Faithful Modeling of Transient Behavior

in Developmental Pathways

Amir Rubinstein
Faithful Modeling of Transient Behavior in Developmental Pathways

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Amir Rubinstein

Submitted to the Senate of the Technion - Israel Institute of Technology
Tishray, 5767 Haifa October 2006
The Research Thesis Was Done Under the Supervision of:

Assoc. Prof. Ron Y. Pinter in the Computer Science Department
And
Assoc. Prof. Yona Kassir in the Biology Department

During the past two years Prof. Ron Pinter and Prof. Yona Kassir have introduced me to the concept of scientific research. I wish to thank them for making this process challenging, fertile - and above all - enjoyable.

The Generous Financial Help Of
The "Leonard and Diane Sherman Interdisciplinary Graduate School Fellowship" and
The Center for Complexity Science (CCS)
Is Gratefully Acknowledged.
Contents

Abstract 1

Abbreviations 2

Chapter 1: Introduction 3
  1. Motivation ........................................... 3
  2. Background .......................................... 4
     2.1. Gene regulation .................................. 4
     2.2. Modeling .......................................... 5
        2.2.1. Modeling and simulation .................. 5
        2.2.2. Modeling regulatory networks ............ 5
  3. Previous work ........................................ 6
  4. Highlights of results ................................ 8
  5. Outline ................................................ 8

Chapter 2: The Early Meiosis-Specific Genes system in yeast 9

Chapter 3: Method 11
  3.1 The Boolean network model .......................... 11
  3.2 Modeling of the yeast EMG system: Extending the Boolean network model .................. 12
  3.3 Data Analysis ........................................ 13
  3.4 Implementation ...................................... 13

Chapter 4: Evolution of the Network and Results 16
  4.1 Overview ............................................ 16
  4.2 Initial networks ..................................... 17
  4.3 Excluding redundant elements ...................... 31
  4.4 Introducing gating by conjunction ................ 35

Chapter 5: Discussion and Predictions 40

Chapter 6: Conclusions and Future Work 47

Bibliography 48
List of Figures

2.1. A schematic illustration of the transcriptional cascade governing meiosis in S. cerevisiae. ........................................ 10
2.2. Experimental data showing the transient transcription of IME1 and IME2 ............... 10
3.1. A simple network, showing oscillatory behavior .............................................. 11
4.1. The initial working hypothesis network describing the relationship between the expression of IME1 and IME2 ...................................................... 17
4.2. The modified network, based on the network illustrated in Fig. 4.1 and the modifications described in the text ......................... 19
4.3. The trajectory of the initial vector 00000000 in the simulation of the network illustrated in Fig. 4.2 .................................................. 21
4.4. The trajectory of the initial vector 10101100 in the simulation of the network illustrated in Fig. 4.2 .................................................. 22
4.5. The trajectory of the initial vector 10000010 in the simulation of the network illustrated in Fig. 4.2 .................................................. 22
4.6. The trajectory of the initial vector 01110010 in the simulation of the network illustrated in Fig. 4.2 .................................................. 23
4.7. The trajectory of the initial vector 00111100 in the simulation of the network illustrated in Fig. 4.2 .................................................. 23
4.8. Karnaugh-like maps: Visualizing the functional dependence between initial states and the trajectories of behavior for the network in Fig. 4.2 .......................................................... 24
4.9. Distribution of the lengths of the trajectories for the network in Fig. 4.2 .................. 25
4.10. The trajectory of the initial vector 00010000, with N=9 (top) and N=15 (bottom) in the simulation of the network illustrated in Fig. 4.2 .................................................. 26
4.11. Distribution of the lengths of the trajectories for the network in Fig. 4.2, with N=15 ................ 27
4.12. The trajectory of the initial vector 00000000: Original results (top), addition of self loop on Im1 (bottom) .................................................. 28
4.13. The trajectory of the initial vector 1000010, original results (top), addition of self loop on Im1 (bottom) .................................................. 29
4.15. Results of the simulation on the network illustrated in Fig. 4.14 .......................... 33
4.16. Grouping the initial vectors into two types .................................................. 34
4.17. Karnaugh-like maps: Visualizing the functional dependence between initial states and the behavior of trajectories (results for network 4.14) .................................................. 34
4.18. The final working hypothesis network .................................................. 35
4.19. Simulation of the network in Fig 4.18 (A) vs. experimental (B) results ............... 36
4.20. The computational model was validated by experimental results ...................... 39
5.1. Six working hypothesis networks ................................................................. 42
5.2. Distribution of the lengths of the trajectories for the alternative networks illustrated in Fig. 5.1 .................................................. 44
5.3. The functional dependence between the initial states and the trajectories of behavior for the various networks .................................................. 45
5.4. The transition graph for the major attractor (v) of Network 6. ......................... 46
List of Tables

3.1 The 3 final vectors obtained by the simulation of the network in Fig. 4.14 using the Boolean model........................................... 12
4.1. A vector represents the state of the network (illustrated in Fig. 4.2) at each time step.................................................. 20
4.2. Basic statistics about the final vectors in the simulation done on the network illustrated in Fig. 4.2.................................................. 20
4.3. Statistics about the effect of adding a self loop on Imc1.................................................. 29
4.4. Statistics on the final vectors in the simulation with the network in Fig. 4.15................................. 33
4.5. Final vectors obtained by the simulation of the network in Fig 4.18.................................................. 36
4.6. Final vectors obtained when omitting the self loop on Imc2 in the network illustrated in Fig. 4.18.................................................. 38
5.1A final vectors obtained by the simulation with $N=2$, in the simulation on the network illustrated in Fig. 4.18.................................................. 40
5.1B final vectors obtained by the simulation with $N=3$, in the simulation on the network illustrated in Fig. 4.18.................................................. 40
5.1C final vectors obtained by the simulation with $N=15$, in the simulation on the network illustrated in Fig. 4.18.................................................. 41
5.2. The alternative networks and highlights of their simulation results.................................................. 43
Abstract

The modeling and analysis of genetic regulatory networks is essential both for better understanding their behavior as well as for elucidating and refining open issues. Many methods for simulating and inspecting the properties of such pathways have been devised, borrowing from a variety of techniques such as differential equations, algebraic calculi, and Flux Balance Analysis. Most of these methods are quantitative in nature and require data that is often not fully revealed; moreover, some of them are computationally intensive, requiring significant time and resource.

This research proposes a computational model that allows for qualitative analysis of regulatory pathways, enabling the examination of characteristics such as transient behavior, robustness, and sensitivity to initial conditions, in an effective manner. To this end, we have extended the Boolean network model, which has limited modeling power, to a richer albeit discrete network model, while maintaining computational efficiency. Moreover, we have borrowed a simple technique for the representation of functions, namely Karnaugh-like maps, to elucidate and visualize the behavior of the pathways under study. We have applied our method to analyze the transience and robustness of a representative developmental pathway, namely early meiosis in budding yeast. Some of our analytic observations, such as the pathway’s response to premature expression of a key regulator, were validated in the lab and were found to be in agreement with experimental data. Furthermore, our analysis predicts new modes of regulation by which negative feedback loops accomplish their roles. We have prepared a simple user interface mechanism to allow future research to be easily conducted using this model.
Abbreviations

DNA - Deoxyribonucleic acid
RNA - Ribonucleic acid
EMG - Early Meiosis-specific genes
ATP - Adenosine triphosphate
FBA - Flux balance analysis
PCR - Polymerase chain reaction
qPCR - quantitative reverse transcriptase coupled PCR

Throughout this thesis, a name of a gene begins with three capital letters in italics (e.g. IME1), and a recessive mutation is written in lower case letters (e.g. ime1). A protein is given the same name as its gene, but only the first letter is capitalized (e.g. Ime1). The symbol Δ stands for a deletion allele (e.g. IME1Δ).
Chapter 1: Introduction

A biochemical pathway is a series of chemical reactions occurring within a cell. These pathways contain numerous elements that interact with each other in a complex manner. There are two main approaches to the study of biochemical pathways: the classic biological approach and the computational approach. The latter allows the integration of vast amounts of biological data into one comprehensive picture and the efficient analyses, both qualitative and quantitative, of biological systems.

1. Motivation

Our objective was to construct an efficient computational model that qualitatively describes the transient expression evident in developmental pathways. Such pathways are characterized by a transcriptional regulatory cascade that ensures the coordinated expression and activity of a network of genes. These pathways are usually governed by master activators. An important feature is that the expression of the regulatory genes occurs during short specific windows in the differentiation pathway (Freeman, 2000). Such short-lived signals are usually accomplished through positive and negative feedback loops (Bolouri and Davidson, 2002a; Ferrell, 2002; Freeman, 2000). For instance, the development of the eye in Drosophila depends on the transient and brief period of expression of the \textit{sevenless} gene (Bowell et al., 1989); another example emerges from the development of the embryonic node and visceral endoderm in \textit{C. elegans} which depends on a transient expression of master regulators of signaling pathways such as WNT, BMP, Hedgehog and FGF (Bouhon et al., 2005).

The importance of feedback regulation was demonstrated in several systems: in mice, deletion of the negative feedback regulator \textit{SOCS1} resulted in death as neonates (Marine et al., 1999); in yeast meiosis, a non-transient behavior of meiotic regulators was deleterious to meiosis (Shefer-Vaida et al., 1995). In this research we focus on the developmental pathway of meiosis in budding yeast, which is subject to positive and negative feedback loops (Kassir et al., 2003).

Continuous and discrete models were used effectively to simulate developmental pathways. Continuous methods require the availability of exact numeric values of, e.g., concentrations, expression levels, and timings, the data to achieve this level of detail are more often than not unavailable. In addition, some of these methods are computationally intensive, requiring significant time and resource. Moreover, such details are not always necessary when reasoning about the qualitative behavior of a pathway; specifically, some important properties, such as robustness, fail-safety and oscillation, can be gleaned using efficient, discrete methods (Batt et al., 2005; Bernot et al., 2004).

The simplest kind of a discrete method is a Boolean network. Li \textit{et al.} (Li et al., 2004) have shown that using this model several interesting properties, such as stability and robustness, can be observed on, e.g. the cell cycle in yeast. We found, however, that this method – simple and elegant as it may be – has certain limitations when it comes to predicting the properties of developmental pathways, e.g. transient behavior. Hence we devised an extension to this model, which is more expressive in terms of expression levels and transitional rules, but at the same time it preserves the efficiency and ability to effectively refine the way in which specific pathways are modeled.

The goal of this chapter is to supply the necessary background for understanding the concept of modeling regulatory networks, the rationale underlying this research and its results. I start by introducing some key terms in biological regulation (Section 2.1),
continue in describing the need for modeling (Section 2.2), and then give an overview of previous work in this field (Section 3). Section 4 states the highlight of the results of this research.

2. Background

2.1. Gene regulation

The DNA present in every cell of an organism contains genes, which are sequences of DNA that encode for proteins. Gene expression (the process by which DNA enables the formation of these proteins) comprises two main stages: the first, in which a molecule called RNA is formed from the gene, is called transcription; the second, in which RNA turns into proteins is called translation. Proteins have many important roles in the cell: some are used as building blocks of the cell, others function, for instance, as enzymes required for biochemical reactions essential for the survival and duplication of the cell.

The functionality of proteins is regulated at different levels: transcription and stability of mRNA, translation and stability of the protein, and post-translational modification of the protein.

An important role of proteins is to promote and regulate the transcription of genes. The regulated transcription of specific genes is accomplished by proteins designated as transcription factors, which bind to short DNA sequences. Transcription factors may inhibit the transcription of a gene (and thus they are called repressors), or activate it (activators). Also, transcription factors may regulate the transcription of their own genes. Some transcriptional activators and repressors do not bind directly to the DNA and are recruited to the specific genes following their association with a specific DNA-binding protein. The activity of regulators is often affected by internal and external signals, sometimes in the form of small molecules such as metabolites.

