

A SIMULATION MODELING FRAMEWORK TO STUDY TRANSCRIPTIONAL  
REGULATION THROUGH THE DYNAMIC CHANGES IN THE CONFIGURATION  
OF DNA BINDING FACTORS

by

David A. Knox

B.A., University of Colorado Boulder, 1982

M.S., University of Colorado Boulder, 2010

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Computational Bioscience Program

2015

©2015

David A. Knox

ALL RIGHTS RESERVED

This thesis for the Doctor of Philosophy degree by

David A. Knox

has been approved for the

Computational Bioscience Program

by

Katerina Kechris, Chair

Robin D. Dowell, Advisor

Sonia Leach

Michael Strong

Elizabeth Bradley

Date : 6/15/2015

Knox, David A. (Ph.D., Computational Bioscience)

A Simulation Modeling Framework to Study Transcriptional Regulation  
Through the Dynamic Changes in the Configuration of DNA Binding Factors

Thesis directed by Assistant Professor Robin D. Dowell

## **ABSTRACT**

Transcriptional regulation is the complex system behavior arising from the interaction of numerous regulators with the DNA. The DNA sequence contains the information for this complex system to produce precise gene expression at specific times and cellular locations. Despite the tremendous progress in understanding the behaviors of individual components, we are unable to predict transcriptional behavior from the signal encoded in a DNA sequence.

To better understand transcriptional regulation, we need to construct interpretable, quantitative models of the regulation processes derived from fundamental biological principles. Models need to capture the mechanisms of the individual components and the interactions between components, including the transcriptional machinery itself. Models should be capable of capturing the regulation signal found in any DNA sequence and allow interrogation of the interactions that lead to transcription events.

I have developed modeling frameworks that capture the behavior of individual components, the competition between components for interactions with the DNA, and more importantly, the dynamics of regulatory events occurring within individual cells. I can construct biologically realistic computational models that capture the inherent stochasticity and dynamics of regulatory interactions in simulations and visualize the results of configuration changes occurring in the components bound to the DNA.

The form and content of this abstract are approved. I recommend its publication.

Approved: Robin D. Dowell

## DEDICATION

Many people have inspired and assisted me in this academic journey and I thank them all.

Reaching my goals and dreams would never have been possible without the support and understanding of my wife Robyn. Her dedication to supporting my dreams has always kept me progressing through the most desperate times.

I wish to thank my parents, Bill and Judy, for providing me with the ability, environment, and desire to always question the world and search for new answers.

And finally, I want to thank my grandfather Robert Russell, who showed me that age is never a limitation to learning new things as he continued to learn new things even when he was well into his nineties.

*"The greatest pleasure in life is doing what people say you cannot do."*

*Walter Bagehot 1826-1877*

## ACKNOWLEDGMENT

I would like to thank the various funding agencies that have made this research possible: NSF under Grant DGE 084142 (eCSite) (2012-2013), Chateaubriand Fellowship Ecole Normale Supérieure de Lyon Lyon, France (Feb-May, 2013), and NLM Informatics Training Grant 2T15 LM009451-07 (2013-2015). The summer undergraduate intern funding was provided by NSF under Grant ABI 1262410.

I would like to express the deepest appreciation to my thesis advisor Dr. Robin Dowell, who took me into her laboratory with a spirit of adventure. I thank her for providing me with funding, opportunities for writing grant proposals, and mentoring other students. Without her guidance and persistent help, this dissertation would not have been possible.

I would like to thank my thesis committee members, Dr. Kechris, Dr. Leach, Dr. Strong, and Dr Bradley, for guiding me through the process and providing valuable advice. I also want to thank Dr. Bentley and Dr. Luger for helping me understand the biological processes.

I would also like to thank to Chateaubriand Fellowship for their financial support granted through a predoctoral fellowship that allowed me to work collaboratively with my hosts Dr. Cedric Vaillant and Dr. Alain Arneodo at Laboratoire Joliot Curie, Ecole Normale Supérieure de Lyon, Lyon France.

I would like to thank my classmates Alex Poole, Chris Funk, Meg Pirrung, and labmates Tim Read, Jess Vera, Phil Richmond, and Mary Allen who have helped me learn the intricacies of biological processes and the computational means to evaluate biological data. I thank members of the Computational Bioscience staff that have guided my path throughout the journey. I thank Anis Karimpour-Fard for helping through the tough times and showing me the positives gained in this journey.

I would like to thank Phil Richmond for his assistance in creating the programming class and Joe Rokicki for collaborating with me his visualization project.

The summer internship program provided me with running an academic program and produced useful results. I thank the students: Michelle Soult, Catherine Dewerd, Hayden

Berge on the video project, Chad Bryant, Emily Owens, Malcolm Duren on the animation project, Katy Muhlrad on the Cytoscape project, graduate student David Brazel for voice over on the video, and Daniel Mahlmer for his assistance with the mentoring of summer interns.

Finally I thank Dr. Daniel Dvorkin and Dr. Chris Funk for their version of the LaTeX dissertation template.

## TABLE OF CONTENTS

| CHAPTER   |    |
|---|----|
| I. INTRODUCTION . . . . .   | 1  |
| 1.1 Motivation . . . . .  | 2  |
| 1.1.1 FLO11 Transcriptional Regulation . . . . .                        | 2  |
| 1.1.2 Why Model? . . . . .  | 6  |
| 1.1.2.1 Explain the Behavior vs Predict . . . . .                       | 7  |
| 1.1.2.2 Guide Data Collection . . . . .                                 | 7  |
| 1.1.2.3 Illuminate Core Dynamics Between Components . . . . .           | 8  |
| 1.1.2.4 Demonstrate Trade-offs Between Model or Model Choices . . . . . | 8  |
| 1.1.2.5 Educating Expert and Non-expert Alike . . . . .                 | 8  |
| 1.2 Dissertation Organization . . . . .                                 | 8  |
| 1.3 Contributions . . . . .   | 9  |
| 1.3.1 Contributions of Modeling Framework . . . . .                     | 10 |
| 1.3.2 Contributions of Two-state Nucleosome Model . . . . .             | 11 |
| 1.3.3 Contributions of Visualization . . . . .                          | 12 |
| 1.4 Assumptions and Limitations . . . . .                               | 14 |
| 1.4.1 Assumptions . . . . .   | 14 |
| 1.4.2 Limitations . . . . .   | 16 |
| 1.5 Biological Background . . . . .                                     | 18 |
| 1.5.1 Overview of Transcriptional Regulation . . . . .                  | 18 |
| 1.5.2 Components of Regulation . . . . .                                | 21 |
| 1.5.2.1 DNA . . . . .   | 22 |
| 1.5.2.2 Position Specific Scoring Matrix (PSSM) . . . . .               | 24 |
| 1.5.2.3 Nucleosome . . . . .  | 26 |
| 1.5.2.4 Transcription Factors . . . . .                                 | 27 |
| 1.5.2.5 Transcriptional Machinery . . . . .                             | 29 |



|         |   |    |
|---------|---|----|
| 1.5.3   | Dynamics of Regulation . . . . .                            | 32 |
| 1.5.4   | Single Cell Variation . . . . .                             | 32 |
| II.     | MODELING PERSPECTIVES . . . . .                             | 33 |
| 2.1     | Selecting a Modeling Perspective . . . . .                  | 34 |
| 2.1.1   | Molecular Dynamic Perspective . . . . .                     | 36 |
| 2.1.2   | Interaction Network Perspective . . . . .                   | 36 |
| 2.1.3   | Regulatory Dynamic Perspective . . . . .                    | 37 |
| 2.2     | Selecting a Modeling Method . . . . .                       | 38 |
| 2.2.1   | Equation-based Modeling Methods . . . . .                   | 40 |
| 2.2.2   | Statistical-based Modeling Methods . . . . .                | 42 |
| 2.2.3   | Chemical Reaction Modeling Methods . . . . .                | 45 |
| 2.2.4   | Stochastic Models at Single Nucleotide Resolution . . . . . | 46 |
| 2.2.5   | Agent-Based Modeling Method . . . . .                       | 48 |
| III.    | DYNAMIC TRANSCRIPTION MODELING FRAMEWORK . . . . .          | 50 |
| 3.1     | Introduction . . . . .                                      | 50 |
| 3.2     | Contribution . . . . .                                      | 52 |
| 3.3     | Previous Work . . . . .                                     | 53 |
| 3.4     | Methodology . . . . .                                       | 56 |
| 3.4.1   | Framework . . . . .   | 56 |
| 3.4.2   | Representation . . . . .                                    | 57 |
| 3.4.3   | Implementation . . . . .                                    | 59 |
| 3.4.3.1 | Stochastic Rule Builder . . . . .                           | 59 |
| 3.4.3.2 | Visualizing Simulations . . . . .                           | 63 |
| 3.4.3.3 | Coping with Parameters . . . . .                            | 63 |
| 3.4.3.4 | System Overview . . . . .                                   | 65 |
| 3.4.4   | Modeling Details . . . . .                                  | 69 |
| 3.4.4.1 | States in the Model . . . . .                               | 70 |
| 3.4.4.2 | Unbound Nucleotide State . . . . .                          | 71 |

|         |   |     |
|---------|---|-----|
| 3.4.4.3 | Transcription Factor Bound States . . . . .                             | 71  |
| 3.4.4.4 | Nucleosome Bound States . . . . .                                       | 73  |
| 3.4.4.5 | Transcription Machinery Bound States . . . . .                          | 74  |
| 3.4.4.6 | Actions within the Model . . . . .                                      | 75  |
| 3.4.4.7 | Transcription Factor Actions . . . . .                                  | 76  |
| 3.4.4.8 | Nucleosome Actions . . . . .  | 76  |
| 3.4.4.9 | Transcription Machinery Actions . . . . .                               | 76  |
| 3.4.5   | Validation . . . . .  | 76  |
| 3.5     | Results . . . . .   | 78  |
| 3.5.1   | Case Studies . . . . .  | 78  |
| 3.5.2   | Capturing Positional Information (GAL10) . . . . .                      | 79  |
| 3.5.3   | Capturing Positional Information (CLN2) . . . . .                       | 80  |
| 3.5.4   | Capturing Temporal Interaction (IME4) . . . . .                         | 83  |
| 3.5.5   | Tractability . . . . .  | 86  |
| 3.5.6   | Complexity . . . . .  | 87  |
| 3.6     | Discussion . . . . .  | 89  |
| 3.7     | Conclusion . . . . .  | 93  |
| IV.     | DYNAMIC NUCLEOSOME MODEL . . . . .                                      | 94  |
| 4.1     | Introduction . . . . .  | 94  |
| 4.1.1   | Components of Transcriptional Regulation . . . . .                      | 94  |
| 4.1.2   | Steady State Modeling of Transcriptional Regulation . . . . .           | 96  |
| 4.1.3   | Using Biologically Inspired States to Make Better Predictions . . . . . | 99  |
| 4.2     | Contribution . . . . .  | 99  |
| 4.3     | Methodology . . . . .   | 100 |
| 4.3.1   | Extending the COMPETE Model . . . . .                                   | 100 |
| 4.3.2   | Evaluating the Accuracy of Model Predictions . . . . .                  | 101 |
| 4.4     | Results . . . . .   | 102 |
| 4.4.1   | Genome Wide Analysis of Single State vs Two State Nucleosome Models     | 102 |

|       |  |     |
|-------|--|-----|
| 4.4.2 | Analysis of CLN2 Promoter Region . . . . .                         | 105 |
| 4.5   | Discussion . . . . .   | 106 |
| 4.6   | Conclusion . . . . .   | 111 |
| V.    | VISUALIZATION AND EDUCATION . . . . .                              | 112 |
| 5.1   | Introduction . . . . .   | 113 |
| 5.2   | Contribution . . . . .   | 115 |
| 5.3   | Educational Videos . . . . .                                       | 117 |
| 5.3.1 | Inverted Classroom . . . . .                                       | 117 |
| 5.3.2 | Transcriptional Regulation Video . . . . .                         | 118 |
| 5.4   | Visualization of Simulation Results . . . . .                      | 120 |
| 5.4.1 | Character Graphics . . . . .                                       | 121 |
| 5.4.2 | Animations . . . . .   | 122 |
| 5.5   | Discussion . . . . .   | 123 |
| 5.5.1 | Visualizing the Interaction Network . . . . .                      | 123 |
| 5.5.2 | Automated Conversion of Graphic Representations for Abstract Rules | 123 |
| 5.5.3 | Syntax and Semantics for Abstract Rules . . . . .                  | 128 |
| 5.5.4 | CodaChrome - a Proteome Conservation Visualization Tool . . . . .  | 129 |
| 5.6   | Conclusion . . . . .   | 129 |
| VI.   | CONCLUSION . . . . .   | 133 |
| 6.1   | Modeling of Transcriptional Regulation . . . . .                   | 133 |
| 6.2   | Visualization and Education . . . . .                              | 136 |
| 6.3   | Future Work . . . . .  | 137 |
| 6.4   | Final Remark . . . . .   | 139 |
|       | REFERENCES . . . . .   | 140 |
|       | APPENDIX   |     |
| A.    | PETRI NET GRAPHS . . . . .   | 149 |

## LIST OF TABLES

Table

|      |  |     |
|------|--|-----|
| 1.1  | Examples of transcription factor binding motifs . . . . .                | 30  |
| 1.2  | RNA types and function . . . . .   | 31  |
| 1.3  | Polymerase used to generate different types of RNA . . . . .             | 31  |
| 3.1  | Modeling parameters for Stochastic Rule Builder . . . . .                | 66  |
| 3.2  | Modeling parameters for Stochastic Rule Builder (cont) . . . . .         | 67  |
| 3.3  | Nucleotide States . . . . .  | 72  |
| 3.4  | States of Transcriptional Machinery bound DNA . . . . .                  | 75  |
| 3.5  | Actions of Transcription Factors . . . . .                               | 76  |
| 3.6  | Actions of Nucleosomes . . . . .   | 77  |
| 3.7  | Actions of Transcriptional Machinery . . . . .                           | 78  |
| 3.8  | Actions of Transcriptional Machinery Interference . . . . .              | 78  |
| 3.9  | Typical run time and memory usage for GAL10-GAL1 models . . . . .        | 79  |
| 3.10 | Non-default parameters used for GAL10-GAL1 models . . . . .              | 80  |
| 3.11 | Typical run time and memory usage for CLN2 models . . . . .              | 83  |
| 3.12 | Non-default parameters used for CLN2 models . . . . .                    | 83  |
| 3.13 | Typical run time and memory usage for IME4 models . . . . .              | 86  |
| 3.14 | Non-default parameters used for IME4 models . . . . .                    | 86  |
| 3.15 | Model size is dependent on components . . . . .                          | 88  |
| 3.16 | Component attributes that influence the span of some actions . . . . .   | 88  |
| 3.17 | Complexity analysis for transcription factor abstractions . . . . .      | 89  |
| 3.18 | Complexity analysis for nucleosome abstractions . . . . .                | 89  |
| 3.19 | Complexity analysis for transcriptional machinery abstractions . . . . . | 90  |
| 4.1  | Correlations for different models . . . . .                              | 104 |
| 4.2  | Correlation values for Two State vs One State Nucleosome Model . . . . . | 104 |

## LIST OF FIGURES

| Figure |  |
|--------|--|
| 1.1    | Regulation of the FLO11 gene . . . . . 4   |
| 1.2    | The Central Dogma of Biology . . . . . 19  |
| 1.3    | Transcriptional Regulation depends on the state of the DNA . . . . . 20            |
| 1.4    | Representation of a DNA segment with a gene and a non-coding transcript . . 23     |
| 1.5    | TF affinity can be scored for any sequence by using the TF's PSSM. . . . . 25      |
| 1.6    | Nucleosome formation dynamics . . . . . 28   |
| 1.7    | Chromatin packaging . . . . . 29   |
| 2.1    | Modeling Development Cycle . . . . . 34  |
| 2.2    | Modeling Perspectives . . . . . 35   |
| 2.3    | Differential Equation example model . . . . . 41                                   |
| 2.4    | Positional information described by Hidden Markov Models . . . . . 43              |
| 2.5    | Biological behaviors are modeled by computational components . . . . . 45          |
| 3.1    | Biological behavior can be modeled by action oriented local descriptions. . . . 57 |
| 3.2    | Simultaneous transitions at multiple states of an HMM . . . . . 58                 |
| 3.3    | Petri net description of transcription factor binding . . . . . 59                 |
| 3.4    | Flowchart depicting the Stochastic Rule Builder and visualization pipeline. . . 60 |
| 3.5    | The Stochastic Rule Builder (SRB) generates a set of biochemical rules . . . . 61  |
| 3.6    | Biochemical interaction rates are a function of the specific sequence. . . . . 62  |
| 3.7    | Visualizing the DNA configurations at each time step of a simulation. . . . . 64   |
| 3.8    | Each DNA position influences a limited number of rules. . . . . 68                 |
| 3.9    | Model size grows linearly with the size of the DNA sequence being modeled. . 69    |
| 3.10   | Nucleosome formation takes multiple states . . . . . 74                            |
| 3.11   | Nucleosome occupancy of GAL10-GAL1 region. . . . . 80                              |
| 3.12   | Known nucleosome depleted region at CLN2 . . . . . 81                              |
| 3.13   | IME4 transcription regulation. . . . . 85  |

|      |  |     |
|------|--|-----|
| 3.14 | Runtime of the SRB application . . . . .   | 91  |
| 4.1  | HMM states and transitions for individual nucleotide position of DNA . . . . .   | 97  |
| 4.2  | HMM states describing binding to transcription factors and nucleosomes . . . . . | 98  |
| 4.3  | Histogram of Two State Nucleosome Model correlation values . . . . .             | 105 |
| 4.4  | Histogram of Single State Nucleosome Model correlation values . . . . .          | 106 |
| 4.5  | Experimental nucleosome occupancy data . . . . .                                 | 107 |
| 4.6  | Nucleosome occupancy for One and Two state nucleosome models . . . . .           | 108 |
| 4.7  | One state, Two state, and Experimental nucleosome occupancy . . . . .            | 109 |
| 5.1  | Video screenshots . . . . .  | 119 |
| 5.2  | ASCII visualization of a single time point . . . . .                             | 121 |
| 5.3  | Visualizing the DNA configurations at each time step of a simulation. . . . .    | 122 |
| 5.4  | Animation of the component configuration over time along a DNA segment . . . . . | 124 |
| 5.5  | Pipeline for creating simulation results . . . . .                               | 125 |
| 5.6  | Sample abstract Petri net for the unbinding of factor from DNA . . . . .         | 126 |
| 5.7  | Model definition includes variables at each level of description . . . . .       | 127 |
| 5.8  | Model definition syntax . . . . .  | 128 |
| 5.9  | The CodaChrome Graphical User Interface . . . . .                                | 130 |
| A.1  | Transcription Factor Bind. . . . .   | 150 |
| A.2  | Transcription Factor Unbind. . . . .   | 151 |
| A.3  | Transcription Factor Unbind by Watson strand Transcriptional Machinery . . . . . | 152 |
| A.4  | Transcription Factor Unbind by Crick strand Transcriptional Machinery . . . . .  | 153 |
| A.5  | Recruit Crick strand Transcriptional Machinery Upstream . . . . .                | 154 |
| A.6  | Recruit Watson strand Transcriptional Machinery Downstream . . . . .             | 155 |
| A.7  | Nucleosome Binding and Unbinding . . . . .                                       | 156 |
| A.8  | Nucleosome Stabilization and Eviction . . . . .                                  | 157 |
| A.9  | Nucleosome Linker Maintenance: Bound - Binding . . . . .                         | 157 |
| A.10 | Nucleosome Linker Maintenance: Bound - Binding with Linker . . . . .             | 158 |
| A.11 | Nucleosome Linker Maintenance: Binding - Binding . . . . .                       | 159 |

|   |     |
|---|-----|
| A.12 Nucleosome Linker Maintenance: Binding - Binding with Linker . . . . .           | 159 |
| A.13 Nucleosome Linker Maintenance: Bound - Bound . . . . .                           | 160 |
| A.14 Nucleosome Linker Maintenance: Bound - Bound with Linker . . . . .               | 160 |
| A.15 Transcriptional Machinery Initiation and Eviction . . . . .                      | 161 |
| A.16 Transcriptional Machinery Initiation stages and transition to Elongation . . . . | 162 |
| A.17 TM movement one position . . . . .   | 163 |
| A.18 Transcriptional Machinery aborting from transcribing or transcribed . . . . .    | 163 |
| A.19 Terminating and eviction of Transcriptional Machinery . . . . .                  | 164 |
| A.20 Collision of two elongating Transcriptional Machinery . . . . .                  | 164 |
| A.21 Collision of elongating and initiating Transcriptional Machinery . . . . .       | 165 |
| A.22 Elongating Transcriptional Machinery evicting a Nucleosome . . . . .             | 166 |

## CHAPTER I

### INTRODUCTION

Transcriptional regulation is the system behavior arising from the interaction of numerous regulators with DNA. This complex system produces precise gene expression at specific times and locations. Experimental studies of gene expression have unlocked the function of many proteins involved in regulating the transcription process (Bai et al., 2011; Bradley et al., 2010; Darzacq et al., 2007; Farnham, 2009; Hahn and Young, 2011; Lickwar et al., 2012b; Mack et al., 2012; Mirny, 2010; Palmer et al., 2011; Segal et al., 2006; Venters et al., 2011). New experimental techniques are being developed to understand transcriptional regulation at unprecedented temporal and molecular detail, ultimately even at single-cell resolution (Galburt et al., 2009; Larson et al., 2011; Levsky et al., 2002; Taniguchi et al., 2010). Yet much is still to be learned. There is growing evidence that transcription emerges not solely from the individual components, but rather from the collective behavior (including competition and cooperation) between the components (Larson et al., 2011; Sanchez et al., 2011; Segal and Widom, 2009a; Struhl and Segal, 2013; To and Maheshri, 2010; Wasson and Hartemink, 2009; Zeevi et al., 2011). Three major classes of protein regulators, transcription factors, nucleosomes, and the transcriptional machinery, interact with DNA in both a competitive and cooperative fashion. DNA undergoes millions of interactions every second, constantly changing the configuration of the molecular components bound. It is the stochastic, temporal, and spatial interactions of these regulators that controls the transcription process in each individual cell.

Encapsulating our understanding of these interactions into a computational model is integral to understanding transcriptional regulation (Lander, 2010). Models allow us to explore a system, create testable hypotheses, and identify when key details are missing in our current knowledge. To date, most modeling frameworks have either focused on the detailed molecular behavior of a single specific regulator or the interaction of a small subset of regulatory components (Barnes et al., 2011; Cantone et al., 2009; Greive et al.,



2011; Kim and Gellenbe, 2012; Lubliner and Segal, 2009; Ribeiro, 2010; Segal et al., 2006). Yet, few models have approached the problem of simultaneously capturing the behavior of all three major regulators. In part, this is because most models either focus on the positional information of each component, the binding locations along the DNA (Segal et al., 2008; Wasson and Hartemink, 2009), or the temporal behavior of their inherent dynamics (Ribeiro et al., 2009; Roussel and Zhu, 2006). Integrating both the positional information and temporal information often leads to computationally expensive models. As experimental techniques continue to improve, modeling approaches must also evolve to represent increasingly realistic molecular details while still remaining computationally tractable. We need new methods to construct biologically realistic computational models that capture not only the positional binding of transcription factors and nucleosomes, but also the underlying temporal dynamics, such as the behavior of transcriptional machinery during initiation and elongation.

Therefore, I have developed a new modeling framework that can automatically generate rule sets describing the possible molecular interactions implied by a given DNA molecule, extended a current state-of-the-art positional method to capture some dynamics of nucleosome formation, and explored methods for visualizing the stochastic and dynamic behavior of the complex system known as transcriptional regulation.

## 1.1 Motivation

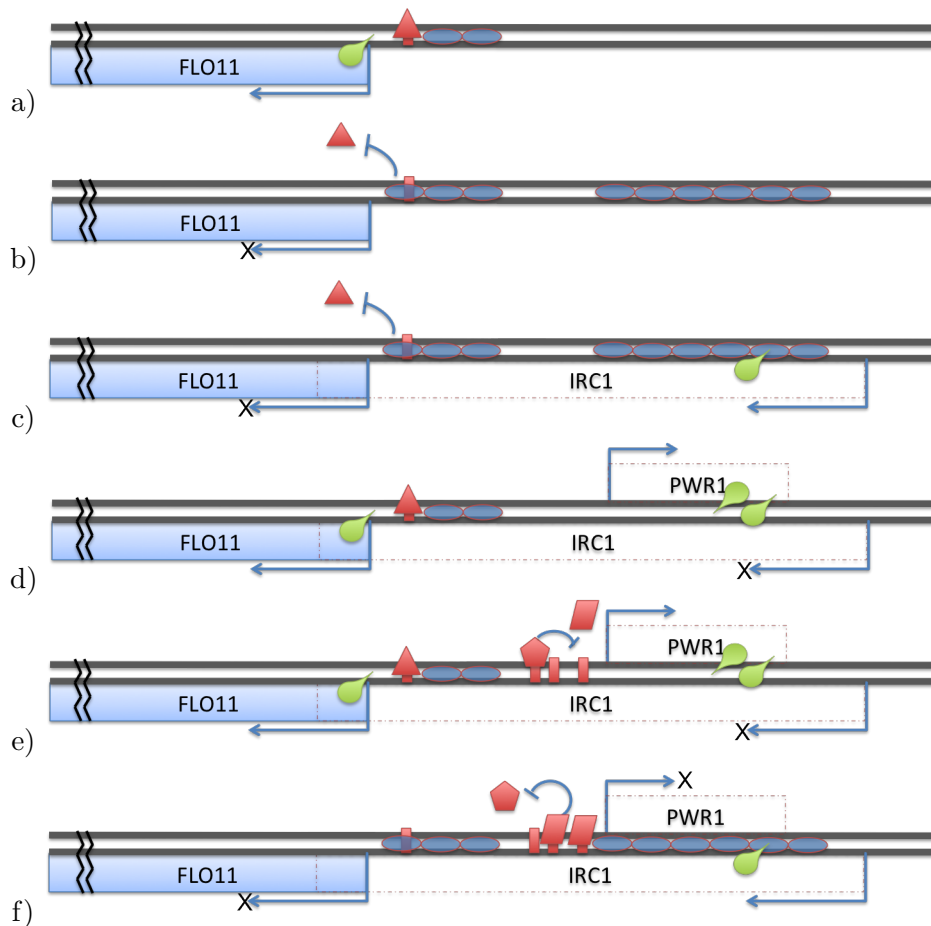
### 1.1.1 FLO11 Transcriptional Regulation

The motivation for this work derived from studies of the regulation of transcriptional switches. These are regions in the genome with complex regulation that depend not only on the individual components, but also the complex cooperation and competition between multiple independent components. For example, in yeast (*Saccharomyces cerevisiae*), regulation of the well studied Flo11 transcriptional switch (Bumgarner et al., 2012; Hongay et al., 2006) is dependent not only on transcription factor binding, but also the transcription process itself. The regulation mechanisms used in these switches are also found in higher eukaryotes, including human, and the same mechanisms may apply to the recently

discovered enhancing RNAs (also known as eRNA) that have been shown to regulate the genes in which they reside (Kornienko et al., 2013).

The transcriptional regulation of the FLO11 gene is very complex, but that complexity is gained through a collection of simpler regulatory mechanisms (Bumgarner et al., 2012). FLO11 is activated by transcription factors binding in its promoter region (Figure 1.1-A). However, access to the FLO11 promoter DNA is restricted by nucleosome formation at the specific TF binding sites. Nucleosome formation thereby suppresses transcription (Figure 1.1-B). The nucleosome binding pattern in this region is maintained by the transcription of a non-coding RNA, Interfering Crick RNA (ICR1), across this region shown in Figure 1.1-C. Presumably factors bound to the DNA are removed as the transcriptional machinery traverses the DNA, which leads to an environment where all the factors compete to rebind after the machinery moves on. In this competition, nucleosomes are the most likely structure to form as the concentration of histone proteins is very high compared to other factors. To allow expression of the Flo11 transcript, a second non-coding transcript, Promoting Watson RNA (PWR1), must be transcribed to interfere with ICR1 transcription through transcriptional interference (Figure 1.1-D). Transcription of PWR1 prohibits ICR1 transcription through the promoter region of FLO11, thereby allowing key transcription factors to eventually access the promoter DNA of FLO11 and initiate transcription. To further complicate matters, the PWR1 transcript is regulated by the competition between two transcription factors (SFL1 and FLO8) that bind in the PWR1 promoter. Stochastic interactions determine which transcription factor binds first, setting the local configuration, which determines if PWR1 transcription occurs and ultimately determines when transcription of FLO11 can occur. (Figures 1.1-E & F).

Experimental data shows that either PWR1 or ICR1 is being transcribed in a cell, but not both (Bumgarner et al., 2012). Once a cell is in one of these states of expression, it stays in that state until a seemingly random event causes the transcription to switch to the other state. The competition for the promoter of PWR1 is dependent on the concentration of the transcription factors and the stochastic behavior of their binding. The experimental



**Figure 1.1: Regulation of the FLO11 gene.** a) FLO11 is activated by binding of transcription factors. The red triangle represents a transcription factor bound on the DNA and the blue oval shows DNA bound within a nucleosome. The green teardrop represents transcriptional machinery directionally processing along the DNA strand it is touching. b) Repression occurs when nucleosomes block access to a TF binding site. The red rectangle representing the transcription factor binding site is occluded by nucleosome formation. c) The repressing nucleosome configuration is maintained by the transcription of a non-coding transcript (ICR1) which is constitutively on at low levels and therefore inhibits FLO11 transcription. d) To allow transcription of FLO11, the non-coding ICR1 must be inhibited. This is accomplished by transcription of a second non-coding transcript PWR1 which interferes with ICR1 transcription through transcriptional interference. e) Transcription of PWR1 is regulated by competition between two factors. When one factor binds it will activate PWR1 which interferes with ICR1 allowing FLO11 to be activated. The two transcription factors, shown as a parallelogram and pentagon, compete for the overlapping binding sites. When the pentagon binds, occluding the parallelogram, PWR1 transcription is activated. f) When the other ICR factor binds in the promoter of PRW1, it inhibits the transcription of PWR1, allowing ICR1 to transcribe and keep FLO11 inhibited.

data shows, in rich media, that the FLO11 repressing state is most likely. When the activating factor out competes inhibiting factor at the PWR1 promoter, the cell switches to transcribing PWR1 and will stay in that state until unbinding of the activation factor. ICR1 is always being transcribed and when the transcriptional machinery escapes the interference of PWR1, the activating factors in the PWR1 promoter will be unbound. At that time, the competition of the transcription factors at the PWR1 promoter will determine the next configuration and the state of the transcriptional switch.

These type of switches are difficult to study via molecular biological experimentation as their state fluctuates seemingly at random. Most experimental techniques measure the behavior of a population of cells, not the behavior of an individual cell. At any specific time point, some cells within a population express the ICR1 ncRNA and a separate set of cells are expressing the PWR1 ncRNA and FLO11. Unfortunately, the switching is stochastic, meaning even if the cells in one state were sorted, by the time the segregation is completed, many of the cells will have switched states (private comm with R. Dowell). This provides an environment where computational models may be the only reasonable method to explore the systems behavior of regulation.

Traditionally, transcriptional regulation models have captured the steady-state behavior of competition and cooperation in cell populations (Figures 1.1 A,B,E, & F) using hidden Markov models (HMM) (Segal et al., 2006; Wasson and Hartemink, 2009) or Ordinary Differential Equations (ODE) (Sanchez et al., 2013). The mechanism of transcriptional interference (Figure 1.1-D) has been described with ODEs (Sneppen et al., 2005) and the behavior of the transcriptional machinery has been modeled using rule based systems (Ribeiro et al., 2006). These methods have been successful in describing the regulatory behavior within a population of cells. However, they are not designed to capture all the spatial and temporal behavior of an individual cell at the FLO11 locus. The competition of transcription factors and nucleosomes for DNA is captured by these methods, but the transcriptional machinery passing through the same regions will alter the probabilities of remaining in that state. This dynamic change in the transitions cannot be captured easily in

these methods. The transcriptional interference that is required by this switch is temporally dependent on the locations of the transcriptional machinery.

To explore and understand the regulation mechanisms of these switches, a new computational modeling method must be developed. These new models must capture the independent behavior of each component, as well as capture the combined behavior of the whole system. The system behavior will be compared to the experimental data to understand which behaviors of the individual components are required to achieve the system level regulation. Understanding this regulation requires a systems modeling method that captures not only the spatial aspects, but also the temporal aspects of the system because the temporal and spatial events can have a profound effect on the other positions of the DNA and effect the behavior of the entire system, as seen in the FLO11 example.

### **1.1.2 Why Model?**

Knowledge does not imply understanding. Often we have data that is confusing or is contrary to the data that has been collected before. As experimentation techniques continue to collect more, better, and higher resolution data, our ability to understand this new complex data often becomes increasingly more difficult. Yet, we can use all the data as facts about the behavior of the system.

How can we create understanding of all this factual knowledge? When new concepts are being discovered, in our minds we implicitly create simplified models of the complex processes and intuitively analyze the data to “see if it fits” that model. When new data does not conform to the expected results, it will often lead to new ideas and a new understanding of the behavior of the system. Computational models explicitly define the assumptions and behavior the model components so we can communicate the models’ behaviors, quantify how well the data matches the model, allow colleagues to replicate the results, and explore the range of component behaviors or parameters to identify the robustness or sensitivity of model perturbations (Epstein, 2008).

When enough knowledge has been collected, we can specify a model accurately enough to predict the behavior of the system. While many people think all models must be able

to accurately predict behavior to be useful, these models become available only after we deeply understand the system and have collected the relevant knowledge. As new concepts are being explored for the first time, most of the relevant data is not available and we create models that simplify the complex processes to match the data acquired thus far.

There are many reasons to build models even when accurate prediction is not possible. Models can be used to explain behavior, guide data collection, illuminate core dynamics between components, demonstrate trade offs between modelling choices, and educate both the non-expert researcher and the general public (Epstein, 2008).

#### **1.1.2.1 Explain the Behavior vs Predict**

Models can be used to explain how a behavior arises from the components and their interactions. The models can be used to learn and understand the behavior without being able to predict when those interactions will occur. The standard example is that we understand the reasons that earthquakes occur (plate tectonics), but we are still unable to predict when they will occur. The same is true in biology, we understand and model how the flu virus works, but we cannot accurately predict the strain that will affect us this year. Most of what we learn is via models, which help us understand, even when it will not allow us to accurately predict the future.

#### **1.1.2.2 Guide Data Collection**

Developing models is an iterative process of collecting data, formation of a model, running that model, and comparing the results to the collected data. When creating a new model, we may identify the knowledge that we are missing and can guide the direction of the next round of experimentation. An example of model guiding data collection can be seen in the search for Higgs boson. For many years a model in modern particle physics envisioned a previously unseen particle. The Higgs particle is an elementary subatomic particle hypothesized by a group of theoretical physicists. The model was continually improved over the years, driven by both theoretical and experimental particle physicists working together. The Higgs boson was recently discovered only because there was an active billion dollar search for the particle. Another example of models guiding the experimentation is

the bending of light by gravitation that was proposed by the general relativity model, which also was later confirmed by experimentation.

### **1.1.2.3 Illuminate Core Dynamics Between Components**

Models are always an abstraction of the real underlying system, but because they focus on the essential features and hide the unnecessary details for understanding the system behavior, they are useful. The famous quote by George Box, “essentially, all models are wrong, but some are useful,” shows that even when a model is inherently wrong, it can still be used to understand how the different components interact. The fact that a model does not match the data is in itself helpful for guiding the experimental data that needs to be collected. In silico experiments allow the behavior of components to be manipulated in order to search the alternative possibilities for thresholds, robustness, and sensitivity of the model and its components.

### **1.1.2.4 Demonstrate Trade-offs Between Model or Model Choices**

Competing models can be compared to see which model performs better. We can use the areas where the two models differ to understand why and maybe find a third model that combines the best of both competitors.

### **1.1.2.5 Educating Expert and Non-expert Alike**

The actual systems being modeled are too complex for most people to comprehend without extensive study. However, a model that is not technically correct can be used to convey the concepts to the non-expert researcher or to the general public. How many of us actually understand the General Relativity Theory or the Standard Model in theoretical physics? Yet, when given a simpler model of the concepts, we are able to obtain a basic understanding of the behavior it is describing.

