-0.15	-0.83	0.00	0.95	0.95
S.dev.	0.57	0.07	0.21	0.14	0.17	0.14	0.05	0.15	0.00	0.07	0.09

311333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	1.90	0.14	-0.53	0.45	-1.40	0.14	-0.15	-0.89	0.15	0.97	0.97
S.dev.	0.70	0.09	0.15	0.15	0.14	0.06	0.06	0.13	0.06	0.06	0.06

321133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.32	0.13	-1.15	0.13	-0.95	0.83	0.00	-0.77	-0.49	0.94	0.70
S.dev.	0.56	0.04	0.17	0.08	0.21	0.19	0.00	0.20	0.32	0.09	0.22

321233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}

Mean	2.54	0.15	-1.18	0.11	-0.90	0.88	0.00	-0.87	0.00	0.95	0.59
S.dev.	0.42	0.10	0.14	0.07	0.18	0.16	0.00	0.15	0.00	0.09	0.19

321333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.07	0.11	-1.17	0.12	-0.95	0.85	0.00	-0.84	0.18	0.94	0.61
S.dev.	0.57	0.02	0.16	0.07	0.19	0.18	0.00	0.17	0.10	0.10	0.21

331133	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.43	0.16	-1.08	0.20	-1.05	0.61	0.34	-0.82	-0.31	0.95	0.71
S.dev.	0.45	0.11	0.35	0.14	0.27	0.35	0.23	0.18	0.21	0.09	0.26

331233	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.09	0.12	-1.26	0.09	-0.88	0.83	0.29	-0.83	0.00	0.94	0.57
S.dev.	0.55	0.02	0.19	0.07	0.20	0.19	0.20	0.18	0.00	0.10	0.21

331333	γ_1	γ_2	ω_{10}	ω_{20}	ω_{30}	ω_{12}	ω_{13}	ω_{21}	ω_{23}	ω_{31}	ω_{32}
Mean	2.52	0.14	-1.24	0.11	-0.93	0.77	0.45	-0.82	0.15	0.94	0.59
S.dev.	0.45	0.07	0.25	0.09	0.22	0.28	0.24	0.19	0.06	0.09	0.24

Appendix E: Exploring First Passage Times over multiple iterations.

Earlier, in Table 3.2, we recorded the First Passage Times (FPTs) for all IFFL-1 and IFFL-4 topologies run with micromutations-only. Again, FPT refers to the generation in which the mean score of a parental set crosses 10. We now repeat the procedure 100 times for each topology so that we can calculate average FPTs. All 100 runs start with different, randomly chosen, initial parameters. We collect all parameter sets that score above 10, across these 100 runs, and calculate the hyper-ellipsoid volume for each topology using these high-scoring sets.

The maximum number of generations allowed per run, T_{max}, is 400. If a run goes all the way to this upper limit of generations without crossing $Z = 10$, we stop the run and record that FPT as not found.

Also, we set a condition under which a run can abort early. In the Methods section, we introduce the parameter β which controls how many of the progeny sets survive to become parental sets for the next generation. We increase β if we want to decrease the number of survivors, and vice versa. In some cases, despite repeated increases to the value of β, a run is not able to narrow down the number of survivors, perhaps due to a flat distribution of all the progeny scores. If, in this process, β exceeds 250, the run is immediately aborted.

Due to the time-consuming nature of these new simulations, we have so far only been able to get statistics on 19 out of the 27 IFFL-1 topologies.

In Table E.1, "Total Iters" refers to the number of runs out of 100 that do not abort. Of these, the number of runs that find a FPT are recorded under "FPT Iters". Therefore, the number of non-aborted runs that go all the way till T_{max} without finding an FPT is simply Total Iters - FPT Iters. The average FPT, standard deviation, along with the minimum and maximum FPT are also recorded for a topology.

We were looking for an inverse correlation between the average FPT and hyperellisoid volume, with the hypothesis being that a larger high-scoring parameter space would be found more quickly, on the average. However, the collected statistics do not show such a correlation.

Table E.1. First Passage Time (FPT) statistics on the 19 IFFL-1 topologies run for 100 iterations each.