The external signals which regulate transcription of specific genes are transmitted to the nuclei through elaborate signal transduction pathways.

The abovementioned interactions give rise to genetic regulatory systems, structured by networks of regulatory interactions between DNA, RNA, proteins and small molecules. Such networks of interacting molecular components direct many biochemical processes. Reconstruction of a regulatory network from existing experimental data is essential for the analytical examination of the system at hand. An example of such a systematic reconstruction appears in (Forster et al., 2003): A metabolic network in the budding yeast *Saccharomyces cerevisiae* was reconstructed based on available biological data. These data, which consist of genomic, biochemical and physiological information, were collected from online genome and pathway databases, key references and journal publications. The set of chemical reactions was formulated and solved using an approach mentioned in more detail later, termed flux balance analysis (Schilling et al., 1999).
2.2. Modeling

2.2.1 Modeling and simulation

Models are abstractions of reality: they are mathematical, logical, or some other structured representation of a real system. They are created from available data on the behavior of the real system, and are usually based on various simplifying assumptions. The process of modeling almost always deliberately emphasizes specific aspects of the reality at the expense of others, taking into consideration the type of problems being researched. For civil engineers, a diagram of a building is a geometric representation of a real building. It captures the spatial relations between walls, doors, stairs etc., while neglecting the color of the wall, for example. This is based on the assumption that when the stability of a building is being examined, the color of the walls is irrelevant. In another example, from Biology, genes are modeled by sequences of letters from the alphabet \{A, T, C, G\}. These letters represent four building blocks of genes. This abstraction emphasizes the different chemical character of each of those building blocks, at the expense, for instance, of their specific chemical structure, polarity, molecular weight, and so on. The assumption that lies here is that when we are interested in the data that genes contain for the formation of proteins, only the sequence of the gene is important.

Simulations are one specific application of models to arrive at some outcome. Simulations allow the validation of a model: a model is suggested based on available knowledge (this model should be consistent with existing data); then, the behavior of the system is simulated in various conditions, and these predictions are validated or refuted by appropriate experiments, as an indication of the reliability of the model. The model is refined and revised this way until an adequate model is obtained. For example, Chen et al. constructed a network for the control of cyclin molecules activities during the budding yeast cell cycle (Chen et al., 2000). They convert this network into a set of differential equations, estimate the various constants that appear as parameters in these equations, and examine the behavior that this model describes in relation to lab experiments. Where there are inconsistencies between the model and experiments, either a better parameter set is provided, or other hypotheses regarding the mechanisms that describe the system are suggested, leading to modifications in the model.

2.2.2. Modeling regulatory networks

Due to the nonlinear character of regulatory networks, and the multiple, simultaneous interactions within them, it is hard to intuitively understand their dynamics. The need to mathematically model these networks emerges in order to integrate the variety of features into a coherent picture. Such computational models are necessary to elucidate qualitative and quantitative aspects of complex biochemical systems. The use of such models may help both for better understanding their behavior as well as for refining open issues. They allow manipulations, and thus allow prediction of different effects on the behavior of the network, and also integration of large gene-expression datasets. Many aspects of regulatory networks can be investigated by formulating them in terms of a computational model. Examples for such aspect are robustness, fail-safety, convergence, stability, transient expression, threshold effects and more. Such models
may predict missing regulatory elements in the network, or enable the examination of
different hypotheses concerning the biochemical pathways in the system, suggesting a
preferred hypothesis. Also, they may focus future research on interesting directions.
The decision on the level of detail and the type of features that a model should
represent is dictated by the characteristics of the biological system being studied, the
type of biological data that is available, and the type of questions we wish to address
through modeling. Different models focus on different aspects and require different
kinds of data for their constructions.

3. Previous work

In recent decades, numerous modeling techniques and analysis paradigms have been
suggested for the elucidation of regulatory pathways, borrowing from a variety of
computational methods (Barkai and Leibler, 1997, Bolouri and Davidson, 2002b;
Schilling et al., 1999, Smolen et al., 2000; Stelling et al., 2002). Such formalisms
allow modeling and simulation of regulatory systems in quite different ways. Some
methods are quantitative, while others are qualitative; some are discrete while others
are continuous; some are deterministic while others are stochastic; also, models can
be static or dynamic, and differ in the level of granularity in which they describe
regulatory processes.

Next is an overview of the main approaches to modeling and simulation regulatory
processes.

Continuous networks
Ordinary or partial differential equations can describe regulatory interactions as
functional and differential relations between elements. In this approach the expression
levels of elements are simulated by continuous (real) variables, and is thus considered
accurate. A set of differential equations is constructed according to the known
regulatory interaction. This set of equations cannot always be solved analytically, and
numerical methods are required to approximate a solution.
Continuous approaches, in addition to insights regarding qualitative characters (such
as stability, robustness, transient signals, etc.) are sometimes appropriate for obtaining
quantitative conclusions on the system studied, such as inferring kinetic constants,
rates of biochemical processes, etc. However, qualitative aspects can also be studied
using continuous approaches. An example for continuous modeling of signaling and
regulatory pathways appears in (Tyson et al., 2003). They create mathematical
modeling of some simple signal–response elements, in the shape of nonlinear
differential equations. They show how certain signals such as positive and negative
feedback and feed-forward loops, can create diverse types of responses: sigmoidal
switches, transient responses, and oscillation.
A technique termed bifurcation analysis, applied to nonlinear continuous networks,
allows determining ranges of values of kinetic parameters that support a specific
behavior of the system. Bifurcation analysis assures us that there are only a few types
of signal–response relationships such as oscillations and irreversible transitions.

Logical-Boolean networks
In this approach, the expression of each gene or protein is assumed to be either ON or
OFF, and intermediate levels of expression are neglected. This is a first approximation
of the levels of expressions of genes. A transition rule determines the trajectory of an element, i.e. its sequence of states (ON, OFF) throughout time. Sometimes, logical combinations that affect the transition rule (AND, OR, NOT, IMPLY) can be assumed between elements in the system. A binary vector, whose length is the number of elements in the system, can describe the state of the network at a specific time. When applying the transition rule step by step, a process which constructs a simulation, any initial vector either converges to a fixed vector, or a set of vectors which form a cycle. Boolean networks allow some interesting qualitative examinations. As an example of qualitative conclusions that can be derived from such models, Li et. al. (Li et al., 2004) conclude that the yeast cell cycle network is robust (not affected dramatically by several small variations in the network) and stable (changes in the state of the system at the onset of the simulation does not alter its fundamental behavior). This is an example for dynamical analysis performed using Boolean networks. In addition, one can use Boolean networks statically to investigate e.g. the correlation between local properties of a network (e.g. a positive feedback loop) and the characters of a large-scale network. However, as will be explained in more detail below, the low resolution of Boolean models with respect to levels of expression is their main limitation: quantitative study is rarely possible, and simulations in such models may produce infinite loops or steady states that are merely artifacts, and do not appear in models with higher granularity.

Stochastic models
Stochastic formalizations of regulatory networks, such as Bayesian networks, have a solid basis in statistics, and enable to deal with stochastic aspects of gene expression and noise in biological systems. Bayesian networks are often used when only partial biological data is available. Zhang et al. (Zhang et al., Elsevier Science, 2006, preprint) used a stochastic model to examine the stability of the yeast cell cycle to noise. Their model is based on the Boolean model of Li et al (Li et al., 2004) (described in detail in Chapter 3), but the difference is that their transition function is probabilistic, and the evolution of the network has the Markov property. They found that the system is stable to levels of noise below a certain threshold, but behaves randomly when the noise increases above that threshold.

Flux balance analysis (FBA)
FBA is based on flux balancing in a metabolic steady state. The fundamental principle underlying FBA is conservation of mass. A set of dynamic mass balance equations is written, such that the balance of flux for each metabolite is preserved. This set of equations is solved as a linear programming problem, to find the optimal flux distribution that minimizes or maximizes a particular objective function (e.g. minimize ATP production, maximize biomass production, etc.). FBA can be used to predict metabolic phenotypes under different growth conditions, by constraining the appropriate fluxes. It can also be used to examine the architecture and topology of a network (i.e. the connectivity of the elements in a regulatory network). Other aspects suitable for analysis with FBA are effects of genetic deletions, and shifts in metabolic routing under various conditions (e.g. loss of functionality of some elements). In Schilling et al. (Schilling et al., 1999) for example, a stoichiometric matrix $S$ of metabolic interactions in E.coli is constructed, in which the $S_{ij}$ element corresponds to the coefficient of the reactant $i$ in the reaction $j$. Once this matrix has been constructed, FBA was used to study properties such as the effect of gene deletions in E.coli. To simulate genetic deletion, the fluxes through reactants associated with the
genes in question are constrained to equal zero. In that research, a surprising number of genes were found to be redundant for growth in glucose minimal media.

**Network motifs**
In this static approach the focus is on finding meaningful structures within the large-scale network. These topological structures, termed motifs, may elucidate a certain behavior or functional characteristic of the system. For example, Mangan and Alon, 2003 (Mangan and Alon, 2003) study the role of the feed forward loop (FFL) in regulatory networks. This is a pattern of three elements, the first regulates the second, and both regulate the third. There are eight possible configurations for the FFL motif, since each relation can either be activation or repression. Their analysis shows that these patterns may act as accelerators for the response of the cell to stimuli, while others delay the response. Moreover, some of the configurations of the FFL motif are more frequent than others, i.e. some of them are selected against for some reason. In general, this approach deals with identifying the basic building blocks of the processes at hand, and employing topological analysis of the underlying graph structure of the networks.

**4. Highlights of results**

We present a novel, discrete computational model that is based on Boolean networks which allows for qualitative analysis of regulatory pathways without the need for detailed quantitative data. It enables the effective examination of characteristics such as transient behavior, robustness, and sensitivity to initial conditions. We have applied our method to analyze the transience and robustness of a representative developmental pathway, namely the transcription of early meiosis-specific genes (EMG) in budding yeast *Saccharomyces cerevisiae*. Some of our analytic observations, such as the pathway's response to premature expression of a key regulator, were validated in vivo. Furthermore, our analysis predicts new modes of regulation by which negative feedback loops accomplish their roles. In general, it provides insights into how to distinguish among alternative hypotheses concerning the function and structure of regulatory networks.