## **1.2 Dissertation Organization**

In this work, I describe the modeling perspective and methods (Chapter II), the design, implementation, and validation of a modeling framework (Chapter III), an extension to a current state-of-the-art hidden Markov model to include dynamics of nucleosome formation

(Chapter IV), and visualization of the spatial and temporal results of the framework's models (Chapter V).

The rest of this chapter provides an overview of all the contributions in this work (1.3), a set of assumptions and limitations for this work (1.4), and a section describing the biological concepts used throughout this work. (1.5). Anyone familiar with the mechanistic concepts of transcription can safely ignore this background section as I quickly introduce the relevant concepts in each chapter.

### 1.3 Contributions

To address the problem of modeling the systems behavior of transcriptional regulation at a single cell resolution, I needed to address two main issues: Building models and interpreting the results of model simulations or analysis. The first issue addresses how to create a model that could capture the spatial and temporal behavior of a system, while remaining flexible enough to capture the behavior for any arbitrary DNA sequence. To this end, I have created a modeling framework that captures these features within a biochemically inspired, rule-based description of individual components and combines them into a single simulation model. I have also explored an alternative approach, namely, adding some of the dynamics inherent in nucleosome formation to a state-of-the-art positional model.

The second issue focuses on how to convey information between the computer and the human user. This includes both the specification of component behavior or kinetics parameters, as well as conveying the results of model simulations or analysis. I have designed a method to visualize specific component interactions with the modeling framework as a graph or more specifically, as a Petri net. I have created a visualization of the DNA configuration at each time point within a simulation and produced animations from those results. Finally, I have also directed the creation of a short video to introduce the dynamic and stochastic nature of transcriptional regulation that can be used in undergraduate courses.

Below I have listed the contributions for all chapters of this manuscript. These contributions are reiterated within their individual chapters to maintain each chapter's autonomy.



### 1.3.1 Contributions of Modeling Framework

Part of this work is currently in the review process for IEEE Transactions on Computational Biology and Bioinformatics as “A modeling framework for generation of temporal and positional simulations.” Additionally, this work comprises Aim 1 of a National Science Foundation grant (ABI 1262410) on which I was a co-author.

- **I created a modeling framework to automatically generate a model comprised of a collection of biochemical based rules describing the individual behavior of each model components for any given sequence of DNA.** The generated models capture the details of interactions at nucleotide resolution, which allows the behavior of individual cells to be captured. The models can describe both steady state processes, such as transcription factor binding, and dynamic processes, such as the transcriptional machinery moving along the DNA. These generated models capture both the spatial and temporal behavior of the system being modeled.
- **My framework allows the models to not only capture the population averaged steady-state behavior, but also capture the dynamic behavior of individual components, as well as the emergent behavior arising from the components working together in a coordinated system.** My framework focuses on modeling at the nucleotide level within single cells. Instead of providing a single population averaged result as current modeling methods produce, I can capture the progression over time through different configurations of factors bound to the DNA in a single cell.
- **The interactions between components can be specified using spatially abstract descriptions.** The abstract descriptions of each component interaction is described independently of the actual DNA sequence or positions. Each interaction description is applied to each position of the given DNA sequence to generate the model. Details of the interaction, such as an interaction rate, can be tied to a function based on the local sequence at each position. Each abstract interaction can be described using a graphical form.

- **The DNA sequence is not a single molecule or component, but many interdependent nucleotide components.** The framework is designed to consider each individual nucleotide as its own component. Each nucleotide is not completely independent, as interactions that occur at a neighboring nucleotide have a large effect on the behavior on other nearby nucleotides. However, the extent of the effects are limited in scope at any specific nucleotide position and therefore makes the simulation of interactions along a large DNA segment trackable.

### 1.3.2 Contributions of Two-state Nucleosome Model

A Manuscript covering part of this work is currently in preparation: “Dynamic Nucleosomes in a Steady State Model”. This work was supported by a Chateaubriand Fellowship awarded to David Knox in 2012 for working collaboratively with Laboratoire Joliot Curie, Ecole Normale Supérieure de Lyon, Lyon France.

- **Extended a state-of-the-art positional model to include some of the dynamics of nucleosome formation by adding multiple nucleosome states and transitions.** The classical nucleosome is formed by eight histone proteins stably binding to ~147 nucleotides of DNA. However, there are additional stable intermediate formations (Luger et al., 2012). Pairs of histones H3 and H4 are bound first to form a core nucleosome and then pairs of other histones (H2A and H2B) combine with the core to capture the additional DNA of entry and exit arms and form a stable canonical nucleosome. I have enhanced a state-of-the-art positional model from using a single nucleosome state to one including both a core and full nucleosome state.
- **The two state nucleosome model correlates with the experimental nucleosome occupancy better than the single state nucleosome model across whole chromosomes.** The enhanced model showed an increased correlation of genome wide nucleosome occupancy values between simulated and experimental data.

### 1.3.3 Contributions of Visualization

This work comprises part of Aim 2 in a National Science Foundation grant (ABI 1262410) on which I was a co-author. The video was entered into National Science Foundations' Visualization Contest in 2014 ([www.nsf.gov/news/special\\_reports/scivis](http://www.nsf.gov/news/special_reports/scivis)). The visualization of proteome conservation was published in BMC Genomics: Rokicki, J., Knox, D., Dowell, R. D., and Copley, S. D. (2014). "CodaChrome: a tool for the visualization of proteome conservation across all fully sequenced bacterial genomes" (doi:10.1186/1471-2164-15-65).

- **Produced a short introductory video to describe the stochastic nature of the transcription process and the dynamics of the transcription process to be used as teaching material for undergraduate introductory biology courses.** There are two concepts that this video uniquely addresses: the stochastic nature of the transcription process and the dynamic behavior of the transcriptional machinery that contributes to transcriptional regulation. This video explains the basic concepts behind transcription as a necessary biological process that is the first step in creating proteins in a cell. It visualizes the complex interactions of transcription factors, nucleosomes, and the transcriptional machinery with the DNA, including the often ignored mechanism of transcriptional interference. The video was created in conjunction with a summer internship program for undergraduate Computer Science students (Michelle Soult, Catherine Dewerd, Hayden Berge) and submitted to the National Science Foundations' Visualization Contest in 2014 ([www.nsf.gov/news/special\\_reports/scivis](http://www.nsf.gov/news/special_reports/scivis)).
- **Created a language to abstractly describe component interactions as graphs.** The Petri net graphs represent the component states and state changes that occur with each interaction. The language describes the syntax and semantics for abstract templates and variable substitutions used to generate multiple related interactions for each abstraction. This allows complex models to be generated from less complex abstract interactions.

- **Created an ASCII visualization of configuration of components bound to the DNA at each time step of a simulation.** Every time point of the simulation provides a snapshot of the configuration of factors bound to the DNA, which can be used to reveal patterns of behavior not seen in summary results. From the output of the model simulations, I generate an ASCII visualization of the configuration of factors bound to each nucleotide of the DNA at each time step. The state of each position of the DNA is uniquely represented as a single character linearly in a line of text. The movement of factors along the DNA can be inferred by the movement of factor positions in consecutive display lines.
- **Created an animation of the component interactions based on the intermediate simulation results.** The simulation results can be interpreted as a script for component movement. Each time point specifies the configuration of components along the DNA. By inferring the movement of components between time points, a trajectory for individual components can be calculated. A visualization framework, provided by Unity, was used to manage the virtual environment and display of individual component movements. The goal of the animations was to provide visual feedback on the cellular behavior based on the modeling parameters. Ultimately, the animations would be used in a teaching tool for students studying transcriptional regulation. The animations were created in conjunction with a summer internship program for undergraduate Computer Science students (Chad Bryant, Emily Owens, Malcolm Duren).
- **Mentored a fellow graduate student (Joe Rokicki) in the creation of a tool for the visualization of proteome conservation among bacterial genomes.** The relationships between bacterial genomes are complicated by rampant horizontal gene transfer, varied selection pressures, acquisition of new genes, loss of genes, and divergence of genes, even in closely related lineages. As more and more bacterial genomes are sequenced, organizing and interpreting the incredible amount of relational information that connects them becomes increasingly difficult. CodaChrome is

a user-friendly and powerful tool for simultaneously visualizing relationships between thousands of proteomes recorded in GenBank. The relationships between a bacterial proteome of interest and the proteomes of every other bacterial genome are visualized as a massive interactive heat map. published in BMC Genomics: Rokicki, J., Knox, D., Dowell, R. D., and Copley, S. D. (2014). “CodaChrome: a tool for the visualization of proteome conservation across all fully sequenced bacterial genomes” (doi:10.1186/1471-2164-15-65).

## 1.4 Assumptions and Limitations

Every modeling system makes general underlying assumptions and includes a set of limitations. While many of these assumptions are common knowledge, they must be embedded into the computational models to generate the behavior described. Therefore, in this section I have explicitly listed many of the assumptions and limitations of this work.

### 1.4.1 Assumptions

- **Each component behaves independently of other components.** The behavior of each component is modeled as an independent molecule. Complexes must therefore be defined separately as independent components. It is assumed that binding of a factor with the DNA is independent of other factors binding elsewhere along the sequence beyond the direct binding positions. However, it is known that binding of factors will change the DNA structure through bending or torsional forces changing the width of major and minor grooves along the DNA helix. These behaviors can be modeled as occlusions of the DNA extending beyond the direct component binding positions.
- **The configuration of components regulates the transcriptional events.** It is my contention that every component exhibits simple behavior, such as binding and unbinding, similar to the states in Wasson and Hartemink (2009). Activation and repression occur because of the configuration of the components along the DNA sequence. In one configuration the probability of transcription is greater than in other configurations and is interpreted as the activating configuration.

- **More resolution of DNA components (nucleotide level) will elucidate the mechanisms used to regulate transcription.** Most of the current modeling methods for transcriptional regulation are focused on explaining “which” genes are affected by changes in transcription of a particular gene. They have been migrating towards models that also use the transcription factor binding within regulatory modules (located in the promoter regions of genes) to enhance the model. My work is focused on asking the question of “how” the change in a factor’s concentration can affect transcription of another gene. This means finding the mechanisms used for regulation. These mechanisms are directly related to the DNA sequence and therefore the models must be able to be defined at nucleotide resolution. However, there are many interactions that may affect the regulation that are still completely hidden even at this resolution. Chromatin and DNA modifications (epigenetics) are some of the details that are still being ignored.
- **Each simulation represents a cell in an quasi-steady state.** Meaning that concentrations (or molecule counts) of components do not vary during the simulations. Each quasi-steady state will have many different patterns of factors bound to the DNA. By collecting the distribution of the time spent in each configuration, I gain insight into the underlying dynamics. Changing the initial molecule counts for the model simulation will create different distributions of the configurations that occur.
- **Sequence affinity (motif descriptions) can be used to approximate the strength of binding for any component.** The affinity of each component is used to determine the on-rate and the off-rate for interactions with DNA. The interaction rates are calculated by applying the position specific scoring matrix to the local sequence at each position. It is assumed that the closer the sequence matches the motif, the more likely binding will occur at that position. It is also assumed that better motif matches imply a longer residency time for the factor bound to the DNA.

- **I have selected yeast (*Saccharomyces cerevisiae*) as the model organism.**

Modeling transcriptional regulation in bacteria is simpler than in eukaryotes, although in bacteria the transcription and translation processes can occur simultaneously, which adds additional interactions affecting the behavior of the translational machinery. In bacteria, the simultaneous transcription and translation has an immediate effect on the local concentrations of the transcription factors driving the transcription. Eukaryotes isolate the transcription and translation processes in separate compartments. There is a pseudo steady-state environment in eukaryote transcription regulation until the cell can translate the mRNA and move the proteins back into the nucleus. Eukaryotes also contain additional factors for maintaining their larger genomes. Bacteria lack the nucleosomes and chromatin structures found in eukaryotes. As I wanted to create a framework that could be extended to model higher eukaryotes, I selected a well-studied and relatively simple eukaryote genome of *Saccharomyces cerevisiae*.

There are many data sets providing data that can be used to describe the individual component behavior. Transcription factor motifs for many of the over 150 known transcription factors are available. There are many well studied loci within the genome that provide me with regions for validating my generated models (see Hahn and Young (2011) and Rando and Winston (2012) for reviews on transcriptional regulation).

#### 1.4.2 Limitations

- **Conversion of graphical representations is currently a manual process.** The interactions of all the components are described in Appendix A. I used these conceptual graphs to build the code that generates rules. This is currently a manual process that must be applied for changes or additions of components to the model. Section 5.5 describes the work I have begun to automate this process.
- **Many modeling parameters are required.** Most of these parameters are currently unknown (or at least unmeasured), so values must be estimated from current knowledge. The models are therefore not capable of accurately predicting the total

system behavior. The interactions of the components and the relative behavior can be explored using the models generated by the framework.

- **Current implementation is tied to the stochastic simulator and its reaction syntax.** Although it has been widely used as a stochastic simulation engine, DIZZY is a simple implementation that is not designed for the large set of rules and reactants that are generated by the framework. DIZZY uses a simple and straightforward syntax, however the design of the framework application could be extended to produce the more complex syntax of other modeling description languages. These other languages can be used in many different simulation engines which would extend the size and complexity of models that could be simulated. DIZZY was selected because of its easy access and usage. DIZZY requires large amounts of RAM to store all the data in memory until simulation is completed. This limits the scalability of the model simulations.
- **The stochastic simulation engines are not designed to efficiently handle the sparse rule sets.** The stochastic simulation algorithm (Gillespie, 1976) has been implemented in a number of simulation engines. These engines were designed to handle dense networks of interactions. My modeling generates very sparse networks as most rules only interact with a few components and there is only one molecule shared across many component states. This means that many rules transition between active and inactive as each interaction is applied. The current engines consider the entire model as a single system, but the events occurring along one segment of DNA are independent of events at other locations. The current engines do not support parallel simulation of DNA segments across multiple processors to allow modeling large human genomes. To handle the sparseness of active rules and scaling up to human genomes may require the design of a new simulation engine.
- **The Mediator complex and its interactions are ignored.** Mediator is a large complex of many different molecules and is found at or near most transcription sites.



However, its exact role and behavior is still being explored. As we gain knowledge of its behavior, it can be added into the components of the modeling framework.

- **The behavior of chromatin remodelers are ignored.** There is a class of transcription factors that can actively change the configuration of factors bound to the DNA. Chromatin remodelers are known to evict or move nucleosomes to less favorable locations. These factors are important players in the regulation of transcription. While the behavior of simultaneous movement of multiple components is easily described in my modeling framework, the when and where for remodeler binding is still unknown and therefore are excluded from the framework.
- **The modifications to histone tails are ignored.** The histones that comprise the core proteins of a nucleosome have tails that wrap around the DNA. There are many post-translational modifications, such as methylation, acetylation, phosphorylation, ubiquitination, etc, that can change their interaction with DNA. Some modifications stabilize the nucleosome, while others will destabilize the nucleosome. Although many of the modifications are associated with transcription, most of the rules for how these marks change the nucleosome behavior are still poorly understood. Likewise, DNA methylation, another epigenetic mechanism, is ignored.

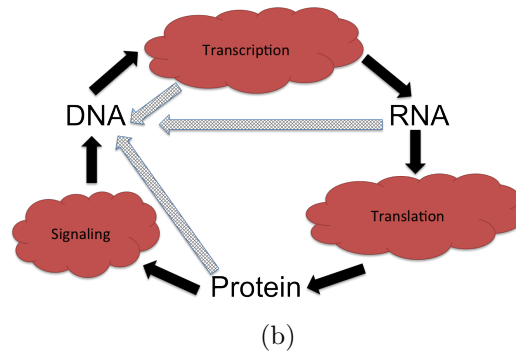
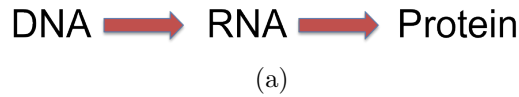
## 1.5 Biological Background

This section reviews the key concepts and components of transcriptional regulation. Section 1.5.1 provides an overview of the transcription process. The interactions among key factors involved in regulation are described in Section 1.5.2. Transcription is an inherently stochastic process, as highlighted in Section 1.5.3. Finally, in Section 1.5.4, I discuss the state of the art experimental techniques for single cell biological experiments that highlight the need for a new modeling approach.

### 1.5.1 Overview of Transcriptional Regulation

All living things on this planet are made up of cells, which, despite the great diversity between them, all behave using the same basic mechanisms (Alberts et al., 2007). The common processes that support life have been distilled into the central dogma of Biology:

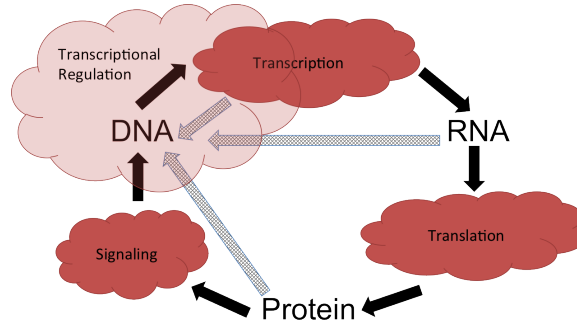
DNA is used to produce messenger RNA that in turn are used to produce proteins (Figure 1.2-a). The information for cellular behavior is stored in DNA and can be replicated and passed on to offspring. Transcription is the biological process of reading a DNA template and producing an RNA copy of that template. Translation reads the RNA to produce the proteins that are required to maintain all cellular activities.



**Figure 1.2: The Central Dogma of Biology as seen from viewpoint of Biologist and Computational Biologist.** a) The central dogma of Biology embodies the notion that a DNA template is used to create the RNA copies that are used to create the proteins used in biological processes. b) In Computational Biology, the focus is changed to the processes, where DNA, RNA, and proteins are the inputs and outputs of the processes. Transcription is the process that takes the current DNA state as input and produces an RNA as the output. The translation process converts the RNA input and outputs a set of amino acids that form the protein represented by the RNA. Many of the proteins do not immediately interact with the processes, but are sequestered away from the processing environment, waiting until the cell detects an internal or external signal that causes the proteins to return to the processing environment and change its behavior. Current biology textbooks usually show the dogma as a linear process, but we show it as a circular set of processes that feedback upon each other. The processes are shown as clouds and the solid black arrows show the inputs and outputs of the processes. The grey hatched arrows depict the influence on the DNA state by the proteins, the RNA products, and even the transcription process itself.

Biological textbooks usually show the central dogma as a linear process from DNA to protein (Figure 1.2-a). The arrows represent the processes to convert between the different types of molecules. If we take an alternative view of the dogma, which focuses on the processes, the DNA, RNA, and proteins are now the inputs and outputs of the processes. In figure 1.2-b, I have redrawn the dogma as a circular process that includes feedback from

other processes as input to the transcription process. The grey hatched arrows depict the influence on the DNA state by proteins, the RNA products, and even the transcription process itself.



**Figure 1.3: Transcriptional Regulation depends on the state of the DNA.** Transcription only occurs when the DNA is in the correct state. Modeling of transcriptional regulation must include all the influences on state of the DNA. There are different methods by which the DNA state can be modified. Some proteins are known to influence the transcription of genes and are collectively referred to as transcription factors. Some of these factors immediately influence regulation, while others are sequestered away until the cell receives signals to allow them to influence the DNA state. There are RNAs that directly interact with the DNA and modify the DNA state. The transcriptional machinery is a large RNA complex that directly interacts with the DNA, changing the DNA state as it transcribes along the DNA sequence. My framework is flexible and can encompass all of these influences and apply them to a given DNA sequence to build a model of the system of biochemical interactions.

To produce cellular response and activity, the information encoded in the DNA must be read and converted into functional molecular machines. Some proteins regulate the transcription process, creating a feedback loop. A detailed molecular understanding of regulation is one of the major interests of biology. Understanding how transcription is regulated, namely when (temporal) and where (positional) RNA is produced, is the underlying goal of transcriptional modeling systems.

There are a number of regulation mechanisms to control the resulting concentration of any protein. Regulation of the transcription process is the first, followed by regulation of post-transcriptional processes, transportation of resulting RNA, translation, and the degradation of the protein. In this work, I focus on the transcription process and its regulation. The regulation of transcription is the resulting behavior of a complex system of interactions between a number of different components (Figure 1.3).

### 1.5.2 Components of Regulation

The transcription process in all species uses fundamentally similar mechanisms and requires the coordination of hundreds of different molecules. To transcribe a region, the DNA is first bound by factors that reconfigure the DNA structure and allow formation of the transcriptional machinery complex. RNA polymerase is composed of many different proteins that together constitute an active cellular machine to transcribe DNA into an RNA copy. The transcriptional machinery separates the strands of the DNA helix and proceeds to read one strand. The machinery transcribes along that strand until a signal indicates the end of transcription and starts the disassembly of the machinery into individual molecules.

Transcriptional regulation is the system behavior arising from the interaction of numerous regulators with DNA and the transcriptional machinery complex. The complex system of transcriptional regulation produces precise gene expression at specific times and locations. Experimental studies of gene expression have unlocked the function of many proteins involved in regulating the transcription process (Bai et al., 2011; Bradley et al., 2010; Darzacq et al., 2007; Farnham, 2009; Hahn and Young, 2011; Lickwar et al., 2012b; Mack et al., 2012; Mirny, 2010; Palmer et al., 2011; Segal et al., 2006; Venters et al., 2011). New experimental techniques are constantly being developed to understand transcriptional regulation at unprecedented temporal and molecular detail, ultimately even at single-cell resolution (Galburt et al., 2009; Larson et al., 2011; Levsky et al., 2002; Taniguchi et al., 2010). Yet, much is still to be learned as the behavior of the system cannot be explained solely by the behavior of the individual components.

There is growing evidence that transcription emerges not solely from the individual components, but rather from the collective behavior (including competition and cooperation) between the components (Larson et al., 2011; Sanchez et al., 2011; Segal and Widom, 2009a; Struhl and Segal, 2013; To and Maheshri, 2010; Wasson and Hartemink, 2009; Zeevi et al., 2011). DNA undergoes millions of interactions every second, constantly changing the configuration of the molecular components bound. Transcription is simply the controlled recruitment and processivity of the transcriptional machinery. Regulation of the

transcription process involves four major classes of components: the DNA, transcription factors, nucleosomes, and the transcriptional machinery. These factors interact in complex ways, both cooperatively and competitively, to induce transcription. It is the stochastic, temporal, and spatial interactions of these regulators that control the transcription process in each individual cell (Coulon et al., 2010).

### 1.5.2.1 DNA

DNA is the central molecule of transcriptional regulation. Whereas the concentration of all other components varies based on condition or cell type, the number of copies of the DNA per cell is largely defined by the organism or differentiated cell type. The DNA not only encodes the blueprints for the creation of proteins, but also encodes the instructions for when, where, and how much of each transcript to produce.

DNA can be envisioned as a linear string of nucleotides that acts as a stable information storage molecule within cells. Each nucleotide of DNA has two building blocks. One part forms a sugar-phosphate backbone structure that links the linear sequence (strand), while the second part is a side chain of either adenine (A), cytosine (C), guanine (G), or thymine (T). These side chains, known as bases, form hydrogen bonds with their counterparts: adenine binds with thymine and cytosine with guanine (see figures 4-3 and 4-4 in Alberts et al. (2007)). Two strands of complementary DNA bind to form the highly stable double helix with between 10 and 11 nucleotides per turn of the helix. There are the common helix patterns (right handed A- and B-forms) and the less common left handed helix (Z-form) that have individual linking and compaction characteristics. Although the Z-form has been associated at regions of transcription, I use the common B-form for illustrations in this work (Alberts et al., 2007).

The strength of the binding between strands of DNA is sequence dependent as one of the pairs of nucleotides has more bonds than the other. The stability of the pair bonds comes into play when the transcriptional machinery is trying to separate the strands. The amount of DNA required to describe even the simplest of cells is too large to be stored as a straight structure, which requires the DNA backbone to be flexible. The collection



**Figure 1.4: Representation of a DNA segment with a gene and a non-coding transcript.** Most of the figures of transcriptional regulation include the basic concepts depicted here. A gene's protein coding region is represented by a rectangle (blue) on the DNA. The sense (reading direction by translation process) direction of transcription is indicated with an arrow extending from one end of the coding region. There is also non-coding transcription indicated by an arrow extending from the DNA sequence. Here we show an anti-sense transcript in an opposite direction. The locations of transcription factor binding sites are indicated using a small rectangle (red) on the DNA.

of all DNA (genome) in complex organisms is so large that it requires additional layers of organization (known as chromatin) to pack the DNA into the nucleus. The first level of structure is the nucleosome, which wraps DNA around a set of histone proteins. When a segment of DNA is populated with nucleosomes, it forms what looks like pearls on a string. Higher orders of chromatin organization coalesce the DNA into tighter structures to further condense and protect the DNA. Ultimately, the DNA for a genome is divided into a set of chromosomes at the highest level of organization.

In this work, I use the yeast *Saccharomyces cerevisiae* as the model organism. The *S. cerevisiae* genome has sixteen chromosomes that are numbered using roman numerals. The individual strands of complementary DNA are named in yeast. The forward strand (as defined by the reference genome) is named 'Waston', while the complementary strand is named 'Crick'. Canonically, the DNA is usually depicted as a line, a gene that encodes a protein depicted as a box, and the direction of transcription is indicated with an arrow (Figure 1.4). There can also be non-protein-coding transcripts produced along the DNA that are depicted as an arrow without a box. The DNA contains a start and stop signal for the translation process, which in yeast defines the bounds of the gene box. Transcription usually begins upstream (before the gene start) and continues past the gene stop before terminating.

The DNA also contains specific sequences that are recognized by DNA binding proteins that are involved in the process of transcriptional regulation. These proteins recognize specific patterns or sequences of DNA with different affinities often represented by position

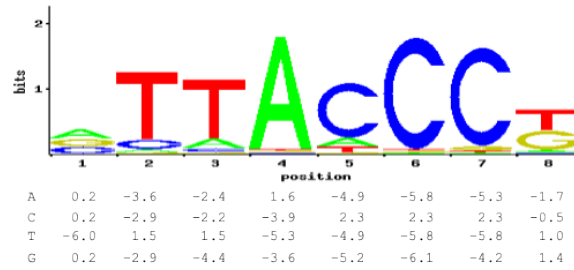
specific scoring matrices (PSSMs). These motifs are depicted as rectangles on the DNA in our cartoons (red rectangle in Figure 1.4). When factors bind to the DNA, they typically occlude a local region of DNA, preventing nearby binding by other proteins. This creates a competition for the DNA between the factors. The cell uses the competition to regulate the DNA accessibility.

### 1.5.2.2 Position Specific Scoring Matrix (PSSM)

Position specific scoring matrices, also known as position weight matrices, have been used to capture the affinity of an individual factor for different sequences of DNA for decades (Stormo, 2000, 2013). DNA binding factors interact differently with different sequences of DNA. If an individual factor would only recognize a single unique sequence, then we could represent the interaction with two parameters: the on-rate and off-rate. This works well for free floating, well mixed environments where the molecules have a limited number of interactions depending on the molecule counts and volume of the environment. However, the positively charged transcription factors have a natural attraction to the negatively charged DNA, which leads to non-specific binding along the DNA. The non-specific binding may only be transient, but sequences that partially match the intended sequence will have stronger affinity for the factor than for random DNA sequences. High affinity binding sites will more likely be bound at low molecule counts, while low affinity sites will require high molecule counts before the site will be highly bound. The high affinity sites also have longer residency times than the transient binding of the low affinity sites.

A PSSM captures the affinity of a DNA binding factor for a sequence, position by position. The simplest method for determining a PSSM is to collect a set of sequence fragments that have been bound by a factor. These sequences are usually large fragments (20-100 nucleotides) and the factor generally only recognizes a few nucleotides (4-20), therefore a common sub-sequence within the collection would be expected. The most common or likely sequence would be the consensus motif that most biologists report as the binding sequence. However, the factor usually only has a few mandatory positions within the sequence and allows multiple different nucleotides to occupy the intervening positions. Aligning the pattern

## PSSM for REB1



| Sequence        | Score | % of max score | Strength |
|-----------------|-------|----------------|----------|
| <b>ATTACCCG</b> | 13.1  | 100%           | 1.00     |
| ATAACCCT        | 8.8   | 67%            | 0.85     |
| ATTCCCGG        | 1.1   | 8%             | 0.02     |

**Figure 1.5: TF affinity can be scored for any sequence by using the TFs PSSM.** An example PSSM, here for Reb1, describes the probability of binding at each nucleotide for any possible sequence. Positions are assumed to be independent and therefore the probability of a TF binding to any arbitrary sequence can be easily calculated.

across multiple sequences, the number of times each nucleotide is found at each position can be calculated. Using the counts, we can calculate the log-odds of seeing a particular nucleotide at a particular position within a bound sequence of DNA. Storing the values for each nucleotide for each position results in a matrix. We can set the values of the matrix to be probabilities of each nucleotide at each position, or we can assign a score for each nucleotide at each position. The scores could be positive for a matching nucleotide or negative for an unlikely nucleotide, resulting in a PSSM.

The PSSM contains the consensus motif as the highest scoring sequence, but also allows any sequence to be scored. Some alternative sequences to the consensus motif will still be able to score highly, indicating that even at low concentrations of the binding factor, sites with the alternate DNA sequence would also be bound.

PSSMs provide a powerful tool for predicting the likely binding sites for any DNA sequence. We can iteratively pass sub-sequences to the PSSM and use the high scores to



predict not only the binding sites, but also the sites of high transcription factor residency times.

Consider the REB1 transcription factor as an example (Figure 1.5). The transcription factor has particular sequences to which it preferentially binds, described by its PSSM. This matrix specifies the number of positions, as well as the nucleotides that are preferentially bound at each position. For any given sequence, we can calculate a score by adding up all the values for the sequence's nucleotide at each position. The resulting score indicates how well the given sequence matches the best sequence for the factor.

### 1.5.2.3 Nucleosome

The most prevalent DNA binding factor is the nucleosome. Nucleosomes are an additional regulatory component in eukaryotes, which form stable structures by wrapping ~147 nucleotides of DNA around a core of eight histone proteins. The pliability of the DNA helix is sequence dependent, which creates an implicit probability of nucleosome formation depending on the energy required to bend the specific sequence of DNA (Drew and Travers, 1985; Morozov et al., 2009). Additionally, the histone proteins have a binding affinity for sequence pairs across large sequence segments (see figure 4-28 in (Alberts et al., 2007)). Therefore, conceptually, nucleosomes can be thought of as recognizing a large segment of DNA with different affinities for preferential binding (Lowary and Widom, 1998).

The formation of a nucleosome steps through a number of intermediate stages. The first stage binds pairs of H3 and H4 histones with ~80-90 nucleotides to form the core of the nucleosome (Figure 1.6). The histone pairs H2A and H2B are then included to bind the entry/exit DNA to the nucleosome core. It is the nucleosome core that exhibits an affinity for a sequence pattern. The DNA bound within a nucleosome wraps ~1.7 times around the histone core and DNA pair binding creates structural features that preferentially allow the histones to bind in the minor groove of the helix. The preferential pattern of DNA for binding has AT pairs along the inside, separated by a single turn of the DNA (10-11 nucleotides), and GC pairs along the outside of the turns. The histone proteins have

tails that are able to wrap around the DNA to form a stable nucleosome. The tails allow modifications that will change the ability of the tail to maintain binding.

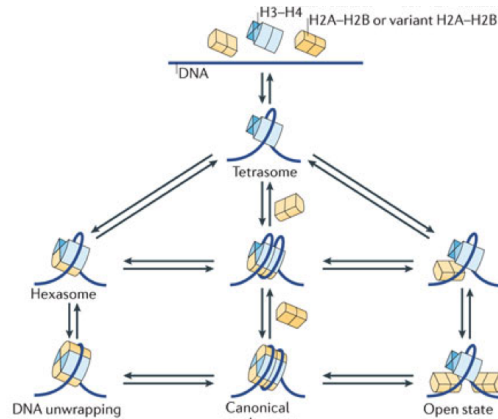
Although any DNA sequence can be folded into a nucleosome, each sequence of DNA requires energy to bend around the histones. Different sequences require different levels of energy for nucleosome formation. The competition between histone affinity, DNA bending energy, and the occlusion by other DNA binding factors allows the cell to regulate the nucleosome positioning, sometimes with great precision. For example, there is often a nucleosome well positioned at the transcription start site (Struhl and Segal, 2013).

The affinity of histones for specific patterns of DNA is captured in an alternative PSSM. The histone affinity is for pairs of nucleotides as opposed to the single nucleotides of transcription factors (Wasson and Hartemink, 2009). This means the PSSM must represent all 16 possible pairs of nucleotides across the entire length of the nucleosome. The current steady state models use a PSSM to describe the probabilities for nucleotide pairs at 127 positions centered on the dyad of the core histones. My extended model uses the central 87 nucleotide pair positions of the PSSM to calculate the state transitions for the core nucleosome.

Nucleosomes along the DNA are the first level of chromatin, which packages and protects the DNA. Due to the physical size of the binding proteins, not all the DNA is bound within nucleosomes. There is always a linker of DNA between individual nucleosomes. The linker length is variable and dependant on the different chromatin remodelers that are active (Struhl and Segal, 2013). Higher orders of chromatin condense the DNA further and require more regulation to access that DNA (Rando and Winston, 2012). Nucleosomes are compacted into denser structures and ultimately form the recognizable chromosome structures.

#### **1.5.2.4 Transcription Factors**

Transcription factors are proteins that recognize small segments of DNA (typically 4-20 nucleotides) (Badis et al., 2009; Bulyk et al., 2001). They often work in groups or complexes, allowing for varying degrees of control over the transcriptional process. Transcription factors



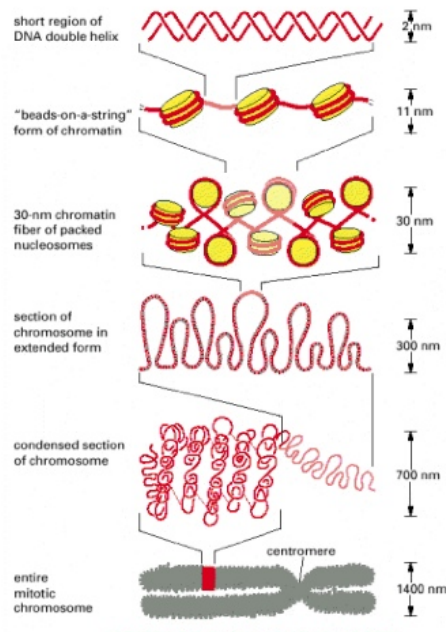
**Figure 1.6: Nucleosome formation dynamics.** Structural states of the nucleosome that are likely to be interchangeable. These include the tetrasome, which is formed by the wrapping of 80 bp DNA around a quartet of (two H3 and two H4) histones. Nucleosomes may undergo spontaneous structural transitions that are characterized either by the transient release of the DNA ends or by a transient opening of the interface between histone subcomplexes. Some states may be favoured by DNA sequence, histone variant incorporation or post-translational modifications. This figure is adapted from Luger et al. (2012).

can function as activators or repressors of transcription. The influence of transcription factors on the activity of different target genes has often been captured in gene regulatory networks (Karlebach and Shamir, 2008).

The ability of a regulator to bind to DNA may be influenced by the competition with other proteins for a sequence, the nearby positioning of nucleosomes, the presence of co-factors, and the post-translational state of the transcription factor itself. Particular configurations of interacting molecules are necessary for the recruitment of the transcriptional machinery and activation of transcription.

When a factor binds, it occludes the DNA and alters the possible interactions in that region of the sequence. The longer the residency time (time an individual factor is bound), the more effect it has over the local configuration of other factors bound to the DNA. Although many transcription factor affinities and binding rates are known, the residency times have not yet been measured for many factors (Lickwar et al., 2012b).