Topology	*n*	Ellip Vol	Total Iters	FPT Iters	Avg FPT	SDev FPT	Min FPT	Max FPT
123131	10	6.68E-05	19	19	49.32	27.35	16	114
123231	9	0.00269685	46	46	38.63	26.32	12	134
123331	10	0.005257436	41	41	37.46	34.49	8	232
133131	11	0.000647999	30	30	45.33	21.21	10	105
133231	10	0.00834577	36	36	31.56	16.72	11	73
133331	11	0.01674151	37	37	22.57	9.88	8	51
213131	10	0.01387888	55	10	72.9	67.73	19	245
213231	9	0.02556658	63	24	100.71	81.17	27	349
213331	10	0.02899178	58	35	89.69	77.11	7	314
223131	9	0.009330953	21	13	60.92	42.95	16	149
223231	8	0.08485527	26	12	70.92	53.74	19	195
223331	9	0.08700994	36	29	54.17	41.36	14	164
233131	10	0.001573423	20	12	77.5	31.95	38	136
233231	9	0.1110491	34	22	46.59	24.52	6	95
233331	10	0.134142	29	22	64.59	63.49	12	317
313231	10	0.01564209	56	18	68.22	69.69	17	326
323331	10	0.05660934	28	13	31.62	14.49	10	62
333231	10	0.1564729	41	24	70.46	48.36	17	192
333331	11	0.08528936	46	26	53.23	45.66	9	211

n is the number of dimensions of the hyper-ellipsoid whose volume is recorded.

Also, to get an idea of the distribution of the FPTs for each simulated topology, we construct plots with the number of generations t as the independent variable, and the probability P(t) that FPT > t as the dependent variable. Therefore, P(0) = 1, and the probability curve steadily decreases as the increasing FPT values are taken into account. Figure E.1 shows these curves for the first 12 topologies.

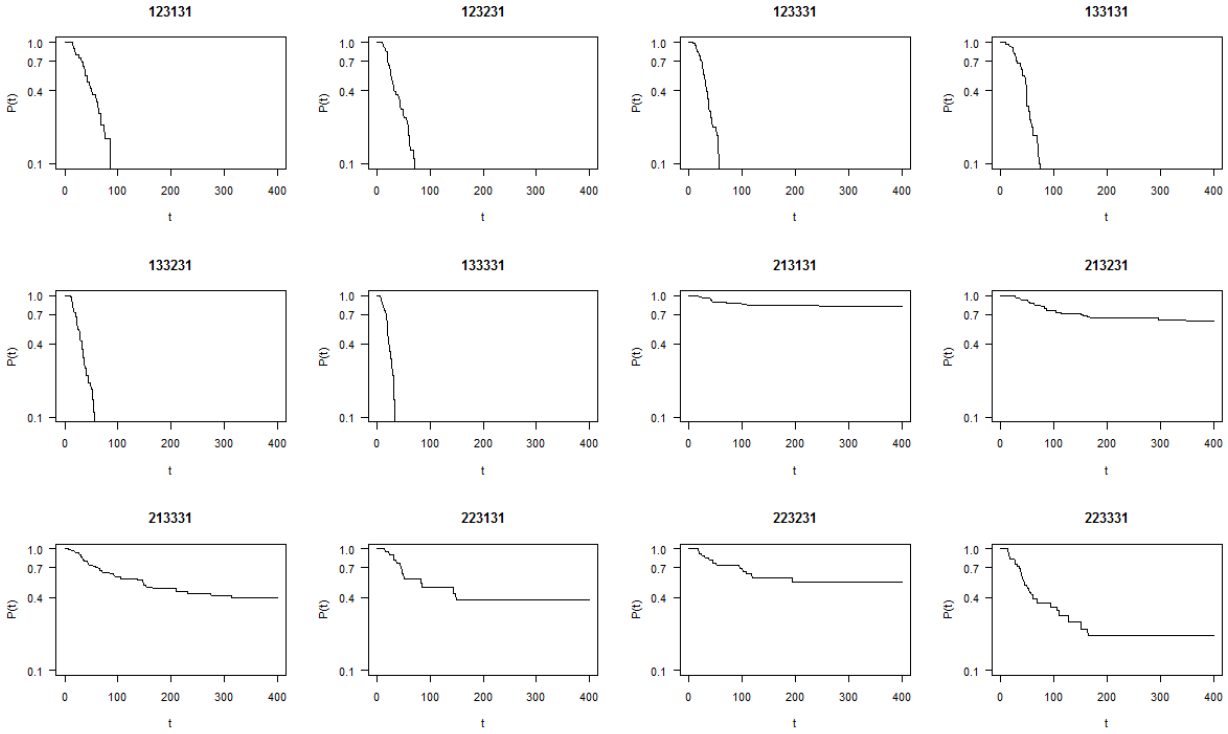


Figure E.1. Probability curves for 12 IFFL-1 topologies.