**5. Outline**

The rest of this thesis is organized as follows: In Chapter 2 the yeast EMG system is presented. This was the biological system with which our model was developed and analysis was conducted. In Chapter 3 we present our computational model, and the Boolean model it is based on. Chapter 4 describes the results of various simulations that we did on three networks representing the yeast EMG system. Chapter 5 discusses some predictions achieved with these simulations, and their biological meaning. In Chapter 6 we present a simple user interface we built, that should ease the use of our tool by biologists. This report is concluded in Chapter 7, where some future work issues are mentioned.
Chapter 2: The Early Meiosis-Specific Genes system in yeast

Our model system is meiosis in the budding yeast *Saccharomyces cerevisiae*. Meiosis in diploid cells is initiated upon nitrogen depletion in cells grown with a non fermentable carbon source such as acetate. A transcriptional cascade that consists of a network of many genes governs the initiation and progression through meiosis (Chu and Herskowitz, 1998; Kassir et al., 2003). Fig. 2.1 (adapted from (Kassir et al., 2003) and according to (Pnueli et al., 2004)) illustrates a schematic representation of this pathway. This transcriptional cascade consists of a master transcriptional activator, Ime1, which is essential for the transcription of the early meiosis-specific genes (EMG) (Mandel et al., 1994; Smith et al., 1993). Ime1 is recruited to EMG by the DNA binding protein, Ume6 (Pnueli et al., 2004). Under conditions leading to vegetative growth Ume6 recruits two repression complexes (Isw2 and Rpd3/Sin3). Isw2 complex functions as a chromatin remodeling complex, whereas Rpd3/Sin3 complex deacetylates histones H3 and H4 (Goldmark et al., 2000; Kadosh and Struhl, 1997). When present on DNA Ime1 possesses two activities, it is required to relieve Rpd3 repression, and to activate transcription (Rubin-Bejerano et al., 2004; Washburn and Esposito, 2001). Relief of Rpd3 repression depends on Tyr359 phosphorylation of Ime1 by Rim11, a kinase whose activity is inhibited in the presence of nutrients (Rubin-Bejerano, 2004 #1511). Furthermore, under meiotic conditions Rpd3 is transiently removed from DNA, depending on the kinase activity of Rim15 (Pnueli, 2004 #1352). Rim15 is active only upon nutrients depletion such as meiotic conditions (Reinders, 1998 #641); (Pnueli, 2004 #1352). Ime1 functions as both a positive and negative regulator of its own transcription (Shefer-Vaida et al., 1995; Shenhar and Kassir, 2001). The positive role of Ime1 was indicated by the increase in the expression of *ime1-lacZ* in cells carrying *IME1* on a multicopy plasmid (Shefer-Vaida et al., 1995). Moreover, Ime1 is required to relieve repression of Sok2, a DNA-binding protein that binds to a specific element in *IME1* promoter and represses *IME1*’s transcription in the presence of glucose (Shenhar, 2001 #860). Upon nitrogen depletion the transcription of *IME1* is transiently induced in wild type cells, but not in cells deleted for *IME1*, in which transcription is induced without any decline (Shefer-Vaida et al., 1995). This suggests a negative feedback loop. The direct role of Ime1 in the negative autoregulation was indicated by a specific mutation in *IME1* (*ime1-3*), which is defective in the negative feedback regulation (expression without decline) but is effective in promoting meiosis (Shefer-Vaida et al., 1995). *IME2* is an early meiosis-specific gene encoding a protein kinase (Yoshida et al., 1990). Ime2 functions as a positive regulator of EMG transcription, and as a negative regulator of *IME1* transcription. Moreover, Ime2 affects the stability of Ime1: phosphorylation of Ime1 by Ime2 tags it to degradation by the proteosome, illustrated in Fig. 2.1 as 26S (Guttmann-Raviv and Kassir, 2002; Smith and Mitchell, 1989; Yoshida et al., 1990). It is not known if the negative effect of Ime2 on the transcription of *IME1* is direct, or achieved through its effect on Ime1’s stability.

We have suggested various networks to represent the described regulatory elements and their function. These networks are described in details later in Chapter 4 (Fig. 4.1, 4.2, 4.14, 4.18).
Fig. 2.1. A schematic illustration of the transcriptional cascade governing meiosis in S. cerevisiae. See text for details. The transcription of the middle meiosis-specific genes (MMG) is regulated by two EMGs, the transcriptional activator Ndt80, and the protein kinase Ime2. The transcription of a subset of the MMG as well as NDT80 is also negatively regulated by Sum1, whose stability and/or activity is antagonized by Ime2.

As mentioned earlier, developmental pathways are characterized by transient expression of genes. In Fig. 2.2, the results of Real Time qPCR show the transient expression of the above-mentioned genes IME1 and IME2. An important issue that will be addressed later on, is the interrelation between the expression of these two genes: IME2 rises and falls in delay with respect to IME1, but their expression overlaps. This is not surprising, since IME1 activates IME2, while IME2 is required to shut down transcription of IME1, i.e. there is a negative feedback loop.

Fig. 2.2. Experimental data showing the transient transcription of IME1 and IME2. Biological data courtesy of Vyacheslav Gurevich.
Chapter 3: Method

3.1 The Boolean network model

A Boolean network model was used to describe a portion of the cell cycle pathway (Li et al., 2004) as follows. The pathway is represented by a graph whose nodes are proteins. Each node can be in one of two states: 0 (off) when there is no protein or no protein activity, or 1 (on) which means the existence thereof. The weighted edges of the graph denote regulations; edges can be positive (with a weight of +1), denoting activation, and negative (with weight -1), meaning repression; the absence of an edge means there is no regulation activity that we know of. Self-loops are allowed, usually denoting natural degradation (i.e., their weights are negative). Note that whereas node states must be 0 or 1, edge weights are not intrinsically restricted to +1 or -1 in the model, but these were the specific values used (Li et al., 2004).

The states of the nodes in the pathway, reflecting its overall activity, change over time. To model this discrete time steps are used: $t = 0, 1, 2, \ldots$, and a transition function determines the states of all nodes in time step $t+1$ given their states at time $t$. If $a_{ij}$ is the weight of the edge from node $j$ to node $i$, and $s_i(t)$ is the state of node $i$ in time step $t$, let us denote the scalar product $\sum_j a_{ij} s_j(t)$ as $k_i(t)$, then

$$s_i(t+1) = \begin{cases} 
1, & k_i(t) > 0 \\
0, & k_i(t) < 0 \\
s_i(t), & k_i(t) = 0 
\end{cases}$$

\textbf{Equation 3.1.} Transition function in the Boolean model.

Starting from an initial state, which can be described as a binary vector whose elements are the states of the nodes, the transition function is applied repeatedly until either no more changes are observed (in which case the system is said to have reached a “steady state”), or the network enters an infinite loop of repeated states, in which case it is oscillatory (see example in Fig. 3.1 for such a network; the set of weights on the edges, as shown in that figure, leads to oscillations from 2 of the 4 initial vectors). We call this process a simulation, and the series of states that the network assumes is called a trajectory.

![Diagram](image.png)

\textbf{Fig. 3.1.} A simple network, showing oscillatory behavior, when the initial vector is $(X,Y) = (0,1)$ or $(1,0)$.  

11
3.2 Modeling of the yeast EMG system: Extending the Boolean network model

We used the above Boolean model to analyze the various EMG networks presented later in Chapter 4 (illustrated in Figs. 4.1, 4.2, 4.14 and 4.18). These networks represent the relations between the main “players” regulating entry into meiosis in budding yeast, as described in Chapter 2. Simulation of these networks did not display the transient behavior of the real biological system (as reported in Figure 2.2), and sometimes even resulted in the formation of an infinite loop. Table 3.1 shows the 3 final vectors that were obtained in a simulation conducted on the network in Fig. 4.14. There are $2^6=64$ initial vectors which represent biologically logical states (see Chapter 4 Section 4.3 for details). More than half (57.81%) of the initial vectors lead to a final vector in which the states of IMEI and Imel are 1 (and not 0), indicating that these nodes do not “turn off” and therefore the transient expression is not captured. The trajectories leading to the two other final vectors do not show transient expression either. Thus, when trying to elucidate transient behavior, it seems that two states (0 and 1) are not enough to convey the extent of the phenomenon.

<table>
<thead>
<tr>
<th>State of nodes in the final vectors</th>
<th>IMEI</th>
<th>Imel</th>
<th>IME2</th>
<th>Imel</th>
<th>Ume6</th>
<th>Ume6/Rpd3</th>
<th>X</th>
<th>Y</th>
<th>#initial vector</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>37</td>
<td>57.81%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>10.94%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1.56%</td>
</tr>
</tbody>
</table>

Table 3.1 The 3 final vectors obtained by the simulation of the Network in Fig. 4.14 using the Boolean model on 26–64 initial vectors. 45 initial vectors converged to the 3 final vectors in the table, and the rest of the 19 initial vectors oscillated and did not reach steady state.

Hence we have extended the basic model used by Li et al. (Li et al., 2004) as follows:

- **Nodes’ states.** We allow node states to be any integer number between 0 and $N$ ($>1$). For example if $N=9$, we allow the nodes to be in states 0 to 9 throughout the execution. The selection of a specific value for $N$ will be discussed later. In addition, nodes in our model may represent not only proteins, but also RNA or metabolites.

- **The transition function.** The transitional effect is made cumulative (within the range 0 to $N$) rather than merely pulling the state up to 1, down to 0, or leaving it the way it was. The new transition function is (again, we denote the expression $\sum a_j s_j(t)$ as $k(t)$):

$$s_i(t+1) = \begin{cases} 
\min(N, s_i(t)+1), & k_i(t) > 0 \\
\max(0, s_i(t)-1), & k_i(t) < 0 \\
s_i(t), & k_i(t) = 0
\end{cases}$$

Equation 3.2. Transition function in the extended (discrete) model.

- **Gating transitions by a conjunctive condition (AND).** The activity of a regulator is sometimes enabled only if some other element in the system is active. The opposite may also hold: a regulator may be active only when another element becomes inactive or absent. To model such dependencies we allow the effect of a
node on another node to be dependent on the state of a third node. Formally, we
denote by \( c(i,j) \) the dependency of an edge from node \( i \) to node \( j \) on the state of
node \( k \). We allow two possibilities for this dependency: if the effect of \( i \) on \( j \)
requires that node \( k \) is in state \( >0 \) (active), then \( c(i,j) = 1 \) (positive dependence), if
this effect requires that node \( k \) is in State 0 (inactive), then \( c(i,j) = -1 \) (negative
dependence).

When performing simulations, the choice of the initial state of the network is
important. In the Boolean case a network of size \( n \) nodes has \( 2^n \) states that can all
serve as potential initial states. For small values of \( n \) (e.g. 11 in the case of (Li et al.,
2004)) this number is manageable; when \( N > 2 \), even with a small \( n \), the number of
possible initial states becomes quite large – \((N+1)^n\). Still, the set of interesting and
biologically feasible initial states is often limited to a subspace thereof. For example,
some proteins can only assume certain values as initial states; in most cases it is
indeed enough to look at the initial states 0, 1 (reflecting basal levels) and \( N \)
(representing significant levels of activity).

3.3 Data Analysis

Although the number of feasible initial states of a network may be limited, when
trying to observe the trajectories that lead from initial states to final ones, the number
of intermediate states does increase considerably (since now all \((N+1)^n\) states are
plausible); the number of final states is also potentially very large. In order to analyze
the large variety of outcomes, we look at specific and relevant properties of the
simulations, such as the lengths of the trajectories and the signature of the trajectories
of specific initial states (e.g. rising followed by falling).
Moreover, in order to identify the pattern of the behaviors and establish a functional
dependence between the initial states and their outcomes, we use Karnaugh-like maps
(Kohavi, 1978) to cluster these behaviors in a systematic fashion. This will be
explained in detail and exemplified in Chapter 4.

3.4 Implementation and interface

The implementation of the computational model was written in the programming
language C. The graph describing the network is represented as an adjacency matrix
weight, in which the entry \( (i,j) \) contains the weight of the edge from node \( i \) to node \( j \).
Another matrix, cond, contains the conjunctive conditions (AND) mentioned earlier:
the entry \( (i,j) \) in this matrix specifies the node (if one exists) on which the edge from \( i \)
to \( j \) depends on. The biologically logical initial vectors are kept as a list of logical
initial states (over the set \( \{0, \ldots, N\} \) ) for each node.
The transition function is implemented as a conditioned scalar product calculation: in
each time step, for each node \( j \), the following expression is calculated:

\[
\text{sum}_j = \sum_{i=1}^n \text{state}(i) \cdot \text{weight}(i, j) \cdot \text{AND}(i, j)
\]
where $\text{state}(i)$ is the state of node $i$ at a specific time step of the simulation, $\text{weight}(i,j)$ is the weight of the edge from $i$ to $j$, and $\text{AND}(i,j)$ is the following condition, calculated using the matrix $\text{cond}$:

$$\text{AND}(i,j) = \begin{cases} 
1 & \text{if } \forall k(c_k(i,j) = 0) \\
\text{pos}(\text{state}(k)) & \text{if } \exists k(c_k(i,j) = +1) \\
\text{neg}(\text{state}(k)) & \text{if } \exists k(c_k(i,j) = -1)
\end{cases}$$

and $\text{pos}(x)$ ($\text{neg}(x)$) equals 1 if $x>0$ ($x<0$, respectively), 0 otherwise. Note that in our implementation each edge may be dependent on at most 1 node.

The main loop of the program starts with all the biologically logical initial vectors, and applies the transition function upon each of them repeatedly, until two consecutive vectors are identical, or an infinite loop occurs.