The nucleosomes condense and prevent the DNA from binding with transcription factors. Once a nucleosome is evicted, the transcription factors can access that DNA. If a



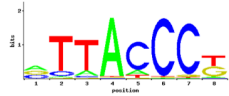
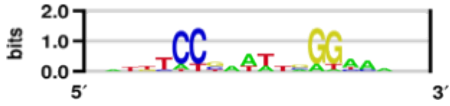
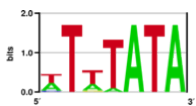
**Figure 1.7: Chromatin packaging.** Models for the different levels of chromatin packaging, from the bare DNA strands through the most condensed packaging seen during mitosis. This figure is adapted from Alberts et al. (2007).

factor binds with the newly accessible DNA, the DNA cannot be re-incorporated into another nucleosome, leaving the entire region accessible. Transcription factors bind to much smaller regions of the DNA than nucleosomes. This allows multiple factors to bind in the newly accessible DNA. Keeping the DNA open and accessible can be obtained either from explicit cooperation between factors (e.g. Ste12-Tec1) or through implicit cooperation between independent factors (e.g. Mcm1-Reb1-Rsc3 at CLN2 nucleosome depleted region). There is a constant competition for the DNA, with the nucleosome formation closing off access and transcription factor binding opening up access to the DNA.

### 1.5.2.5 Transcriptional Machinery

This machinery is a large complex composed of a diverse set of protein subunits that assemble on the DNA (see (Hahn and Young, 2011) for a detailed review of transcriptional machinery components and behavior). The transcriptional process is separated into three stages: initiation, elongation, and termination. The many components of the transcriptional machinery assemble a pre-initiation complex on a sequence of DNA. It must separate

**Table 1.1:** Examples of transcription factor binding motifs.

| Transcription Factor | Consensus Motif | Logo (Schneider and Stephens, 1990)  |
|----------------------|-----------------|--|
| REB1                 | TTACCCG         |  |
| MCM1                 | CC...T..GGAAA   |  |
| SPT15                | ATATATA         |  |

the strands to obtain access to a single strand, and primes the first few nucleotides of matching RNA. The energy required to separate the double stranded DNA is dependant on the sequence, as the segments with high number of GC pairs is stronger than with a corresponding high number of AT pairs. Although the machinery can assemble in either orientation on the double stranded DNA, once the components are assembled, transcription progresses in only one direction on a single stand.

Once the complex has completed initiation, it can begin the elongation stage. Elongation is the stage where the DNA template is transcribed into an RNA copy. The transcriptional machinery must deal with obstacles in its path along the DNA and presumably will evict nucleosomes and other DNA binding factors as it traverses the DNA. Elongation is not a simple continuous process. It can pause, arrest, edit mistakes, or abort before reaching the termination signal. Elongation continues until a transcription stop sequence is encountered and the transcriptional machinery transitions to the termination stage.

Termination of the RNA is most often performed when a signal within the sequence is encountered. The pre-mRNA can be further processed, such as a poly-A tail being appended, to create a messenger RNA that is ready for transportation to translation process.

**Table 1.2:** Types of RNAs produced in cells.

| RNA type       | Function of RNAs  |
|----------------|---|
| mRNA           | Messenger RNAs, instructions for producing proteins.  |
| rRNA           | Ribosomal RNAs, the basic structure of the ribosome that translates mRNA into protein.  |
| tRNA           | Transfer RNAs, adaptors used to match mRNA nucleotides and amino acids.   |
| snRNA          | Small nuclear RNAs, variety of functions, including the splicing of pre-mRNA.   |
| non-coding RNA | Used in diverse cellular processes, including telomere synthesis, X-chromosome inactivation, and sometimes only the process of transcription is required to regulate other transcription. |

Once the machinery has terminated the transcription, it releases the DNA and breaks up into its individual components.

There are many different types of RNA produced in cells and they require different sets of transcriptional machinery (Tables 1.2 and 1.3). In this work we are only concerned with

**Table 1.3:** Polymerase used to generate different types of RNA.(Alberts et al., 2007)

| RNA Polymerase     | Type(s) of RNAs transcribed   |
|--------------------|---|
| RNA Polymerase I   | Ribosomal RNA (rRNA - 5.8S, 18S, and 28S).                          |
| RNA Polymerase II  | All protein-coding genes, plus snoRNA genes and some snRNA genes.   |
| RNA Polymerase III | tRNA genes, rRNA genes (5S), some snRNA genes and other small RNAs. |

transcription of the gene coding regions and focus exclusively on the RNA polymerase II behavior.

### 1.5.3 Dynamics of Regulation

Recent experimental work has highlighted a number of inherently dynamic events that contribute to transcriptional regulation. Transcription factor residency times (how long a factor remains bound at an individual site) may be an important but previously overlooked aspect of regulation (Lickwar et al., 2012b). The histone proteins within a nucleosome are modified, swapped, or displaced during transcription, which changes the behavior of individual nucleosomes (Workman, 2006; Kulaeva et al., 2013). Finally, the movement of the transcriptional machinery along DNA is a highly dynamic process that pauses, shows variable processivity, and likely evicts DNA binding factors that impede its forward progress (Coulon et al., 2013). In fact, when one transcriptional machine directly impacts a second transcriptional process, this interaction is referred to as transcriptional interference (Prescott and Proudfoot, 2002). The kinetics of these events is often inherently local and intrinsically stochastic.

### 1.5.4 Single Cell Variation

The dynamic aspects of regulation are most apparent when examining single cell transcription. These studies have been made possible by recent technological innovations, such as fluorescent protein tracking and real-time nascent transcription observations (Taniguchi et al., 2010; Pelechano et al., 2010). In these studies, variability in transcription and protein expression is widely observed, likely stemming from fluctuations in cellular abundances of proteins, the stochastic nature of molecular interactions, and microenvironments within a cell (Elowitz et al., 2002; Kaufmann and van Oudenaarden, 2007). Transcriptional regulation is inherently dependent upon the biochemical interactions of many different molecules, but robust enough to handle the stochastic fluctuations inherent in any molecular system. The resulting cell-to-cell variability is likely fundamental to most, if not all, molecular cellular processes (Huang et al., 2009; Schwabe et al., 2011).

## CHAPTER II

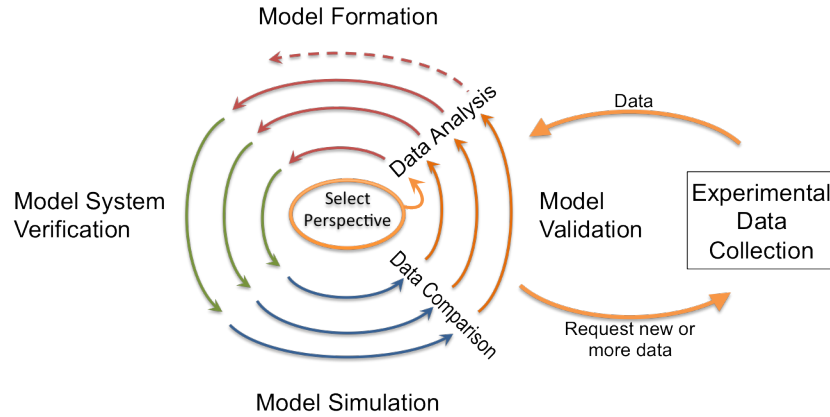
### MODELING PERSPECTIVES

The process of creating biological models is an iterative process of model formation, model verification, model simulations, and model validation (Figure 2.1). During model formation we take the accumulated data and knowledge to expand or abstract the details within the model. Once the model has been described, the model is verified to ensure the implementation is correctly representing the behaviors that have been formulated. The verified models are used to simulate the behavior of the modeled system. The results are collected and compared against the real world experimental data during the validation stage. The simulation of a model may only require a single simulation for deterministic models or many simulations for stochastic models that are non-deterministic to provide a distribution of possible trajectories for the model. The comparison analysis of the predicted behavior and the experimental data provides insight into which areas of the model could be enhanced, either by collecting more data, collecting new data, or modifying the model. Each interaction takes a step towards better models, but as the models increase the ability to match the experimental data, they become more complex and usually require more computational resources to complete each stage.

At the center of this development process is the selection of a perspective (Figure 2.1). The formation of a good model is predicated on knowing what kinds of answers are needed for the relevant questions that are being asked. This means that there are many methods and different perspectives from which researchers could look for answers to their questions. Selecting a single perspective and method depends on those questions, as well as understanding the available data. Choosing a perspective for biological systems is difficult because of the large number of components, each with its own behavior.

We build models of what we think we understand. Often we do not have enough data for all the behaviors to be well understood. Yet, those behaviors must be included in the model and we add a new component to represent the behavior. Other times, we have too



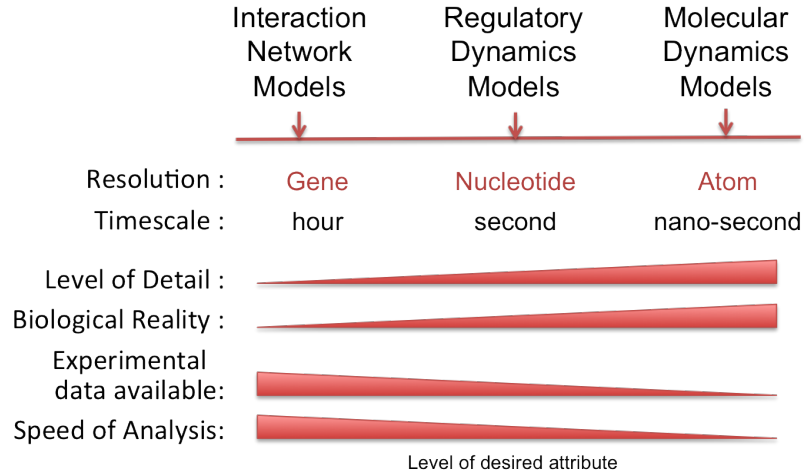


**Figure 2.1: Modeling development cycle is an iterative and collaborative process.** Developing models of biological behavior requires an iterative and collaborative process of data analysis leading to model formation, verification, simulation based predictions, and validation of model results against the experimental data. **Model Formation** encapsulates the data and knowledge about the real world behavior of a biological system as a computational model. **Model System Verification** ensures the computer implementation of the model is correct. **Model Simulation** uses the verified model to simulate and capture the predicted biological behavior. **Model Validation** compares the simulation results with the experimental data to determine the degree to which the model matches the real world behavior. The results of the simulations are compared to the experimental data to understand how closely the model is matching the real world data. This comparison and analysis may lead to directing the collection of new experimental data. The development cycle spirals outward as the model is improved. As it spirals outward the model becomes more complex and assimilates more data. However, as the model becomes more complex, it requires an increase in the computational resources required to process each step. This figure is based on Thomas H Jordon’s “Inference Spiral of System Science.”

much knowledge about components that is not relevant for the questions being asked. These details are abstracted away by combining all the behaviors into a less complex component. And finally, we may be given higher resolution data that requires a new level of detail within the models, which can be accomplished by replacing or adding more components and interactions. Determining which details to hide and which to define depends on the questions being asked, the data available, and the types of answers required.

## 2.1 Selecting a Modeling Perspective

Current models of transcriptional regulation vary tremendously in their underlying perspective on the biological process. In Figure 2.2, I have arranged some of the major modeling perspectives along an arbitrary axis, followed by a list of the major attributes we want to achieve. It is not possible to maximize for every attribute as they can be mutually exclusive.



**Figure 2.2: Modeling Perspectives.** The modeling perspectives are arranged along the x-axis from least to most detail of the models. Below them are wedges representing the level of an attribute in each of the modeling perspectives. We would like to maximize for all the attributes, however as the graphic indicates, model selection is a trade-off between the different attributes. Figure adapted from (Karlebach and Shamir, 2008).

Therefore, selection of the perspective is a trade-off between the attributes and is dependent on the questions to be asked of the model. Each perspective focuses on different levels of resolution or detail within the model. If we are interested in understanding the effect of a signaling pathway or knowing what genes are affected by changes in levels of other genes (gene regulatory networks), then interaction network models can be used. These models are able to analyze thousands of genes and millions of interaction possibilities on standard computer hardware. On the other hand, if we are interested in the details of binding between a transcription factor and the DNA, we must select a perspective that has details of the individual bonds between atoms of the molecules. The molecular dynamics models are very detailed, but can only produce milliseconds of simulation time by using many hours of today's multiprocessor supercomputers. These trade-offs of computational resources and level of detail must be weighed against the ability of the models to produce results for the questions being asked.

### 2.1.1 Molecular Dynamic Perspective

The arbitrary axis of the modeling spectrum in Figure 2.2 differentiates between three different modeling perspectives. At one end of the axis are molecular dynamics models that focus on the physics behind atomic interactions between molecules. These models are used to answer questions about the shapes, structures, and bonds between atoms in multiple molecules (see Karplus and McCammon (2002) for review on molecular dynamics models). Molecular dynamics models are very detailed models that provide exact physical behavior over very short timescales using quantum or molecular mechanics. However, they are very detailed, which requires tremendous amounts of computational resources to model two molecule interactions for a few nanoseconds. There are modeling methods that attempt to reduce the computational costs by coarse-graining, which abstracts the individual atoms into pseudo atoms and calculates the behavior of the pseudo atom at each time step. Even with this abstraction, the models are limited in number of atoms and typically can only model 10 milliseconds of time. It is not possible for these models to be scaled up even to the relatively small numbers of molecules that interact with a small sequence of DNA.

### 2.1.2 Interaction Network Perspective

At the other end of this arbitrary axis are the interaction networks representing the knowledge (both qualitative and quantitative) of relationships between components of a system. In transcription regulation modeling, a primary example of these networks are Gene Regulatory Networks (GRNs), which seek to capture the logic of a circuit by describing the behavior between genes (see (Hecker et al., 2009) for review on inferring gene regulatory networks). Interaction networks capture the correlation between changes in the level of one component and the changes in levels of all the other components. These correlations are usually captured in a graph where all the components are nodes and the relationship between nodes is represented by an edge. Both the components and the relationships can have associated attributes that further describe the relationships.

There are a range of different modeling methods that can represent the relationships captured from experimental data (Kerlebach and Shamir, 2008). These methods are fo-

cused on the behavior at the gene or regulatory module level of detail. As almost all the experimental data has been collected as population averaged behavior, the models answer questions about the population average behavior and not the behavior of any single cell.

Interaction network models are great for asking questions about which other components would be affected by changes in a specific component. However, these models represent only the correlations between levels and not causation. If we want to ask "why does the change in level of a component cause the probability of another component's transcription?", then we must change our modeling perspective to one that contains the hidden details beyond the interaction networks.

### **2.1.3 Regulatory Dynamic Perspective**

Regulatory dynamics models focus on the dynamic interactions between DNA binding factors and a sequence of DNA at nucleotide resolution to study the spatial and temporal behavior of the transcriptional regulation mechanisms.

Transcriptional regulation in eukaryotes is not regulated by a single factor as is often observed in prokaryotes. Eukaryotic regulation is a complex balance of many individual factors behaving independently, but creating a higher order system control. This concept is known as emergent behavior and is often described using examples of the behavior in a flock of birds or a school of fish. Each individual bird or fish is behaving independent of the others, but because of the dynamics of the group, they exhibit a higher order behavior. The same is true at the microscopic level within a cell. Each individual molecule is performing the task for which it was designed. Factors that bind DNA are constantly binding and unbinding the DNA, staying longer at sequences where it obtains a stronger bond. Cooperativity can be obtained through actual binding of two molecules to each other or by independently binding in close proximity along the DNA. The independent binding by factors to produce cooperative binding is an example of an emergent behavior.

To capture the emergent behavior of the complex system of interacting molecules in transcriptional regulation, we must focus on a level of detail beyond the gene or regulatory module of the DNA. The behavior of each individual molecule must be modeled to

understand the diversity of behavior in individual cells. In this work, I focused on the understanding of the mechanisms of transcriptional regulation at a high level of detail, while maintaining computability for large sequences of DNA. Instead of focusing on the atomic interactions of molecular dynamics models or the gene interactions of interaction networks, regulation dynamic models focus on the nucleotide interactions between DNA and the DNA binding factors. Regulatory dynamics models capture the temporal and spatial dynamics between components that culminates in transcription.

## 2.2 Selecting a Modeling Method

Models within each of these perspectives can be built using different modeling methods. Each method focuses on different aspects of the the system and represents the model in different ways. The methods can be categorized as logical (Boolean network, probabilistic network, Petri net, Hidden Markov Models), continuous (linear, differential equations, flux balance analysis), and single molecule (rule-based). Each method has different strengths and selecting which one to use depends on the questions being asked, the data available, and the type of answer required for the questions. Several recent reviews discuss the trade-offs inherent in choosing any method (Ay and Arnosti, 2011; Karlebach and Shamir, 2008; Tenazinha and Vinga, 2011). The details of the different modeling methods have been reviewed in more detail elsewhere, see (Hecker et al., 2009) for review on inferring gene regulatory networks, (Ay and Arnosti, 2011) for general mathematical modeling methods, and (Karplus and McCammon, 2002) for molecular dynamics models. As the modeling methods for molecular dynamics do not scale well to the large number of components used in transcriptional regulation, I will not discuss those methods in any further detail and focus on the methods used in interaction networks and regulatory dynamics perspectives.

The division of the perspectives along a continuous scale is to categorize the abstract differences between the questions the models are addressing. The theoretical distinction between the different modeling perspectives is along a scale to generalize the categories. However, this is a continuous scale and many of the transcriptional regulation interaction

networks are becoming so detailed that they are approaching the single nucleotide resolution of the regulatory dynamics models.

Most current modeling methods for describing models of transcription regulation focus either on the positional details (Greive et al., 2011; Segal et al., 2008; Wasson and Hartemink, 2009) or the temporal dynamics (Chaouiya, 2007; Dresch et al., 2013; Ribeiro, 2010; Sanchez et al., 2013) of the system of interest. Here I will categorize the methods into Statistical and Analytical, although some methods can span both categories by changing the level of detail used to describe the models (Ay and Arnosti, 2011).

Statistical models are often used when the system to be modeled has a large number of components. Graph based methods can represent the probabilistic models as neural, Boolean, or Bayesian networks. However, these methods only allow an overview or big picture of the system because the details are obscured to handle a large number of components and to see the high level relationships. They are not able to explain the details of the relationships between the DNA, transcription factors, nucleosome formation, and the transcriptional machinery.

Analytical models are used to describe a smaller number of components in more detail. Models describing detailed component behavior require more knowledge, which may be unknown and must be estimated from current knowledge. The detailed models also required more computational resources to process and therefore may limit the number of components in a model to keep the computation tractable.

The analytical models can be deterministic or stochastic, discrete or continuous, and are either solved or simulated. Most of the biological processes we are trying to model are noisy (meaning that even the most improbable events can and do occur) and regulated by a small number of molecules (a small change in the number of molecules will have large effects). If the noise is ignored, deterministic models can be used to describe the behavior of the system and a system of equations can be solved by applying the experimental data to determine unknown parameters. Stochastic models incorporate the inherent noise as part

of the system. These models explicitly describe the behavior and the variability of those behaviors.

The behavior of any component can be described with population averaged behavior or at single molecule detail. The general behavior within a population of cells can be described as continuous values because it is a description of the fractions of cells with a behavior. When the detail level is at single molecule resolution, the number of molecules is discrete, as you cannot have only part of a molecule behaving one way and the rest behaving differently. The molecular counts are discrete and every individual molecule has a distinct state influencing the overall behavior of the system.

### **2.2.1 Equation-based Modeling Methods**

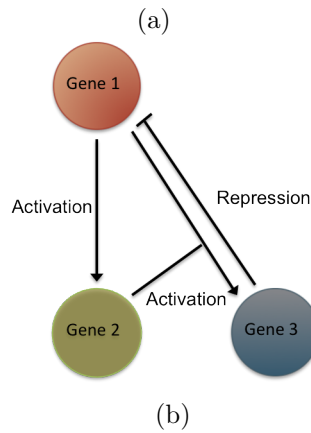
Mathematical models have been successfully applied to biological modeling for many years. The most common are Boolean and differential equation models, which have been used for decades to describe gene regulatory networks. Boolean models have abstracted all the details of the interactions away and only use the existence of a component in its computations. The differential equation models define a detailed relationship between the current levels of the components and the level of an individual component. Generally, these models represent how the state of the system changes over time. The current state of the system is recorded as a set of component concentrations or molecule counts. The state of the system in the next time period can be calculated from a set of equations that uses the current state of the system as parameters. The parameters of the model can be determined either from direct experimental work or by solving for the unknown parameters by using general experimental results data. However, often there are too many unknown parameters to uniquely solve the set of equations using the limited experimental results.

Each equation in the model specifies how to calculate the next state of one component using a function that considers the current state of all the components. In Figure 2.3, the equations define the behavior of the levels of each gene over time. Each equation is dependent on the levels of the genes at the current time point. For example, the first equation represents the changes in levels of  $gene_1$  based on the constitutively expressed rate

$$\frac{d(\text{gene}_1)}{dt} = k_{1,s} * \frac{1}{1 + k_{1,3} * \text{gene}_3} - k_{1,d} * \text{gene}_1$$

$$\frac{d(\text{gene}_2)}{dt} = k_{2,s} * \frac{k_{2,1} * \text{gene}_1}{1 + k_{2,1} * \text{gene}_1} - k_{2,d} * \text{gene}_2$$

$$\frac{d(\text{gene}_3)}{dt} = k_{3,s} * \frac{k_{3,1} * \text{gene}_1 * k_{3,2} * \text{gene}_2}{(1 + k_{3,1} * \text{gene}_1) * (1 + k_{3,2} * \text{gene}_2)} - k_{3,d} * \text{gene}_3$$



**Figure 2.3: Differential Equation models are explicitly defined.** This network of three genes is modeled using ordinary differential equations (ODEs). (a) The level of each gene can be calculated from the current levels of genes and each of the known reaction rates (specified as 'k') between components. (b) Graphical representation of the interactions between the genes. Each circle represents a single gene. Arcs between genes represents a correlation between the level of the arc's originating gene and the destination gene. An arrow head represents one gene having a positive effect (activating) on the level of the other gene. A 'T' end on an arc represents a negative effect (repressing) on the level of the other gene. Figure from (Karlebach and Shamir, 2008).

of  $\text{gene}_1$  minus the repressive effect of  $\text{gene}_3$  and the degradation of the current level of  $\text{gene}_1$ .

The rates of each of these interactions may be known from experimental data or may be discovered using the experimental data. For many decades, gene regulatory networks such as these have been discovered from the large data sets for gene expression (see (Hecker et al., 2009) for review on inferring gene regulatory networks).

This type of differential equation model is deterministic, using continuous values to represent the population averaged levels of components. Given the same initial state, the behavior over time is always the same. This means that the equations could be 'solved', where every rate is explicitly defined. This has been accomplished for systems such as the



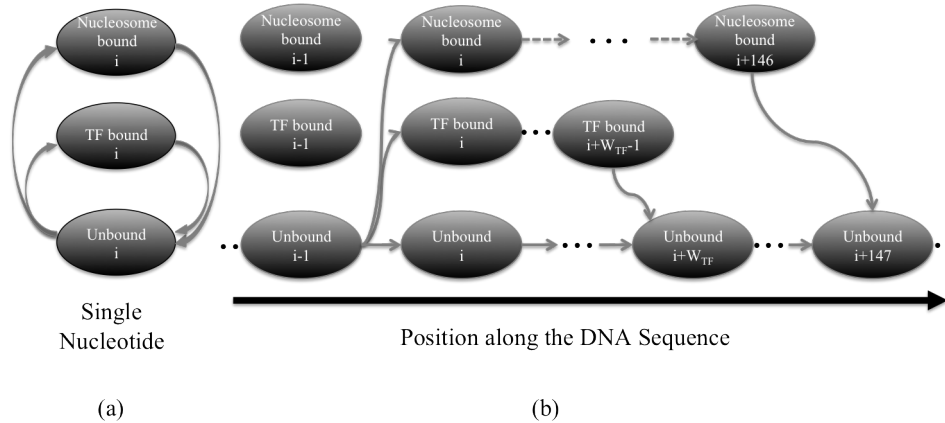
metabolism of bacteria (Edwards and Palsson, 2000; Förster et al., 2003). The method of Flux Balance Analysis (FBA) places constraints on the behavior of the system to help limit the possible solutions. FBA constrains the model by using the fact that matter must be conserved throughout the interactions. This means that the overall mass within the system must remain consistent. The constraints limit the possible values of the parameters and aid in solving the system for a data set.

Unfortunately, most biological systems are not very deterministic. There is a seemingly random behavior that causes individual cells within clonal populations to behave differently. The equations can be modified to include the stochastic behavior by including another term representing the noise or randomness of the system. Adding another parameter for each equation makes it much more difficult to uniquely solve the set of equations.

These models have been integral to my understanding of how the act of transcription in one region can regulate nearby and overlapping transcription through a process known as interference (Sneppen et al., 2005).

### **2.2.2 Statistical-based Modeling Methods**

There is another set of modeling methods that focuses on the configuration of the system as a whole. These configuration based modeling systems (also known as thermodynamic or fractional occupancy models) have been developed to describe the positional binding configuration of proteins along a segment of DNA (reviewed in (Ay and Arnosti, 2011)). Briefly, the models capture the probability of each configuration of bound components based on the probability of each component binding to a segment of DNA. The binding probabilities are calculated from the concentrations and thermodynamic binding probability of the component and specific DNA sequences. The method is a simple three step process: 1) list all the possible configurations of components bound to the DNA, 2) calculate the probability of each configuration based on the statistical weight of each configuration compared to all



**Figure 2.4: Positional information described by Hidden Markov Models.** Focusing at the nucleotide level, the model captures the fact that each nucleotide can be in only one of a limited number of states. (a) Conceptually, any nucleotide ( $i$ ) can only be in one of three states: unbound, bound to a glstf transcription factor (TF bound), or bound in a nucleosome (Nucleosome bound). Competition between binding factors is determined by the relative difference in probability of each factor binding to the DNA at that position. This figure is adapted from Wasson et. al. (Wasson and Hartemink, 2009). (b) Practically, each factor extends over multiple nucleotides, which is managed by the transition probabilities. Nucleosomes are formed with 147 nucleotides ( $i \dots i+146$ ) and each transcription factor has a unique interaction length ( $i \dots i+W_{TF}-1$ ). The probability of transition depends on the different affinity each factor has for the underlying sequence.

possible configurations, and 3) assign the expression level possible for each configuration and predict expected expression based on sequence and concentrations of components.

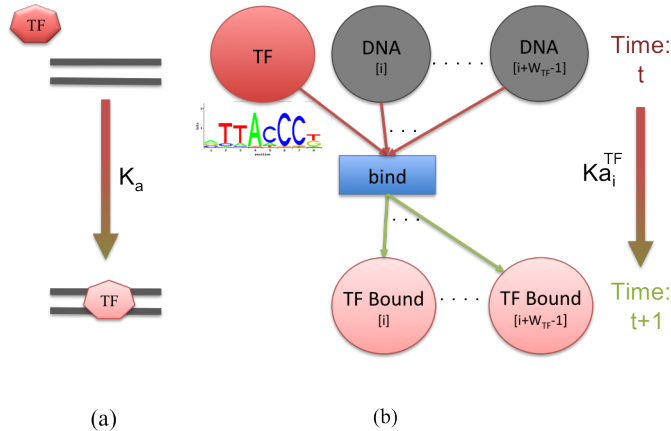
Conceptually, these states define a model where the transitions between the states depend on both the affinity of the component to the DNA sequence and its concentration. As both nucleosomes and transcription factors bind to multiple nucleotide positions, the actual connectivity between states varies depending on the identity of the component in order to capture the specificity and length of binding.

Configuration based models of transcriptional regulation focus on the state of each nucleotide of a given sequence. Currently, these models consider only nucleosome positioning and transcription factor binding. With these two components, a given nucleotide can be unbound, bound in a nucleosome, or bound in a particular transcription factor (Figure 2.4-a). Conceptually, these states define a model where the transitions between the states depend on both the affinity of the component to the DNA sequence and its concentration

(Wasson and Hartemink, 2009). As both nucleosomes and transcription factors bind to multiple nucleotide positions, the actual connectivity between states varies depending on the identity of the component in order to capture the specificity and length of binding (Figure 2.4-b). This additional complexity is still elegantly captured by a hidden Markov model (HMM) and allows large DNA sequences to be quickly modeled on conventional computer resources (Segal et al., 2006; Wasson and Hartemink, 2009). These models have proven to be quite successful at elucidating key regulatory principles inherent in the competition between nucleosome and transcription factors (Segal and Widom, 2009a). However, these methods describe only the population averaged behavior of a DNA region and do not address the inherent temporal variation in configurations within a single cell.

Another modeling method that has been used to capture the transitions between component states uses the well studied formalism of Petri nets. Petri nets are an intuitive representation of the biochemical networks that uses a graphical representation to describe the transition between states as actions (Figure 2.5). Briefly, every action has inputs and outputs, which are the components of the system. When molecules of all the inputs are available, the action fires and consumes molecules of the inputs and produces molecules of the outputs. During simulations, the molecule counts of each component is tracked as actions consume and create molecules. Extensions have been added to standard Petri net descriptions to account for time delayed or stochastic actions. Often the Petri net descriptions are transformed into a set of differential equations for analysis. There are also a number of simulators available that accept a Petri net and initial conditions to simulate the system.

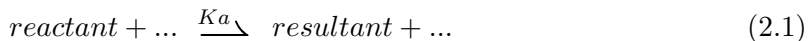
Petri nets can capture the same competition between components as the HMM by creating molecule definitions for each state of each nucleotide. Transcription factor binding is represented as an action that takes a molecule of the transcription factor and a contiguous set of unbound nucleotides as inputs, producing a set of transcription factor bound nucleotides as output (Figure 2.5).



**Figure 2.5: Biological behaviors are modeled by computational components.** (a) The core of our modeling framework is to capture biological interactions, such as the behavior of a transcription factor (TF) binding to the DNA (at rate  $K_a$ ). (b) Abstract computational description of the biological interaction using Petri net notation. These descriptions are action based (blue rectangle), with the preconditions (circles listed above the action for a molecule of the TF (red) and multiple positions ( $i \dots i+W_{TF}-1$ ) of unbound DNA (grey)) required to apply the action, and the post-conditions (listed below the action with DNA positions ( $i \dots i+W_{TF}-1$ ) bound by that TF (pink)) that are true after the action is applied at time  $t$ . The position specific scoring matrix (PSSM, example shown below the TF as a sequence logo [99]) describes a specific TF’s affinity for DNA, which is used to calculate the TF specific rate depending on the actual DNA being bound ( $Ka_i^{TF}$ ).

### 2.2.3 Chemical Reaction Modeling Methods

All biological behavior can be viewed as a result of a system of chemical interactions between biological components. Transcriptional regulation is a result of all the reactions that occur along a specific sequence of DNA, but each individual interaction of a component can be described as a straightforward chemical interaction rule (concepts reviewed in (Karlebach and Shamir, 2008) and (Faeder et al., 2005)). Each rule specifies the reactants being combined at a specified rate to produce the resultants (Eq. 2.1). An arrow indicates the direction of the interaction and the rate ( $K_a$ ) of the reaction when the components are available.



In many cases, the resulting network of interactions can be converted into a set of differential equations. It is important to point out that all of the methods described above

(except Petri net models) for equation, statistical, and chemical reaction models, are population averaged methods. They describe the average behavior of a cell within a population of identically defined cells. They do not describe the behavior of any individual cell.

Simulations treat all components as continuous variables and represent the behavior of a population of cells. These continuous values do not describe the discrete quantity of components within a single cell and most likely do not describe the behavior of any single cell within that population. The mass action assumptions of large quantities of reactants does not hold when the number of molecules is small. Transcriptional regulation within individual cells uses a small number of molecules of important components where small changes in the quantity of these components have enormous influence in the cellular behavior.

One of the advantages of modeling the individual interactions is being able to capture the temporal dynamics of the system. Temporal dynamics models seek to capture the key molecular interactions occurring during transcription process and typically describe the behavior of molecules through a series of biochemical rules. In many cases, the Gillespie stochastic simulation algorithm (SSA) can be applied to explicitly capture every interaction in a discrete and stochastic simulation (Gillespie, 1976). The algorithm is a dynamic Monte Carlo method that stochastically simulates a system of biochemical reactions to produce one possible trajectory (solution) of the interaction rules. Multiple simulations can be run to explore the distribution of trajectories given the stochastic nature of the system. This approach has been used to model a variety of stochastic systems, from ecosystems to cells (Black and McKane, 2012; Takahashi et al., 2004).

#### **2.2.4 Stochastic Models at Single Nucleotide Resolution**

Another method has recently emerged that uses the simplicity of defining rule based systems to increase the model resolution to the nucleotide interactions (Roussel and Zhu, 2006; Ribeiro et al., 2009; Mäkelä et al., 2011). These models consider the DNA as a serial sequence of nucleotides that interact with other components. Each nucleotide is defined to have a set of states representing each interaction possible. For example, Roussel et. al.

describes how modeling of the transcription process can be abstracted from the complex interactions of many molecules in the biological process into a set of interactions for the transcription stages of initiation, elongation, and termination (Roussel and Zhu, 2006). Their model of prokaryotes defined a promoter and a gene region. The promoter must be in an open conformation to allow the transcriptional machinery to bind (initiation). Once the machinery is bound and activated, it can process each nucleotide in the sequence (elongation) until it unbinds from the DNA (termination). This description provides a small set of components (DNA, transcription factor, and transcriptional machinery) and interactions (promoter activation and deactivation, transcriptional machinery binding, activation, moving, and unbinding), which can easily be described with simple reaction rules.

The details of the system were enhanced to include transcription factor binding to the promoter region for both initiation and repression, as well as detailed interactions of the transcriptional machinery, including pausing, arresting, and premature termination (Ribeiro et al., 2009). Another very important enhancement by Ribeiro et. al. was to model the transcriptional machinery simultaneously bound to multiple nucleotides. This enables the modeling of cooperative behavior between multiple copies of the transcriptional machinery along one DNA segment.

These models of the prokaryotic transcription were further enhanced to include the translation process that is simultaneously happening on the output from the transcription process (Mäkelä et al., 2011). Makela et. al. created a more complex model of gene regulator networks to produce both the RNA and the proteins from the sequence of DNA.

However, all these models are focused on the simple transcription activation of prokaryotes. They are interested in the behavior of the protein levels in the system over time. They have abstracted the behavior of the transcription factors because of the simple regulation in prokaryotes has often evolved to have a single factor binding the promoter and sometimes the promoter also contains a single repressor.

Transcriptional regulation in eukaryotes is both simpler and much more complex than prokaryotes. The fact that transcription and translation occur independently in different

cellular compartments keeps translation from affecting the transcription process. It is more complex because there are many factors competing for the same DNA sequences and each of the factors has a different affinity for each DNA sub-sequence. Each factor has a preferred series of nucleotides to which it binds, but will also bind with any DNA sequence. The difference in binding at each site is in the residency time of the factor on the DNA. Sequences closely matching the preferred DNA sequence will remain bound by the factor longer than the non-preferred sequence sites. Besides the many transcription factors competing for DNA in eukaryotes, there are histone proteins that bind with DNA to form nucleosomes. Nucleosomes are used to package and protect the DNA from damage and prohibits a transcription factor or the transcription machinery from binding.

### **2.2.5 Agent-Based Modeling Method**

Despite the growing understanding of transcriptional regulation mechanisms, the complexity of the transcription process is still a modeling challenge. Equation based models do not provide insight into the behavior of individual cells or molecules within a biological system. Observations represent the averaged value and assume the homogeneity and well-mixed components. However, biological processes within single cells are not that well behaved, with stochastic events and localized concentrations or behaviors of components.

Insight into the biological systems can be provided by modeling the individual biological components. It is theoretically possible to model any system by defining the behavior rules for all the individual components. This is the modeling concept used in molecular dynamics where the behavior of molecules interacting can be modeled through rules for interactions between every atom. However, depending on the size of the molecules and the purpose of the model, this may not be computable or even useful. The behavior of the transcription process components must be aggregated to keep the models computationally feasible, which results in a loss of resolution for the model.

Agent based modeling is a relatively new modeling method that has become possible with recent advances in computational power and memory capacity. An agent is a self contained entity defined by a set of interactions and states. Each instantiation of a component

individually maintains and controls its states and interactions. An agent's autonomy allows it to perform tasks within an environment without external control.

Agent based models are comprised of agents interacting in a simulated environment. By specifying the low-level or local rules for each component, complex behaviors that have not been explicitly programmed can be observed from the agent interactions. These emergent behaviors are common within the real world. Flocks of birds or schools of fish have many individuals behaving with simple interactions, but exhibit different behaviors when viewed as a group.

The ability to simulate the actions of the individual components and observe the resulting system behavior over time allows the models to be useful tools for studying the behavior of a complex system. These models can be used as a laboratory for exploring the effects of changes to the behavior of an individual component within a complex system.