Only the first six topologies in the table (which happen to be IFFL-1 + NFLB-1's) have all their valid iterations find a FPT, and only their curves are amenable to be modeled using the exponential distribution e^{-kt}. The k's range between 0.03 and 0.095.

For the other six topologies, the reason for many of the non-aborted iterations not having a FPT may be that the threshold we set for the mean score, 10, is too high. Indeed, a few iterations did have maximum mean scores between 7 and 9. Therefore, we tested the already collected simulation profiles with a lower threshold of 5 to see how many more of them find a FPT under the new relaxed criteria. Those results are shown in Table E.2. As evident from the table, a significantly higher number of non-aborted total iterations find an FPT with the lower threshold of 5.

We went on to construct similar plots as shown in Figure E.1 with the FPT threshold being 5. Again, we calculated k and average FPT for every topology which showed an exponential distribution of the FPTs, but could not find any correlation between those two parameters and the (re-calculated) hyper-ellipsoid volumes. Those results are listed in Table E.2, sorted by the number of dimensions n over which the volumes are calculated. The volumes are used to sub-sort the table.

Table E.2. Comparing First Passage Time (FPT) statistics from FPT thresholds 10 and 5.

Topology	Total Iters	FPT Iters (10)	FPT Iters (5)
123131	19	19	19
123231	46	46	46
123331	41	41	41
133131	30	30	30
133231	36	36	36
133331	37	37	37
213131	56	10	32
213231	64	24	59
213331	59	35	56
223131	22	13	22
223231	27	12	25
223331	37	29	36
233131	20	12	19
233231	35	22	31
233331	30	22	29
313231	58	18	46
323331	28	13	28
333231	43	24	28
333331	46	26	31

For each topology, we list the "Total Iters" and "FPT Iters" just as in Table E.1. Listed in brackets are the FPT thresholds used.

Table E.3. Relevant statistics from Table E.1 re-calculated with threshold 5.

Topology	*n*	Ellip Vol	k	Avg FPT
223231	8	0.189	0.01	44.32
123231	9	0.0669	0.016	32.43
223131	9	0.1223	0.006	67.36
213231	9	0.1342	0.006	84.76
223331	9	0.5551	0.019	25.72
233231	9	0.645	0.01	36.03
123131	10	0.0051	0.012	45.68
213331	10	0.0595	0.005	54.68
123331	10	0.1005	0.036	32.15
133231	10	0.1678	0.026	24.72
313231	10	0.1942	0.003	115.46
323331	10	0.2228	0.016	36.48
233131	10	0.356	0.003	115.32
233331	10	0.4956	0.024	37.9
133131	11	0.0073	0.025	40.97
133331	11	0.1067	0.09	18.24

Appendix F: Tournament selection runs.

Tournament Selection Criteria

In lieu of assigning survival probabilities to each progeny parameter combination, another method we tested to select N parents for the next generation was the tournament selection criteria. This criteria yielded mixed results – a few topologies managed to find a high-scoring region, and a few did not. Two versions of it were tried, a more primitive "short shuffle" method, and a more sophisticated "long shuffle" method.

"Short Shuffle" Method
In this method, the M progeny were divided into $M/2$ sets (all cases had an even number of M progeny) of 2 progeny each. Next, the higher scoring progeny in each pair was selected to go to the next round, where it would pair up with another "winner", and so on. These multiple rounds of selection were continued until N progeny were left. As shown in the results section later, this method was much less effective in finding a high-scoring region in parameter space for a given topology, and even more importantly, in staying in it even after finding one.