**Pseudo code for the main loop of the program:**

For each ($IV[n]$)  
   // $IV[n]$ = initial vector of size $n$
   trajectory($IV[n]$)  
      // apply transition function repeatedly, thus calculating
      the trajectory of $IV$

End

trajectory($IV[n]$) {
   $IV\_next[n] = IV[n]$
   do
      $IV[n] = IV\_next[n]$
      $IV\_next[n] = \text{apply\_transition\_function}(IV[n])$
      while ($IV[n] \neq IV\_next[n]$ OR $\text{detect\_loop}()$)
   End
}

The function $\text{apply\_transition\_function}(IV)$ applies the transition function on the vector $IV$ as explained earlier. The function $\text{detect\_loop}()$ is a function that determines whether we have entered an infinite loop of vectors.

Since the graph was represented as an adjacency matrix, the time complexity of computing a single transition step for the whole network is $O(n^2)$, where $n$ is the number of nodes. However, representing the graph as an adjacency list improves the complexity to $O(m+n)$, where $n$ is the number of nodes and $m$ is the number of edges (including the ones representing conjunctive conditions). Obviously, the test whether we have reached a steady state (i.e. a situation where the next state is identical to the current one) can be absorbed within this complexity. Theoretically, the number of iteration needed to test for an infinite loop is $O((N+1)^n)$, but practically, in our analyses we used specific properties of our simulations (such as the average length of the trajectories) to detect infinite loops more efficiently (sometimes we detected them manually).

We wrote Visual Basic Macros to allow the automatic drawing of trajectories and histograms of trajectories lengths. In addition, a simple textual user interface was created. This should allow Biologists to easily conduct computational experiments.
using this model, by defining a network of interactions and by controlling the free parameters of the model. The user creates a text file which includes the following data: The value of $N$, the positive and negative thresholds for the transition function, a list of nodes (their labels) in the network, a matrix of weights, a matrix of the initial states allowed for each node, and a matrix of AND gates.
Chapter 4: Evolution of the Network and Results

4.1 Overview

This chapter is divided into three parts, each presents the results of the simulation of a different network: in section 4.2 we describe how and why the initial network illustrated in Fig 4.1 was changed and refined, leading to the network shown in Fig 4.2, and we show in detail the results of the simulations on the latter network. In Section 4.3 we present an alternative network (Fig. 4.14), which improves the fitness of the simulations to the experimental data. In Section 4.4 we describe the final network we examined, depicted in Fig 4.18, which includes the conjunctive condition (AND) feature, an extension to the Boolean model mentioned earlier. This final network was the favorable one, since it captures best the behavior of the real biological EMG system (see Chapter 5 for further analyses conducted with this network).

The fact that throughout our research the networks we have worked with have been modified and refined reveals a basic issue in modeling of biological networks. Indeed, it is – more often than not – unclear which network best describes the biological relations that are being examined. Furthermore, a major issue was to what resolution we should aim with respect to the amount of details in the network: higher resolution (a more detailed network) may have a greater potential in capturing the biological mechanisms that govern the studied processes; on the other hand, simplifying the network may have the advantage that only first order relations are needed to be taken into consideration, and therefore the amount of noise (unreliable, unclear insufficient data) on which the network is based is limited, and the network is more “sterile”.

The process of fitting the network to the biological behavior being modeled is somewhat like a process of evolution: we began with the simplest network we thought was appropriate and consistent with biological data. This was not trivial since many biological details were missing. Then we modified the network, sometimes by extending it, and sometimes by simplifying it, each time examining what we have lost and gained through these modifications, using different mutations in the lab.

One thing is common to all the networks presented here (and some others that we have examined): They all include the 4 nodes that are representatives of the four basic elements in the EMG “arena”. These nodes are IME1, Imel, IME2 and Ime2, which represent IME1 RNA, Imel protein, IME2 RNA, and Ime2 protein, respectively. These 4 elements and the interactions between them are the focus of this research, and they are the ones for which we are specifically interested in learning the mechanisms that govern their expression and/or activity. As a result, we try to model them in greater detail, and to be more exact in the way we formalize their activities and the regulation applied to them. The other nodes present in any of the networks are modeled at the level of detail that is necessary for completing the picture of the main 4 nodes.
4.2 Initial networks

Initial network

Fig. 4.1 shows an initial working hypothesis network that captures the key activities of the EMG pathway. This network was constructed according to parts of the available biological data on the EMG activity, as summarized in Fig 2.1 and described in Chapter 2. There are six nodes in this network: the 4 main nodes (IME1 and IME2, RNA and protein) and two other nodes which represent metabolites that affect the regulation of meiosis – Nitrogen (N) and Glucose (G). In this diagram, green arrows represent activation, red edges with a knob are repressors, and yellow edges reflect intrinsic stability (natural degradation).

![Diagram of initial network]

Fig. 4.1. The initial working hypothesis network describing the relationship between the expression of IME1 and IME2.

When we applied the our model to this relatively simple network, no transient expression was seen, regardless of the specific set of parameters we used (the value of N and the weight on the edges). A possible explanation is that this is a result of the simplicity of the network, or that some important features were missing in it, so we needed to refine the network. The modifications are described below, leading to the network depicted in Fig. 4.2.

Modifications to the initial network

- **Conditions of N and G**
  We are mainly interested in the conditions where Nitrogen and Glucose are absent. In the model, this would be simulated by setting the initial states of nodes N and G to 0, and they will remain in this state throughout the execution of the model since they have no incoming edges. In this case the outgoing edges from N and G have no effect on the other nodes. We therefore removed the nodes N and G from the network, and replaced them by a single node S (for "signal"), which has a green outgoing edge to IME1. This node does not stand for a specific protein or RNA, its role is simply to simulate the signal that exists when cells are shifted
to the meiotic conditions, i.e. absence of nitrogen or glucose. Since our simulations start at that point, the signal node is set to the initial state of 1 ("on"), and stays in this state all the time (since it has no incoming edges).

- **Adding Rpd3**
  Rpd3 is a main negative regulator of IME2’s transcription (see Chapter 2 for details), and therefore we introduced it into the network.

- **Rim11 and Rim15 are constantly “on”**
  We added the nodes Rim11 and Rim15, whose activities are required to relieve Rpd3 repression (see Chapter 2 for details). These proteins are active throughout the process of meiosis, so they were set to the initial values of 1. Since these nodes have no incoming edges (as far as we currently know), they remain in State 1 all the time.

- **Splitting Ime1**
  The node of Ime1 protein was split into two nodes: Ime1 and Ime1* to represent the two functions of Ime1 – transcriptional activation and relief of Rpd3 repression, both of which are required for the transcription of IME2. The latter function depends on the phosphorylation of Ime1 by Rim11.

- **X and Y – "putative proteins/complexes" that are negative regulators for the transcription of IME1 and IME2**
  Biological results suggest that Ime1 is both a positive and a negative regulator for its own transcription. In order to model this effect, we had to add a new node, X, that can simulate several possible mechanisms through which Ime1 represses its transcription: an unknown protein or protein complex that is activated by Ime1 and represses the transcription of IME1, a specific modification of Ime1 that switches its activity from an activator to a repressor, or time delay between the positive and negative effects of Ime1 on its transcription. Unpublished data demonstrates that Ime1 recruits Rpd3 to its own promoter, and that Rpd3 functions as a negative regulator for the transcription of IME1.

  Another node that was inserted into the network is Y. The network assumes that the decline in the transcription of IME2 is through auto-regulation. Because IME2 encodes a kinase, we suggest that its negative role depends on an additional, unidentified protein, designated Y in the network. The importance of this node will be discussed in Section 4.4.
**Results**

We now describe the results we obtained with the network in Fig. 4.2. The parameters used for the basic results are: Nodes’ states can vary between 0 and 9 (i.e. \(N=9\)), and the weights of the edges are +1 for positive edges (green), -2 for negative edges (red), and -1 for degradation edges (yellow). Our reason to assign the weight of -2 to red edges is that in many regulatory mechanisms the effect of repression is stronger and more influential than that of activation on the behavior of the system (for example, a protein is not expressed even when positive regulators take effect until a negative regulator is removed).

**“Legal” initial vectors**

Even though the state of each node in the extended model may be any integer between 0 and \(N\), we restrict the states at the beginning of the simulation to be either 0 or 1. In other words, the initial vectors are binary ones, even though each entry in those vectors may change throughout the simulation to higher values. The biological justification to this restriction is that the RNAs and proteins represented in the network exist in low (basal) levels at the points when nitrogen is removed and entry to meiosis begins (which is the starting point of the simulation). So, binary vectors in the case of the network in Fig. 4.2 represent the biologically feasible initial conditions.

In the rest of this section we show various experiments, in each of which we examined one aspect of the model by changing one of its features, and analyzed its effect on the behavior of the network.

**Basic results**

We have 11 nodes in our network, but 3 of them are degenerate: the source node \(S\), Rim11 and Rim15 remain always in State 1. Thus, we are interested in examining the
behavior of only 8 nodes. The state of the network at a specific time step can be described as a vector of size 8, which consists of the states of the 8 nodes. The order in which they appear in the vector is this:

<table>
<thead>
<tr>
<th>IME1RNA</th>
<th>Imel</th>
<th>Imel*</th>
<th>IME2RNA</th>
<th>Ime2</th>
<th>Rpd3</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
</table>

Table 4.1. A vector represents the state of the network (illustrated in Fig. 4.2) at each time step.

Let us examine some overall statistics of this network’s behavior. Table 4.2 summarizes basic information on the final (steady state) vectors. Near each final vector appears the percent of initial vectors that converged to it, and the average length of the trajectories (sequences of vectors going from initial to final vector) that lead to this final vector.

<table>
<thead>
<tr>
<th>Final vector</th>
<th>% initial Vectors that converged to it</th>
<th>Average trajectory length</th>
</tr>
</thead>
<tbody>
<tr>
<td>00100022</td>
<td>8.59%</td>
<td>4</td>
</tr>
<tr>
<td>00100099</td>
<td>7.03%</td>
<td>17.44</td>
</tr>
<tr>
<td>01200092</td>
<td>7.03%</td>
<td>11</td>
</tr>
<tr>
<td>00100012</td>
<td>6.25%</td>
<td>4.25</td>
</tr>
<tr>
<td>01200091</td>
<td>6.25%</td>
<td>10</td>
</tr>
<tr>
<td>01100014</td>
<td>5.47%</td>
<td>5.86</td>
</tr>
<tr>
<td>00100032</td>
<td>5.47%</td>
<td>4.86</td>
</tr>
<tr>
<td>01100087</td>
<td>4.69%</td>
<td>13</td>
</tr>
<tr>
<td>00100013</td>
<td>4.69%</td>
<td>4.33</td>
</tr>
<tr>
<td>00100023</td>
<td>4.69%</td>
<td>4.33</td>
</tr>
<tr>
<td>00100024</td>
<td>3.91%</td>
<td>5.6</td>
</tr>
<tr>
<td>00100046</td>
<td>3.13%</td>
<td>10.5</td>
</tr>
<tr>
<td>00100021</td>
<td>3.13%</td>
<td>2.75</td>
</tr>
<tr>
<td>03400092</td>
<td>3.13%</td>
<td>10.75</td>
</tr>
<tr>
<td>03400099</td>
<td>2.34%</td>
<td>22.33</td>
</tr>
<tr>
<td>00100098</td>
<td>2.34%</td>
<td>15.67</td>
</tr>
<tr>
<td>00100008</td>
<td>2.34%</td>
<td>12.67</td>
</tr>
<tr>
<td>00100069</td>
<td>1.56%</td>
<td>16.5</td>
</tr>
<tr>
<td>00100047</td>
<td>1.56%</td>
<td>10.5</td>
</tr>
<tr>
<td>00100056</td>
<td>1.56%</td>
<td>11</td>
</tr>
<tr>
<td>04300093</td>
<td>1.56%</td>
<td>10</td>
</tr>
<tr>
<td>01200099</td>
<td>1.56%</td>
<td>21</td>
</tr>
<tr>
<td>01200093</td>
<td>1.56%</td>
<td>12</td>
</tr>
<tr>
<td>00100053</td>
<td>1.56%</td>
<td>5</td>
</tr>
<tr>
<td>00100079</td>
<td>0.78%</td>
<td>14</td>
</tr>
<tr>
<td>00100026</td>
<td>0.78%</td>
<td>9</td>
</tr>
<tr>
<td>00100048</td>
<td>0.78%</td>
<td>11</td>
</tr>
<tr>
<td>00100063</td>
<td>0.78%</td>
<td>7</td>
</tr>
<tr>
<td>00100076</td>
<td>0.78%</td>
<td>10</td>
</tr>
<tr>
<td>00100010</td>
<td>0.78%</td>
<td>1.5</td>
</tr>
<tr>
<td>00100036</td>
<td>0.78%</td>
<td>9</td>
</tr>
<tr>
<td>00100020</td>
<td>0.78%</td>
<td>2</td>
</tr>
<tr>
<td>00100078</td>
<td>0.78%</td>
<td>13</td>
</tr>
<tr>
<td>00100052</td>
<td>0.78%</td>
<td>5</td>
</tr>
<tr>
<td>00100011</td>
<td>0.78%</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 4.2. Basic statistics about the final vectors in the simulation performed on the network illustrated in Fig. 4.2.