There are many advantages to the agent based models over equation based models: ability to capture emergent behaviors, provide a method to study systems that are difficult to experimentally validate, flexibility in describing the behavior of biological components in different detail or timescales, deriving the model from fundamental behaviors of the biological components (e.g. binding/unbinding of molecules), interactions are easier to visualize and understand, and finally, the models are naturally stochastic as the interactions can be based on probabilities or probability distributions.

The granularity of the agent based models comes at a cost. The computational resources required to simulate these complex models are much greater than the equation based models. The system being modeled must be understood at greater detail to allow individual behaviors to be described. A limited knowledge of component kinetics or limited computational resources require modelers to make trade-offs in the detail of the models while still addressing the questions being asked.



## CHAPTER III

### DYNAMIC TRANSCRIPTION MODELING FRAMEWORK <sup>1</sup>

#### 3.1 Introduction

Transcription is the biological process of reading a DNA template and producing an RNA copy of that template. Transcriptional regulation is the system behavior arising from the interaction of numerous regulators with DNA, which allows transcription to occur. In this work, I have developed a new modeling framework that can automatically generate rule sets describing the possible molecular interactions implied by a given DNA molecule to produce a model that captures both the stochastic and dynamic behavior of the complex system known as transcriptional regulation.

The central dogma of biology is that DNA is used to produce RNA copies that in turn are used to produce the proteins described by the DNA. Biological textbooks usually show this as a linear process from DNA to RNA to protein (Figure 1.2-a). However, the process is a never ending cycle as the proteins generated will influence the state of the DNA. Therefore, I have redrawn the dogma as a circular process that includes feedback from other processes on the input to the transcription process (Figure 1.2-b). See section 1.5 for more detailed background on the biological processes and the individual components involved in transcriptional regulation.

The process of transcription is regulated by complex interactions of the DNA with binding proteins, the RNA products, and even the transcription process itself. This complex system of transcriptional regulation produces precise gene expression at specific times and locations. Experimental studies of gene expression have unlocked the function of many proteins involved in regulating the transcription process and new experimental techniques are being developed to understand transcriptional regulation at unprecedented temporal and molecular detail down to single-cell resolution. (section 1.5.2). Yet, much is still to be

---

<sup>1</sup>Part of this work is currently in the review process for IEEE Transactions on Computational Biology and Bioinformatics as "A modeling framework for generation of temporal and positional simulations." This work comprises Aim 1 of a National Science Foundation grant (ABI 1262410) on which I was a co-author.

learned because the behavior of the system cannot be explained solely by the behavior of the individual components.

There is growing evidence that transcription emerges not only from the behavior of individual components, but rather from the collective behavior (including competition and cooperation) between the components (section 1.5.2). DNA undergoes millions of interactions every second, constantly changing the configuration of the molecular components bound. Transcription is simply the controlled recruitment and processivity of the transcriptional machinery. Regulation of the transcription process involves four major classes of components: the DNA, transcription factors, nucleosomes, and the transcriptional machinery. These factors interact in complex ways, both cooperatively and competitively, to induce transcription. It is the stochastic, temporal, and spatial interactions of these regulators that control the transcription process in each individual cell (Coulon et al., 2010).

Encapsulating our understanding of these interactions into a computational model is integral to understanding transcriptional regulation (Lander, 2010). See Chapter II for a more detailed review of the modeling perspectives and computational modeling methods. Models allow us to explore a system, create testable hypotheses, and identify when key details are missing in our current knowledge. To date, most modeling frameworks for transcriptional regulation have either focused on the detailed molecular behavior of a specific regulator or the interaction of a small subset of regulatory components (Barnes et al., 2011; Cantone et al., 2009; Greive et al., 2011; Kim and Gelenbe, 2012; Lubliner and Segal, 2009; Ribeiro, 2010; Segal et al., 2006). Few models have approached the problem of simultaneously capturing the behavior of all the major regulator classes. In part, this is because most models either focus on the positional information of each component (Segal et al., 2008; Wasson and Hartemink, 2009) or the temporal behavior of their inherent dynamics (Ribeiro et al., 2009; Roussel and Zhu, 2006). Integrating both the positional information and temporal information often leads to computationally expensive models. As experimental techniques continue to improve, modeling approaches must also evolve to represent increasingly realistic molecular details while still remaining computationally tractable. We

need new methods to construct biologically realistic computational models that capture not only the positional binding of transcription factors and nucleosomes, but also the underlying temporal dynamics, such as the behavior of transcriptional machinery during initiation and elongation.

### 3.2 Contribution

This section reiterates the contributions for this chapter. See Section 1.3 for a complete list of my contributions.

- **I created a modeling framework to automatically generate a model comprised of a collection of biochemical based rules describing the individual behavior of each model components for any given sequence of DNA.** The generated models capture the details of interactions at nucleotide resolution, which allows the behavior of individual cells to be captured. The models can describe both steady state processes, such as transcription factor binding, and dynamic processes, such as the transcriptional machinery moving along the DNA. These generated models capture both the spatial and temporal behavior of the system being modeled.
- **My framework allows the models to not only capture the population averaged steady-state behavior, but also capture the dynamic behavior of individual components, as well as the emergent behavior arising from the components working together in a coordinated system.** My framework focuses on modeling at the nucleotide level within single cells. Instead of providing a single population averaged result as current modeling methods produce, I can capture the progression over time through different configurations of factors bound to the DNA in a single cell.
- **The interactions between components can be specified using spatially abstract descriptions.** The abstract descriptions of each component interaction is described independently of the actual DNA sequence or positions. Each interaction description is applied to each position of the given DNA sequence to generate the model. Details of the interaction, such as an interaction rate, can be tied to a func-

tion based on the local sequence at each position. Each abstract interaction can be described using a graphical form.

- **The DNA sequence is not a single molecule or component, but many interdependent nucleotide components.** The framework is designed to consider each individual nucleotide as its own component. Each nucleotide is not completely independent, as interactions that occur at a neighboring nucleotide have a large effect on the behavior on other nearby nucleotides. However, the extent of the effects are limited in scope at any specific nucleotide position and therefore makes the simulation of interactions along a large DNA segment trackable.

### 3.3 Previous Work

The different modeling perspectives and methods are discussed in greater detail in Chapter II. Here I briefly introduce the previous work on which I have built my framework.

Traditionally regulation of transcription has been modeled as gene regulatory networks (GRNs). These graph based systems capture the correlation between levels of one gene on the levels of other genes. Many methods exist for capturing the networks from the experimental data (reviewed in (Hecker et al., 2009)). These models answer the question of "which" genes are affected by change in levels of another gene. They cannot answer the questions of "why" or "how" the change in the level of one gene can affect levels of other genes. To answer these types of questions, the level of detail in the model must be increased to examine the mechanisms of transcriptional regulation.

Transcription in eukaryotes is complex with many factors contributing to the regulation of transcription at each gene promoter. Each factor has different affinities for different sequences of DNA. Factors bind to different lengths of the sequence and may be competing for overlapping sections of the sequence, but only one factor can be bound to each nucleotide of DNA at any one time.

A new class of models have recently emerged to capture the complex configurations of regulators bound to the DNA at nucleotide resolution. These models are designed to either capture the probability of all the different configurations using HMMs (Wasson and

Hartemink, 2009; Segal et al., 2006) or capture the molecular interactions occurring along DNA (positional) throughout time (temporal) (Coulon et al., 2013) using stochastic simulations.

Configuration based modeling systems have been developed (concepts reviewed in Segal and Widom (2009a)) to describe the positional binding configuration of proteins using either probabilities or thermodynamics. Currently, these models consider only transcription factor and nucleosome binding. With these two components, a given nucleotide can be unbound, bound in a nucleosome, or bound by a particular transcription factor (Figure 2.4). Conceptually, these states define a model where the transitions between the states depend on both the affinity of the component to the DNA sequence and its concentration (Wasson and Hartemink, 2009). As both nucleosomes and transcription factors bind to multiple nucleotide positions, the actual connectivity between states varies depending on the identity of the component in order to capture the specificity and length of binding. This is captured elegantly by an HMM and allows large DNA sequences to be quickly modeled on conventional computer resources (Segal et al., 2006; Wasson and Hartemink, 2009). These models have proven to be quite successful at elucidating key regulatory principles inherent in the competition between nucleosome and transcription factors (Segal and Widom, 2009a). However, these methods describe only the population averaged behavior of a DNA region and do not address the inherent temporal variation in configurations within a single cell.

Another class of stochastic models of gene regulation are just emerging (Mäkelä et al., 2011; Ribeiro et al., 2009; Zhu et al., 2007) that are specifically tailored to deal with complex configurations of regulators at individual loci within single cells. These models also seek to capture the molecular interactions occurring along DNA (positional) throughout time (temporal) using stochastic simulations.

One set of models focuses on the behavior of the transcriptional machinery in bacteria (Roussel and Zhu, 2006; Ribeiro et al., 2006). These model a generic DNA sequence containing a promoter and gene region. The transcriptional regulation in the highly evolved

promoters of bacteria usually have a single transcription factor that will activate transcription of the gene. Only when the factor is bound can the transcriptional machinery bind and progress through the initiation, elongation, and termination stages. These positional and temporal models capture competition at the all nucleotides of the promoter region by creating multiple states for the whole promoter. The tri-state promoter can be unbound, bound by activator, or bound by repressor.

Because the framework of these rule based systems is flexible, it is relatively easy to extend the models. Extensions have created more complex details of the transcription process, such as transcriptional machinery pausing, arrest, and early termination or have added more processes. Including processes such as the translation in the models easily extends the model's coverage to more biological behavior (Mäkelä et al., 2011).

However, there is a big drawback to these rule based models. They are computationally expensive, as every molecule requires rules to describe the set of all possible interactions. This leads to a combinatorial explosion of possibilities as the number of components in the model increases. This is particularly true for the central molecule of transcriptional regulation: DNA. The models are explicitly built to abstract the promoter region of bacteria and ignore the diversity of DNA binding factors, which increase the complexity of regulation found in eukaryotes.

Another drawback is the manual creation of models for every gene of interest. While this is simple to manually manage models with a small set of genes and transcription factors, it becomes tedious, monotonous, and error prone when many factors are required to regulate each gene. Models containing many genes are required to understand the system behavior. Manual curation of the models is no longer feasible for creating or maintaining models for explicit sequences of genomic DNA.

I use the Petri net formalisms to describe the individual interactions of each component. Stochastic Petri nets have been utilized successfully to model diverse biological processes, including metabolic networks, signaling pathways, and gene regulatory networks (Lei, 2011; Genrich et al., 2001; Mura and Csikász-Nagy, 2008; Ovacik and Androulakis, 2008; Ruths

et al., 2008). The focus of this work is on the generation of the biochemical rules and not on the use of Petri nets as a description language.

## 3.4 Methodology

### 3.4.1 Framework

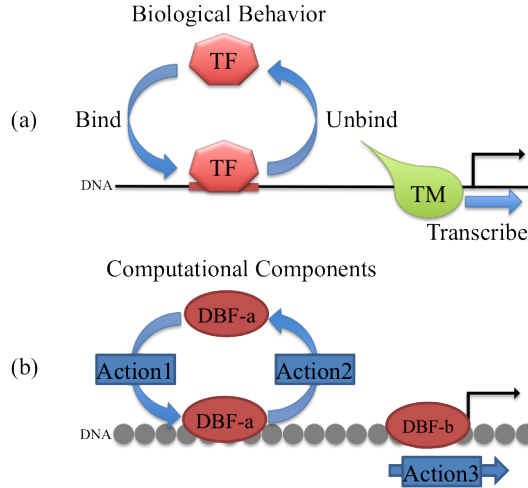
I sought to construct a modeling framework that integrates the sequence dependent positional information of DNA binding factors with the inherent temporal dynamics of the transcriptional machinery. The configuration based modeling paradigm can capture steady state positional interactions, but must be extended to capture dynamic events, such as the movement of the transcriptional machinery. These dynamic events are not only positional dependent, but also temporally dependent (Figure 3.1). For example, consider polymerase traversing DNA in the process of transcription. The location of polymerase is time dependent and its ability to move along the DNA is influenced by the state of the nucleotides ahead (Kireeva et al., 2005).

Capturing the temporal and positional dimensions simultaneously within the HMM framework leads to an explosion of alternative states because each possible movement in the temporal dimension impacts multiple positional states (Figure 3.2). Consequently, I sought an alternative representation that simplifies capturing both dimensions.

In reality, DNA is a chain of nucleotides. The key observation is that individual DNA binding proteins interact with a short contiguous string of bases. Because of steric constraints, only a single binding protein can interact with a given nucleotide at a given time. Therefore, by treating DNA not as a single molecule but as a string of entities, I can decouple the actions of one DNA binding protein from another, so long as they interact sufficiently far away from each other.

Where the factors interact along the DNA is determined by the sequence. Each of the DNA binding factors has specific affinities for different sequences. The locations and probability of each factor binding can be calculated using the PSSM for that factor.

The rule based models capturing transcriptional machinery dynamics can be extended to include the DNA binding factors. Like the HMM models, the binding positions and the



**Figure 3.1: Biological behavior can be modeled by action oriented local descriptions.** (a) Biological detail of individual factors, such as transcription factors (TF) and the transcriptional machinery (TM), has been well studied experimentally. Each biological component interacts with a small region of the DNA molecule. (b) Creation of a positional and temporally realistic model can be achieved by focusing on the actions of individual components (DBF-a, DBF-b). These actions have a localized effect, influencing a finite number of nearby nucleotides within the underlying DNA. My framework captures the interactions between individual computational components, such as the binding and unbinding of transcription factor (TF, DBF-a) to consecutive DNA nucleotides (gray circles) and the movement of the transcriptional machinery (TM, DBF-b) along the DNA.

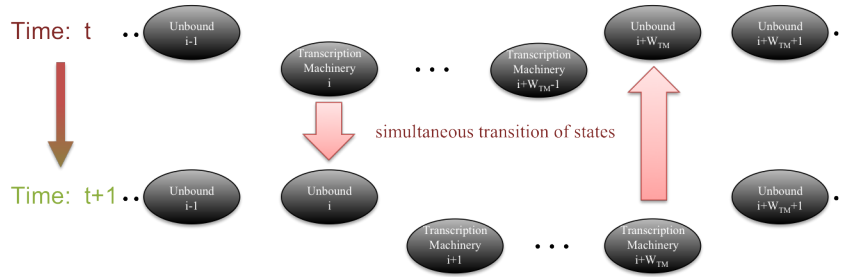
probability of binding can be calculated for any given DNA sequence. These can be used to generate the rules for binding.

### 3.4.2 Representation

My framework uses the rule-based methodology that defines all the interactions between the DNA and factors as chemical equations. Equations 3.1 and 3.2 show a reversible interaction as two interactions: the first combines two components to create another component and the second is the opposite action that separates the components back into individual factors. Each of the interactions has a rate ( $K_a$  and  $K_d$ ) at which that interaction occurs. These actions need to be defined for every interaction possible between components of the model.





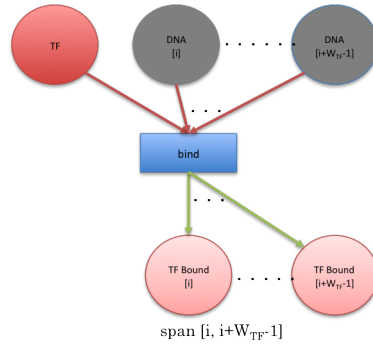


**Figure 3.2: Adding temporal information to the positional HMM requires simultaneous transitions at multiple states.** At any point in time  $t$ , the configuration of factors bound to the DNA places each nucleotide  $i$  in a single state, as described in Figure 2.4. When a component moves through time, such as the movement of the transcriptional machinery along the DNA, this requires the simultaneous transition of two positions,  $i$  and  $i+W_{TM}$ , in the positional HMM.

In my modeling perspective, the rules define transitions of the nucleotides from one state to another. To build a realistic model, every transition possible must be defined for every nucleotide position. This would limit the size of the models, if they are built by hand. But, the interaction rates can be calculated for each position in the sequence and the rules can be automatically generated.

I use Petri nets to focus on describing processes from an event centric perspective (Figure 3.3). This shifts the focus squarely to the dynamic temporal events permissible for a particular component, while still allowing the events to be decomposed into their local positional effect. The use of the graphical representation creates easy-to-understand interactions and still allows the framework to be highly flexible and extensible.

The full set of abstract rules (described in Appendix 1) are applied to a specific DNA sequence to obtain a complete set of model rules that can be simulated by off-the-shelf stochastic simulation engines. Each temporal event, such as binding, depends on the current state or local configuration of the DNA (preconditions). Petri net figures in the appendix also specify the positions of the DNA influenced by each rule (its span). Execution of an event results in the described change to the local configuration. To create a runnable simulation, these abstract descriptions are applied to a specific DNA sequence to generate a set of rules defining the permissible molecular interactions. An additional advantage of



**Figure 3.3: Petri net description of transcription factor binding.** This Petri net representation describes a binding action that takes a transcription factor and unbound DNA positions to produce a set of TF bound positions of DNA. The actions in a Petri net are represented as rectangles and the components as circles. Arrows from a component into an action are the required inputs for that action. The arrows from an action to a component are the output components of the action. Each TF binds to consecutive nucleotides for a length matching the area occluded by the TF. This binding action consumes a molecule of the TF and unbound DNA for each occluded position. When the action is performed, it produces a molecule of TF bound DNA for each position.

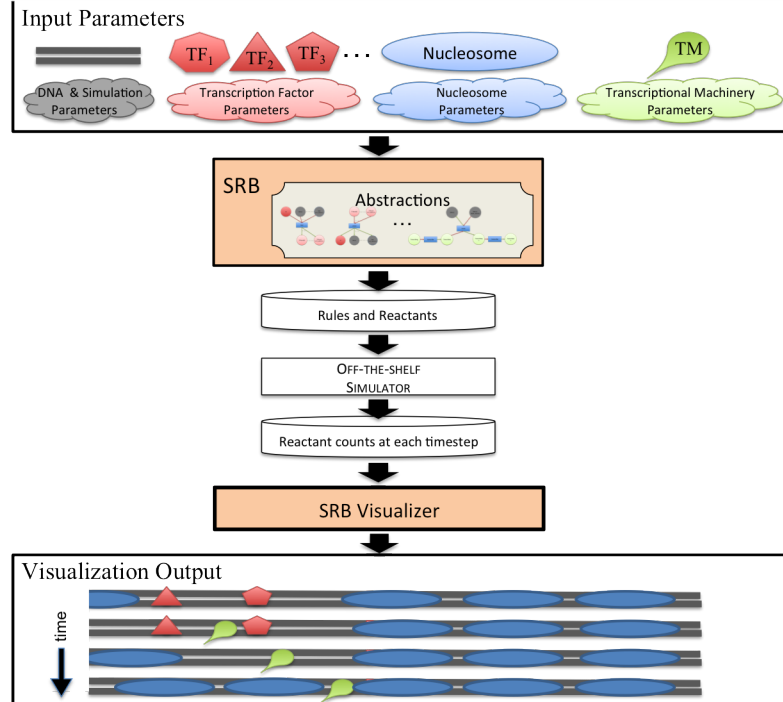
using simulations is it allows us to visualize the configuration of each position at each time step, as well as infer the dynamic movement of factors along the DNA (Figure 3.7).

### 3.4.3 Implementation

The representational framework defines a set of interaction rules that are independent of the DNA being modeled. In this section, I describe how to apply the rules to a specific DNA sequence, simulate the resulting system of interactions, and analyze the results. The complete overview of this process is shown in Figure 3.4.

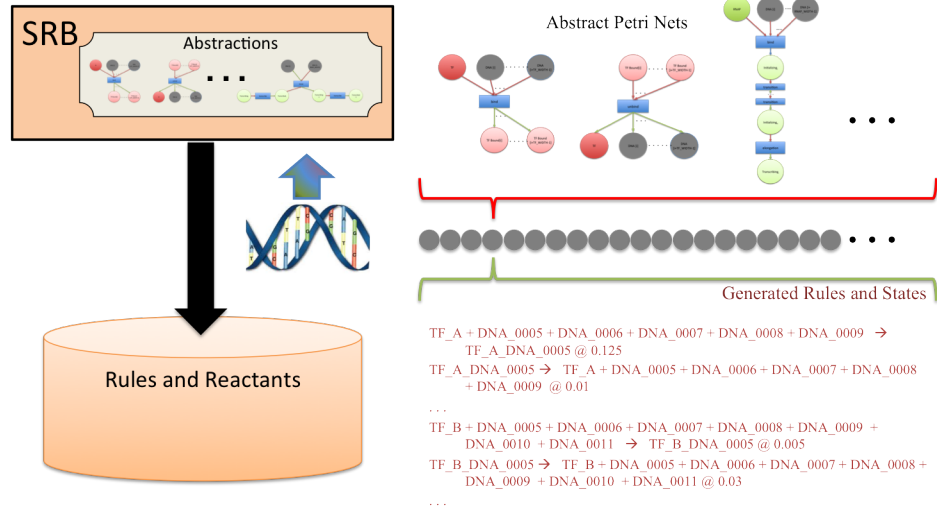
#### 3.4.3.1 Stochastic Rule Builder

The stochastic rule builder (SRB) converts the abstractions into a runnable set of rules specific to a particular DNA sequence. Given my framework, the creation of a specific executable model becomes a conversion process, similar to a compiler. The SRB takes as input the specific DNA sequence, a configuration file, and the necessary parameters to build a complete set of rules for the simulator (Figure 3.5). The SRB utilizes the generic action descriptions to generate rules in the form of chemical reactions where each rule specifies a set of reactants that are transformed at some rate ( $K_a$ ) to produce a set of resultants (Figure 3.3).



**Figure 3.4: Flowchart depicting the Stochastic Rule Builder (SRB) and visualization pipeline.** The SRB encapsulates the abstractions (Petri net descriptions, Appendix I) for all the interactions and applies those interactions to specific sequences of DNA (grey bars) using the user provided component parameters (input parameters). Output of the SRB is both a set of reactants representing all the different states of the components and the rules using those reactants. The rules are simulated using an off-the-shelf simulation engine, which produces an output file containing each reactant’s molecular count at each time step. These molecular counts are interpreted by my Visualizer to generate the configuration of the components along the DNA at each time step (visualizer output).

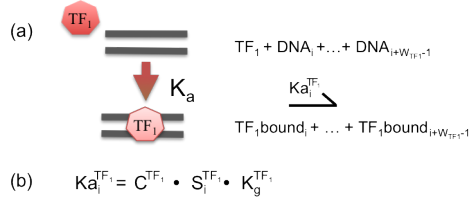
For each generic component, the SRB creates sequence specific rules for each nucleotide of the sequence. Consider a transcription factor as an example. A transcription factor has particular sequences to which it preferentially binds, described by a PSSM. This matrix specifies the number of positions, as well as the nucleotides that are preferentially bound at each position. At every position of a specified DNA sequence, the SRB utilizes the PSSM to calculate how well the transcription factor could bind. Figure 3.6 shows an abstract rule for the binding of a transcription factor to the DNA. When a rule is generated for the binding of this transcription factor to a given DNA position, a binding rate ( $K_a$ ) must be specified. The theoretical rate ( $K_a^{TF_1}$ ) for a specific sequence can be defined as a function that takes into account the factor’s concentration ( $C^{TF_1}$ ), its strength of binding the specific sequence



**Figure 3.5: The Stochastic Rule Builder (SRB) generates a set of biochemical rules for a given DNA sequence.** The logical flow (on the left) shows how the SRB applies its internal Petri Net representations of actions to a specific DNA sequence to produce an output file containing the reactants of the model system, as well as the rules for interactions among those reactants. Each set of possible actions (the Petri Nets) are applied at every position along the DNA, as depicted on the right. In this case, the rules are generated for an off-the-shelf simulation engine Dizzy (Ramsey et al., 2005).

( $S_i^{TF_1}$ , inferred from the PSSM), and a generic association rate ( $K_g^{TF_1}$ ) (Figure 3.6 b). The SRB applies a similar process to the nucleosome abstraction using the nucleosome affinity scoring matrix (Wasson and Hartemink, 2009) to define sequence preferences. It should be noted that the concentration is not used in specifying the rate for each rule generated, because the simulation engine uses the supplied rule rate for each set of molecules available at any time point in the simulation.

Each rule generated by the SRB uses a set of reactants and resultants, which define the possible states of each nucleotide. Each nucleotide can only be bound by a single factor at a given time, which is a temporal constraint in the framework that is based on biophysical constraints. The SRB must ensure that each bound state of the nucleotide is mutually exclusive by augmenting the state names with the bound factor name. This is conceptually similar to each nucleotide having a limited number of possible states in the HMM formalisms (Figure 2.4).



**Figure 3.6: Generation of the biochemical interaction rules requires interaction rates to be a function of the specific sequence.** (a) The biochemical interaction rules generated by the binding of a specific transcription factor (TF1) at position  $i$  of a specific DNA sequence ( $DNA_i + \dots + DNA_{i+W_{TF}-1}$ ) to produce bound factor ( $TF1\_bound_i + \dots + TF1\_bound_{i+W_{TF}-1}$ ). (b) The position and TF specific reaction rate ( $Ka_i^{TF1}$ ) is a function of the transcription factor concentration ( $C^{TF1}$ ), the transcription factor's affinity for a specific sequence of DNA ( $S_i^{TF1}$ ) and the generic association rate ( $K_g^{TF1}$ ) of the transcription factor.

I have also extended the behavior of the transcriptional machinery. In previous rule based models the machinery was modeled in only a single direction along the DNA and ignored the possibility of transcription along the opposite strand. My framework specifies the interactions for transcription in both directions. This includes the behavior of the transcription machinery moving along the DNA, encountering DNA binding factors, and other transcription machinery moving in the opposite direction (known as transcriptional interference).

The transcriptional machinery is capable of loading onto DNA and starting transcription at any position. However, it also traverses DNA and terminates transcription at given rates. I have set the default rates for transcribing based on the in vitro experimental data (Tolić-Nørrelykke et al., 2004). In addition, there are cases where specific transcription factors can bind to DNA and recruit the transcriptional machinery to a nearby position (Bryant and Ptashne, 2003). For example, TATA binding protein (TBP) recruits the transcriptional machinery to a DNA position roughly 35 bases downstream of its position. Therefore, the SRB allows each transcription factor to recruit the transcriptional machinery to a position at a specified distance (shown in appendix Figure A6).

An interesting modeling issue arises when the transcriptional machinery encounters obstacles (other components) during its movement. Given the stochastic nature of factor binding, the transcriptional machinery could just wait until the obstacle removes itself. However,

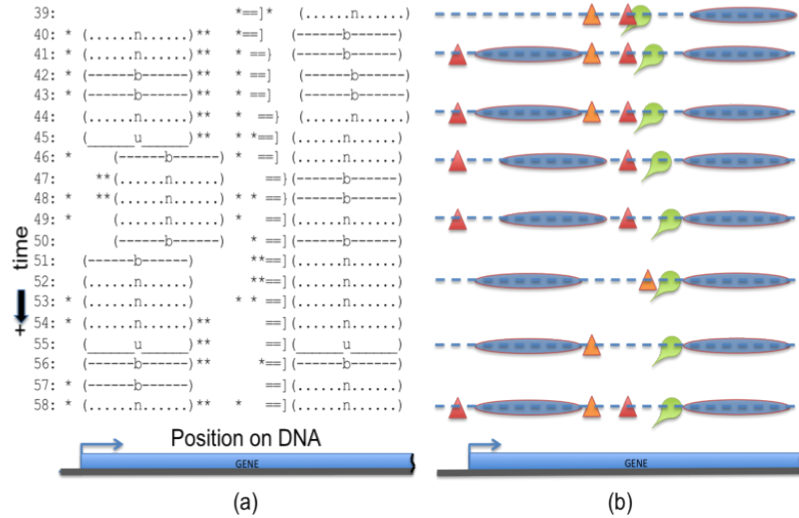
it is more likely that the transcriptional machinery actively removes obstacles (Schwabish and Struhl, 2004). Consequently, I allow the transcriptional machinery to modify the eviction probabilities of a protein when it encounters an obstacle. When the obstacle is another transcriptional machine traversing in the opposite direction, this results in transcriptional interference (Gullerova and Proudfoot, 2010; Sneppen et al., 2005). For simplicity, in my current implementation transcription will abort in both directions when interference occurs.

### **3.4.3.2 Visualizing Simulations**

The models generated by SRB are simulated using DIZZY (Ramsey et al., 2005), which produces an output containing all the molecule counts of each component state at each time step in the simulation. The number of states generated can quickly grow into the thousands for short sequences. The counts must be interpreted to understand the behavior of the system. I created a second application to visualize the results of simulation and collect statistics to be reported. The simulation results are interpreted to show the configuration of factors bound to the DNA at each time point (Figure 3.7). The visualization application also calculates the occupancy of factors by summarizing the percentage of time each factor occupies an individual nucleotide.

### **3.4.3.3 Coping with Parameters**

An inherent obstacle to my approach is the large number of necessary parameters (Tables 3.1 and 3.2). Whenever possible, I obtain the default parameters from the literature. For example, my modeling of the yeast genome uses protein counts obtained from Ghaemmaghami (Ghaemmaghami et al., 2003). The PSSMs for transcription factors were obtained from MacIsaac (MacIsaac et al., 2006) and Badis (Badis et al., 2009). Nucleosome affinity is from Kaplan (Kaplan et al., 2009) and Wasson (Wasson and Hartemink, 2009). In all of these cases, the datasets were generated from population-averaged experiments, typically with cells grown in standard yeast media (YPD) and measured during log phase growth. Therefore, they are considered baseline default values that can be overridden by the user in my modeling configuration file. User specified custom motifs are also supported, allowing newly discovered proteins to be quickly incorporated.



**Figure 3.7: Visualizing the DNA configurations at each time step of a simulation.** The state of the simulation is recorded for each time step of the simulation (y-axis) and can be interpreted in light of my Petri Net descriptions of actions to determine the current configuration of factors bound to the DNA (x-axis). Here I show an example of a simulation that included 3 transcription factors, nucleosomes, and the transcriptional machinery. (a) A simple ASCII representation where each character summarizes 10 nucleotides of the DNA. ASCII symbols are: \* bound transcription factor; ==} bound transcriptional machinery (position not transcribed); ==] bound transcriptional machinery (position transcribed, waiting to advance); (.....n.....) bound nucleosome. Nucleosomes are further labeled (binding (b), unbinding (u), or stable (n)) to reflect the intermediate states of nucleosome formation. When the image is viewed as a whole, scanning from top to bottom, I observe movement of a single transcriptional machine along the DNA and pausing at a nucleosome with perhaps strong sequence affinity. (b) An alternative representation of the ASCII art, redrawn into cartoon representations of the DNA configuration in relation to a given gene region (blue rectangle). Each representation of an associated time step shows triangular transcription factors (in red and orange), blue oval nucleosomes, and a green teardrop for the transcriptional machinery.

The temporal events also require rates (Figure 1.5). In general, these rates are largely unknown and currently difficult to estimate from the literature. While recent advances in experimental approaches (Coulon et al., 2013) show tremendous promise in rapidly addressing this parameterization issue, currently many of the kinetic parameters have not been experimentally determined. Some of the factor ‘on’ rates are known, but very few ‘off’ rates (or residency times) are known (Lickwar et al., 2012b). I estimate these rates by tying them to other well-studied parameters. Experimental studies indicate that the temporal information is unlikely to be fully independent of the positional information, as the sequence being bound influences the kinetics of the reaction (Lickwar et al., 2012b; Lieb et al., 2001).

Therefore, I assume that higher affinity sites will also have higher residency times (Lickwar et al., 2012b). A site with strong sequence preference will bind more frequently and for a longer period of time. These rates can be overridden by user defined global rates or data derived position dependent information provided in the configuration file.

#### **3.4.3.4 System Overview**

The SRB generates potentially thousands of chemical reaction rules for small DNA sequences. These rules can then be fed to off-the-shelf simulators (Figure 3.4). In my case, the SRB output is formatted for the third-party stochastic simulation engine Dizzy (Ramsey et al., 2005). Briefly, I use the Next Reaction Method (Gibson and Bruck, 2000), which is an extension of an exact stochastic simulation algorithm (Gillespie, 1976). In these simulations, the state of the system is represented by the set of current molecule counts of each reactant and the time until each of the reactions will occur. The times are selected from the probability distributions for each reaction, which are based on the rate of the reaction and the current molecular counts of the reactants. After initialization of the internal data structures, the algorithm repeatedly selects the next reaction to occur and applies that reaction. Each reaction changes the molecule counts, which may affect the probability distribution of many other reactions using the same reactants. The algorithm recalculates, as needed, the time to next reaction for all the affected reactions. As many independent reactions could occur nearly simultaneously, many reactions may be applied during each time step. At the end of each time step, the current molecule counts for all reactants are reported. The algorithm continues to apply reactions until the user specified number of time steps has been reached.

In my models, each possible state of a nucleotide becomes a possible reactant. The size of my models therefore depends on the size of the DNA segment being modeled and the number of different protein components being used. Because each action influences only a localized span of nucleotides, the worst case is always a linear growth in the number of rules (Figure 3.8). When all possible rules are applied at every position, this growth is still substantial (Figure 3.9).