"Long Shuffle" Method
The procedure in this more successful method was to divide the M progeny into N sets, or brackets, of R progeny each, and select the best-scoring parameter combination from each set. The R progeny in each set were chosen randomly from the M progeny to avoid the scenario in which only the progeny from one parent compete against each other, which would result in mediocre-scoring combinations being almost certain of surviving. Competition amongst progeny from different parents favors the best scoring combinations' survival, while still giving the mediocre-scoring ones a slim chance to survive to become parents for the next round. Collectively, this would ensure that the mean score of all N surviving parameter combinations would continue to stay high over successive generations once, of course, a high-scoring region in parameter space had been found.

These methods were tried only with the 27 IFFL-1 topologies starting with completely random initial conditions. The short shuffle results are presented first in Table F.1.

Table F.1. Short shuffle tournament selection runs.

IFFL-1	*N*=20 X *R*=10		*N*=10 X *R*=20		*N*=16 X *R*=8	
Code	< *Z* >	FPT	< *Z* >	FPT	< *Z* >	FPT
113131	11.05	210	15.74	46	12.23	92
113231	13.41	49	12.12	27	12.47	46
113331	14.12	135	15.41	49	11.84	61
123131	15.93	73	0.95	(250)	13.29	197
123231	13.39	80	15.12	200	12	47
123331	10.37	59	0.95	(250)	12.92	29
133131	13.48	143	15.29	135	15.77	52
133231	12.77	23	15.65	93	15.04	87
133331	12.06	153	13.19	59	15.94	77
213131	1.46	(250)	1.82	(250)	1.6	(250)
213231	1.01	(250)	1.79	(250)	10.79	57
213331	2.31	(250)	4.75	(250)	5.91	(250)
223131	1.52	(250)	0.95	(250)	11.51	61
223231	1.45	(250)	2.9	(250)	12.54	36
223331	5.38	(250)	6.71	(250)	8.45	(250)
233131	9.17	148	0.95	(250)	13.37	40
233231	0.94	(250)	5.38	(250)	12.61	58
233331	2.08	(250)	2.65	(250)	10.94	21
313131	6.27	152	4.33	(250)	1.78	(250)
313231	7.86	25	15.04	9	12.52	10
313331	5.24	(250)	2.07	(250)	7.39	(250)
323131	4.89	(250)	1.62	(250)	2.29	(250)
323231	10.19	14	16.21	6	13.17	7
323331	7.4	(250)	6.64	(250)	12.57	25
333131	0.89	(250)	0.92	(250)	1.9	(250)
333231	10.57	12	15.69	3	11.83	12
333331	5.75	(250)	8.77	87	12.97	34

< *Z* > represents average score. FPT stands for First Passage Time. Cases where the FPT is not found are indicated by the maximum number of generations in brackets.

Again, the First Passage Time (FPT) indicates the generation at which the mean score of the topology crosses 10; the simulation is followed for another 49 generations; and, the average score is calculated from the last 50 generations. The topologies were given a maximum of 250 generations to find a high-scoring region, and if they were unsuccessful, their FPT is simply recorded as (250).

In some cases, when we looked into the mean score vs. generation plot of some of these topologies, we found that the topology does not necessarily stay in a high-scoring region. For example, in the plot below (Figure F.1) for topology 323231 (an Incoherent Feed-Forward Loop Type 1), the mean score crosses 10 at generation 14, but then eventually it claws back to nearly 5 by the time the simulation ends.