We observe that there is a correlation between the final states of X and Y and the lengths of the trajectories: X and Y increase to high-valued states in the longer
trajectories (in the above table, when the average trajectory length in relatively high, X and Y are in states closer to 9), and remain in lower states when the trajectories are shorter. Rpd3 always ends up in State 0. Also, the first 5 entries (which represent the level of expression/activity of IME1 and IME2) are either 0 (IME1 RNA, IME2 RNA, Ime2) or in a low state (Ime1, Ime1*) in the final vectors. This is either the result of transient behavior in which these nodes’ states rise and fall, or simply a decrease in their state without rising (no peak). To distinguish between these 2 possibilities we need to look at the trajectories leading to these final vectors.

Below are several examples (Fig. 4.3 - 4.7) of the behavior of nodes’ states starting with several initial vectors. In these trajectories, the change over time in the states of each node is depicted in a different color.

Let us look in detail at the first example: the initial vector is 00000000, which represents a biological condition in which all the elements in the system are not expressed at the beginning of the simulation. It takes 17 steps for this initial vector to converge, and the final vector is 00100089.

![trajectory of 00000000](image)

**Fig. 4.3.** The trajectory of the initial vector 00000000 in the simulation of the network illustrated in Fig. 4.2. The "p" in Ime1p, Ime1p* and Ime2p stands for protein.

The trajectory in Fig 4.3 shows transient expression of all nodes besides X, Y and Rpd3. The first node to increase its state is IME1 RNA, followed by Ime1, Ime1*, then IME2 RNA and Ime2. Rpd3 does not show transient expression, and stays in State 0 throughout. Finally, the states of X and Y increase to 8 and 9 (respectively) monotonically. The transient behavior of the expression of IME1 and IME2 shown in this trajectory is in agreement with the reported expression as seen in Fig. 2.2.

In the second example with the initial vector 10101100 (Fig. 4.4), in which the length of the trajectory is smaller (only 14 time steps), the behavior is qualitatively similar to the previous example although some changes can be observed: the peaks are lower, the final states are 0 or 1 (besides those of X and Y), and rising start times are earlier in the case of IME2 RNA and Ime2.
In the third example (initial vector 10000010, Fig 4.5), the behavior of IMEI RNA, Imel and Imel* seems even more restrained: their peaks are lower and by time step 6 all three of them are shut down. Here, the peaks of IME2 RNA and Ime2 are higher than those of IMEI RNA and Imel.

In these examples we clearly see transient expression of IMEI and IME2, in both RNA and protein levels. It seems that the longer the convergence path is, the more evident the transient pattern is. Specifically, when this path is longer than 10, some transient pattern is observed. This is the case in −40% of the initial vectors. In the other −60% we see shorter (≤10) trajectories, and no transient pattern.

In the following example (initial vector 01110010, Fig 4.6), the initial vector converges after 9 steps. Ime2 shows transient expression, but this pattern is less clear with Imel: IMEI RNA is always 0, Imel starts at State 1 and decreases without rising, and Imel* has one step of increase before decreasing to 0.
In the final example (initial vector 00111100, Fig 4.7), the path is even shorter (of length 4), and no transient expression can be observed whatsoever.

These were several examples out of the $2^8 = 256$ initial vectors. Clearly, the initial vector determines the shape of the ensuing trajectory. To examine this relation we used Karnaugh-like maps. Each cell in the 5 dimensional matrix in Fig. 4.8 matches a single initial vector, and the number in that cell is the length of the trajectory of that initial vector. Each table matches a different combination of the states of X, Y and Rpd3. Each line in a table matches a different combination of $IME1$ RNA, Ime1, and Ime1*. Each column describes a different combination of $IME2$ RNA and Ime2. For example, the upper leftmost table describes the initial vectors in which Y = 0, Rpd3 = 0, and X = 0. The upper leftmost cell in that table matches the initial vector (0,0,0,0,0,0), whose trajectory was shown before. We painted in yellow cells which correspond to trajectory of length greater than 10, and orange indicates length equal to or shorter than 10.
### Fig. 4.8

Karnaugh-like maps: Visualizing the functional dependence between initial states and the lengths of trajectories of behavior for the network in Fig. 4.2. For easy visualization, cells corresponding to trajectories whose length is more than 10 were colored in yellow, the other were colored in orange.

<table>
<thead>
<tr>
<th>Rpd=0, X=0</th>
<th>lme2</th>
<th>Rpd=0, X=1</th>
<th>lme2</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>17</td>
<td>00</td>
<td>2</td>
</tr>
<tr>
<td>01</td>
<td>17</td>
<td>01</td>
<td>5</td>
</tr>
<tr>
<td>00</td>
<td>23</td>
<td>00</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>lme1 011</td>
<td>11</td>
<td>lme1 011</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>00</td>
<td>10</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rpd=1, X=0</th>
<th>lme2</th>
<th>Rpd=1, X=1</th>
<th>lme2</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>12</td>
<td>00</td>
<td>2</td>
</tr>
<tr>
<td>01</td>
<td>12</td>
<td>01</td>
<td>3</td>
</tr>
<tr>
<td>00</td>
<td>6</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>lme1 011</td>
<td>13</td>
<td>lme1 011</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>00</td>
<td>7</td>
<td>00</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rpd=0, X=0</th>
<th>lme2</th>
<th>Rpd=0, X=1</th>
<th>lme2</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>18</td>
<td>00</td>
<td>2</td>
</tr>
<tr>
<td>01</td>
<td>18</td>
<td>01</td>
<td>3</td>
</tr>
<tr>
<td>00</td>
<td>20</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>lme1 011</td>
<td>13</td>
<td>lme1 011</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>5</td>
<td>00</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rpd=1, X=0</th>
<th>lme2</th>
<th>Rpd=1, X=1</th>
<th>lme2</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>11</td>
<td>00</td>
<td>2</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>01</td>
<td>3</td>
</tr>
<tr>
<td>00</td>
<td>4</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>lme1 011</td>
<td>13</td>
<td>lme1 011</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>4</td>
<td>00</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
A few observations emerge from these tables:

- The convergence is faster when $X-1$ in the initial vector (all the numbers in the right tables are equal to or smaller than their matching cells in the left tables). This is explained by the fact that $X$ is a negative regulator of $IME1$ RNA, and its role is therefore to shut the system down. When $X-1$ in the initial vector, the system shuts down more quickly.
- The level of $Ime1^*$ in the initial vector does not affect the length of convergence (odd lines are almost always equal to the next even lines).
- The length of convergence is hardly affected by $Y$ (the four upper tables and the four lower ones are not very different).

A histogram of trajectories' lengths is shown in Figure 4.9. It supports what was mentioned earlier: it seems that there are two groups of trajectories with respect to their lengths: the long trajectories (length of 9 or more), and the short ones (7 or less).

![Trajectories Lengths](image)

**Fig. 4.9.** Distribution of the lengths of the trajectories for the network in Fig. 4.2.

**Number of node states**

As mentioned earlier, to capture the transient behavior of our network we needed to set the parameter $N$ to a value larger than 2. Now we wanted to examine the effect of the number of node states on the behavior of the simulations.

The trajectories for the initial vector 00010000 is shown below (Fig. 4.10) for $N=9$ (top) and $N=15$ (bottom). A comparison between the two graphs reveals the following differences (until time step 14 they are identical):

- The trajectory length is 25 when $N=15$, instead of 23 when $N=9$.
- The peak of $IME2$ RNA and Ime2 protein is higher when $N=15$ (4 instead of 3).
- It is interesting to see that $Ime1$ and $Ime1^*$ do not reach the maximal state when $N=15$ as they do in the case when $N=9$ (reached State 9). In other words, $Ime1$ begins to decrease before they reached the “ceiling”. This means that they are indeed restrained by other negative regulators, and that
this is not an artifact of the number of states (i.e. the beginning of the decrease is not an artifact of the model).
- Beside X and Y whose final states are higher than before, all other nodes reach the same state in the end.

![Diagram 1](image1)

**Fig. 4.10.** The trajectory of the initial vector 00010000, with $N=9$ (top) and $N=15$ (bottom), in the simulation of the network illustrated in Fig. 4.2.

The examples of other trajectories shown before are identical or very similar if $N=15$ and $N=9$, and thus are not shown.
Fig. 4.11. Distribution of the lengths of the trajectories for the network in Fig. 4.2, with \( N=15 \).

The histogram in Fig. 4.11 also reveals similarity between the behavior when \( N=9 \) and \( N=15 \). Again, there is a separation between the long trajectories and the short ones. Note that with \( N=15 \) the trajectories tend to be longer (average length is higher). The conclusion of these simulations is that the choice of \( N \) does not change the behavior qualitatively.

**Self loops**

In our network, the nodes of Ime2 and Ime1\(^*\) are subject to degradation (yellow self loops). Experiments in the lab show that the levels of Rim11 and Rpd3 are constant whereas that of Rim15 is reduced in acetate (in comparison to glucose) but also remains constant throughout meiosis. Therefore we have not put yellow arrows on these nodes. However, for Ime1, the degradation rate may be similar to that of Ime2 and Ime1\(^*\), and therefore a self-negative loop may be needed there as well. We wanted to check this hypothesis through simulations.

In this section we analyze the results of adding a self loop on Ime1. Fig. 4.12 shows the trajectory for the initial vector 00000000 in the original results in comparison to the new mode in which Ime1 has a self loop. The final vector at steady state is 00100058 and not 00100089 as seen before. The length of the trajectory is 14 and not 17. The main difference in the behavior of the network is that the expression is more restrained with the addition of the self loop. This tendency was observed in other initial vectors as well.
Fig. 4.12. The trajectory of the initial vector 00000000: Original results (top), addition of self loop on Ime1 (bottom).

Below is another example (Fig 4.13) for the initial vector 10000010 which we have also seen before (Fig 4.5). In this example the differences are less prominent, but still the peaks are now lower and narrower in most cases.
From these examples, which represent the effect of adding a self loop on Ime1, it seems that this additional negative regulation causes the shutting down of the whole system more rapidly, and that the conditions for Ime2’s transient expression may not exist.