**Table 3.1:** Modeling parameters for SRB.

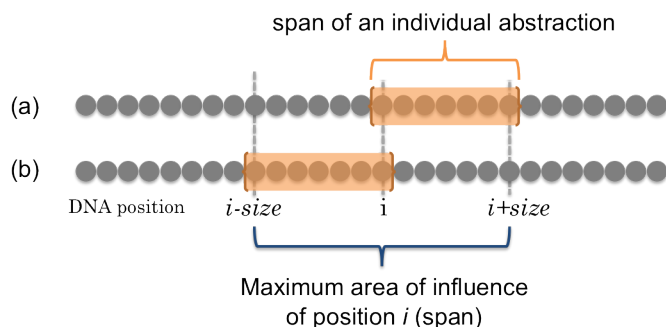
| Attribute Name                               | Example of value expected | Notes   |
|--|---------------------------|---|
| SRB parameters                               |                           |   |
| MODEL_NUCLEOSOME                             | 1                         | If non-zero, add nucleosomes to the model.  |
| MODEL_RNAP                                   | 0                         | If non-zero, add RNAP complex to the model.   |
| MODEL_TATA                                   | 0                         | If non-zero, add TATA to the model. TATA recruits and stabilizes the RNAP complex.  |
| MOTIF_THRESH                                 | 0.8                       | Threshold (percent of max PSSM score) that must be met to generate TF rules at a position.  |
| GROUPING                                     | 10                        | Number of nucleotides being grouped into a single position within the model.  |
| DNA parameters                               |                           |   |
| FILE   | DATA/S288C_R64.fasta      | FASTA formatted file containing DNA sequence.   |
| CHR  | chrVII                    | Chromosome within FASTA file.   |
| START  | 140000                    | Start position within chromosome.   |
| END  | 146000                    | End position within chromosome.   |
| LENGTH                                       | 6000                      | If END not specified, LENGTH nucleotides from the start position is used.   |
| Transcription Factor parameters              |                           |   |
| TF_LIST                                      | REB1, MCM1, RSC3          | Names of TFs to include in the model.   |
| MOTIF_FILE                                   | DATA/yeast.tamo           | (Gordon et al., 2005) Specify the file to be used to define motif.  |
| MOTIF_THRESH                                 | 0.7                       | Threshold that must be met to generate TF rules at any position. Overrides the global SRB settings.   |
| PROTEIN_COUNTS                               | YeastProteinCounts.txt    | (Ghaemmaghami et al., 2003) File from which default molecule counts are initialized.  |
| LOCAL_CONCENTRATION                          | 1                         | Adjust the local concentration of every component [0..1].   |
| OCCLUSION_5                                  | 3                         | Every TF has a footprint along the DNA extending beyond the motif. I assume there could be a non-symmetrical footprint. These set the generic defaults for used for all TFs in the model. Number of nucleotides beyond the 5' end of the motif match. |
| OCCLUSION_3                                  | 3                         | Number of nucleotides beyond the 3' end of the motif match.   |
| Parameters below are replicated for each TF: |                           |   |
| INITIAL_COUNT                                | 4                         | Number of TF complexes in the model   |
| MOTIF_THRESH                                 | 0.8                       | Threshold that must be met to generate this TF's rules at a position. Overrides the global SRB and TF settings.   |
| MOTIF  | TGTNNNNNNNACATCA          | Explicitly defines motif for a TF. (MOTIF_THRESH is ignored)  |
| MOTIF_PWM                                    | DATA/PWM/Gal4.pwm         | (Gordán et al., 2011) Loads motif from given PWM file.  |
| OCCLUSION_5                                  | 3                         | Number of nucleotides beyond the 5' end of the motif match. Overrides the generic TF default value.   |
| OCCLUSION_3                                  | 3                         | Number of nucleotides beyond the 3' end of the motif match. Overrides the generic TF default value.   |
| ON_RATE                                      | 0.965                     |   |
| OFF_RATE                                     | 0.035                     |   |
| RNAP_EVICTS                                  | 1                         | If non-zero, add TF eviction by RNAP complex to the model.  |
| RNAP_RECRUIT                                 | 0                         | If non-zero, the TF recruits the RNAP complex.  |
| RNAP_OFFSET                                  | 30                        | If TF recruits RNAP, the offset from the TF where RNAP is bound.  |
| RNAP_RECRUIT_RATE                            | 0.00001                   | Rate to recruit the RNAP complex to TF bound.   |

**Table 3.2:** Modeling parameters for SRB (cont).

| Attribute Name                       | Example of value expected | Notes   |
|--------------------------------------|---------------------------|---|
| Nucleosome parameters                |                           |   |
| N_HISTONES                           | 100000                    | Number of histone complexes in the model.   |
| SIZE                                 | 147                       | Size in nucleotides of the complete Nucleosome complex.   |
| MIN_LINKER_SIZE                      | 18                        | Size in nucleotides of the minimum space between nucleosomes.                                     |
| ON_RATE                              | 0.0001                    |   |
| OFF_RATE                             | .1                        |   |
| ABORT_RATE                           | 1000.                     |   |
| NUC_PROB_FILE                        | DATA/experimental.txt     | Use the experimental data to set the relative probability at each position.                       |
| DL_NT_NUC_PROB_FILE                  | dinucleotide_probs.txt    | Use the dinucleotide probabilities to set the relative probability at each position.              |
| Transcriptional Machinery parameters |                           |   |
| INITIAL_COUNT                        | 4                         | Number of TF molecules in the model.  |
| RNAP_SIZE                            | 25                        | Size in nucleotides of the RNAP complex.  |
| N_INIT_STAGES                        | 5                         |   |
| INIT_RATE                            | 0.1                       |   |
| INIT_ABORT                           | 0.002                     |   |
| ON_RATE                              | 0.05                      |   |
| OFF_RATE                             | 0.0008                    |   |
| TRANSCRIPTION_RATE                   | 30                        |   |
| POSITION_FILE                        | DATA/experiment.bed       | File containing positions of initiation.  |
| POSITIONS                            | [141850,w],[144100,c]     | List of explicit positions at which initiation occurs (only used if POSITION_FILE not specified). |

The number of rules produced can be reduced by the application of biologically realistic and reasonable heuristics. For example, while in theory a transcription factor can bind to any location, it is anticipated that significant interactions will only occur with sequences with reasonably good fit to the PSSM (Qi et al., 2006). Applying a score cutoff to the PSSM reduces the number of rules significantly, but makes the number of rules strongly dependent on the underlying sequence. I also optionally allow rules to be applied to groups of nucleotides, a heuristic that drastically reduces the number of rules produced for a specified length of DNA, but also reduces the precision of the simulation. My default system typically generates hundreds of thousands of interaction rules for a typical gene and millions of rules for chromosomes.

There are practicalities with using a simulation method that must also be considered. First, I initialize all the DNA states as unbound and allow the system to populate binding



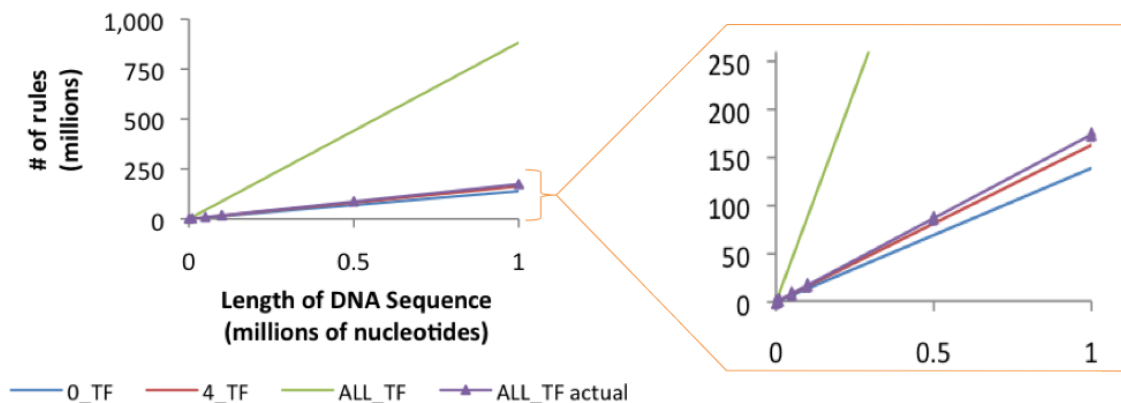
**Figure 3.8: Each DNA position influences a limited number of rules.** Focusing at the nucleotide level, the model captures the fact that each nucleotide (grey circle) can be in only one of a limited number of states. (a) Each abstraction (orange box) has a span of nucleotides that are influenced when the Petri net is applied at position  $i$ . (b) Therefore, a chance at any position  $i$ , influences a fixed window size  $[i-size, i+size]$  of nearby nucleotides, defined by the span of the largest abstraction.

factors until a quasi-equilibrium is reached. Through observation, I have found 500 steps to be sufficient for this burn-in period. Each simulation is run for a minimum of 8000 steps, to allow for sampling of a variety of pathways. Finally, because I frequently simulate small sections of DNA, there are edge effects. The behavior of the model is altered at the edges of the model (beginning and end of the DNA region) due to an absence of defined rules for positions outside the modeled region. The natural barrier of the edges of the region can be felt for a distance along the DNA. In my experience, padding the edges by two thousand nucleotides, a distance in excess of three times the largest rule span, reduces edge effects.

The stochastic simulation engine, Dizzy, requires the rules and reactants description file as input, as well as specification of the method, number of time steps, and the sampling rate. Each simulation results in a different series of events and describes one possible trajectory of the given DNA through time. Dizzy's implementation is a Monte Carlo implementation and scales logarithmically with the number of rules (Ramsey et al., 2005). While I have chosen Dizzy because of its availability and relative ease of use, the SRB output could easily be revised to use any off-the-shelf SSA simulator (FERN-Erhard et al. (2008), NFsim-Sneddon et al. (2011), DYNSTOC-Colvin et al. (2009)).

Finally, the output results of the simulations can be summarized or visualized. Running the simulation multiple times or for an extended period of time will stochastically select

## Size of Generated Models



**Figure 3.9: Model size grows linearly with the size of the DNA sequence being modeled.** The main graph shows the maximum size of models generated for 124 transcription factors (TFs) (green), a set of 4 TFs (Reb1, Rim101, Ste12, and Tec1) (red) and with only the nucleosome and transcriptional machinery (TM) interactions being modeled (blue). These worst-case scenarios generate every rule at every position for every TF, even when the TF binding would only be weak, rare and transient. By default, my system only generates rules when the likelihood of binding exceeds a strength threshold filter, resulting in much smaller models even when using all 124 TFs (purple). The inset graph is a close up of the deviation of actual from the theoretical as the length of the DNA increases.

a different sequence of events, leading to alternative trajectories. Combining many time points allows for summary statistics, such as the distribution of binding configurations, to be obtained and compared to experimental data. In addition, the simulation results can be interpreted in light of the possible actions described by my Petri nets, allowing for the visualization of the configuration of each position at each time step, as well as inferring the dynamic movement of factors along the DNA (Figure 3.7).

### 3.4.4 Modeling Details

Transcriptional regulation is inherently dependent upon the biochemical interactions of many different molecules, but robust enough to handle the stochastic fluctuations inherent in a molecular system. The resulting cell-to-cell variability is likely fundamental to most, if not all, molecular cellular processes (Huang et al., 2009; Schwabe et al., 2011). I wanted to develop a quantifiable, interpretable, and flexible model of transcription regulation. My framework strives to capture the distinct behaviors and interactions among the components

being modeled as abstract rules that may be applied to any DNA sequence. These abstract rules are then applied to the specific DNA sequence to obtain a complete set of model rules that can be simulated by off-the-shelf stochastic simulation engines. Here I describe, in more detail, the Stochastic Rule Builder (SRB). The SRB applies the abstractions (Appendix 1) to a specific DNA sequence, resulting in a set of rules defining the permissible molecular interactions on that DNA sequence.

Most biochemical models have many molecules of each reactant interacting. But, in my generated models most of the reactants have a molecule count of zero because there is only one molecule being shared among the set of reactants for a single position. Consequently, my models are very sparse as most of the reactions specify mutually exclusive states of each particular nucleotide of DNA. This implies that most of the reactions within the model are not applicable at any time. Therefore, for a given sequence of unbound DNA, the simulation is selecting one of the possible reactions available at those positions and changing the single molecule counts from the unbound DNA reactant to the bound by a specific component reactant. Application of any reaction to the DNA immediately blocks all other previously possible reactions (implementing competition). Now the only reactions available at these positions are reactions based on that bound component. My models can be viewed as two models in one:

- 1) when DNA is unbound, the rules are using population averaged behavior of multiple molecules interacting within the system;
- 2) once a generic molecule has bound to the DNA, the model switches to modeling the specific behavior of that component bound to DNA.

#### **3.4.4.1 States in the Model**

The framework must apply all the action abstractions to a specific DNA sequence and insure that the generated rules all reference the correct nucleotide positions and states. Here I discuss the mechanism for generating the unique state names, the set of actions currently being modeled, and the method for calculating the interaction rates when they are dependent on the specific DNA sequence.

The DNA is central to my modeling, as it is also central to transcriptional regulation. While the concentration of all other components varies based on condition or cell type, the number of copies of the DNA per cell is largely defined by the organism and tissue identity. The DNA encodes the instructions for when, where, and how much of each transcript to produce, while the other components bind with the DNA to carry out those instructions. My models focus on the behavior of individual nucleotides, which can be in only a small number of different states. I must represent the state of each DNA nucleotide as a unique reactant in my biochemical interaction system. Each nucleotide can only be in a single state at a given time and my framework must maintain this constraint while building the rules for a specific model. I accomplish this by generating unique, but deterministic, names for each state at each position of the modeled DNA sequence.

My models are built from the four abstract classes of factors: nucleotides, transcription factors, nucleosomes, and transcriptional machinery. Below are the abstract names used to create unique names for each nucleotide position. I create placeholders for the variables that are substituted during model building by placing them within special characters. For example, %position% is replaced with the position identifier.

#### **3.4.4.2 Unbound Nucleotide State**

The modeling framework treats DNA as a linear sequence of nucleotides. Each nucleotide can be in only one of a limited number of states (Table 3.3). Most of the interactions require each nucleotide to return to the unbound state before interacting with another component.

#### **3.4.4.3 Transcription Factor Bound States**

Transcription factors are proteins that recognize small segments of DNA (typically 4-20 nucleotides) (Badis et al., 2009; Bulyk et al., 2001). They often work in groups or complexes, allowing for varying degrees of control over the transcriptional process. Transcription factors can function as activators or repressors of transcription. A position specific scoring matrix (PSSM) is typically utilized to describe the sequence affinity preferences of each DNA binding protein. The interaction of a regulatory protein with DNA is transient, as these

**Table 3.3:** Nucleotide States.

|  |
|--|
| Unbound States<br>DNA_-%position%  |
| TF Bound States<br>%tf-name%_bound_-%position%<br>TF_bound_-%position%   |
| Nucleosome Bound States<br>nuc_bound_-%position%<br>nuc_binding_-%position%<br>nuc_stable_-%position%<br>nuc_unbinding_-%position% |

factors are thought to bind and release frequently (Berg et al., 1981). The physical binding of a regulator to DNA depends on not only its PSSM, but also its cellular concentration. At higher concentrations, the best matches to the PSSM will become saturated and the protein will more likely bind to lower affinity sites. By default, I have utilized the TAMO data, which contains 124 transcription factors (Gordon et al., 2005). I must create a reactant for a nucleotide bound to each named transcription factor.

For simplicity, I only maintain the transcription factor bound states at a single position even though the transcription factor spans multiple positions. As shown in Figure 3.8, when the transcription factor binds at position  $i$ , the state of position  $i$  is changed from unbound to bound by a transcription factor, and positions  $i+1$  through  $i+W-1$  (in this case  $W=4$ ) are no longer available for other interactions. The rules for the unbinding interaction will restore all the nucleotides to the unbound state and remove the transcription factor bound molecule at position  $i$ .

Each DNA binding protein interaction involves multiple nucleotides of the DNA. The binding of a transcription factor or a nucleosome is not limited to a single nucleotide, but a number of connected nucleotides. Each factor has a specific number of nucleotides that are

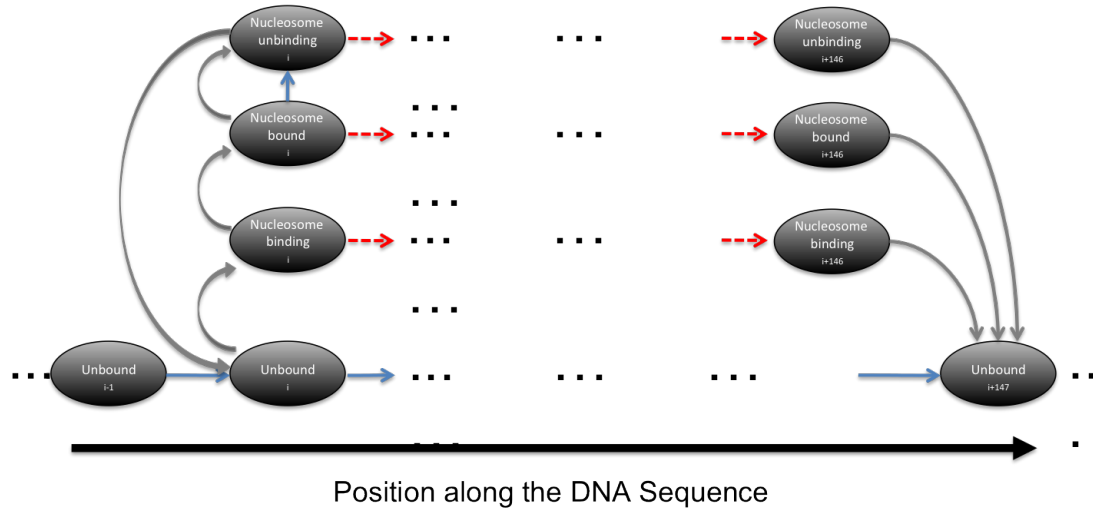
affected when it is bound. Physically, a bound transcription factor molecule will keep other molecules from accessing more nucleotides than just the length of its motif. Therefore, the length of each factor's interaction is determined by its PSSM and an additional number of nucleotides that are occluded (Hesselberth et al., 2009). I model these occluded regions by defining the window of nucleotides to be impacted by a binding event. The size of the additional nucleotides occluded can be generically set for all transcription factors and/or specifically for individual transcription factors.

#### **3.4.4.4 Nucleosome Bound States**

Nucleosomes are an additional regulatory component in eukaryotes, which form stable structures by wrapping DNA around histone proteins. The pliability of DNA is sequence dependent, which creates an implicit probability of nucleosome formation (Drew and Travers, 1985; Morozov et al., 2009). Therefore, I can treat its binding in a similar fashion to a transcription factor, only the size of the binding ( $W$ ) is much greater (typically 147 nucleotides). However, nucleosome formation is known to take more time than transcription factor binding and unbinding (Hager et al., 2009). To represent the time between initial binding of histones to the DNA and formation of a stable nucleosome, I introduce intermediate states whose purpose is to extend the time necessary for complete formation of the complex. There are two different stages in the formation of a stable nucleosome. The first stage encompasses the binding of the nucleotides onto the DNA and occluding other interactions. At this point, however, the nucleosome is not yet a stable state. The transition to a stable nucleosome takes time, which is enforced by the use of multiple states, each with individual rates for proceeding to the next state (Figure 3.10).

Nucleosome formation transitions from unbound to binding when forming a stable nucleosome. The removal of a nucleosome follows a similar trajectory through an unbinding state before reaching the fully unbound state. Similar to the transcription factor, only the first nucleotide position is transitioned to new state and all other 146 positions are assumed to be in same state.





**Figure 3.10: Nucleosome formation takes multiple time steps to occur, which is implemented by using multiple states.** Additional states are introduced to extend the time necessary for a nucleosome to bind. Initial binding makes the window of nucleotides unavailable for other interactions, but is not stable. A separate transition is necessary for the formation of a stable nucleosome. Likewise, removal of the nucleosome is a two step process through an unbinding state.

### 3.4.4.5 Transcription Machinery Bound States

The final component in my system is the transcriptional machinery. The movement of the transcriptional machinery along DNA is a highly dynamic process that pauses, shows variable processivity, and likely evicts DNA binding factors that impede its forward progress (Coulon et al., 2013). Consequently, this component differs from transcription factors or nucleosomes. As it travels along the DNA, it can influence the state of other bound factors. In fact, when one transcriptional machine directly impacts a second transcriptional process, this interaction is referred to as transcriptional interference (Prescott and Proudfoot, 2002). The kinetics of these events is often inherently local and intrinsically stochastic.

Once the transcriptional machinery binds, it transcribes along the DNA in a single direction. Therefore, there must be components for each direction (or strand) of DNA. Using the yeast convention, I call one strand the Watson (w) strand and the other the Crick (c) strand. All the possible transcriptional machinery states must exist in both Watson and Crick form to allow for movement in either direction. Note that I differentiate

the state names using a lower case c or w for the specific transcriptional machinery direction (Table 3.4).

Biologically, the transcriptional machinery is thought to exist in at least four distinct forms: the loading form, the initiation form, the elongating form, and the terminating form. These forms differ in the composition of subunits present in the larger transcriptional machinery complex. For simplicity, I have abstracted the transcriptional machinery into a single component and, similar to nucleosomes, instantiated multiple states to simulate the time needed to form and activate a transcriptional machine. For flexibility, the SRB can be told the number of initiation stages that must be passed before the elongation state is entered. At each stage, the transcriptional machinery is less likely to spontaneously abort and more likely to move to the next stage. Once the transcriptional machinery transitions to elongation, it can move in its predetermined direction along the DNA. Practically, as the transcriptional machinery moves to a new nucleotide, that nucleotide is in the un-transcribed state. Once the nucleotide transitions to the transcribed state, the transcriptional machinery can be moved to the next position.

**Table 3.4:** States of Transcriptional Machinery bound DNA.

|  |
|--|
| <pre> rnap_w_init_%stage%_%position% rnap_w_scribing_%position% rnap_w_scribed_%position% rnap_w_abort_%position%  rnap_c_init_%stage%_%position% rnap_c_scribing_%position% rnap_c_scribed_%position% rnap_c_abort_%position%</pre> |
|--|

#### 3.4.4.6 Actions within the Model

My modeling framework's main focus is on the actions that occur to change the configuration of nucleotide states along the DNA. See Appendix 1 for a graphical representation of all the actions currently encoded within the SRB. Here I provide additional information on these actions.

### 3.4.4.7 Transcription Factor Actions

Transcription factors interact with the transcriptional machinery. Some transcription factors can recruit the transcriptional machinery to adjacent positions on the DNA. Likewise, an elongating transcriptional machinery can evict bound transcription factors it encounters in its path. These interactions are represented as distinct actions. Note that I differentiate the action names using an uppercase C or W for the specific transcriptional machinery direction.

**Table 3.5:** Actions of Transcription Factors. (see Appendix 1 for the figures)

| Action Name     | Description  | Figure # |
|-----------------|--|----------|
| TF_BIND         | Binding of TF to positions of DNA                        | A1       |
| TF_UNBIND       | Return DNA to UNBOUND state                              | A2       |
| TF_UNBIND_W_TM  | Increase probability of TF unbinding when TM is upstream | A3       |
| TF_UNBIND_C_TM  | Increase probability of TF unbinding when TM is upstream | A4       |
| TF_RECRUIT_C_TM | Increase probability of TM_C binding when TF is bound    | A5       |
| TF_RECRUIT_W_TM | Increase probability of TM_W binding when TF is bound    | A6       |

### 3.4.4.8 Nucleosome Actions

In addition to the actions involved in nucleosome formation, I enforce a minimum distance between stable nucleosomes by including actions capable of taking into account the status of adjacent nucleosomes.

### 3.4.4.9 Transcription Machinery Actions

The transcriptional machinery binds and protects both strands of the DNA, but only moves (transcribing) in a single direction. The strands are represented as Watson (W) and Crick (C) by the yeast convention. As many intermediate states are necessary to capture the temporal behavior of transcriptional machinery, I provide the appropriate actions to transition between these states.

## 3.4.5 Validation

To evaluate the predictive capability of each of the models, we compared the model’s predicted nucleosome occupancy with an experimentally measured data set. We use Pearson correlation of occupancy values at all the positions within a given region to score the

**Table 3.6:** Actions of Nucleosomes. (see Appendix 1 for the figures)

| Action Name               | Description  | Figure # |
|---------------------------|--|----------|
| NUC_BIND                  | Binding of nucleosome to DNA   | A7       |
| NUC_UNBIND                | Transition from STABLE to UNBINDING  | A7       |
| NUC_STABLE                | Transition from BINDING to STABLE  | A8       |
| NUC_EVICT                 | Return DNA to UNBOUND state  | A8       |
| NUC_BOUND_BOUND_LEFT_0    | BOUND nucleosome checking for BOUND nucleosome to the LEFT (increasing position) in the BOUND state at 0 space between   | A9       |
| NUC_BOUND_BOUND_RIGHT_0   | BOUND nucleosome checking for BOUND nucleosome to the RIGHT (decreasing position) in the BOUND state at 0 spaces between | A9       |
| NUC_BOUND_BOUND_LEFT_L    | BOUND nucleosome checking for BOUND nucleosome to the LEFT (increasing position) in the BOUND state at L spaces between  | A10      |
| NUC_BOUND_BOUND_RIGHT_L   | BOUND nucleosome checking for BOUND nucleosome to the RIGHT (decreasing position) in the BOUND state at L spaces between | A10      |
| NUC_BOUND_BINDING_LEFT_0  | BOUND nucleosome checking for BINDING  | A11      |
| NUC_BOUND_BINDING_RIGHT_0 | BOUND nucleosome checking for BINDING  | A11      |
| ...                       |  |          |
| NUC_BOUND_BINDING_LEFT_L  | BOUND nucleosome checking for BINDING  | A12      |
| NUC_BOUND_BINDING_RIGHT_L | BOUND nucleosome checking for BINDING  | A12      |
| NUC_BINDING_BOUND_LEFT_0  | BINDING nucleosome checking for BOUND  | A13      |
| NUC_BINDING_BOUND_RIGHT_0 | BINDING nucleosome checking for BOUND  | A13      |
| ...                       |  |          |
| NUC_BINDING_BOUND_LEFT_L  | BINDING nucleosome checking for BOUND  | A14      |
| NUC_BINDING_BOUND_RIGHT_L | BINDING nucleosome checking for BOUND  | A14      |

accuracy of each model. Although each position of the DNA is not independent of the adjacent positions (because components bind multiple positions), we feel that this simple statistic is adequate for comparison between models.

There are many different experimental data sets available for nucleosome occupancy (Field et al., 2008; Kaplan et al., 2009; Lee et al., 2007). Each data set uses different protocols to measure the nucleosome occupancy across the entire genome by matching the fragments of DNA bound in a nucleosome to the genomic sequence. I chose the Lee et al. experimental data set because of its high resolution (4 bp). To compare the experimental data set to the single nucleotide resolution model predictions, the model prediction data is averaged over each of the experimental data's positions ( $\pm 2$  base pairs).

**Table 3.7:** Actions of Transcriptional Machinery. (see Appendix 1 for the figures)

| Action Name             | Description  | Figure # |
|-------------------------|--|----------|
| TM.W.INITIATE           | enter first initiation state   | A15      |
| TM.W.NEXT_STAGE         | move from stage s to s+1   | A16      |
| TM.W.ACTIVATE           | transition from initiation stage to transcribing                     | A16      |
| TM.W.TRANSCRIBE         | transcribe the current location, enter transcribed state             | A17      |
| TM.W.MOVE               | transition from transcribed [i] to transcribing [i+1]                | A17      |
| TM.W.EVICT_NUCLEOSOME   | increases the probability of the nucleosome entering unbinding state | A22      |
| TM.W.TERMINATE          | completed the transcription of the mRNA                              | A19      |
| TM.W.ABORT_INITIATION   | abort from an initiation stage                                       | A16      |
| TM.W.ABORT_TRANSCRIBING | abort from an transcribing stage                                     | A18      |
| TM.W.ABORT_TRANSCRIBED  | abort from an transcribed stage                                      | A18      |
| TM.C.INITIATE           | enter first initiation state   | A15      |
| TM.C.NEXT_STAGE         | move from stage s to s+1   | A16      |
| TM.C.ACTIVATE           | transition from initiation stage to transcribing                     | A16      |
| TM.C.TRANSCRIBE         | transcribe the current location, enter transcribed state             | A17      |
| TM.C.MOVE               | transition from transcribed [i] to transcribing [i+1]                | A17      |
| TM.C.EVICT_NUCLEOSOME   | increases the probability of the nucleosome entering unbinding state | A22      |
| TM.C.TERMINATE          | completed the transcription of the mRNA                              | A19      |
| TM.C.ABORT_INITIATION   | abort from an initiation stage                                       | A16      |
| TM.C.ABORT_TRANSCRIBING | abort from an transcribing stage                                     | A18      |
| TM.C.ABORT_TRANSCRIBED  | abort from an transcribed stage                                      | A18      |
| TM.EVICT                | shared by both TM molecules, changes DNA to unbound state            | A15      |

**Table 3.8:** Actions of Transcriptional Machinery Interference. (see Appendix 1 for the figures)

| Action Name   | Description  | Figure # |
|---|--|----------|
| TM.COLLISION  | two convergent TM colliding, current implementation causes both TM to abort (Sneppen et al., 2005) | A20      |
| The following actions check for TM in any of the initiation stages ahead of transcribing TM |  |          |
| TM.W.SITTING_DUCK_TM.C.STAGE.0<br>...   |  | A21      |
| TM.W.SITTING_DUCK_C.STAGE.%S%   |  | A21      |
| TM.C.SITTING_DUCK_TM.W.STAGE.0<br>...   |  | A21      |
| TM.C.SITTING_DUCK_W.STAGE.%S%   |  | A21      |

## 3.5 Results

### 3.5.1 Case Studies

In this section I describe case studies that verify the simulations capture both the positional and temporal aspects of regulation at well-studied yeast loci. Each case study is focused on validating a distinct property of my framework. The first study examines my ability to capture positional information in a manner similar to the best positional models. The second case study focuses on the ability to capture temporal information, such as the effects of transcriptional interference as a regulatory mechanism. Finally, I consider

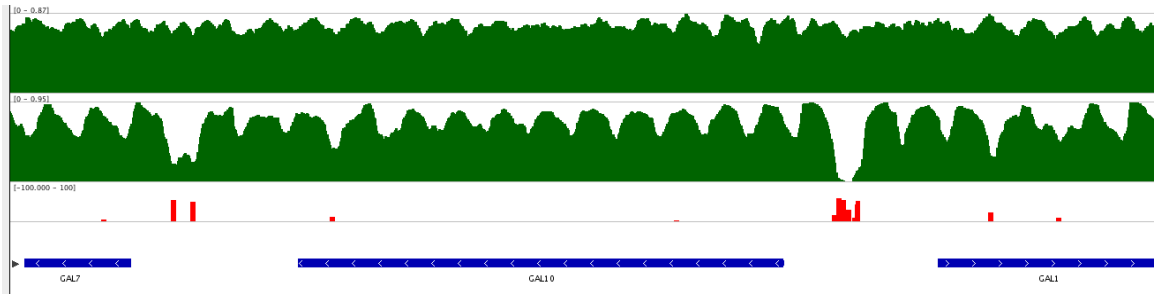
the scalability of my framework by modeling an entire chromosome. The end goal of these simulations is to provide validation of my modeling framework, not to present new biological insight. Here I describe the overall results of these case studies and interesting observations arising from my work, as well as a description of the parameters, runtimes, and memory usage for each simulation. To evaluate each of the models, I compared the model’s predicted nucleosome occupancy with an experimentally measured data set (Lee et al., 2007). While my model is at base pair resolution, the Lee experimental data is measured at a 4 base pair resolution. Therefore, I summarize the predicted values over each Lee data probe (+/-2 bp) and calculate a Pearson correlation value.

### 3.5.2 Capturing Positional Information (GAL10)

The first case study focuses the model’s ability to capture accurate positional information. Occupancy is a population averaged measure of the time that a position of the DNA is occupied. Changes in transcription factor concentration can alter the identity and occupancy of factors within the region. For example, the GAL locus is one of the most well studied regions within the yeast genome. This locus has a known activator (GAL4), which has multiple binding locations in the promoter region. By running my simulation at distinct concentrations, my model shows a parallel increase in the occupancy of GAL4, causing shifts in nucleosome positioning that opens the chromatin near both GAL10 and GAL1 transcription start sites (Figure 3.11). My model and the best positional models, as exemplified by the COMPETE model (Wasson and Hartemink, 2009), both predict the changes in nucleosome occupancy in the region (using GAL4 and nucleosomes only).

**Table 3.9:** Typical run time and memory usage for GAL10-GAL1 models (laptop, 2.8 GHz cpu, 8 GB ram; 110,000 rules using 19500 reactants)

|                 | SRB    | DIZZY   |
|-----------------|--------|---------|
| Execution Times | 10 sec | 200 sec |
| Memory Usage    | 60 MB  | 3.7 GB  |



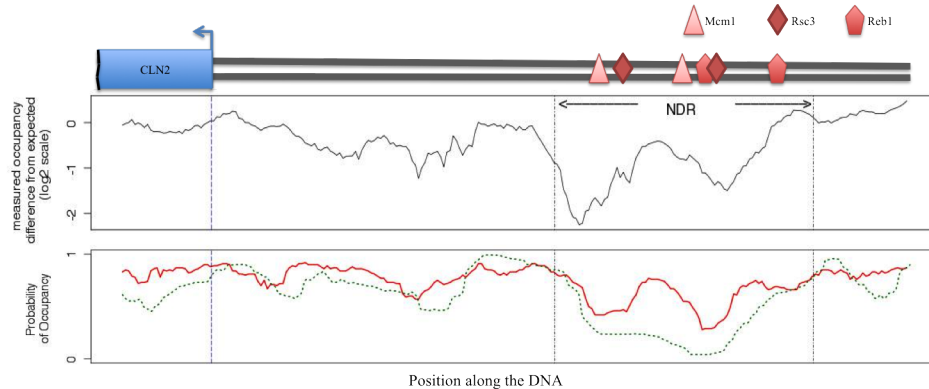
**Figure 3.11: Nucleosome occupancy of GAL10-GAL1 region.** My model was run without (top track) and with (second track) the Gal4 transcription factor. The results largely recapitulate the findings in Wasson and Hartemink (2009) Figure 1, showing how random nucleosome positioning is transitioned to a phasing of nucleosomes when barriers (Gal4 in red third track) are added.

**Table 3.10:** Non-default parameters used for GAL10-GAL1 models

| Attribute Name  | Value used                     | Notes  |
|---|--------------------------------|--|
| DNA<br>chr<br>START<br>END<br>GROUPING                            | chrII<br>270000<br>283000<br>4 |  |
| Nucleosome<br>ON_RATE<br>OFF_RATE<br>INITIAL_COUNT<br>LINKER_SIZE | 0.0001<br>1.0<br>50000<br>30   |  |
| Transcription Factor<br>TF_LIST<br>MOTIF_FILE<br>THRESHOLD        | GAL4<br>DATA/yeast.tamo<br>.5  |  |
| GAL4<br>INITIAL_COUNT   | 500                            | (Count of 0 used when modeling nucleosomes only) |

### 3.5.3 Capturing Positional Information (CLN2)

A more interesting case is the CLN2 locus, where experiments show that multiple factors work cooperatively, without explicit protein-protein interaction, to maintain a known nucleosome depleted region (NDR) (Bai et al., 2011). Bai and colleagues determined that the NDR of CLN2 was dependent on three transcription factors: Mcm1, Reb1, and Rsc3. I sought to confirm that my modeling framework could capture this nucleosome depleted region in a manner similar to the state-of-the-art positional models, again using COMPETE (Figure 3.12). In this case, I quantitatively compared each model to Lee’s experimentally determined nucleosome occupancy profile and obtained at 4-nucleotide resolution (Lee



**Figure 3.12: Modeling nucleosome occupancy at gene CLN2 recapitulates the known nucleosome depleted region (NDR).** The yeast gene CLN2 (blue box) contains a well-studied NDR upstream of the start site. The known location of three key transcription factors (Mcm1, Rsc3, and Reb1) are shown as red geometric shapes (Bai et al., 2011). The experimentally determined nucleosome occupancy (Lee et al., 2007) is shown in black, as observed from ChIP signal (normalized, log<sub>2</sub> scale). The bottom panel shows the results from my model (red line) and the COMPETE model (green dotted line), plotted as the probability of a nucleosome (y-axis) as a function of DNA position (x-axis).

et al., 2007). Giving each model (COMPETE and my framework) the three key transcription factors (Mcm1, Reb1, and Rsc3), both models correlated well with the experimental data (COMPETE  $r=0.59$ ; my SSA model  $r=0.57$ ), showing that, similar to GAL4, my framework captures the same steady state behaviors as a state of the art positional model.

Upon inspection of my initial results in this region (3 transcription factors + nucleosomes), I noted a sharp drop in nucleosome occupancy proximal to the CLN2 transcription start site that corresponds to a well-defined TATA box sequence. Upon adding the TATA-binding protein to the simulation, I obtained a stronger nucleosome depleted region around the TATA box that more closely mimicked the experimental data ( $r=0.62$  for entire region for both models).

I next sought to understand the underlying dynamics implied by my simulations between the transcription factors, the nucleosomes, and the DNA sequence at this locus. When my SSA model uses only nucleosomes, the correlation to the experimental data is little better than random ( $r=0.10$ )(data not shown). This result is consistent with previous experimental work which showed the NDR had lower nucleosome occupancy than was predicted by positional models that use only nucleosome affinity (Kaplan et al., 2009). Next, I



added in only the transcription factor Mcm1 and observe the emergence of a well-positioned nucleosome at the far edge of the NDR. This nucleosome influences the positioning of adjacent nucleosomes, consistent with studies that indicate that a single transcription factor can impact many nucleosomes (Jansen et al., 2012). As I add additional transcription factors to the model, they improve the correlation with experimental data, showing that my framework can capture the implicit cooperation of multiple binding factors to maintain the NDR.

Finally, I sought to determine how sensitive my model results were to the particular configuration of parameters chosen for the less well determined parameters within my model, namely binding rates and molecule counts. I found that small changes in molecule counts for Mcm1 have an initial dramatic effect on configurations, but that saturation is quickly reached (data not shown). My stable NDR results were obtained using three transcription factors (Mcm1, Rsc3, Reb1) at the default molecule counts (Ghaemmaghami et al., 2003) by adjusting the OFF rate, which controls the residency time. If the OFF rate is reduced to 10%, the molecule count must be increased by a factor of 10 to maintain the NDR. This implies that the balancing act between molecule counts and residency time is consistent with intuition about how the model should perform. Furthermore, it has been speculated that modulation of residency times may allow for more precise regulation (Lickwar et al., 2012b).

It should be noted that the single trajectory nature of my simulations leads to a number of interesting observations at the CLN2 locus. First, all three transcription factors are not necessarily bound at the same time to maintain the NDR. I observe cases where zero, one, two or three transcription factors are bound. Likewise, the NDR shows some limited binding of nucleosomes, consistent with the fact that a depleted region does not imply the absence of binding, but rather less binding than expected. Lastly, I observe that a transcription factor with longer residency time has a stronger effect on nearby nucleosome positioning (data not shown). This effect propagates out from the transcription factor position over time, leading to speculation that longer residency times may be necessary for long range

effects. My models allow for the exploration of the temporal patterns of interactions that lead to the NDR result, observations that could be experimentally validated.