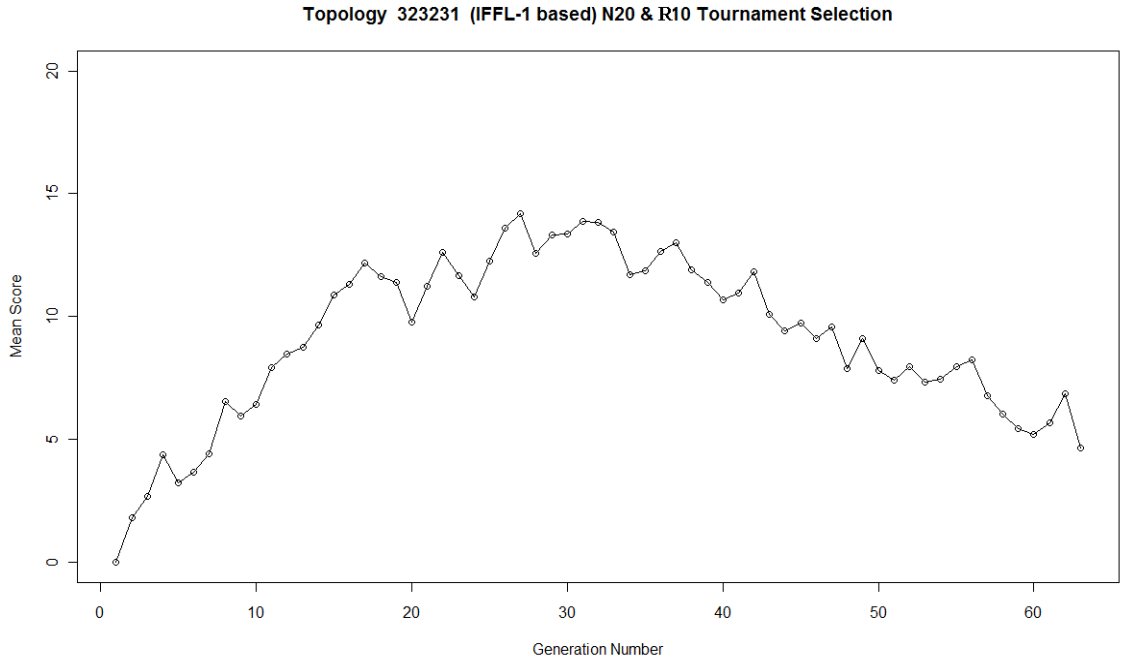


Figure F.1. Mean score vs Generation Number plot for a "short shuffle" tournament selection run with *N*=20 and *R*=10.

Let us now examine the results generated using the "long shuffle" tournament selection criterion, as shown in Table F.2.

In this set of simulations, with *N*=16 and *R*=16, the topologies that did not cross a mean score of 10 even after 500 generations are marked with a (500) in their FPT column. Only 5 such topologies are found out of a total of 27.

Fewer progeny runs, with *N*=20 and *R*=5 and 2, are also shown. Many of the topologies in these runs do not have a FPT. The maximum number of generations with *R*=5 is 1000, and that with *R*=2 is 2500.

Table F.2. Long shuffle tournament selection runs.

IFFL-1 Code	*N*=16 X *R*=16		*N*=20 X *R*=5		*N*=20 X *R*=2	
	< *Z* >	FPT	< *Z* >	FPT	< *Z* >	FPT
113131	17.01	33	14.8	29	9.6	110
113231	16.83	22	15.19	29	9.76	72
113331	17.03	9	15.15	20	10.19	47
123131	17.43	23	15.33	71	11.07	85
123231	17.64	109	14.8	28	9.95	100
123331	17.26	291	14.11	20	11.14	55
133131	0.95	(500)	0.95	(1000)	12.47	128
133231	17.3	107	15.53	312	9.3	182
133331	17.14	19	16.6	23	13.19	53
213131	2.04	(500)	13.51	42	6.23	(2500)
213231	13.67	15	13.11	40	7.97	156
213331	16.04	31	14.08	44	7.09	616
223131	9.5	67	6.92	(1000)	4.67	(2500)
223231	7.4	(500)	6.66	(1000)	7.86	96
223331	16.12	9	13.28	63	4.48	(2500)
233131	16.6	20	8.95	95	6.44	(2500)
233231	15.9	10	14.18	18	4.52	(2500)
233331	16.61	7	14.04	30	6.2	(2500)
313131	14	19	2.49	(1000)	3.98	(2500)
313231	16.46	10	14.07	17	5.63	(2500)
313331	6.87	(500)	6.09	(1000)	4.31	(2500)
323131	15.5	18	13.5	24	8.88	70
323231	15.72	10	14.57	5	8.44	44
323331	14.43	55	14.07	16	7.95	242
333131	4.55	(500)	2.31	(1000)	3.7	(2500)
333231	16.44	4	14.17	6	7.52	117
333331	12.32	86	7.6	(1000)	4.51	(2500)

< *Z* > represents average score. FPT stands for First Passage Time. Cases where the FPT is not found are indicated by the maximum number of generations in brackets.

Overall, the tournament selection criteria do show a fair degree of success in helping topologies score well, although compared to the Beta selection criterion, they sometimes fail to help a topology reach a high-scoring region, even with exactly the same initial parameters.