In general, with the addition of Ime1’s self loop, the lengths of the trajectories are clearly smaller, and the variety of final vectors is also smaller. This is summarized in Table 4.3:

<table>
<thead>
<tr>
<th>Property</th>
<th>Original (self loops are on Ime1* and Ime2) results</th>
<th>Self loops on Ime1, Ime1* and Ime2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average length of trajectories</td>
<td>9.0</td>
<td>6.4</td>
</tr>
<tr>
<td>Number of final vectors</td>
<td>35</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 4.3. Statistics about the effect of adding a self loop on Ime1.
Summary of Section 4.2
The computational results of the simulations on the network in Fig. 4.2, which were described in Section 4.2, are in partial agreement with the biological data, as shown in Fig 2.2. In both the experimental results as well as in the simulations we see transient behavior of IME1 and IME2, RNA and protein, in the expected order (RNA precedes the protein expression, IME1 precedes IME2).
However, a closer inspection reveals that Rpd3 is always at State 0. Therefore this node does not affect any other node, and is thus redundant. Moreover, Rim15 is also redundant, since it affects only Rpd3. In addition, the reason we separated Imel into 2 nodes was to simulate the effect of Imel* on Rpd3. The fact that Rpd3 is always at State 0 weakens the rationale behind this splitting. Therefore, we decided to look for a modification in the network which will exclude the redundant elements from it, and will keep it consistent with biological data.
4.3 Excluding redundant elements

As mentioned above, a simplified network was suggested. This network does not include the nodes Rpd3, Rim15, and Rim11. Also, Ime1 is represented by a single node only. However, there are two new nodes: Ume6 which represents a protein with that name, and Ume6/Rpd3 which stands for the complex that Ume6 forms with Rpd3 (see Fig. 4.14). The two functions of Ime1 mentioned earlier were modeled in a different mode than before. Ime1 has a positive effect on the conversion of Ume6 to an activator, and on the other hand represses the Ume6/Rpd3 negative activity.

![Diagram of network](image)

Fig. 4.14. An alternative network.

Simulations were performed with the following parameters: \( N = 10 \), weights are: +1 on green nodes, -2 on red nodes, and -1 on yellow. At entrance to meiosis, which is the starting point to our simulations, we know that Ume6 exists mostly in a complex with Rpd3. Therefore the initial state of the node representing Ume6 was set to 0, and that of the complex Ume6/Rpd3 to 10. All the other nodes are allowed to be either 0 or 1, as the levels of expression of the elements they represent are basal levels. Thus we have only \( 64 = 2^6 \) initial vectors.

Table 4.4 summarizes the basic information on the final vectors. First, note that the states of IMEI and IME2, RNA and protein in steady state is 0, which means that the simulations fit the biological data that shows that they do not accumulate. Whether they are transiently expressed or not is another question, for whose resolution we need to draw the trajectories. An example of the trajectory of the initial vector 00000a00 (the letter 'a' represents State 10) is shown in Figure 4.15. In this trajectory a clear transient expression is observed.

Note that in this simulation most (80%) of the final vectors were short (Fig. 4.16). This implies that this model describes a non robust system which may not be correct. We next asked if there is a correlation between the lengths of the trajectories and the state of the network in steady state: In the long trajectories the final states of Ume6, X
and Y are relatively high (8, 9 or 10), whereas that of Ume6/Rpd3 is low (0, 1 or 2). In most of the other (short) trajectories, the situation is the opposite in this respect. In Fig 4.16 we see the partition that exists between the initial vectors whose trajectories are long (right hand side, red circle, about 20% of the initial vectors) and short (left hand side, yellow circle, about 80% of the initial vectors). This figure further shows the correlation between the length of a trajectory and its behavior: only the long ones express transient behavior of IME1 and IME2, RNA and protein. This means again that the system is sensitive to perturbations, i.e. initial values. What we need to examine is how the initial vector affects the following characteristics: the length of a trajectory, the form of the final vector, and the transience of the trajectory. For this, again, we draw Karnaugh-like maps (Fig 4.17). The conclusions from these maps are that the system is sensitive to the initial level of Imel protein (premature expression). Furthermore, it is not sensitive to the initial levels of Ime2 and Y. Finally, premature expression of X “shuts down” the system.

The results of our simulations were found to be affected by the value of N. When N=9, we observed that in 2 initial vectors Imel and Ime2 increased without decreasing, and in the corresponding final vectors their state was 9. This phenomenon was not seen with N=10. Besides the fact that most of the initial vectors did not show transient expression, there is inconsistency with experiments even in the long trajectories: there is no overlap between the transient expression of Imel and Ime2, as should be (see Fig. 2.2). This means that Imel’s decrease does not depend on Ime2. Also, biological data suggest that the activation of Ume6 on IME2 is enabled only with the presence of Imel. This led us to the conclusion that in order to simulate this dependency we need to introduce conjunctive conditions into the network, specifically AND gates.
<table>
<thead>
<tr>
<th>final vector</th>
<th>% of source initial vectors</th>
<th>Average length of trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td>IME1</td>
<td>IMe1</td>
<td>IME2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

12 additional final vectors total 64 100% 7.8

Table 4.4 Statistics on the final vectors in the simulation with the network in Fig. 4.14.

![Trajectory of 00000a00](image)

Fig. 4.15 Results of the simulation on the network illustrated in Fig. 4.14. The trajectory of the initial vector in which all state are 0 besides that of Ume6/Rpd3, which is initially set to 10 ("a" in hexadecimal notation).
**Fig 4.16.** The initial vectors can be grouped into 2 types: The ones with long trajectories (length 22 or more) showing clear transient expression, and short ones (11 or less) with no such behavior.

**Fig 4.17.** Karnaugh-like maps: Visualizing the functional dependence between initial states and the behavior of trajectories.
4.4 Introducing gating by conjunction

As a conclusion of the results in the last section, we introduced gating by conjunction (AND) into the network describing EMG, in the form of blue edges going from a node to an edge (see section 3.2 for details). For instance in Fig. 4.18, Ume6 activates $IME2$ only if Imel exists. In addition, there is a blue edge from Ime2 and one from Imel, indicating a hypothesis that Imel and Ime2 depend on each other in their negative role on $IME1$. The rest of this section is dedicated to the results of the simulations performed on the network in Fig 4.18.

We used the model to simulate the EMG network as presented in Fig. 4.18 with the following parameters: $N = 9$, the green and red edges were assigned weights of $+1$ and $-1$, respectively, and $c_{Imel}(Ume6,IME2) - c_{Ime2}(IME1,X) - c_{Imel}(IME2,X) = +1$. As for the yellow edges (self-loops): the one on Imel has weight $-1$, and the weight of the one on Ime2 is $-2$ (for reasons to be explained below).

The following set of initial states was used in the simulation: the initial state of Ume6 was 0, and that of Ume6/Rpd3 was 9; all other six nodes were either 0 or 1. The reasons for this limitation were explained above.

The $64-2^6$ initial vectors converged to 8 different final vectors (attractors); no oscillations were encountered. Table 4.5 shows the full list of the 8 final vectors.

Noticeably, 46 of the initial vectors (72%) converged to the same steady-state, denoted $v_6$, in which both $IME1$ and $IME2$ (RNA and protein) and the Ume6/Rpd3 complex are in State 0; Ume6, X, and Y are in State 9. Moreover, the corresponding 46 trajectories are qualitatively similar: as expected, under starvation conditions our simulation shows transient expression of the genes $IME1$ and $IME2$, their RNA and protein products, with the required overlap between them, i.e. the decline in the expression of $IME1$ RNA and protein is dependent on Ime2. This behavior, a representative of which is shown in Fig. 4.19A, fits the normal behavior of these genes as reported from Northern and Western analysis (Benjamin et al., 2003; Guttmann-Raviv and Kassir, 2002; Kassir et al., 1988; Shefer-Vaiida et al., 1995; Smith and Mitchell, 1989) and as is evident by real time RT-PCR analysis (Fig. 4.19B). Finally, we note that among the 46 initial vectors whose behavior is transient, there is the vector in which all nodes are 0 (i.e. proteins and mRNA’s are absent or
inactive), except for Ume6/Rpd3 which is set to 9 (i.e. the activity of this complex is extensive); this vector, denoted $v_0$, represents initial vegetative growth conditions (see Fig. 4.19A).

<table>
<thead>
<tr>
<th>Final vector</th>
<th>% of source initial vectors</th>
<th>Average length of trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>00009099</td>
<td>71.88%</td>
<td>32.11</td>
</tr>
<tr>
<td>00000910</td>
<td>1.56%</td>
<td>1</td>
</tr>
<tr>
<td>00009111</td>
<td>6.25%</td>
<td>2</td>
</tr>
<tr>
<td>00009035</td>
<td>4.69%</td>
<td>14.33</td>
</tr>
<tr>
<td>00002721</td>
<td>3.13%</td>
<td>4</td>
</tr>
<tr>
<td>00009036</td>
<td>4.69%</td>
<td>14.33</td>
</tr>
<tr>
<td>00002722</td>
<td>3.13%</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 4.5.** Final vectors obtained by the simulation of the network in Fig 4.18.

![Graph of Computed Trajectories and Real-time qPCR Analysis](image-url)

**Fig. 4.19.** Simulation of the network in Fig 4.18 (A) vs. experimental (B) results. In the experimental results, wild type diploid cells (Y 1600) were shifted to meiotic conditions and RNA was isolated at the indicated hours. The level of IME1 and IME2 RNA was measured by qPCR.
Verifying the rationale of the network

The network presented in Fig. 4.18 is based both on observed data as well as on several assumptions (Kassir et al., 2003; Pnueli et al., 2004). The following changes were introduced to the network in order to validate it.

a. Stability of proteins [self-loops and their weights]. The importance of protein degradation in the transient expression of genes was evident for cell cycle genes. In yeast cell cycle, for example, the transcription of $POL1$ and $CLB5$ is periodic, and identically regulated (Lowndes et al., 1991; Schwob and Nasmyth, 1993). Nonetheless, the steady-state level of Poll is constant (Foiani et al., 1995), whereas the level of Clb5 is periodic (Irimiger and Nasmyth, 1997). Therefore, negative self-loops that reflect the intrinsic stability of the proteins in the network are required. Accordingly, the elimination of the self-loop on Ime1 in our network resulted in an infinite trajectory in which both $IME1$ and $IME2$ RNA as well as Ime1 protein were increased with no decline, and Ime2 protein oscillated. When the Ime2 self-loop was eliminated $IME2$ RNA and protein remained high rather than declining to State 0 in about 50% of the final vectors (see Table 4.6). Thus, the simulations suggest that the decline in both $IME1$ and $IME2$ (RNA and protein) requires the negative self-loops on both proteins. Moreover, it suggests that the decline in Ime1 alone is not sufficient for the shutting down of $IME1$ and $IME2$ RNA as well as Ime2 protein.