**Table 3.11:** Typical run time and memory usage for CLN2 models (laptop, 2.8 GHz cpu, 8 GB ram; 43,000 rules using 12,000 reactants)

|                 | SRB   | DIZZY  |
|-----------------|-------|--------|
| Execution Times | 6 sec | 78 sec |
| Memory Usage    | 30 MB | 2.3 GB |

**Table 3.12:** Non-default parameters used for CLN2 models

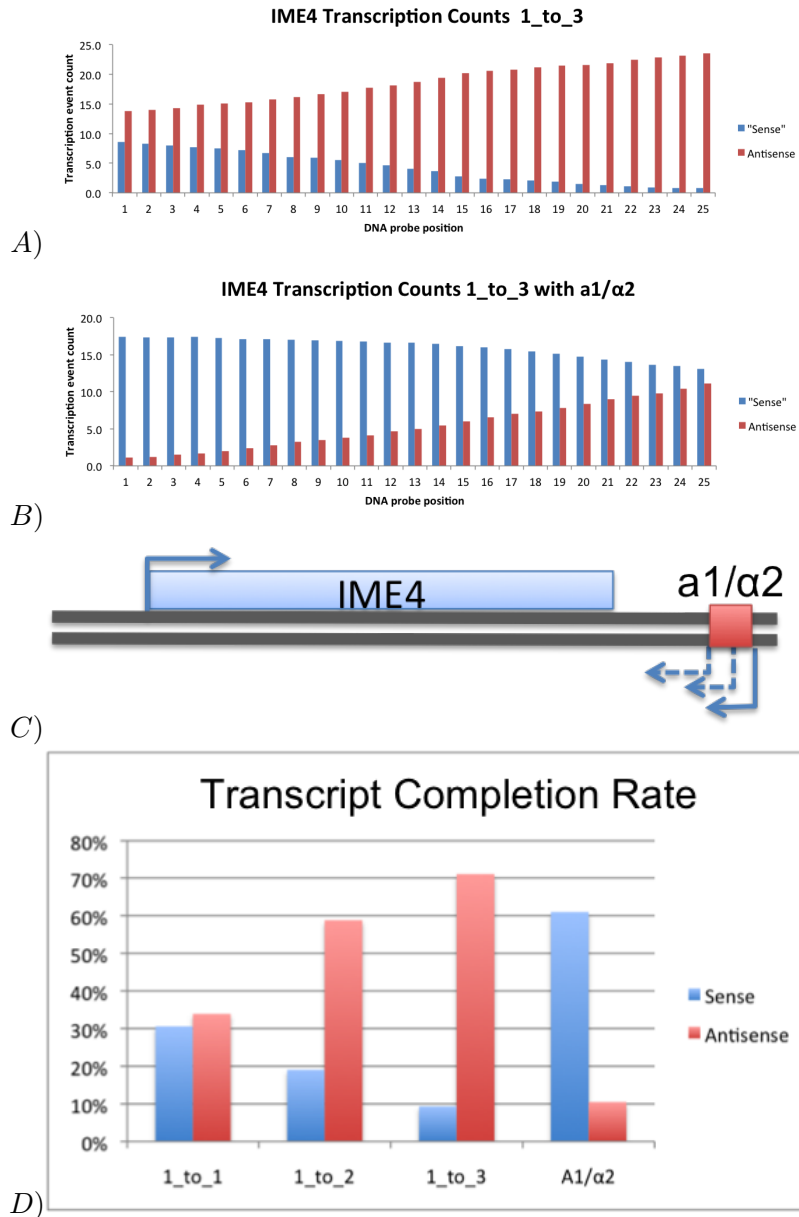
| Attribute Name  | Value used   | Notes |
|---|--|-------|
| DNA<br>chr<br>START<br>END<br>GROUPING  | chrXVI<br>64001<br>72000<br>4  |       |
| Nucleosome<br>ON_RATE<br>OFF_RATE<br>INITIAL_COUNT<br>LINKER_SIZE   | 0.0001<br>1.0<br>100000<br>10  |       |
| Transcription Factor<br>TF_LIST<br>MOTIF_FILE   | REB1, MCM1, RCS3, TBP<br>DATA/yeast.tamo   |       |
| MCM1<br>OFF_RATE<br>OCCLUSION_5<br>OCCLUSION_3<br><br>REB1<br>OFF_RATE<br>OCCLUSION_5<br>OCCLUSION_3<br><br>RSC3<br>ON_RATE<br>OFF_RATE<br>INITIAL_COUNT<br>MOTIF<br><br>TBP<br>OFF_RATE<br>INITIAL_COUNT<br>THRESHOLD<br>MOTIF | .01<br>12<br>12<br><br>.01<br>12<br>12<br><br>.1<br>.8<br>3<br>CGCGC<br><br>.035<br>4<br>.85<br>TTATATAT |       |

### 3.5.4 Capturing Temporal Interaction (IME4)

The second case study focuses on the ability of my framework to capture temporally driven events, such as transcriptional interference. I use a well-studied location, the IME4 locus, where transcription of the gene is constitutive and largely unregulated. Yet, Ime4 is

an N6-adenosine methyltransferase required only for a cell's entry into meiosis. To modulate *Ime4* levels, the cell produces an antisense transcript constitutively that, by transcriptional interference, stops the gene's sense transcript from completing (Hongay et al., 2006). Only in diploids where the antisense transcript is itself repressed can the full length transcript of *IME4* be produced. It is precisely this sort of regulation that motivated the development of my modeling system. I sought to confirm that my model could recapitulate the known transcriptional interference pattern observed in experimental data. In yeast there exists a transcription factor,  $a/\alpha$ , which is unique to diploids. I ran the *IME4* locus both with and without the  $a/\alpha$  transcription factor. In the absence of the transcription factor, I observe robust transcription of the antisense transcript and very little of the sense transcript reaches full length. In the presence of  $a/\alpha$ , the antisense transcript is repressed by occluding the antisense initiation site, allowing most of the sense transcript to reach full length. These simulation results are consistent with the experimental data at *IME4* (Gelfand et al., 2011; Hongay et al., 2006). I then explored a series of interesting what if scenarios at this locus. For instance, I studied the impact of different transcriptional machinery initiation and elongation rates on the transcriptional interference event. From these parameter explorations, it is clear that the initiation rate of the transcriptional machinery must be fast enough to always keep at least one polymerase transcribing in the antisense direction for each initiation of sense strand machinery (Figure 3.13). This pattern is consistent with previous mathematical studies of transcriptional interference (Sneppen et al., 2005).

One interesting observation is worth noting: transcription through any transcription start site causes a transient localized open chromatin conformation that makes initiating transcriptional machinery more likely. Thus, the few sense transcripts that do complete, i.e. traverse through the start location of the antisense transcript, typically trigger the initiation of an antisense transcript. This may illustrate a particularly interesting feedback mechanism where the rate of transcription through a region regulates the open chromatin and therefore transcription at nearby sites.



**Figure 3.13: IME4 transcription regulation.** A) Number of transcription events along the DNA for the sense (blue) and anti-sense (red) transcripts. B) Model results when the same simulation was run with the addition of the  $a/\alpha$  transcription factor, a simulation that captures the diploid state. C) IME4 gene and binding location of the repressing factor. D) Rates of full-length transcripts as percentage of initiated transcripts. Different conditions show the relative rates of initiation (sense\_to-antisense), and when  $a1/\alpha2$  repression factor is included (uses same initiation as 1\_to\_3).

**Table 3.13:** Typical run time and memory usage for IME4 models (laptop, 2.8 GHz cpu, 8 GB ram; 38,000 rules using 22,000 reactants)

|                 | SRB   | DIZZY   |
|-----------------|-------|---------|
| Execution Times | 3 sec | 135 sec |
| Memory Usage    | 30 MB | 3.5 GB  |

**Table 3.14:** Non-default parameters used for IME4 models

| Attribute Name   | Value used  | Notes   |
|--|---|---|
| DNA<br>chr<br>START<br>END<br>GROUPING   | chrVII<br>140000<br>146000<br>8   |   |
| Nucleosome<br>ON_RATE<br>OFF_RATE<br>INITIAL_COUNT<br>LINKER_SIZE                                | 0.0001<br>1.0<br>50000<br>18  |   |
| Transcription Factor<br>TF_LIST<br>MOTIF_FILE  | A1_ALPHA2<br>DATA/yeast.tamo  |   |
| A1_ALPHA2<br>ON_RATE<br>OFF_RATE<br>INITIAL_COUNT<br>MOTIF                                       | .5<br>.01<br>10<br>TGTNNNNNNNACATCA   | Count is 0 for nominal processing. 10 is used when inhibiting transcription                 |
| RNAP<br>MODEL_RNAP<br>N_STAGES<br>INITIAL_COUNT<br>ON_RATE<br>OFF_RATE<br>INIT_RATE<br>POSITIONS | 1<br>5<br>4<br>.05<br>.0008<br>.1<br>[142175, W] [144220, C]<br>[144190, C] [144250, C] | Explicit positions of initiation. 1 sense initiation and 1-3 sense to anti-sense initiation |

### 3.5.5 Tractability

My last case examines the tractability of using my framework to model large systems. One of my modeling goals was to include temporal information while maintaining computability for large sequences. This case study specifically focuses on testing options for scaling the simulations. I chose to model an entire chromosome from the yeast genome. *S. cerevisiae* chromosome I is approximately 230,000 nucleotides containing 92 genes. In the worst case, modeling of a large sequence with many transcription factors present can generate millions of rules and overwhelm the simulation engines. When using transcription

factor threshold cutoffs, a single locus, such as GAL, CLN2, or IME4, generates thousands of rules and reactants (Table 3.15). For chromosome I, a similar approach at single nucleotide resolution would produce approximately 7 million reactants and 40 million rules. To make this simulation tractable, I employed nucleotide grouping, reducing the granularity of the resultant simulation. I found that grouping the DNA into 30 nucleotide units reduced the model to 170,000 reactants and 400,000 rules. At this size, Dizzy simulations were tractable, requiring 98 GB of ram and running for just over 20 CPU hours per simulation. My resource limiting factor in computability is the stochastic simulation engine, currently Dizzy (Ramsey et al., 2005). It computes large tables to efficiently transition between the possible interactions. As these tables grow towards a given machine’s available RAM, performance decreases quickly. Managing the number of rules that are created keeps the models computable. It is possible that other off-the-shelf simulators would be capable of larger simulations using fewer compute resources. Alternatively, it is possible to replace the off-the-shelf stochastic simulation engine with one designed specifically for this application.

### 3.5.6 Complexity

The SRB generates models that are created for simulation using an off-the-shelf simulation engine. This engine is expecting a set of chemical reaction equations describing the interactions between all the different molecules. Each reaction describes the pre-condition molecules, the rate, and the post-condition molecules (Equation 1). My modeling framework can generate a model for any DNA sequence. The size of the model is dependent on the number of different DNA binding factors included, the number of actions being applied, and the length of the DNA sequence being modeled. At every position, a number of rules are generated for each action. Each action can influence a set of additional positions, as defined by the span of the rule. Even in the worst case, when all rules are applied at all positions, the number of states and rules generated grows linearly with the length of the DNA sequence being modeled (Figure 3.9). My modeling framework uses thresholds for factor affinity to manage the number of sequence positions that generate transcription factor binding-unbinding rules, which can drastically reduced the size of these models. The

runtime of the SRB scales linearly with the size of the model being generated (Figure 3.9). Even with the reduction of rules from the TF components, the sizes of these models are beyond the capabilities of the current simulation engines. To combat this problem, my SRB application can group the interactions for a group of DNA nucleotides together into a single position within the model. Using this rule reduction method, I have been able to run a model of a complete chromosome of yeast (chromosome 1, 230,000 nucleotides).

**Table 3.15:** Model size is dependent on components included in the model

| Genomic Region             | Size (nucleotides) | Theoretical    | Actual      |
|----------------------------|--------------------|----------------|-------------|
| Average Gene               | 4,000              | 31,140,000     | 676,985     |
| Smallest Chromosome (chrI) | 230,218            | 1,792,247,130  | 40,044,142  |
| Largest Chromosome (chrIV) | 1,531,933          | 11,926,098,405 | 266,420,938 |

The actual number of rules is only  $\approx 2\%$  of the theoretical limit. Theoretical numbers assume each nucleotide of the DNA is modeled (no grouping) and all rules are applied at every position. The reported values for Actual assumes a minimum score from the PSSM for be obtained at a sequence position before rule generation for that transcription factor. A total of 124 transcription factors defined in TAMO are included in these calculations.

**Table 3.16:** Component attributes that influence the span of some actions

| Component                         | Variable | Size                            |
|-----------------------------------|----------|---------------------------------|
| TF (each has different footprint) | WTF      | [4..25], nominally 12           |
| # of TFs                          | #TFs     | 120<br>depending on source data |
| Nucleosome                        | WN       | 147                             |
| Nucleosome Linker                 | WL       | 1                               |
| TM                                | WTM      | 25                              |
| TM initiation stages              | #stages  | 5                               |

I next consider the maximum number of rules that can be generated for any sequence. For each abstraction (Appendix I) I consider its span of influence (range of other positions that are referenced). The larger the span of the influence, the more positions that are inter-related. The application of a specific rule at position  $i$  can directly impact some number of adjacent nucleotides, giving rise to a formula for calculating the number of rules that

have any specific position as a reactant or resultant. Using the parameter settings from Table 3.16, this produces a theoretical maximum for the number of rules influenced by position  $i$ . A worst-case total of 18,855 rules involve each position of the DNA.

**Table 3.17:** Complexity analysis for transcription factor abstractions.

| Figure | Action Name       | Span of Influence  | Worst-case # of rules impacted by a state change at position $i$ | Max # of rules <sup>1</sup> | # of rules generated per position |
|--------|-------------------|--------------------|--|-----------------------------|-----------------------------------|
| A1     | TF BIND           | $[i, i+WTF-1]$     | $WTF * \#TFs$  | 1440                        | 1                                 |
| A2     | TF UNBIND         | $[i, i+WTF-1]$     | $WTF * \#TFs$  | 1440                        | 1                                 |
| A3     | TF UNBIND by TM-W | $[i-WTM, i+WTF-1]$ | $(WTF+WTM)*\#TFs$  | 4440                        | 1                                 |
| A4     | TF UNBIND by TM-C | $[i, i+WTF+WTM-1]$ | $(WTF+WTM)*\#TFs$  | 4440                        | 1                                 |
| A5     | TF RECRUIT TM-C   | $[i-WTM, i+WTF-1]$ | $(WTF+WTM)*\#TFs$  | 4440                        | 1                                 |
| A6     | TF RECRUIT TM-W   | $[i-WTM, i+WTF-1]$ | $(WTF+WTM)*\#TFs$  | 4440                        | 1                                 |

<sup>1</sup>Max # of rules is calculated using values from Table 3.16.

**Table 3.18:** Complexity analysis for nucleosome abstractions.

| Figure   | Action Name   | Span of Influence     | Worst-case # of rules impacted by a state change at position $i$ | Max # of rules <sup>1</sup> | # of rules generated per position |
|----------|---------------|-----------------------|--|-----------------------------|-----------------------------------|
| A7       | NUC BIND      | $[i, i+WN-1]$         | WN   | 147                         | 1                                 |
| A7       | NUC UNBIND    | $[i, i+WN-1]$         | WN   | 147                         | 1                                 |
| A8       | NUC STABILIZE | $[i, i+WN-1]$         | WN   | 147                         | 1                                 |
| A8       | NUC EVICT     | $[i, i+WN-1]$         | WN   | 147                         | 1                                 |
| A9 - A14 | NUC LINKER    | $[i, i+WN+WN-1+WL+L]$ | $WN+WN+(WL*3)$   | 348                         | 54                                |

<sup>1</sup>Max # of rules is calculated using values from Table 3.16.

### 3.6 Discussion

My goal was to integrate inherently dynamic aspects of transcriptional regulation, such as transcriptional interference, with the intuitive position based models. To this end, I constructed a modeling framework that leverages the power of Petri nets to describe the actions of various regulators and the extent or span of their influence. By treating the DNA as an ordered set of entities (nucleotides or groups of nucleotides) rather than a single molecular entity, I can generate models that grow linearly with the length of the DNA sequence being modeled. At the core of my framework is my stochastic rule builder, an application that can take in an arbitrary sequence and construct the complete set of coherent biochemical rules.

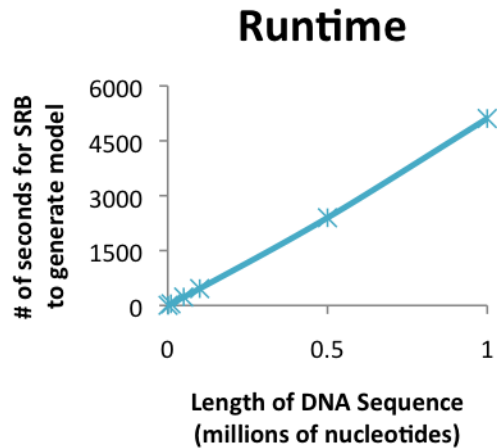


**Table 3.19:** Complexity analysis for transcriptional machinery abstractions.

| Figure | Action Name           | Span of Influence  | Worst-case # of rules impacted by a state change at position $i$ | Max # of rules <sup>1</sup> | # of rules generated per position |
|--------|-----------------------|--------------------|--|-----------------------------|-----------------------------------|
| A15    | TM_W INITIATE         | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A16    | TM_W NEXT_STAGE       | $[i, i+WTM-1]$     | WTM * #stages  | 125                         | 10                                |
| A16    | TM_W ABORT_INITIATION | $[i, i+WTM-1]$     | WTM * #stages  | 125                         | 10                                |
| A16    | TM_W ACTIVATE         | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A17    | TM_W TRANSCRIBE       | $[i, i+WTM]$       | WTM  | 25                          | 2                                 |
| A17    | TM_W MOVE             | $[i, i+WTM-1]$     | WTM  | 25                          | 1                                 |
| A18    | TM_W ELONGATION_ABORT | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A19    | TM_W TERMINATE        | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A22    | TM_W EVICT_NUCLEOSOME | $[i, i+WTM+WN-1]$  | (WTM + WN)   | 172                         | 2                                 |
| A15    | TM_C INITIATE         | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A16    | TM_C NEXT_STAGE       | $[i, i+WTM-1]$     | WTM * #stages  | 125                         | 10                                |
| A16    | TM_C ABORT_INITIATION | $[i, i+WTM-1]$     | WTM * #stages  | 125                         | 10                                |
| A16    | TM_C ACTIVATE         | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A17    | TM_C TRANSCRIBE       | $[i-1, i+WTM-1]$   | WTM  | 25                          | 2                                 |
| A17    | TM_C MOVE             | $[i-1, i+WTM-1]$   | WTM  | 25                          | 1                                 |
| A18    | TM_C ELONGATION_ABORT | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A19    | TM_C TERMINATE        | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |
| A22    | TM_C EVICT_NUCLEOSOME | $[i, i+WTM+WN-1]$  | (WTM+WN)   | 172                         | 2                                 |
| A20    | TM COLLISION          | $[i, i+WTM+WTM-1]$ | WTM + WTM  | 50                          | 1                                 |
| A21    | TM* SITTING_DUCK      | $[i, i+WTM+WTM-1]$ | (WTM + WTM) * 2 * #stages  | 500                         | 10                                |
| A15    | TM EVICT              | $[i, i+WTM-1]$     | WTM  | 25                          | 2                                 |

<sup>1</sup>Max # of rules is calculated using values from Table 3.16.

Off-the-shelf stochastic simulation engines, such as Dizzy, can then simulate these rule sets. I have developed a framework to create biologically realistic models of the mechanisms of transcriptional regulation. Based on this framework, I can model not only the steady-state behavior of transcription factor binding and nucleosome formation (case study 1), but also the dynamics of components, such as the transcriptional machinery (case study 2). My framework scales linearly, making it possible to simulate very large segments of DNA (case study 3). The simulations produce tremendous amounts of positional and temporal data,



**Figure 3.14: Runtime of the SRB application is linear with respect to length of the DNA sequence.** Using chromosome IV, models of varying lengths of DNA (1000, 10,000, 50,000, 100,000, 500,000, and 1,000,000 nucleotides) were generated using 124 TFs. These models were generated on a server: Dell R510 (12 MB Cache, 2.66 GHz), 128 GB ram, and 48 TB storage.

which can be converted into simple visualizations depicting the state of the DNA at each time step (see Figure 3.7). There is an intimate relationship between the development of new experimental techniques and new modeling frameworks. Typically, the level of detail of the modeling abstraction is influenced by both the questions being asked and the resolution of available experimental data. Large-scale experimental techniques are continuously evolving to capture increasingly detailed views of regulation. Recent experimental work is only beginning to highlight the importance of temporal dynamic events, such as transcription factor turnover (Lickwar et al., 2012b), nucleosome turnover (Dion et al., 2007), and transcriptional interference (Palmer et al., 2011), for understanding transcriptional regulation.

As is true with any new modeling system, my framework depends on a number of parameters to capture the distinct behaviors of individual components. I not only require the DNA binding parameters used in positional models, but also rate parameters that capture the underlying temporal aspects of events. Unfortunately, many of the temporal parameters are not currently known. I presently set these parameters using coarse searches for values that reasonably capture desired phenomena or fit available experimental data. I am well

aware that many of these parameters may be overfit, thus detailed parameter explorations and sensitivity analysis remains as future work. However, as new experimental studies uncover these rates or identify new key regulatory mechanisms, my modeling framework is poised to incorporate this information. Ultimately, single cell measurements (Taniguchi et al., 2010) will permit a more precise comparison between my model and biological reality. As my understanding of the molecular details improves, my framework can be easily extended.

Currently, my system simplifies every component in an attempt to capture the essence of its behavior. However, it may be necessary to extend existing components to capture key molecular events. For example, I currently consider the nucleosome as a single large binding component. Yet in reality, the nucleosome is composed of a histone core (H3-H4), which binds first, and two subunits (H2A-H2B) on each edge (Andrews and Luger, 2011). Recent work on nucleosome dynamics indicates the core histone is relatively stable, whereas the edge histones are more dynamic (Böhm et al., 2011; Li et al., 2005). In the future, I may need to model the core and edge components separately to account for their differences in behavior. Likewise, I may need to add additional components, such as histone tail modifications, that can affect the affinity or stability of a nucleosome. While histone modifications are known to be well correlated with transcriptional state (Berger, 2002), the details of how these marks influence binding, when these marks are temporally deposited, and how they function is not well characterized. Therefore, the addition of new components or augmented functionality must be balanced against the parameterization problem. Only as I understand the dynamic behavior of these components in more detail is it realistic to include these within my framework.

My framework results in a large set of rules to describe the chemical reactions within the system. An alternative to simulating the system of equations is to mathematically solve them. Solving the system of differential equations would provide the equilibrium behavior of the system, but requires large-scale system solvers. Even the best of these solvers are limited in their ability to handle tens of thousands of equations (Pahle, 2009). Simulation

can handle much larger sets of equations, but at the cost of increasing computational time. Finally, my SRB produces single trajectory simulations for a single cell, but they are not whole cell simulations. The current models reach a relative equilibrium (cycling through common configurations), as there is little feedback to alter protein concentrations or modify component behaviors within the system. As the parameters of the model become more informed by experimental data, I envision introducing realistic feedback into the model.

### 3.7 Conclusion

I have created a modeling framework that captures both the positional and temporal aspects of transcriptional regulation. My framework uses Petri nets to describe permissible actions and their localized span of influence. I have created an application, the stochastic rule builder, which quickly generates large systems of chemical reactions to model any specific instance of DNA. The resulting equations can be simulated with standard stochastic simulation engines. I confirmed through case studies that my model can capture positional information, temporal information, and is scalable to large segments of DNA.

I consider my framework, at this time, as primarily an exploration tool. The predictive power is limited to the known kinetics of factors and this knowledge is currently limited. As technological advances in single cell experimentation further uncover temporal cellular kinetics, my flexible modeling framework can easily be extended to incorporate new components or additional detail of component behavior. The models will continue to advance towards biologically realistic and predictive models of transcriptional regulation.

## CHAPTER IV

### DYNAMIC NUCLEOSOME MODEL <sup>2</sup>

#### 4.1 Introduction

Transcriptional regulation emerges from the complex system of interactions amongst factors binding to the DNA. Current steady state models capture the competition and cooperation between factors for access to the DNA, predict the occupancy of factors bound to the DNA, and infer transcription rates. Nucleosome positioning plays a key role in transcription and the accuracy of these models. Most models consider nucleosomes as monolithic 147 nucleotide binding factors, but recent work has shown that nucleosome formation is dynamic (Andrews and Luger, 2011). In this work I extend a positional steady state model described by Wasson (Wasson and Hartemink, 2009). The COMPETE model uses a Hidden Markov Model (HMM) to capture the transitions between biologically inspired states and represent the competition between components for DNA. My extension adds multiple states for the dynamic formation of nucleosomes. My extended model achieves better correlation to experimental nucleosome occupancy data over the whole genome. My results show how and where the correlation between model predictions and experimental data have increased within the genome.

##### 4.1.1 Components of Transcriptional Regulation

Transcriptional regulation emerges from the complex interactions between many components vying for the opportunity of binding with the DNA. The patterns within the sequence of DNA control the position and rates at which these components will bind and ultimately initiate transcription. Previous systems biology work has shown how changes in the DNA sequence affect the rate of transcription (Lubliner et al., 2013).

Transcription factors are the best known components affecting transcription. It is only when the correct configuration of factors are bound to the DNA that transcription occurs. Each transcription factor has a unique affinity for specific patterns of DNA sequence

---

<sup>2</sup>A Manuscript covering part of this work is currently in preparation: "Dynamic nucleosomes in a steady state model." This work was supported by a Chateaubriand Fellowship awarded to me in 2012 to work collaboratively with Laboratoire Joliot Curie, Ecole Normale Supérieure de Lyon, Lyon, France.

(Stormo, 2013; Gordon et al., 2005). Transcription factor affinity is represented using a position specific scoring matrix (PSSM), which quantifies the probability of finding the factor bound at a specific DNA pattern. The probability of a transcription factor binding to any position of a DNA sequence is dependent on its affinity for the sequence at that position and the cellular concentration of the factor.

However, transcription factors only bind to a small portion of the DNA. Most of the DNA in a cell is bound within nucleosomes, which are formed by the wrapping of ~147 nucleotides around a set of histone proteins (Luger et al., 2012). Although any sequence of DNA can be found within nucleosomes, histones also have an affinity for patterns of nucleotides (Lowary and Widom, 1998). The probability of histones binding to any position of a DNA sequence is dependent on their affinity for the sequence at that position and their cellular concentration.

All of the DNA binding factors are actively binding and unbinding with the DNA in vivo. Still, each nucleotide of the DNA can only be bound by one component at a time. Therefore, once bound within a nucleosome, most transcription factors are prevented from accessing the nucleosome bound DNA. Combining all the factors into a single system creates a model of competition between the different components for binding with DNA. The configuration of factors bound to the DNA is the balance between the interactions of all the individual components. Some configurations of factors simultaneously bound to the DNA will have higher probability of occurring within a population of cells. Experimental measurements across a population of cells will sample these different configurations.

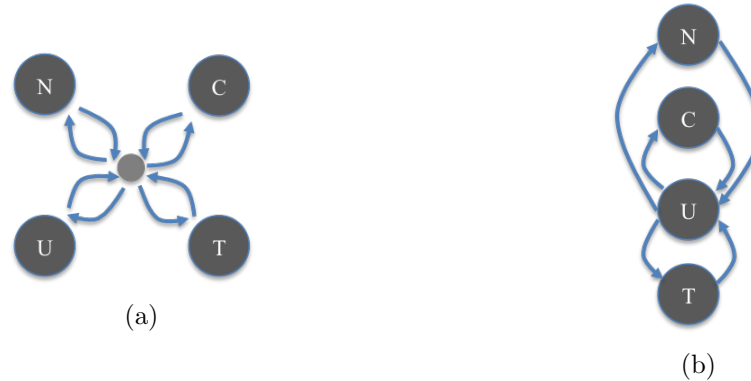
Experimental techniques currently measure the position of factors within a population of cells and averaged across time. Nucleosomes bind promiscuously and can be found at 80-90% of the genome at any time (Lee et al., 2007). The nucleosome positions are inferred from the accumulation of data across a genomic region. Well-positioned nucleosomes have accumulated more data at specific nucleotide positions, as opposed to randomly positioned regions, where nucleosome data is spread with equal frequency across all positions (Struhl and Segal, 2013).

Nucleosome occupancy is related to, but distinct from, nucleosome positioning (Struhl and Segal, 2013). In experimental data, occupancy is a measurement of the fraction of cells within the population that were found to have a nucleosome occupying a specific position. Nucleosome occupancy can be experimentally measured across entire genomes (Lee et al., 2007) and occupancy profiles are used to show the perturbations to complex systems (Badis et al., 2008). Because we are examining populations of cells, the positions of nucleosomes are in a quasi steady state, where positions of individual nucleosomes in individual cells are undergoing continual change, but the proportion of cells with nucleosomes at specific positions remains nearly constant (constant within small variation).

#### **4.1.2 Steady State Modeling of Transcriptional Regulation**

The quasi steady state behavior of a cell population has been captured using steady state modeling methods. Previous work has shown that HMMs can capture these complex steady state systems (Segal et al., 2006; Wasson and Hartemink, 2009). Each nucleotide of a DNA sequence can only be bound to at most one binding factor at a time. These discrete states of the DNA can be represented as states in an HMM (Figure 4.1). The probability of transitioning between the states is a function of the binding affinity of the factors for specific sequences of DNA and the concentrations of those factors. Application of the HMM to specific sequences of DNA and a DNA binding factor set can determine the probability of all the possible configurations of factors bound along the DNA. From the probabilities of each possible configuration, the probabilities of being bound by any single factor can be determined at each nucleotide. HMM analysis is efficient and scalable, allowing the modeling of whole genomes using the forward-backward algorithm.

In yeast there are over 150 known transcription factors (Teixeira et al., 2006). Many of these factors have been well studied and their individual affinities for sequence have been documented and have published PSSMs (Harbison et al., 2004; Gordon et al., 2005). Although the concentrations of factors within individual cells vary depending on the cellular conditions, the population averaged concentrations have been experimentally measured



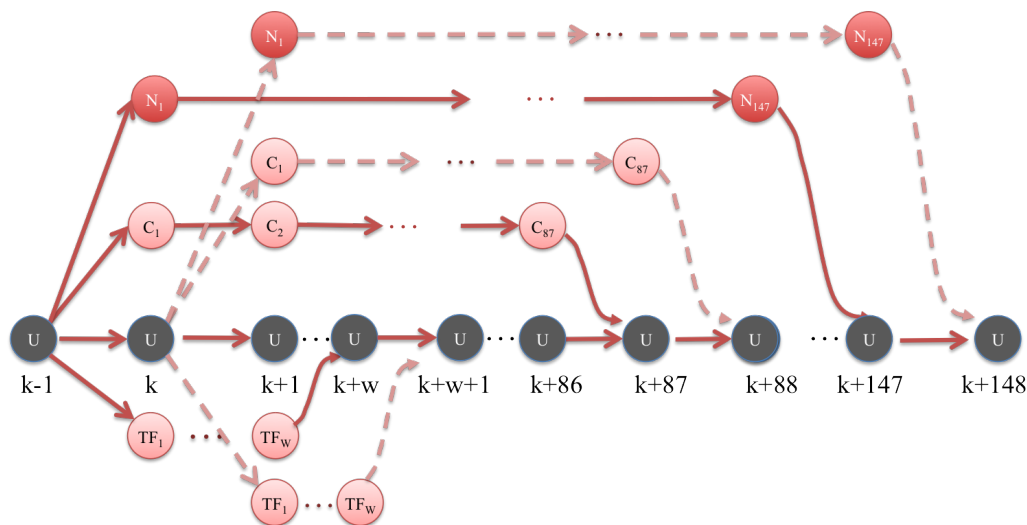
**Figure 4.1: Hidden Markov model states and transitions for individual nucleotide position of DNA.** Each circle is a possible state of an individual DNA nucleotide. It can be unbound (U), bound to a transcription factor (T), bound within a nucleosome (C and N). Arrows represent the transitions from one of these states to another. This model only allows transition from the unbound state to one of the bound states. a) The states and transitions in the Wasson paper use a pseudo state for all transitions. b) A simplified way of imagining the transitions that will be used in other figures.

(Ghaemmaghami et al., 2003). The HMM state transitions to transcription factor bound states are inferred from these affinities and relative concentrations.

Nucleosomes have a sequence affinity that is based on the thermodynamics of bending the DNA into the nucleosome structure (Morozov et al., 2009). This affinity can also be captured in a similar type of scoring matrix. Particular pairs of nucleotides are found at greater frequency at specific positions within a nucleosome (Segal et al., 2006; Wasson and Hartemink, 2009). Some of the bendability characteristics inherent in the DNA can be captured by the probability of di-nucleotides (consecutive pairs of nucleotides). Various patterns of consecutive nucleotides are easier to compress together or be stretched apart when being coiled around the histone core. A di-nucleotide scoring matrix captures these features and can be used to calculate transitions to the nucleosome bound states.

Each DNA position can be in only one state at a time, but all positions are not independent. Transcription factors typically bind from 4 to 20 nucleotides and nucleosomes encompass up to 147 nucleotides. This can be captured in a HMM by explicitly listing all the states for a DNA position and the required states of adjacent positions, where the number of adjacent states is determined by the length of the scoring matrix. A single HMM





**Figure 4.2: Complete HMM states describing binding to transcription factors and nucleosomes.** Each circle is a possible state of an individual DNA nucleotide. It can be unbound (U), bound to a transcription factor (T), bound within a nucleosome (C and N). Arrows represent the transitions from one of these states to another. This model only allows transition from the unbound state to one of the bound states. However, DNA binding factors do not bind to only a single position of the DNA. Transitions of one nucleotide implies the state for adjacent nucleotides (length is dependent on the individual factor).

can be created to represent all the possible states and transitioned for a sequence of DNA (Figure 4.2). This HMM is used to analyze all the possible configurations of components along the DNA and determining the probability of each nucleotide being in each of its possible states using the forward-backward algorithm (Austin et al., 1991).

Verifying the accuracy of model predictions is difficult, as experimental techniques cannot measure all the factors possible bound to the DNA in a single experiment. Current techniques only measure a few components at a time. The most ubiquitous component is the nucleosome. Using high throughput techniques, the DNA bound within a nucleosome is collected and sequenced. From the sequencing data, nucleosome positioning and occupancy can be determined for each nucleotide of the genome. The nucleosome occupancy can also be predicted by the COMPETE model. Correlation of the predicted occupancy with the experimentally determined occupancy is used to evaluate the accuracy of the models.

### 4.1.3 Using Biologically Inspired States to Make Better Predictions

Nucleosomes are usually modeled as monolithic components, but recent work has shown the dynamics of nucleosome formation and turnover (Böhm et al., 2011; Dion et al., 2007). Although steady state models have shown good accuracy in predicting the nucleosome occupancy in vivo, they assume a static 147 nucleotide nucleosome (see Luger et al. (2012)). The dynamics of the nucleosome implies the DNA in the entry and exit arms is more accessible to the other factors than the DNA around the core histones. The core nucleosome consists of the H3-H4 histones binding with the central ~87 nucleotides. The DNA exists in both of these states in vivo and both states should be represented in the models.

Even though nucleosomes can form with any sequence of DNA, the energy needed to bend the DNA when coiling it around the histones is dependent on the sequence. Some sequences are known to require more energy to be wrapped within a nucleosome and are therefore rarely captured in experimental data. Poly dA:dT regions (at least five consecutive A or consecutive T nucleotides) are one of these sequences (Segal and Widom, 2009b). Because they are excluded from nucleosome formation, poly dA:dT tracks form natural barriers for nucleosome positioning. These barriers have an influence on positioning that extends through multiple adjacent nucleosomes and is seen as nucleosome phasing in experimental results (Kornberg and Lorch, 1999). Poly dA:dT sequences are often seen in promoter regions near transcriptional start sites (Segal and Widom, 2009b) and have an influence on the transcription rates (Raveh-Sadka et al., 2012). This work explores how extending the COMPETE model to support other biologically inspired nucleotide states increases the accuracy of nucleosome occupancy predictions across the whole genome. I also show where these models work well and why the extended models are better able to match the experimental data.