Since the half life of Ime2 is 5 minutes whereas that of Ime1 is 30 minutes (Guttman-Raviv and Kassir, 2002), the weights on the corresponding self-loops should differ, specifically, we used the value -2 for Ime2, and -1 for Ime1. When the former weight was changed to -1 (i.e. it became identical to all negative edges), $IME2$ expression behaved in a non-transient manner (increased but did not decline) in 72% of the trajectories. These results reflect the different modes by which Ime1 and Ime2 proteins are degraded: namely, degradation of both proteins emerges from their intrinsic stability, but – in addition – Ime1's stability is also regulated by Ime2, and therefore, a -1 value is appropriate.
<table>
<thead>
<tr>
<th>Final vector</th>
<th>% of source initial vectors</th>
<th>Average length of trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>00999099</td>
<td>40.63%</td>
<td>23.85</td>
</tr>
<tr>
<td>10011819</td>
<td>6.25%</td>
<td>10</td>
</tr>
<tr>
<td>00017279</td>
<td>6.25%</td>
<td>10</td>
</tr>
<tr>
<td>00899099</td>
<td>6.25%</td>
<td>22.5</td>
</tr>
<tr>
<td>00010919</td>
<td>6.25%</td>
<td>10</td>
</tr>
<tr>
<td>00012739</td>
<td>6.25%</td>
<td>10</td>
</tr>
<tr>
<td>00021829</td>
<td>6.25%</td>
<td>9.5</td>
</tr>
<tr>
<td>10021819</td>
<td>3.13%</td>
<td>9.5</td>
</tr>
<tr>
<td>20021819</td>
<td>3.13%</td>
<td>9.5</td>
</tr>
<tr>
<td>10010919</td>
<td>3.13%</td>
<td>9.5</td>
</tr>
<tr>
<td>00011829</td>
<td>3.13%</td>
<td>9.5</td>
</tr>
<tr>
<td>00020919</td>
<td>3.13%</td>
<td>9.5</td>
</tr>
<tr>
<td>10020919</td>
<td>3.13%</td>
<td>9.5</td>
</tr>
<tr>
<td>00000910</td>
<td>1.56%</td>
<td>1</td>
</tr>
<tr>
<td>00000911</td>
<td>1.56%</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.6. Final vectors obtained when omitting the self-loop on Ime2 in the network illustrated in Fig. 4.18.

b. The role of Ime1 in the negative autoregulation of IME1 transcription [elimination of X]. The network assumes that the decline in the transcription of IME1 is through autoregulation. This assumption was based on the observation that in cells expressing the ime1-3 allele meiosis is completed, but the transcription of IME1 is non-transient (Shefer-Vaida et al., 1995). Indeed, deleting the node X from the network led to a non-transient behavior in which an infinite loop was obtained. Changing the weights of all the repression (negative) edges to -2 led to only two final vectors, none of the trajectories leading to them being transient. We conclude, therefore, that Ime1 is a key player in the negative feedback loop. Since IME1 encodes a transcriptional activator, we suggest that its negative role depends on an additional, unidentified protein or complex of proteins, namely X.

c. The role of Ime2 in the negative autoregulation of IME2 transcription [elimination of Y]. The network assumes that the decline in the transcription of IME2 is through autoregulation. Because IME2 encodes a kinase, we suggest that its negative role depends on an additional, unidentified protein, designated Y in the network. Deleting Y from the network led to an infinite trajectory; the level of IME2 RNA was increased in a non transient fashion. This result reinforces the conclusion that the transient expression of IME2 is not due to the elimination of its activator.
Validation of the network

Deletion of either IME1 or IME2 results in a non-transient expression of IME1 mRNA (Fig. 4.20A, right). The ability of the model to observe these effects was examined by removing the outgoing edges from either IME1 or IME2. Fig. 4.20A (left) shows that this simulation led to a non-transient expression of IME1, in agreement with the experimental results.

We also examined the effect of premature expression of IME1 on the network’s behavior. To simulate this we let IME1 start in State 3 (instead of 0 or 1). The model predicted normal transient behavior, which differed from the original results in that the expression of both IME1 and IME2 was slightly earlier (Fig 4.20B, left). This result was validated in the lab (Fig 4.20B, right): premature expression was accomplished by deleting the G1 cyclin CLN3 (Colomina et al., 1999), and indeed the observed levels of expression are qualitatively identical to the predictions.

Fig. 4.20. The computational model was validated by experimental results.
A. The transient transcription of IME1 depends on either Ime1 or Ime2. Deletion of IME1 or IME2 results in non transient expression of IME1. Since ORF (Open Reading Frame) was deleted we checked it indirectly by fusing the IME1 5’ region to lacZ and examining β-galactosidase activity.
B. Premature transcription of IME1 results in advanced expression of Ime1 and Ime2 as predicted by the simulation (left) and validated by experimental data (right).
Chapter 5: Discussion and predictions

In this chapter we discuss and analyze several aspects of the results shown earlier for the network in Fig. 4.18. Specifically, we present some alternative networks and the hypotheses behind them, and conclude that indeed the network from Fig. 4.18 is the most probable one.

5.1 Issues concerning the computational model

The maximal value of a state ($N$). With $N=2$ no transient behavior was observed (similarly to the Boolean model, when $N=1$). See Table 5.1A for a list of the final vectors with $N=2$. The majority of the initial vectors (3 first rows in the table) stabilized on final vectors in which the states of $IME1$ and $IME2$ are not all 0. However, transient expression was evident for several larger values that were tested, e.g. $N=3$ (Table 5.1B) and $N=15$ (Table 5.1C). All the final states of $IME1$ and $IME2$ in Tables 5.1B and 5.1C are 0. The choice of $N=9$ is convenient for technical reasons (one-digit values).

| State of nodes in the final vectors | IME1 | IMe1 | IME2 | IMe2 | Ume6 | Ume6/Rpd3 | X | Y | #initial vector | %
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>24</td>
<td>37.50%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>12</td>
<td>18.75%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>10.94%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6.25%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6.25%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4.69%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4.69%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4.69%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3.13%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.56%</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.56%</td>
</tr>
</tbody>
</table>

Table 5.1A: Final vectors obtained by the simulation with $N=2$ in the simulation on the network illustrated in Fig. 4.18.

<table>
<thead>
<tr>
<th>Final vector</th>
<th>% of source initial vectors</th>
<th>Average length of trajectories</th>
<th>Max length of trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>00003033</td>
<td>81.25%</td>
<td>16.48</td>
<td>19</td>
</tr>
<tr>
<td>00000310</td>
<td>1.56%</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>00003111</td>
<td>6.25%</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>00021211</td>
<td>3.13%</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>00003122</td>
<td>4.69%</td>
<td>2.33</td>
<td>3</td>
</tr>
<tr>
<td>00022122</td>
<td>3.13%</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.1B: Final vectors obtained by the simulation with $N=3$ in the simulation on the network illustrated in Fig. 4.18.
<table>
<thead>
<tr>
<th>Final vector</th>
<th>% of source initial vectors</th>
<th>Average length of trajectories</th>
<th>Max length of trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000ff</td>
<td>71.88%</td>
<td>49.11</td>
<td>51</td>
</tr>
<tr>
<td>000000f10</td>
<td>1.56%</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>000000f11</td>
<td>6.25%</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>000000c335</td>
<td>4.69%</td>
<td>17.33</td>
<td>18</td>
</tr>
<tr>
<td>000020d21</td>
<td>3.13%</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>000000f12</td>
<td>4.69%</td>
<td>2.33</td>
<td>3</td>
</tr>
<tr>
<td>000000c336</td>
<td>4.69%</td>
<td>17.33</td>
<td>18</td>
</tr>
<tr>
<td>000020d22</td>
<td>3.13%</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table 5.1C:** Final vectors obtained by the simulation with N=15 in the simulation on the network illustrated in Fig. 4.18.

**Reducing the sensitivity of the transition function.** To determine the state of a node in the next time step, the original transition function compares the dot-product expression \( k_i(t) = \sum a_j s_j(t) \) to 0. Consider this expression to reflect the “strength” of regulation applied to a node. Rather than comparing it to 0, one could allow only “strengths” of regulation higher than \( \text{thresh}_1 \) or lower than \(-\text{thresh}_2\) to take effect. In this case the transition function will be:

\[
    s_i(t+1) = \begin{cases} 
        \min(N, s_i(t) + 1), & k_i(t) > \text{thresh}_1 \\
        \max(0, s_i(t) - 1), & k_i(t) < -\text{thresh}_2 \\
        s_i(t), & -\text{thresh}_2 \leq k_i(t) \leq \text{thresh}_1
    \end{cases}
\]

**Equation 5.1.** Transition function with thresholds.

We tried several values for both thresholds without much of an impact on the results, possibly due to the sparseness of the networks at hand, or because different nodes may have different thresholds, and not a common threshold to all nodes.

**Additive transitions.** Instead of changing a nodes’ state by 1 each time, that change could be dependent on e.g. the value of the scalar product \( k_i(t) = \sum a_j s_j(t) \). One simple way of doing this is to adopt an additive update, as in:

\[
    s_i(t+1) = \begin{cases} 
        \max(0, \min(N, s_i(t) + k_i(t))), & k_i(t) > \text{thresh}_1 \text{ or } k_i(t) < -\text{thresh}_2 \\
        s_i(t), & -\text{thresh}_2 \leq k_i(t) \leq \text{thresh}_1
    \end{cases}
\]

**Equation 5.2.** Additive transition function with thresholds.

We tried this and it did not have much of an impact either, probably for the same reason as above (sparseness).

**5.2. Modeling the negative feedback loop**

Ime2 plays a pivotal role in shutting down the transcription of IME1: in cells deleted for IME2 the transcription of IME1 is non-transient. However, a major unsolved issue is the manner in which this is accomplished: since IME2 encodes a kinase, its mode of
function is not trivial. Since the negative feedback edge from Ime2 to Ime1 is supported by biological data (the phosphorylation of Ime1 by Ime2 targets Ime1 to degradation (Guttman-Raviv and Kassir, 2002)), it should definitely be part of the network. The remaining question, in terms of network topology, is whether the (positive) edge going from Ime2 to X (as in Fig. 4.18) is correct, or can it be removed? Alternatively, may be this edge should be replaced by a negative edge going to IMEI? Another uncertainty concerns the dependence of Ime1 on Ime2 in its activation of X, i.e. whether such dependence exists or not.

We used our model to examine these issues, and suggest a preferred hypothesis. There are six possible networks that represent the combinations of alternatives described above in which Ime2 shuts down IMEI. We denote these networks as Networks 1 to 6 (see Table 5.2, and Fig 5.1), Network 6 being the one shown in Fig 4.18. In Networks 4, 5 and 6, Ime1 is dependent on Ime2 in its effect on X, whereas in Networks 1, 2 and 3 it is not. In Networks 3 and 6 there is a positive edge from Ime2 to X, which is eliminated in Networks 1 and 4, and it is replaced by a negative edge to IMEI in Networks 2 and 5.

**Independent networks**

Network 1

![Network 1 Diagram](image1)

Network 2

![Network 2 Diagram](image2)

Network 3

![Network 3 Diagram](image3)

**Dependent networks**

Network 4

![Network 4 Diagram](image4)

Network 5

![Network 5 Diagram](image5)

Network 6

![Network 6 Diagram](image6)

**Fig. 5.1.** Working hypothesis networks describing the relationship between the expression of IMEI and IME2.
<table>
<thead>
<tr>
<th>Ime1 negative feedback loop depends on Ime2</th>
<th>Network</th>
<th>Positive edge from Ime2 to X</th>
<th>Ime2 has negative feedback only on Ime1</th>
<th>Negative edge from Ime2 to IME1 RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td># of attractors</td>
<td>19</td>
<td>2</td>
<td>16</td>
<td>22</td>
</tr>
<tr>
<td>size of main attractor</td>
<td>34%</td>
<td>34%</td>
<td>28%</td>
<td></td>
</tr>
<tr>
<td>Ime1 function is independent of Ime2</td>
<td>Network</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td># of attractors</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>size of main attractor</td>
<td>72%</td>
<td>62.5%</td>
<td>72%</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: The alternative networks and highlights of their simulation results.

We have applied the model to the six networks, each time on the same 64 initial vectors. All network executions reached final vectors in which both IME1 and Ime2 RNA as well as proteins are absent, in agreement with the reported data. Running the simulations starting with the initial vegetative growth conditions ($v_0$) gave rise to transient behavior that was indistinguishable for the different networks.

Table 5.2 highlights the results in relation to the formation of final vectors. In all cases, most of the initial vectors converged to the very same final vector, $v_f$ (mentioned earlier). Strikingly, in Networks 4-6 (when the activity of Ime1 on X depends on Ime2), between 62.5% and 72% of the initial vectors converged to the abovementioned steady-state, and the number of different final vectors was low (8-10). On the other hand, in Networks 1-3 the number of different final vectors increased significantly, and the portion of initial vectors that converged to $v_f$ was between 28% and 34%. Thus, the dependence between Ime1 and Ime2 increases the robust behavior of the pathway, namely, regardless of the specific initial vector the trajectories reflect the behavior of the biological system (Fig. 2.2).