## 4.2 Contribution

This section reiterates the contributions for this chapter. See Section 1.3 for a complete list of my contributions.

- **Extended a state-of-the-art positional model to include some of the dynamics of nucleosome formation by adding multiple nucleosome states and transitions.** The classical nucleosome is formed by eight histone proteins stably binding to ~147 nucleotides of DNA. However, there are additional stable intermediate formations (Luger et al., 2012). Pairs of histones H3 and H4 are bound first to form a core nucleosome and then pairs of other histones (H2A and H2B) combine with the core to capture the additional DNA of entry and exit arms and form a stable canonical nucleosome. I have enhanced a state-of-the-art positional model from using a single nucleosome state to one including both a core and full nucleosome state.
- **The two state nucleosome model correlates with the experimental nucleosome occupancy better than the single state nucleosome model across whole chromosomes.** The enhanced model showed an increased correlation of genome wide nucleosome occupancy values between simulated and experimental data.

### 4.3 Methodology

#### 4.3.1 Extending the COMPETE Model

One of the most well described steady state models is the COMPETE model (Wasson and Hartemink, 2009). Briefly, the model captures the possible states of each nucleotide in the sequence. Each nucleotide can only be in one of the following states: unbound, bound within a nucleosome, or bound to a transcription factor. The probability of being in those states depends on the sequence, the affinities of the factors for that sequence, and the relative concentrations of factors. The forward-backward algorithm calculates the probability of every possible configuration of factors along the DNA sequence and allows the probability of each position being in each of the nucleotide states to be determined.

Steady state models have shown respectable accuracy in predicting the nucleosome occupancy in vivo while assuming a static 147 nucleotide nucleosome. The dynamics of the nucleosome implies the DNA in the arms is more accessible to the other factors than the DNA around the core histones. I have extended the COMPETE model to incorporate the

both a full and a core nucleosome. My model includes an additional nucleotide state representing being bound within a core nucleosome. The core nucleosome biologically denotes the binding of H3-H4 histones with the 87 central nucleotides of a conventional nucleosome. I use the central positions of the di-nucleotide scoring matrix to create the transitions to the core nucleosome state.

The COMPETE nucleosome transition is based on the di-nucleotide scoring matrix and only captures sequence dependencies between two consecutive nucleotides. The di-nucleotide scoring matrix does not capture longer sequence features, such as the known exclusion of nucleosomes at poly dA:dT regions. As there are no known transcription factors that explicitly bind to a poly dA:dT sequence, I have created a pseudo transcription factor to bind poly dA:dT regions to preclude nucleosome formation.

The COMPETE model generator is able to include any number of components within the models. I selected a subset of the possible transcription factors with experimentally determined PSSMs to be included in the models for analysis. Some of these transcription factors were selected because of their involvement in regulation at specific loci in the genome: Reb1, Mcm1, Rsc3, Pho1, and Gal4. Reb1, Mcm1, and Rsc3 participate in maintaining a nucleosome depleted region at the CLN2 locus. Gal4 is the major contributor to regulation at the well studied GAL10-GAL1 locus. In addition, Rap1 was selected because of its slow turnover rate (Lickwar et al., 2012a) and Spt15, a subunit of the TATA binding protein (TBP), was selected because it is instrumental in transcription initiation.

### **4.3.2 Evaluating the Accuracy of Model Predictions**

To evaluate the predictive capability of each of the models, I compared the predicted nucleosome occupancy with an experimentally measured data set. I used Pearson correlation of occupancy values at all the positions within a given region to score the accuracy of the model. Although each individual position of the DNA is not independent of the adjacent positions (components bind multiple positions), I feel that this simple statistic is adequate for comparison between models.

There are many different experimental data sets available for nucleosome occupancy (Lee et al., 2007; Kaplan et al., 2009; Field et al., 2008). Each data set uses different protocols to measure the nucleosome occupancy across the entire genome by matching the fragments of DNA bound in a nucleosome to the genomic sequence. The more DNA found at each position, the larger the number of cells within the population have that DNA bound in a nucleosome. Unfortunately, the data sets do not correlate with each other. The Kaplan et al. and Field et al. data sets correlate much more closely to each other than to the Lee et al. data set. I chose the Lee et al. data set because of its nucleotide resolution and experimental protocol. I believe their protocol may allow the smaller core nucleosome bound DNA to be captured within the experimental results.

The Lee et al. experimental data is measured at a 4 base pair resolution and my model predictions are at single base pair. To compare the two data sets, the model prediction data is averaged over a 5 base pair window for each of the experimental data points ( $\pm 2$  base pairs).

## 4.4 Results

I compared my model predictions for nucleosome occupancy to experimental data for the entire yeast genome. I provided a quantitative comparison of nucleosome occupancy between the model predictions at single nucleotide resolution and the experimental data at 4 base pair resolution. I showed that my biologically inspired dynamic two state nucleosome model has achieved better overall correlation with experimental data than previous single state nucleosome models.

I also showed an analysis of a specific locus within the genome where the models match well and discussed the possible biological explanations for why my model performs better than the original model.

### 4.4.1 Genome Wide Analysis of Single State vs Two State Nucleosome Models

I applied the single state and the two state nucleosome models with different sets of components to each chromosome of the yeast genome. Correlations of predicted occupancy as compared to experimental data are shown for each model in Table 4.1. The two state

model consistently correlates better with the experimental data. These correlation values are consistent with previous models compared to in vivo experimental data (Tillo and Hughes, 2009).

The nucleosome only model only includes the positioning from the histones sequence affinity and lacks the competition from other components for binding the DNA. The dinucleotide nucleosome affinity scoring does not capture the aversion to nucleosome formation at the poly dA:dT sites and therefore the addition of the component for excluding nucleosome formation at the poly dA:dT sites shows the greatest increase in correlation. The exclusion sites create a barrier that limits the positioning for neighboring nucleosomes and this in turn limits the positioning of adjacent nucleosomes (Parmar et al., 2014).

The transcription factors included in the final parameter set also increase the correlation scores across the genome. Although the increase is not as large as the sequence exclusion component, the addition of the transcription factors consistently increases the correlation beyond the other parameterized models.

The correlation does not continue to increase with each additional transcription factor. I also correlated the models using additional sets of factors and recorded a reduction in the correlation (data not shown). I did not attempt to optimize the parameter set to achieve the best correlation. The parameters used were selected to highlight the models' predictions at specific regions of the genome.

The correlation to the experimental data varies along the genome and there are regions within each chromosome where the models work much better than the overall chromosome correlation. The nucleosome occupancy differs between gene regions with high nucleosome occupancy and promoter regions where there is depletion in nucleosome occupancy. I selected a set of ~4800 genes from the *Saccharomyces* Genome Database (SGD) gene annotations and correlated predicted occupancy in these gene regions against the experimental data. I created a set of promoter regions (from -500 to +100) around the annotated start position of each gene.

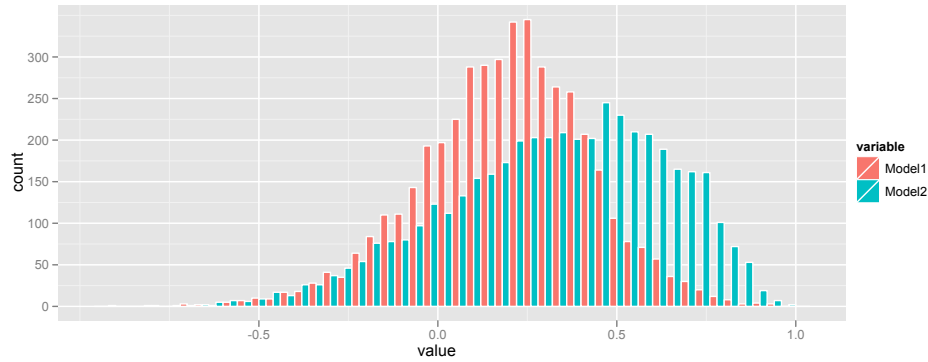
**Table 4.1:** Correlations for different models. The models were applied to multiple yeast chromosomes with different sets of parameters. The nucleosome only model uses only the di-nucleotide affinity matrix to calculate nucleosome occupancy. The Poly dA:dT model adds a pseudo transcription factor that matches only five consecutive A or T nucleotides. The last model adds a small set of transcription factors (TF) to the model (Reb1, Mcm1, Rsc3, Rap1, Phd1, Gal4, Spt15). The addition of a poly dA:dT component shows the greatest improvement in the correlations. The two state nucleosome models are consistently better than single state nucleosome models. (see supplemental for other chromosomes)

| Chromosome | Two State Nucleosome Model |                        |                              | Single State Nucleosome Model |                        |                              |
|------------|----------------------------|------------------------|------------------------------|-------------------------------|------------------------|------------------------------|
|            | Nucleosome Only            | Nucleosome + Poly_dAdT | Nucleosome + Poly_dAdT + TFs | Nucleosome Only               | Nucleosome + Poly_dAdT | Nucleosome + Poly_dAdT + TFs |
| chrI       | .223                       | .355                   | .361                         | .154                          | .335                   | .334                         |
| chrII      | .213                       | .326                   | .355                         | .149                          | .309                   | .327                         |
| chrIII     | .213                       | .344                   | .357                         | .151                          | .311                   | .322                         |
| chrIV      | .207                       | .300                   | .341                         | .145                          | .283                   | .312                         |
| chrV       | .215                       | .305                   | .340                         | .162                          | .281                   | .307                         |
| chrVI      | .211                       | .333                   | .351                         | .154                          | .309                   | .320                         |
| chrVII     | .212                       | .316                   | .359                         | .145                          | .300                   | .330                         |
| chrVIII    | .217                       | .327                   | .346                         | .150                          | .308                   | .315                         |
| chrIX      | .234                       | .315                   | .350                         | .171                          | .297                   | .324                         |
| chrX       | .224                       | .330                   | .363                         | .169                          | .308                   | .333                         |
| chrXI      | .215                       | .311                   | .341                         | -                             | -                      | .316                         |
| chrXII     | .205                       | .313                   | .346                         | .146                          | .298                   | .320                         |
| chrXIII    | .219                       | .315                   | .355                         | .158                          | .296                   | .323                         |
| chrXIV     | .212                       | .312                   | .341                         | .139                          | .292                   | .310                         |
| chrXV      | .216                       | .308                   | .347                         | .160                          | .280                   | .313                         |
| chrXVI     | .223                       | .311                   | .356                         | .157                          | .294                   | .324                         |

For each gene and promoter region, the predicted occupancy was individually correlated to the experimental data. Figures 4.3 and 4.4, I show a histogram of the number of regions with each correlation value. I found that the correlations in the promoter regions are significantly higher than the correlations in the gene regions (Table 4.2). This trend is true regardless of whether the two state or single state nucleosome model is correlated. When I compare the histograms of the two models in the gene regions or the two models in promoters regions, I do not see a significant change in the distribution of the correlation values, indicating neither model is significantly better in either subregion.

**Table 4.2:** Correlation values for Two State vs One State Nucleosome Model

| Regions   | Two State Model | Single State Model |
|-----------|-----------------|--------------------|
| Promoters | $0.33 \pm 0.31$ | $.31 \pm .32$      |
| Genes     | $0.2 \pm 0.23$  | $.17 \pm .22$      |



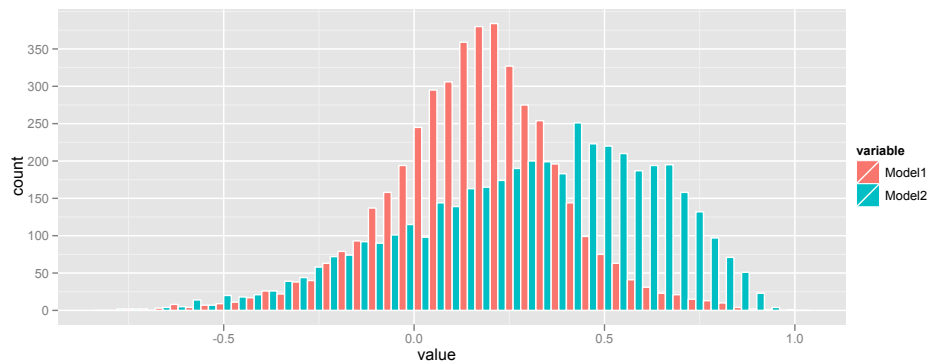
**Figure 4.3: Histogram of correlation values for all gene and all promoter regions using Two State Nucleosome Model.** The distribution of correlation values for a set of ~4800 genes across the entire genome is shown for gene regions and promoter regions. Each region in a data set is correlated with the corresponding experimental data and shown as a histogram of the number of regions with each correlation value (binned in .04 ranges). Gene regions (spanning the annotated regions from SGD) are shown in red (mean=0.2, sd=.23). The promoter regions (-500 to +100 around the annotated start site) are shown in blue (mean=.33, sd = .31).

#### 4.4.2 Analysis of CLN2 Promoter Region

When I zoom in on a specific promoter region within a genome, I can plot each model's predictions for nucleosome occupancy with the experimental data (Figure 4.5). The experimental data shows small peaks within nucleosome depleted regions (see the nucleosome depleted regions near positions 67200, 67600, and 69200 on chromosome XVI in Figure 4.5). These small peaks are too narrow to represent a full nucleosome's stable formation. It is possible these regions represent the capture of a smaller stable configuration within the dynamics of nucleosome formation. The core nucleosome fits in these regions (see Figures 4.5 - 4.7).

Visualizing the predicted occupancy by each of the models, I can visualize where the model matches with the experimental data. The single state nucleosome model fails to place nucleosomes in these three regions, however, the two state nucleosome model captures increased nucleosome occupancy within these regions. Figure 4.7 shows the overall correlation for the region using the single state model is 0.56 and increases to 0.63 using the extended two state model. The small peak regions show the greatest difference between



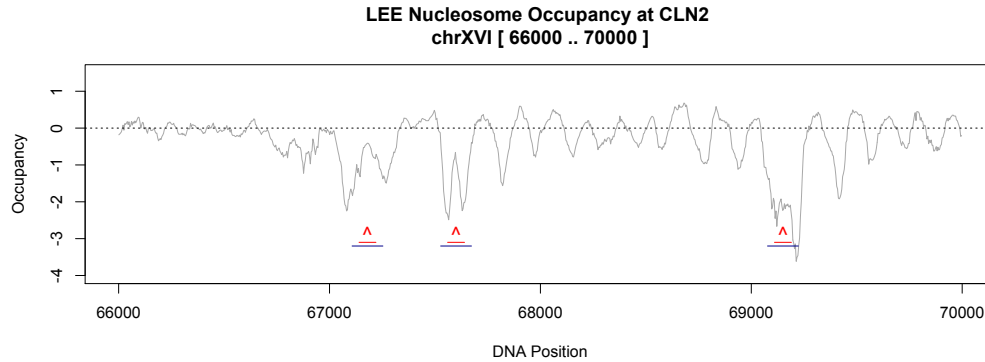


**Figure 4.4: Histogram of Correlation Values for all Gene and all Promoter regions using Single State Nucleosome Model.** The distribution of correlation values for a set of  $\sim 4800$  genes across the entire genome is shown for gene regions and promoter regions. Each region in a data set is correlated with the corresponding experimental data and shown as a histogram of the number of regions with each correlation value (binned in .04 ranges). Gene regions (spanning the annotated regions from SGD) are shown in red (mean=0.17, sd=.22). The promoter regions (-500 to +100 around the annotated start site) is shown in blue (mean=-.31, sd = .32).

the two state nucleosome model and the single state nucleosome model as a result of the smaller core nucleosome binding in this region.

#### 4.5 Discussion

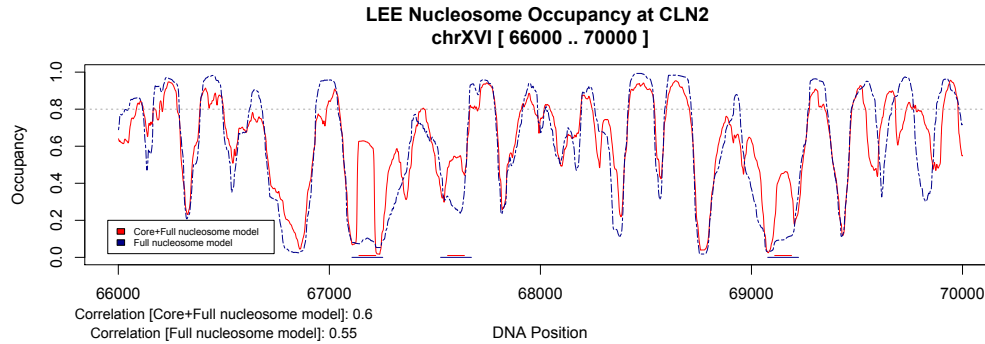
As the field of Biology continues to expand our knowledge and understanding, we must re-evaluate the current models to incorporate the new data or capture alternative theories. I have taken the new knowledge of the dynamics involved in nucleosome formation and re-examined the modeling assumptions used in the current state-of-the-art nucleosome occupancy models. I have extended the models to include the dynamic formation of the nucleosome by creating multiple biologically derived states in nucleosome formation within the steady state COMPETE model. I have shown that the extended model is able to more closely match the experimental data on both the whole genome and at individual promoter regions. Transcriptional regulation is a complex system of dynamic interactions occurring within a cell. Steady state modeling methods attempt to capture the behavior of a population of cells as they alternate between distinct configurations of components bound to the DNA. The models are able to capture the interesting peaks within the nucleosome depleted regions near CLN2 and could represent the dynamics of chromatin remodeling. Nucleo-



**Figure 4.5: Chromosome XVI 66000..70000 showing experimental nucleosome occupancy data from (Lee et al., 2007).** There are three positions highlighted with red arrows where the experimental data shows an internal peak within a nucleosome depleted region. The width of a full nucleosome (blue) and a core+full nucleosome (red) is shown below each of these positions to give a visual indication of the width peak.

somes forming in the region can be shuttled away by remodeling factors. Depending on the efficiency of the remodeling factors, only a few cells will have nucleosomes in that region at the experimental time.

Each nucleosome component uses the same di-nucleotide scoring matrix to calculate the affinity score, but they use different lengths of DNA sequence. The best scoring position for the full 147 nucleotide nucleosome is not always the same position for the 87 nucleotides of a core nucleosome. The extended model shifts the probability of binding to adjacent positions where the smaller nucleosome would bind. This shift results in the nucleosome occupancy predictions having better correlation to experimental data across large regions of DNA. I assumed that there would be a subset of genes or gene regions that would correlate better with one model than the other. However, I was not able to find a definitive subset. I divided the genome into regions by gene expression levels, by gene lengths, by genes known to be regulated by modeled TFs, and by regions containing the nucleosome signature profile shown in Figure 4.7. There was no significant difference between the two models for any of these subcategories. This may be related to another observation about the correlation values. I observed that the correlation values are increased across large regions, but when the correlations are performed over small windows of only a few nucleosomes, the differentiation

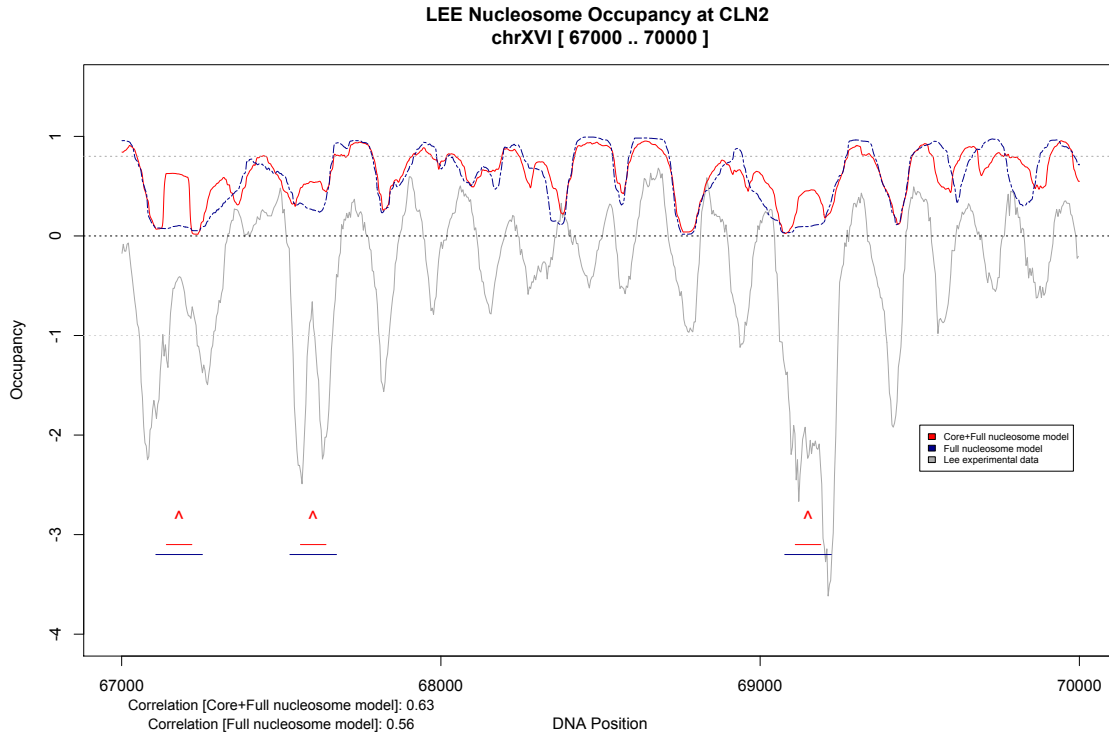


**Figure 4.6: Chromosome XVI 66000..70000 showing model with one state nucleosome only (blue dashed line) and the two state nucleosome model (red solid line).** The three positions (describe in figure 4.5) are highlighted with markers showing width of core nucleosome (red) and full nucleosome (blue) sizes. The full nucleosome model does not capture these sub-peaks, but our core nucleosome model captures the probability of a core nucleosome in these areas. The overall correlation in this region between the experimental data and the one state nucleosome model is 0.55, while the correlation of the two state nucleosome model and experimental data has increased to 0.60.

of the correlation scores between the models is lost. The Pearson correlations measure how well the general patterns or trends of increasing or decreasing occupancy matches across all positions. When correlating a large region, the minor fluctuations are smoothed, but correlation scores across smaller regions are still sensitive to these fluctuations. The gene and gene promoter regions are relatively small when compared to the length of a whole chromosome. The reason that I have not been able to cluster some of these regions may be that the signal is lost when focused on the small regions of only a couple nucleosomes.

The transition between the two nucleosome states is currently modeled as independent transitions from the unbound state (Figure 4.1). This means that DNA bound in a core nucleosome does not have a higher probability of becoming bound in a full nucleosome. Extending the model to allow transitions directly between the two nucleosome states (direct transition from state C to state N) may increase the extended model's prediction capabilities. This modification would allow high affinity sites for the core nucleosome to increase its occupancy, shifting more probability away from the other full nucleosome positions.

The size of the DNA linker between nucleosomes is a fixed length in this HMM. The in vivo length of DNA between nucleosomes is a function of nucleosome affinity and chromatin



**Figure 4.7: Chromosome XVI 66000..70000 showing model with one state nucleosome only (blue dashed line), the two state nucleosome model (red solid line), and experimental occupancy (gray line).** See figures figure 4.5 and figure 4.6 for descriptions.

remodeling activity and therefore is not constant across the whole genome. The correlations between model predictions and experimental data are strongest where the regularity of nucleosome phasing matches the model parameters. When the length of the linker is modified in the models, the locations where the models highly correlate is also modified. The models would need to incorporate the effects of remodelers to account for this dynamic behavior. It is difficult to extend a HMM to model the movement of states through the DNA. It would require multiple copies of the HMM for each position of the moving object, which would create an exponential explosion in the state space and therefore is a computationally unrealistic solution.

Transcription factors change the local nucleosome binding probability at location of high transcription factor affinity. In the modeling examples I have presented, only a small set

of transcription factors were included. Many of the other transcription factors are known to bind and affect the nucleosome occupancy in gene promoter regions. The correlation of prediction to experimental data did not increase as more factors were introduced into the model. This may indicate that the promoter regions are already being occluded by the factors even though the transcription factors may only have weak affinity in the region. Addition of more factors may create an over abundance of nucleosome exclusion which would effect the placement of many adjacent nucleosomes.

As more components are added to the models there is an increase in the number of variables (concentrations and affinities) that can be adjusted to maximize the correlations. These variables allow more flexibility in the models, and as the number of variables increases in a model, there is more potential for overfitting the model to the experimental data. I have not attempted to solve the model and find the parameters that are the best fits to the experimental data.

I have also added the core nucleosome to the models generated by my modeling framework (presented in Chapter III). Using this completely different modeling method, I also observed an increase in correlation values. Although I have not been able to find a distinct subset of regions where the two state model works better or to find the cause for the increase in correlation, it seems that there may be a biological behavior that these models are partially capturing.

While the simple dynamics of nucleosome formation can be captured using steady state models, there are regions along the genome where components actively modify the chromatin structure. Chromatin remodelers can evict otherwise stable DNA binding factors or actively move them along the DNA to less favorable positions. The behavior of these components affects not only the state transitions at a local position, but also the state transitions at many adjacent positions along the DNA. The behavior of this type of dynamics cannot easily be captured using a HMM or other steady state modeling methods. It requires a change in the modeling perspective from population averaged behavior to behavior of a single cell. Simulations of the events occurring along the DNA can easily capture the dynamics of

multiple components simultaneously changing positions along the DNA. This was a major motivation for creating the modeling framework I presented in Chapter III.

#### 4.6 Conclusion

I have extended a state-of-the-art steady state model, COMPETE, to include the dynamic formation of the nucleosome by creating multiple biologically derived states in nucleosome formation. The results show that the extended model is able to more closely match the experimental data on both the whole genome and at individual promoter regions.

Every chromosome correlates better with the two state model than the one state model. There are specific loci within the genome where the two state model clearly captures the diversity within the experimental nucleosome occupancy data better than the one state model. I was not able to find features of a subset of genes or gene promoters where the two state or one state model would out perform the other. This may be due to the fact that smaller regions of DNA do not show the difference in correlation values between the models.

The increased correlations with experimental data was also observed when a additional state for the core nucleosome was added to the modeling framework described in Chapter III. It appears that these models may be capturing a biological behavior as both methods have an increased correlation.

## CHAPTER V

### VISUALIZATION AND EDUCATION

I hear and I forget.  
I see and I remember.  
I do and I understand.  
Confucius (551 BC - 479 BC)

Most of the work summarized here comprises parts of Aims 2 and 3 in a National Science Foundation grant (ABI 1262410) on which I was a co-author. The goal of the grant was to create a teaching tool for transcriptional regulation. In the grant aims, we envisioned a game that allowed student interaction in creating DNA sequences, simulating the regulation for the given sequence, and visualizing the results as animations. Through direct manipulation of the DNA sequence, students would gain an understanding of the difficulty of obtaining desired phenotypes in a complex environment.

Imagine the scenario where a student is given an expression profile for a gene. The profile shows how the expression changes over time given a transcription factor concentration change over time. The student is then asked to recreate the expression level by designing a regulatory circuit and simulating the results for changes in transcription factor concentration. The student would create the regulatory sequence using a drag and drop interface to place different regulatory elements (components) into a custom DNA sequence around the gene. Components for all the different transcription factors, general transcription factors, nucleosome affinity, and RNA polymerase dynamics are available for inclusion in the model. The student can watch animations of the molecule interaction simulations, examine summary plots of the results, and compare these to the desired results. The student can experiment with different mechanisms until the simulated results match the desired profile. As the students interact with the mechanisms, they obtain a much deeper understanding of how these mechanisms work and interact.

The value of tools for educating both researchers and students is dependent on two tasks: 1) achieving reasonably biologically realistic simulations of transcriptional regulation

mechanisms and 2) creating viable dynamic visualizations that convey meaning given the large amount of interactions that occur.

In this chapter I show my work on integrating research and education that addresses the dynamic visualization aim. My approach for understanding transcriptional regulation assumes that all users, from expert researchers to high school students, benefit from tools designed to aid in understanding the complex system of transcriptional regulation. I am a proponent of an interactive, hands-on approach that brings together powerful simulations with quality visualization, which are key to understanding complex systems. My work promotes teaching, training, and learning while providing an outstanding training opportunity to promote interdisciplinary research at the confluence of Computer Science, Molecular Biology, and Education.

My work has included a number of projects that address the use of visualization in all stages of scientific research and education. I was involved in bringing computational thinking into classes at the K-12 level through an NSF funded ECSITE project and working with the University of Colorado undergraduate International Genetically Engineered Machines (iGEM) team, which provides intense interdisciplinary undergraduate internships in systems biology. I also taught an introductory programming class for biologists using an inverted classroom, utilizing video lectures so students can ‘hear’ and ‘see’ the concepts. I have produced a short introductory video that describes the basic concepts of transcriptional regulation that includes the often ignored concepts of stochasticity and the dynamics of the transcriptional machinery. I have also created a graphical representation for describing interactions between model components and visualization of simulation results using character graphics and animations.

## 5.1 Introduction

The best way to understand complex systems is to interact with the system. Experimental approaches allow one to make changes to DNA sequences of genomes and measure the impact of these changes on the transcription process. While powerful, these experiments may take days or weeks in the laboratory. It would be better for teaching purposes



to be able to make changes to DNA and simulate the anticipated transcriptional response. Tools for simulation, visualization, and interaction with models are integral to enriching our understanding of transcriptional regulation. These tools will be critical to advance research in transcriptional regulation and as teaching tools. This approach has been used in Physics (Wieman et al., 2008) and Biology for protein folding (Cooper et al., 2010).

There are many complex systems in the world that we must deal with everyday. We perceive the world through vision and we even communicate via vision, as words alone on a page are often insufficient to communicate or understand complex interactions. To aid the complete understanding of complex concepts, we include images that explain and highlight the important ideas. Our visual sense is a highly developed process that can bring insight and understanding to many complex systems. The old saying “a picture is worth a thousand words” shows that we have understood the importance of images for a long time.

Scientific research is an iterative process of forming a hypothesis from current knowledge, designing experiments to test the hypothesis, collecting data from experiments, analysing the data, and communicating the results. Unfortunately, visualization of scientific data is usually only used in the last step of the scientific process to communicate the results to others. The power of visualization can provide the understanding and inspiration for greater advancements if it becomes an integral part of the iterative scientific process as an exploration tool (Rinaldi, 2012; Fox and Hendler, 2011).

Biology is so complex and multifaceted that almost every new technology and experimental technique requires a new visual framework for representing and presenting the new and highly detailed data. The mistake made by many computer scientists outside of biology is thinking that general principles will solve all the problems in biology, when the real challenge is to adapt those principles to specific experimental situations (paraphrased from O’Donoghue, Rinaldi (2012)).

Because Biology is a complex system and we cannot directly observe its complexity, we create models of how we think a process works and attempt to validate the model through experimentation (Figure 2.1). Visualization can be used at each step of this iterative process,

making it easier to communicate the intentions when building models, as well as interpreting the results.

## 5.2 Contribution

This section reiterates the contributions for this chapter. See Section 1.3 for a complete list of my contributions. This work comprises part of Aims 2 and 3 in a National Science Foundation grant (ABI 1262410) on which I was a co-author. The ASCII visualizations of the DNA states are described in a manuscript under review in IEEE Transactions on Computational Biology and Bioinformatics as "A modeling framework for generation of temporal and positional simulations." The educational video (<http://dowell.colorado.edu/vizgrp/>) was entered into the National Science Foundation's Visualization Challenge for 2014. The application for visualizing proteome conservation across bacteria was published in BMC Genomics: Rokicki, J., Knox, D., Dowell, R. D., and Copley, S. D. (2014). "CodaChrome: a tool for the visualization of proteome conservation across all fully sequenced bacterial genomes" (doi:10.1186/1471-2164-15-65).

- **Produced a short introductory video to describe the stochastic nature of the transcription process and the dynamics of the transcription process to be used as teaching material for undergraduate introductory biology courses.** There are two concepts that this video uniquely addresses: the stochastic nature of the transcription process and the dynamic behavior of the transcriptional machinery that contributes to transcriptional regulation. This video explains the basic concepts behind transcription as a necessary biological process that is the first step in creating proteins in a cell. It visualizes the complex interactions of transcription factors, nucleosomes, and the transcriptional machinery with the DNA, including the often ignored mechanism of transcriptional interference. The video was created in conjunction with a summer internship program for undergraduate Computer Science students (Michelle Soult, Catherine Dewerd, Hayden Berge) and submitted to the National Science Foundations' Visualization Contest in 2014 ([www.nsf.gov/news/special\\_reports/scivis](http://www.nsf.gov/news/special_reports/scivis)).

- **Created a language to abstractly describe component interactions as graphs.** The Petri net graphs represent the component states and state changes that occur with each interaction. The language describes the syntax and semantics for abstract templates and variable substitutions used to generate multiple related interactions for each abstraction. This allows complex models to be generated from less complex abstract interactions.
- **Created an ASCII visualization of configuration of components bound to the DNA at each time step of a simulation.** Every time point of the simulation provides a snapshot of the configuration of factors bound to the DNA, which can be used to reveal patterns of behavior not seen in summary results. From the output of the model simulations, I generate an ASCII visualization of the configuration of factors bound to each nucleotide of the DNA at each time step. The state of each position of the DNA is uniquely represented as a single character linearly in a line of text. The movement of factors along the DNA can be inferred by the movement of factor positions in consecutive display lines.
- **Created an animation of the component interactions based on the intermediate simulation results.** The simulation results can be interpreted as a script for component movement. Each time point specifies the configuration of components along the DNA. By inferring the movement of components between time points, a trajectory for individual components can be calculated. A visualization framework, provided by Unity, was used to manage the virtual environment and display of individual component movements. The goal of the animations was to provide visual feedback on the cellular behavior based on the modeling parameters. Ultimately, the animations would be used in a teaching tool for students studying transcriptional regulation. The animations were created in conjunction with a summer internship program for undergraduate Computer Science students (Chad Bryant, Emily Owens, Malcolm Duren).

- **Mentored a fellow graduate student (Joe Rokicki) in the creation of a tool for the visualization of proteome conservation among bacterial genomes.** The relationships between bacterial genomes are complicated by rampant horizontal gene transfer, varied selection pressures, acquisition of new genes, loss of genes, and divergence of genes, even in closely related lineages. As more and more bacterial genomes are sequenced, organizing and interpreting the incredible amount of relational information that connects them becomes increasingly difficult. CodaChrome is a user-friendly and powerful tool for simultaneously visualizing relationships between thousands of proteomes recorded in GenBank. The relationships between a bacterial proteome of interest and the proteomes of every other bacterial genome are visualized as a massive interactive heat map. published in BMC Genomics: Rokicki, J., Knox, D., Dowell, R. D., and Copley, S. D. (2014). “CodaChrome: a tool for the visualization of proteome conservation across all fully sequenced bacterial genomes” (doi:10.1186/1471-2164-15-65).

### 5.3 Educational Videos

#### 5.3.1 Inverted Classroom

Today’s students have expectations of instantaneous access to small units of information and know that information can be delivered as needed. However, they face learning and combining diverse subjects to solve the problems of the world. Learning programming skills is comparable to learning other laboratory science skills. The concepts can be introduced during a lecture, but learning is obtained only after application of the concepts, usually over and over, until students gain the skills necessary to produce a desired result. The student is expected to try and fail often, spending many hours of hard work only to fail. It is during those hours, when students are working alone and encountering barriers to their progress, where the best teaching opportunities occur.

To take advantage of those teaching opportunities, I used an inverted or “flipped” classroom for teaching computer programming. Traditionally, a lecture is given live in a classroom and is followed by students performing homework, often alone and struggling. An

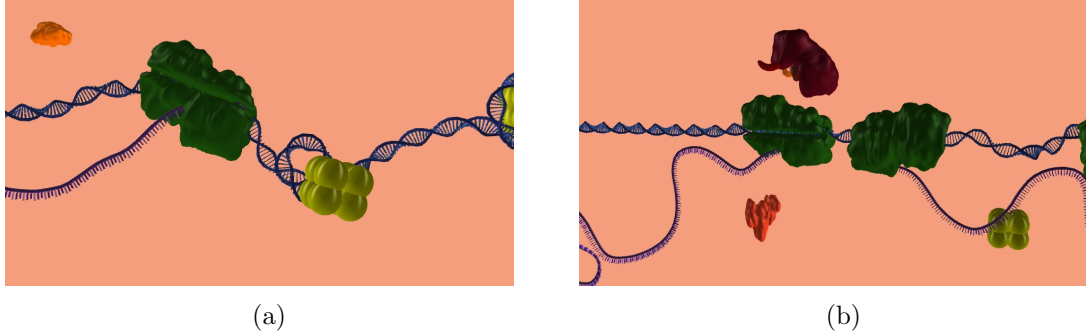
inverted classroom provides prerecorded lectures that are viewed away from the classroom, while class time is used for homework and active learning methods that increase student engagement and result in a deeper understanding of concepts while gaining a mastery of the skills. Together with Phil Richmond, a talented recent graduate and professional research assistant in the Dowell laboratory, I created a set of videos where the students not only hear the information, but can see the the concepts being used. Prerecorded videos allow students to progress at their own rate, re-watching and pausing as needed (<http://dowell.colorado.edu/educationpython.html>).