In order to evaluate the quality of meiosis under abnormal conditions, the number of steps (the length of the trajectory) required to reach a steady-state was used as a primary discriminator, followed by checking the transient behavior displayed by the trajectories themselves. The histograms of these lengths (Fig. 5.2) show two sets of networks exhibiting distinct behaviors: In Networks 4-6, between 40 and 46 of the 64 initial vectors gave rise to long trajectories; careful examination of the actual trajectories revealed that they also showed transient behavior with overlap. Between 0-6 trajectories were moderate in length, and (when examined) showed restrained transient behavior; the remaining 12 to 24 trajectories were extremely short and no meiosis was induced. In contrast, in Networks 1-3 only 18 to 22 trajectories were long and exhibited transient behavior, and the rest were either very short or moderate in length. These results reinforce the conclusion that the dependent networks show a more robust behavior.
We further used Karnaugh-like maps to analyze the functional relation between the initial vectors and their behavior (Fig. 5.3). For simplicity, the maps were painted according to the transience of the trajectories, as follows: white – short trajectories, meiosis is not induced; turquoise – IMEI expression is weak but transient, while IME2 is not expressed or expressed prior to Ime1; yellow – restrained meiosis, weak transience; green – normal behavior. Fig. 5.3 shows a remarkable resemblance among Networks 1-3 (Fig. 5.3A) and among Networks 4-6 (Fig. 5.3B). When IMEI and X are present (even at low levels, i.e. their state is 1) at the onset of meiosis, Networks 1-3 show irregular behavior, i.e. in many cases IME2 is not expressed (marked by turquoise). Moreover, in Network 2, when X is in State 0, premature expression of Ime2 without the expression of IMEI results in a transient induction in IMEI expression, but Ime2 expression is prematurely shut down. This unregulated entry into the meiotic pathway may result in lethality as cells may be arrested infinitely at a specific meiotic stage.

The robust nature of Networks 4-6 in comparison to Networks 1-3 suggests that Ime2 is required to convert Ime1 from a positive to a negative regulator. Current data demonstrates that phosphorylation of Ime1 by Ime2 targets it for degradation. It is possible that the same event also converts Ime1’s function to a repressor, or that the phosphorylation of specific residues is required for these two functions. In order to distinguish between these possibilities, and to confirm that the correct network is indeed one of Networks 4-6, specific mutations in Ime1 that modulate its stability and/or activity should be screened for. The discrimination between Networks 4-6 requires the identification of X and Y, or the demonstration of direct recruitment of Ime2 to IMEI’s promoter.
Finally, following Li et al. (Li et al., 2004) we also looked at the transition graph in which each node is a vector describing a state of the network, and there is an edge from one node to another if the network makes a legal transition between them. Obviously, one component of this graph (see Fig. 5.4) is an inwards tree in which the sink is the final vector \( v_f \) and the sources are a subset of the 46 initial vectors (some initial vectors lie on a path from another source to the sink). We note that this tree, which contains 313 nodes altogether (out of the \( 10^8 \) possible states) for Network 6, is rather balanced in the sense that the paths merge midway between the sources and the sink, indicating that the convergence to \( v_f \) is gradual and there is strong commonality among the 46 trajectories.
Fig. 5.4. The transition graph for the major attractor ($v_j$) of Network 6.
Chapter 6: Conclusions and Future Work

This research proposes a simple and efficient computational model for the elucidation of developmental networks and the results of applying it to the pathway of meiosis in budding yeast. In addition to its ability to faithfully describe the system, i.e. to display transient behavior in such networks, the main advantage of this model is its predictive capabilities to pinpoint missing regulatory elements in the network. We conclude that this model is suitable to the explication of the qualitative behavior of such pathways and that it can be easily adapted to other developmental networks as well. This work gives rise to a rich agenda of further research. A few issues include:

- **Recognition of functional modules.** Note that the subgraph induced by IME1, Ime1 and X is isomorphic to the one induced by IME2, Ime2 and Y. If we could define this subgraph as a module and characterize its behavior as a whole, we might be able to explain the behavior of the network at a higher level of abstraction. Ultimately, this might lead towards an algebra of modules and their interrelations, enabling the analysis of larger and more complex networks.

- **Imposing constraints on the quantities of substances.** Even though our model is not quantitative, it is often desired to reflect the fact that the changes in the concentrations of two substances are interdependent due to conservation of matter throughout the duration of the process. This could be captured by appropriate changes to the transition rules or by assertions on the states.

- **Extending the network.** To include the middle meiosis-specific genes and their regulation by Ime1 and the recombination checkpoint.

- **Refining the model.** To include more features, such as time delay, other combinational logic (such as the AND gate), and specific threshold (for the transition function) for each node.

- **Experimental work.** To identify X and Y, and by this to validate the prediction on their existence.
Bibliography


מידול מודלים של ביטוי ארצי במציאות התחнстית

אמיר רובינשטיין
:CGRectMake על מחקר

לשים מבויו חלקי של הדרישות לקבלי התחור
מגיסטר لمgeois ממדעי המחשב

אמריר רובייטשטיין

ורשה לכנס JsonRequest - מכון טקסטילגרא לאימל
תש"ע"ז דמל 2006
המ timeZone נועש בחותמים פורה/יו, וו פירט ממקומלו של מחזר ת."ה.And/or, write copyright mark.

אני מודע לפורים ואת פירט ופורים, יש קשור של סווהפ אתיי לעולמ macroscopic המmı נושא במשותף

מאגר, פורום או מועי הכל - מחנה.

אנא מודה למא יב' לאאגרד ויקרא שרים (מלגת שרים לשומק בתר תודמה).

The Center for Complexity Science (CCS)

על המוותה המסתי נזרק בחקלאות.
הנקび עיניים

תקציר

קפרים

1. מבוא
2. מוטיבציה
3. קצף
4. בנד
5. מודל הדימוי
6. בדיקות מודלים
7. עבורה קולנוע
8. עקרון הצלחת
9. ראשים פריקס

(EMG) מעריך הנגזרתי הבדליים בשמן בשטח

11.简介
11.1. מודל הפרGetty
12. מדריך מערכות EMG: בושם: הזרמה מודול הבדליים
13. מדריך הפרGetty
13.1. מודול הפרGetty

16. אidders של הזרמה הפרGetty
16.1. הפרGetty
17. התווים הפרGetty
31. הזרמות הפרGetty
35. הזרמות הפרGetty

40. דמיון התוויית
47. מסקנות ביגוד
58. מסקנות ממלוכיות

מִקהַנָּאָּלָאָה
רשימה אientos

1. S. cerevisiae
2. המאובנים המובאים
3. המאובנים
4. המאובנים
5. המאובנים
6. המאובנים
7. המאובנים
8. המאובנים
9. המאובנים
10. המאובנים
11. המאובנים
12. המאובנים
13. המאובנים
14. המאובנים
15. המאובנים
16. המאובנים
17. המאובנים
18. המאובנים
19. המאובנים
20. המאובנים
21. המאובנים
22. המאובנים
23. המאובנים
24. המאובנים
25. המאובנים
26. המאובנים
27. המאובנים
28. המאובנים
29. המאובנים
30. המאובנים
31. המאובנים
32. המאובנים
33. המאובנים
34. המאובנים
35. המאובנים
36. המאובנים
37. המאובנים
38. המאובנים
39. המאובנים
40. המאובנים
41. המאובנים
42. המאובנים
43. המאובנים
44. המאובנים
45. המאובנים
46. המאובנים
47. המאובנים
48. המאובנים
49. המאובנים
50. המאובנים
51. המאובנים
52. המאובנים
53. המאובנים
54. המאובנים
55. המאובנים
56. המאובנים
רשימת טבלאות
3.1 שלושת הוקרות הספירים שיתקבלו בכסמוניות של הרצאה שבאיאר
4.14 באמרת
4.11
4.12
4.2
4.3
4.4
4.5
4.6
4.18
4.18
4.18
4.18
4.15
5.1A
5.1B
5.1C
5.2
**Saccharomyces cerevisiae**

The yeast *Saccharomyces cerevisiae* is a model organism for molecular biology and genetics. Its genome is relatively small, making it easier to study compared to larger organisms. The yeast has a haploid genome, which means it has one copy of each chromosome, simplifying genetic analysis.

The yeast genome contains genes that are essential for viability, such as those involved in cell cycle control, metabolism, and stress response. These genes are conserved across many eukaryotes, making yeast a valuable tool for understanding fundamental biological processes.


אנא בדקר את הירשמה של מדריך הפיתוחottie ובנוסף בהינתן👍 וארפומית המחלקה שלרות תבנית.

rium בинфרא אֶנ איתוס המשותפים בוחרים של מדריך הפיתוחottie. רשת בולאייטס היא יצרנית, שב כל חברת

רייס אניוליס והערת אֶנ איתוס העבריים. ומצעי RNA 👍 וארפומית בוחרים. המחקר והארגוןiniz עקרות קורק. מצא

בעלות משאップ הפרספקטיבה והבעלות משאップ הפרספקטיבה. גורמים מהותיים של המשקלו

מדברים את תParcelable. פסק ה新西ית משותפים עם ה新西ית, ונמצאו כברמ שמות הפיתוחottie 👍 וארפומית.-shadow

אות ביצורים של משטחים Разע ו侴ש. מה המאפיין בעלות משאップ הפרספקטיבה (Oscillation) 👍 וארפומית, בלומרﻊ

מורפזים ביצורים Разע ו与时ש. והמאפיין בעלות משאップ הפרספקטיבה במotonin Students, כמות מרופזים בשתי ימי והמאפיין

נוצר Разע ו与时ש. והמאפיין בעלות משאップ הפרספקטיבה (Robustness) 👍 וארפומית, בציוע אנפר על משטחים של

داعش Разע ו与时ש. והמאפיין בעלות משאﾟ께 הפרספקטיבה (Stability) 👍 וארפומית, בציוע אנפר על משטחים של

משתמש כגרמס החודש Разע ו与时ש, וה妞ות משאפקות שלגר רضرب בתויה וברתן הפיתוחottie על

לאפאנט ימי, אך בדם השמדת אֶנ איתוס הפיתוחottie סיבר ויושמות.
כדי להתקף לאפרת וממלית ישנה, במציעו סילוליצין על רופיאצין של תחושת הבכורה, שמתאימה
ולמתן צעיף ידועו ממרד. בשתי מעצדות חזרו Showing (im1 - Im2) סילוליצין הממלית
הורה המבנה לא אופיינו של Im1 מוקבל שלוש שיוויים בשיערי Im1 פאך Im1
cmpוליציה שלל מיציאות התוקן של Im2 Im1 אנח ממוקמות. Im1

הממס続く לגופים צעמים ההיבנים לשילובים שלמל שלול הממלית. התמקדות ושתייה של Im2 Im1
בנמצאת Im1 ומגוון הדרך של תחנת אחור. Im1 Im2
dים בדרך Im1 וIm2, שתפילים בסרטון תבורי של Im1 Im2

שוכבים לכל מחוז אחין ב-2 מוסר רכיב של המרחבת התמיהדות. Im1 Im2
בנמצאת Im1 Im2 פסגת את העתוןまでの Im1 Im2
mcık צעיף זומרterrorism, או משה מוסר העתון ולא핑בילבוד בחיתת התמה לבר-Novavin1 Im2
הנה מנהיגות האיר פרפדר בפיים בבחינה של התמקדות השורות אגב לא אירוח Im1 Im2
ולתלמי חן Im2 Im1 מיםNovavin2 הבוגרים, מ(mirovא ריכוז פומפרם ב- Im1 Im2

לשתה את פצעדנו על צעיף מדריך לשילובית. התמקדות וה📁ית מ çalışmalar
ל_AMDادي למעון ובנעם: 1)agina פסא ריס בקהחל תחיה מירסער לתמקדות
המותה וארוז,علاج אחר במקצת של מדע מומר אברוס מטר.2) ינתח והằmיתות ב תמקדות
אפרת משלהםו ברגלטרים והתים בהproved.