### **5.3.2 Transcriptional Regulation Video**

Visualization of complex biological systems is difficult because many different components are working simultaneously. While it is difficult to describe these interactions within text, motion pictures provide a platform to convey the behavior of complex interactions. Video can show an interaction in different orientations or from different perspectives and viewers can replay the video to focus on different aspects, until they obtain a thorough understanding of the concepts.

A quick search of teaching videos for transcriptional regulation reports several hundred short videos are available. However, I did not find any that focused on the stochastic nature of transcription and its regulation. In the summer of 2014, I managed a group of talented undergraduate computer science students in an internship program. We produced a five minute video highlighting the stochastic behavior of the transcription process and the often ignored transcriptional regulation behavior of interference (Figure 5.1).

Although the stochastic nature of the cellular systems is understood, it is not usually taught at the undergraduate level and the concepts of transcriptional regulation are typically taught from textbooks with static images. Most of the dynamic and stochastic nature of the transcriptional process, such as factor competition and transcriptional interference, are abstracted away to produce these static pictures. To provide an introduction to these concepts, I lead a team of undergraduate students to produce a short video that specifically portrays the dynamics and stochastic processes and higher level regulation concepts.



**Figure 5.1: Screenshots from video on Transcriptional Regulation.** a) Screenshot from video showing transcriptional machinery (green) and nucleosome (yellow histone octamer) interactions with DNA. b) Screenshot showing transcriptional interference.

The video explains the basic concepts behind the dynamic process of transcription. It visualizes the complex interactions of transcription factors, nucleosomes, and the transcriptional machinery with the DNA. This video is unique as it demonstrates the dynamic regulation of transcriptional interference, an important but often overlooked aspect of regulation.

The animation in the video shows how the interactions of individual molecules within a cell are thought to behave. The combined action of individual components provides insight into the dynamics and stochastics of the transcriptional process. Models and colors were chosen to optimize the learning of concepts while remaining at a level of detail understandable by the target audience.

We designed our video to reach a target audience of undergraduate students studying introductory biology. The terminology has been taken from concepts in the current textbooks and extended to include the new ideas conveyed in this video. The narrative builds in complexity and reinforces new concepts with visuals.

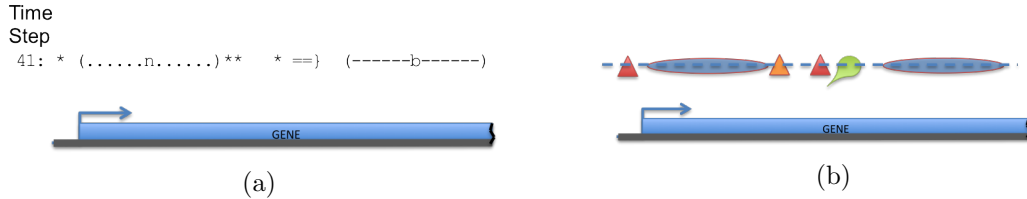
The development of the video followed the same process as used in the creation of animated films. A storyboard was created to address the concepts that were to be covered. From the storyboard, the individual scenes were staged, action and movements of components defined. Next the individual components were created as simple 3D objects that could be manipulated using Autodesk Maya 3D computer graphics software. These simple

representations were used to create the movement and interaction of components to capture teaching concepts. Once the general flow within a scene was defined, the representation of the components was refined to create more realistic complex structures, lighting and shading were added, and the voice over text was recorded. The integration of voice and animations requires the adjustment of timings of both the narrator and the animations to produce the final video.

#### 5.4 Visualization of Simulation Results

The traditional paradigm for modeling transcriptional regulation has focused on the individual components involved and is usually achieved using mathematical models. These models are based on a population averaged response to the initial conditions and predict changes of individual component concentrations over time. But, there is growing evidence that transcription is regulated not only by the individual components, but also by the competition among all components for the DNA. It is the stochastic, temporal and spatial interactions of these components that ultimately control the transcription process within each individual cell. Understanding this new interaction model of transcriptional regulation requires a paradigm shift from modeling the population average cell to modeling the molecular events within a single cell. We need tools that can model the molecular interactions for any DNA sequence and visualize those interactions along that DNA.

The modeling framework in Chapter III describes a tool for generating models and simulating the transcriptional regulation behavior for any DNA sequence. The simulation engine produces a list of molecule counts for each reactant species at each time point in the simulation. The reactant species are created by the modeling framework to represent the different states of each DNA position. Each of the DNA positions can only be in one state at a time, therefore the molecular counts represent a binary value for each possible state of the DNA position (Figure 5.2). The SRB-Visualizer interprets the state counts to determine the higher level abstraction of components bound to the DNA.



**Figure 5.2: ASCII visualization of a single time point.** The state of the simulation is recorded for each time step of the simulation (y-axis) and can be interpreted by understanding the states generated by the framework to determine the current configuration of factors bound to the DNA (x-axis). Here I show a single simulation time point example from a simulation that included 3 transcription factors, nucleosomes, and the transcriptional machinery. (a) A simple ASCII representation of the high level cartoon depicting the gene (blue rectangle) and transcription start site and direction (arrow). Each character summarizes 10 nucleotides of the DNA.

| Symbol  | Description  |
|---------|--|
| *       | bound transcription factor   |
| ==}     | bound transcriptional machinery (position not transcribed)   |
| ==]     | bound transcriptional machinery (position transcribed, waiting to advance)   |
| (.....) | bound nucleosome. Nucleosomes are further labeled as binding (b), unbinding (u), or stable (n) to reflect the intermediate states of nucleosome formation. |

(b) An alternative representation of the ASCII characters, redrawn into cartoon representations of the DNA configuration with transcription factors (triangles in red and orange), nucleosomes (blue oval), and the transcriptional machinery (green teardrop).

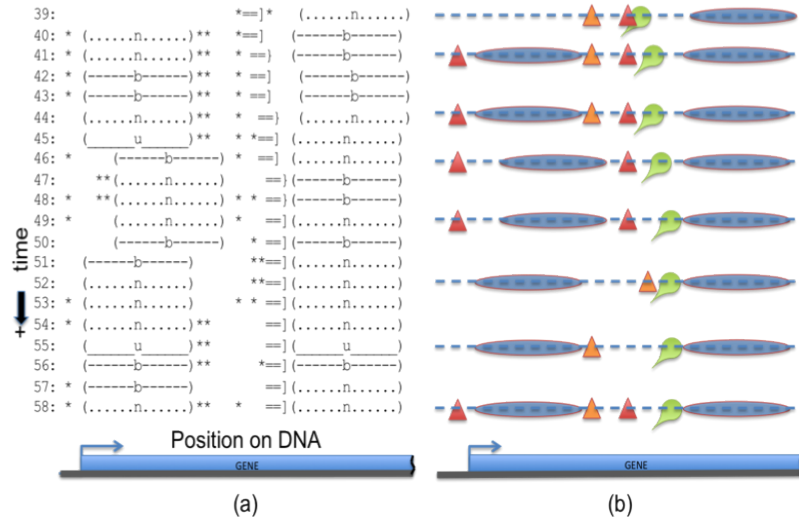
### 5.4.1 Character Graphics

An underlying concept of my modeling framework is that DNA is a long connected serial sequence of positions and each of those positions can only have a single component bound at any particular time point. Using a single character to represent the state of each DNA position allows the entire sequence configuration to be represented in a single line of characters (Figure 5.2). The SRB-Visualizer produces a series of component configuration lines, one for each time point of the simulation (Figure 5.3).

Using character graphics to represent the state of the DNA at any time point requires the dynamics of binding and the movement of components along the DNA to be inferred by the viewer. When the time points are viewed as a series (Figure 5.3), the movement of a component along the DNA is seen as a positional shift of the component from one time point to the next. Over a large number of time points the movement and dynamic behavior can be conceptualized. Figure 5.3(a) shows a character representation for the configuration



of factors bound to a segment of the DNA sequence for individual time points. The representation for the transcriptional machinery can be seen to move to adjacent positions when scanning down through the stacked time points. The movement can be inferred from the consecutive time points and knowledge of the component behavior.



**Figure 5.3: Visualizing the DNA configurations at each time step of a simulation.** The state of the simulation is recorded for each time step of the simulation (y-axis) and can be interpreted by understanding the states generated by the framework to determine the current configuration of factors bound to the DNA (x-axis). Here I show an example of a simulation that included 3 transcription factors, nucleosomes, and the transcriptional machinery. (See 5.2 for legend of the symbols). (a) A simple ASCII representation where each character summarizes 10 nucleotides of the DNA. When the image is viewed as a whole, scanning from top to bottom, I observe movement of a single transcriptional machine along the DNA and pausing at a nucleosome with perhaps strong sequence affinity. (b) An alternative representation of the ASCII character representation in a cartoon form.

#### 5.4.2 Animations

An animation of the movement would provide an explicit visualization of the dynamics. Using the location of components at each time step, I can build a script for explicit instances of components to appear on stage, move to specific locations over time, and move off stage. The script can then be used to create an animation of components interacting with the DNA.

Working with a group of undergraduates during a summer internship program, a prototype application to generate animations was developed. Our solution interprets the results

of the simulations to generate a set of component locations for each time point. These locations are used to generate a set of movements for each component between time points, which are fed to our display routines implemented in the Unify game engine. The prototype animations renders simple 3D objects for each component (Figure 5.4).

## 5.5 Discussion

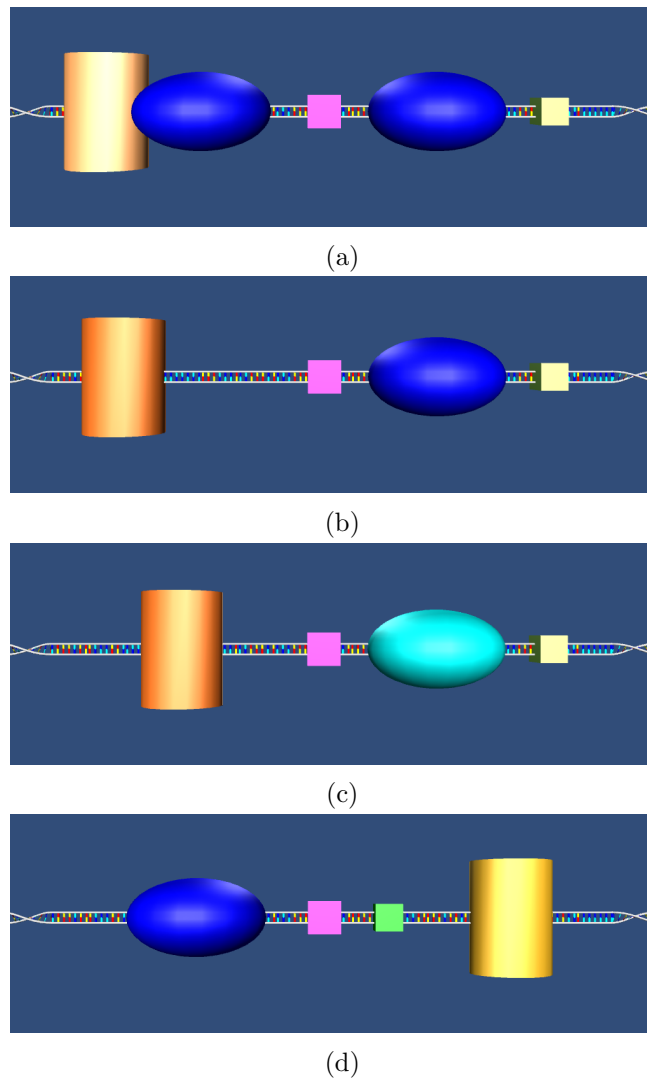
The visualization projects described above focus on conveying transcriptional regulation concepts and displaying the results of simulations using either static images or animations. These are still active research projects and I have reported the current status of these ongoing projects. The creation of an interactive teaching tool for teaching the dynamics of transcriptional regulation is the next step. The processing pipeline (Figure 5.5) has been prototyped and needs to be integrated into a web frontend to handle students and lessons.

### 5.5.1 Visualizing the Interaction Network

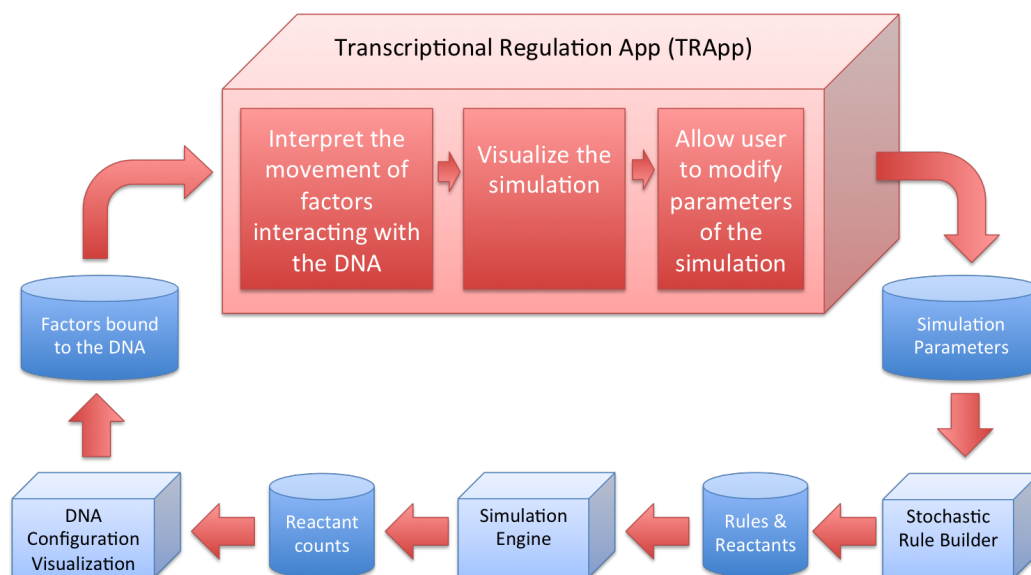
I have described visualization of the simulation results, but there is also the visualization of the abstract Petri nets and the resulting interaction networks. The modeling framework creates a network of states and transitions between states that can be represented as a graph. With the help of a talented high school student, I have explored using a popular visualization application, Cytoscape, to visualize the graphs. Cytoscape provides a programmatic interface for network layout and adjusting the display features. It can even be used to create new Petri net interaction abstractions with all the appropriate attributes or parameters. We found that Cytoscape displays did not scale well to the millions of states and interactions generated for large DNA segments and could only display a small subset at any one time. Cytoscape does provide a good interface for displaying, creating, and editing the individual interaction descriptions in a graphical environment.

### 5.5.2 Automated Conversion of Graphic Representations for Abstract Rules

Another ongoing research project is the automated conversion of a graphic representation of an interaction into a form that can be applied to a specific DNA sequence. In my modeling framework, I use Petri net representations to describe the abstract interactions used to generate the model rules (see Chaouiya (2007) for review of Petri nets). Petri nets



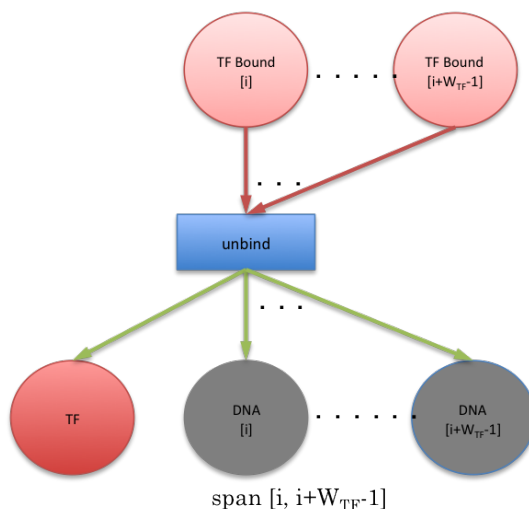
**Figure 5.4: Animation of the component configuration over time along a DNA segment.** The prototype animation using a game engine shows the same DNA segment at four distinct time points. The nucleosomes are shown as ellipsoids (light and dark blue depending on state). The transcription factors are small cubes (different colors for each factor). Transcriptional machinery is shown as a cylinder (different shades of gold depending on state) can be seen to move along the DNA. a) Transcriptional machinery stalled, waiting for removal of nucleosome in its path (right). b) Nucleosome has been removed, transcriptional machinery resumes elongation. c) Transcriptional machinery continues transcription. d) Factors bind again after the machinery has moved through the region.



**Figure 5.5: Pipeline for creating simulation results for a given set of parameters.** Given a set of parameters (right edge of figure) to the backend server processes (in blue), a model is built by the stochastic rule builder modeling framework and simulated by an off-the-shelf simulation engine. The results of the simulation are interpreted to provide component positions for each of the simulation time steps, which is passed to the frontend TRApp application. TRApp interprets the movement of individual components along the DNA from the time step component positions. An animation script is generated to describe the movement of individual components that can be used to control the animation display. The user can modify the parameter settings and resubmit new parameters to generate another animation of the system behavior.

are designed to describe the inputs and outputs of an action in a graphical form (Figure 5.6). The process of converting the individual graph descriptions of abstract interactions into code is currently a manual process. To modify or enhance the model component behavior requires programming skills. However, we can describe the abstractions in a computer readable form and therefore the creation of rules directly from the descriptions can be automated.

Languages for describing biochemical interactions have been well defined for decades (Hucka et al., 2008; Faeder, 2011). Petri nets are just one way of describing the interactions from an action point of view (Figure 5.6). Standardized descriptive languages have been developed to aid the development of Petri net description and simulation (Billington et al., 2003; Iec et al., 2009). However, each modeling framework is creating an abstraction of an explicit biological system using the standardized nomenclature.

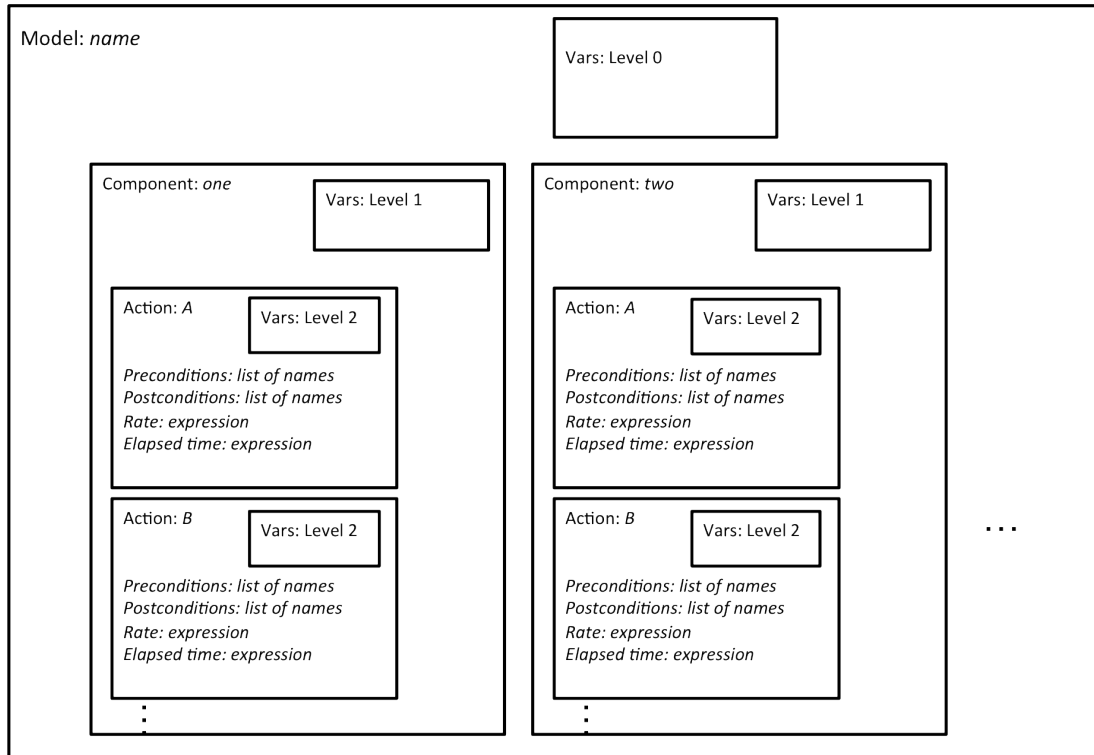


**Figure 5.6: Sample abstract Petri net for the unbinding of a factor from DNA.** Each TF binds to consecutive nucleotides for a length matching the area occluded by the TF. Transcription factor unbinding consumes molecules of TF bound DNA for each position and produces a molecule of the TF and unbound DNA for each position.

Here I describe the syntax and semantics of a description language for converting graphical representations into explicit model rules. The syntax provides a means to parse the form of the descriptions and the semantics provides the interpretation of the parsed instructions to ensure the correct resolution of variables to generate explicit rules.

The Petri net descriptions provide a graphical description of the rules to be generated based on abstract inputs and outputs that are dependent on the position within a DNA sequence. To keep the model developer from needing to explicitly list the same interaction at every position of the DNA, I create abstract rules that describe an interaction based on a variable representing the current position of the DNA. When describing more complex interactions that are replicated over a range of other variables, again I can write the simple interaction with the assumption that a variable value will be iterated over a given range.

Each graph represents an interaction that consumes a set of nucleotides of DNA in a particular state and produces the same set of nucleotides in another state. The abstract interaction defines positions of the nucleotides from a single base position. Other abstractions may need to be applied multiple times at each base position for a range of variable values.



**Figure 5.7: Model definition includes variables at each level of description.** The model definition is comprised of a set of individually defined components, which are described as a set of independent interactions. Each description there may be a set of variables on which the descriptions depend for generating the rules. The scope of the variables is only within the object for which the variable is defined and that object’s descendants.

This preliminary syntax below describes how to specify the abstractions and variable substitutions for each interaction description. The syntax was designed to handle all the abstractions encountered in producing rules for the interactions described in Appendix I. The model definition is comprised of a set of individually defined components, which are described as a set of independent interactions (Figure 5.7). Along with each description there may be a set of variables on which the descriptions depend for generating the final rules. The scope of the variables is only within the object for which the variable is defined and that object’s descendants. Therefore, different components or actions can use the same variable name and override the local values.

```

var_expr := string | var_name | '(' var_expr ')' | '\(' | '\)'
          | var_expr op var_expr
          | var_expr '(' ')'
          | var_expr '(' var_list ')'

var_name := '%' string '%'
var_list := var_expr | var_expr ',' var_list

var_value :=          var_expr | "RANGE" '(' var_list ')' | "SET" '(' var_list ')'

```

**Figure 5.8: Preliminary Model definition syntax.** The syntax describes the variables and their values for each abstract interaction description. Three types of variables are supported: Constants, Range variables that step incrementally through a series of values, and Sets from which a variable is iteratively set to each value.

### 5.5.3 Syntax and Semantics for Abstract Rules

The syntax for describing the interactions with variables specifies where to place the resolved value for each variable and how to specify the iteration of variable values. The semantics describe how the variables values are set and how iteration of multiple values is accomplished.

There are three types of variables in our framework: Constants that are defined once, but used in a number of interaction rules; Range variables that step incrementally through a series of values; and Sets from which a variable is iteratively set to each value.

Constants can be used to specify a value that is used in a number of independent interactions, but may change depending on the model environment. For example, a rate of interaction shared between many factors or group of rule definitions could be defined as a constant. This is similar to using constants as a programming style to make it easier to modify a single variable than to modify all the abstract rules in which the value is used.

Ranges of variables allow iteration through the values and generation of rules with each variable value. The framework defines an explicit global variable, *POS*, that iterates from 1 to length of the DNA, walking along the DNA and generates all the rules for each position. Some interactions require the generation of multiple rules per position of the DNA. For example, the rules for nucleosome linker maintenance (see Figures A.9-A.14) capture the behavior of nucleosome formation that inhibits formation near another nucleosome.

Therefore, a rule assigned to a binding nucleosome needs to look to see if there is another nucleosome nearby. The same rule is being applied to multiple locations relative to the current position along the DNA to make sure there is not another nucleosome within the linker distance. We can define a RANGE variable to iterate through each distance for each position of the DNA.

The linker maintenance for a nucleosome also uses a variable to vary the interaction rate depending on the distance from the other nucleosomes. It is assumed that formation immediately adjacent to another nucleosome is less likely than if the nucleosome is further away. The rule rate can be described using the current distance to mediate the rate of eviction for a forming nucleosome.

#### **5.5.4 CodaChrome - a Proteome Conservation Visualization Tool**

I assisted and guided a fellow graduate student in creating a visualization tool for relationships between bacterial genomes. We developed CodaChrome , a one-versus-all proteome comparison tool that allows the user to visually investigate the relationship between a bacterial proteome of interest and the proteomes encoded by every other bacterial genome recorded in GenBank in a massive interactive heat map (Figure 5.9). This is an open source project and the software is freely available ([www.sourceforge.com/p/codachrome](http://www.sourceforge.com/p/codachrome)).

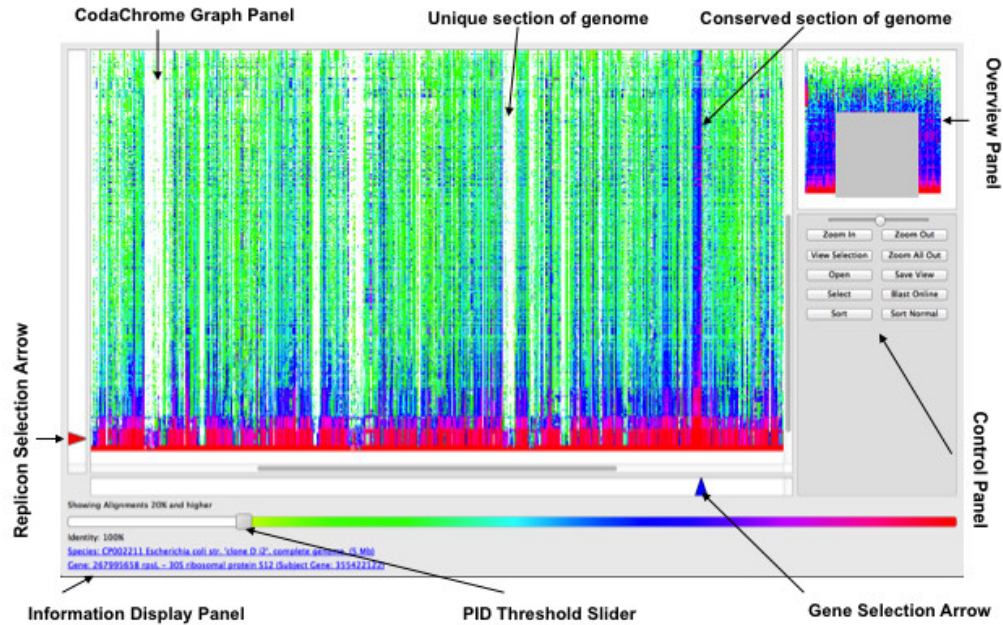
Rokicki, J., Knox, D., Dowell, R. D., and Copley, S. D. (2014). CodaChrome: a tool for the visualization of proteome conservation across all fully sequenced bacterial genomes. BMC Genomics, doi:10.1186/1471-2164-15-65

### **5.6 Conclusion**

The goal of my work has been to facilitate the understanding of the stochastic nature of transcriptional regulation to provide a better understanding of how molecular interactions lead to gene expression, how sequence changes cause transcriptional changes, and how complicated combinations of regulation mechanisms can be designed to achieve desired results.

In my work on integrating research and education, I have explored different visualization methods for all aspects of the modeling process. I have provided a teaching video





**Figure 5.9: The CodaChrome Graphical User Interface.** *Salmonella enterica* 14028S was loaded as the seed organism. The rows were sorted by average proteome identity. The portion of the heat map visualized in the CodaChrome Graph Panel is indicated by the grey box in the Overview Panel. Lighter-than-normal vertical stripes represent large clusters of proteins unique to the seed organism. Dark vertical stripes represent clusters of highly conserved proteins. Buttons embedded in the control panel allow the user to interact with the visualization of the matrix file. A slider at the top of the control panel allows the user to zoom in or out. Replicon arrows and gene selection arrows indicate the alignment selected and described in the information display panel. Finally, the percent identity threshold slider allows users to filter alignments below a specified threshold. For this image, the threshold was set to 20%. The slider also functions as a legend indicating how percent identities are translated into color.

Rokicki et al. BMC Genomics 2014 15:65 doi:10.1186/1471-2164-15-65

for the stochastic and dynamic behaviors of transcriptional regulation. I have developed methods for visualizing the behavior of individual components during a simulation through animation. I investigated the automation of generating model rules directly from graph descriptions, visual displays for defining the model interactions and viewing the fully generated network of interactions, and visualization of proteome conservation over all bacterial genomes.

There are two concepts that the teaching video uniquely addresses: the stochastic nature of the transcription process and the dynamic behavior of the transcriptional machinery. A stochastic process is any process or series of events that occur with a particular probability.

These processes are described by a series of random variables that describe the likelihood of each event. Stochastic processes are not deterministic, but can be analyzed statistically. The transcription machinery is dynamic because it changes its local environment and the probability of localized stochastic events. Until our video, these concepts were not previously shown together in a teaching video.

Most modeling frameworks for transcriptional regulation traditionally produce static charts and graphs as the visualization output. These graphs represent the behavior of cellular populations over time. A modeling framework that explores the details of the regulation mechanisms of single cells must increase its complexity to create realistic models. To make these complex models accessible by the general research community, alternative methods for visualizing both the simulation results and model behavior are required.

Simulation of the generated models provides molecule counts for each of the component states at each time step of the simulation. I have created an application to interpret the state counts and produce a visualization of the configuration of factors bound to the DNA at each time step. The results can be viewed as an ASCII character representation of current state of each nucleotide of the DNA at each time step of the simulation. I have used this representation to produce a cartoon animation of the sequence of interactions along a segment of DNA.

While the individual interactions of the model system can be represented as animations, the overall results of a simulation must be summarized over all the time steps. The simulation visualization application can also provide the time averaged results of each simulation, such as the summary of component occupancy at each position along the DNA, or the summary across multiple simulations to allow comparison to the population averaged experimental data.

The first step in building a model is to define the behavior of the components. I have developed a graphical notation to describe the behavior of each component as interactions with the individual DNA nucleotides. In my modeling framework (Chapter III), the conversion of graphic descriptions into code representing the abstract rules is a manual process.

I have developed a language syntax and semantics to describe the abstract interactions of individual components, allowing automatic generation of code that will produce interaction rules for specific DNA sequences. I have explored using Cytoscape for creating and editing interaction definition in a graphical interface, as well as using it to visualize the full interaction network. Automatic generation of modeling rules from the graphical representation of the interactions would allow any researcher to draw a graphical representation of any abstract behavior and immediately build a model with that behavior.

## CHAPTER VI

### CONCLUSION

The goal of my work has been to facilitate the understanding of the stochastic nature of transcriptional regulation by switching from modeling of the individual mechanisms to a model that encompasses the whole stochastic system. This approach allows for a better understanding of how molecular interactions lead to gene expression, how sequence changes cause transcriptional changes, and how complicated combinations of regulation mechanisms can be designed to achieve desired results. My work has focused on two areas: modeling the system behavior of transcriptional regulation mechanisms and the communication of these behaviors to researchers and students through visualization and education.

#### 6.1 Modeling of Transcriptional Regulation

I have captured the population averaged behavior for some of the nucleosome dynamics during formation by using an extension to current steady state models. Current state-of-the-art modeling methods are capable of capturing the population averaged behavior of component competition and cooperation. These steady state models are successful at capturing the competition and implicit cooperation among static components that bind and unbind DNA sequences. The question was, can they capture the dynamic behavior of the components? Components, such as the nucleosome, are not static since they are comprised of multiple sub-components that bind and unbind stochastically. I extended one of the steady state models, COMPETE (Wasson and Hartemink, 2009), to capture the dynamic states of nucleosome formation. The two state nucleosome model was able to predict nucleosome occupancy better than the single state nucleosome model across large regions of the genome.

The nucleosome dynamics are confined to the interaction of the histones with DNA and do not affect the other components bound to the DNA. There are other components that actively change the configuration of DNA bound components by moving along the DNA and modifying the behavior of adjacent components. Chromatin remodelers can

move nucleosome positions on the DNA, even moving them to less favorable positions than they would bind to based on their inherent affinity. This movement may allow other DNA binding factors access to DNA from which they would otherwise be occluded. Another dynamic component is the transcriptional machinery that binds and progresses along a strand of the DNA, presumably removing nucleosomes and transcription factors impeding its progress along the DNA.

The population averaged behavior of dynamic nucleosome formation can be captured by extending the current steady state models. However, these steady state methods cannot easily be extended to model the dynamics of the transcriptional machinery or chromatin remodelers. The dynamic changes of transition rates are both spatial and temporal to the actual location of the dynamic component. This type of dynamics is difficult to capture using an HMM as the movement of a component along the DNA would require a coupled copy of the HMM to capture the transitions between configurations over both time and space.

To capture the behavior of individual components, the competition between components for interactions with the DNA, and more importantly, the dynamics of regulatory events occurring within individual cells, I have developed a new modeling framework. The models I constructed are biologically realistic representations and capture the inherent stochasticity and dynamics of regulatory interactions.

I created a modeling framework to automatically generate a model for any DNA sequence. The modeling framework describes each individual interaction between components as spatially abstract rules. The abstract rules are applied to any specific DNA sequence to produce a collection of biochemical based rules describing the individual behavior of each model component across that specific DNA sequence. My framework allows the models to not only capture the population averaged steady-state behavior, but also capture the dynamic behavior of individual components and the emergent behavior arising from the components working together in a coordinated system.

The behavior of each component is independently defined as a positionally abstract interaction. These abstract interactions define the generic behavior and a means of calculating the interaction rates depending on each specific sub-sequence of DNA. The abstractions are applied to the specific DNA sequence being modeled to produce a complete set of interaction rules across the entire DNA sequence. The stochastic simulations of the rule-based models will capture the configurations of the DNA at each successive time point and can be used to visualize the dynamic behavior of components.

My work was motivated by the transcriptional regulation of the FLO11 gene in *Saccharomyces cerevisiae*. The regulation at that locus is a combination of simple regulation interactions: competitive binding, nucleosome positioning, remodeling through transcription machinery, and transcriptional interference (Figure 1.1). The abstract interaction rules are able to capture each of the regulatory behaviors of the components as shown in the case studies at different loci within the yeast genome (Section 3.5.1). I have shown case studies for models at specific loci to verify that the generated models can capture the known behaviors at those locations. Each of these models combined the interactions of several components and interactions were simulated over a time period to observe the patterns of DNA configurations.

Unfortunately, the predictive power of my modeling framework is currently limited by the relatively few known kinetics of DNA binding factors. While my original motivation was to model the transcriptional switch of regulation at FLO11, there are many unknown parameters required to model this circuit. Most of these kinetics parameters are difficult to measure at single cell resolution. As technology advances, single cell experimentation will further uncover component cellular kinetics. My flexible modeling framework can easily be extended to incorporate new components or additional details of component behavior as they are discovered or hypothesized.

My models will continue to advance towards biologically realistic and predictive models of transcriptional regulation with the addition of the kinetic rates. However, at this time, I consider my framework as primarily an exploration tool. It is designed to rapidly create

models for different DNA sequences and components. This allows exploration of the effects on the simulation results given changes in component concentrations, component behaviors, or small changes in the DNA sequence (single nucleotide polymorphisms and structural variants).

## 6.2 Visualization and Education

I have created animated visualizations to aid researchers and students in learning and understanding the complex interactions of transcriptional regulation. The complex interactions of biological processes are dynamic and depend on both spatial and temporal events within a cell. Population averaged behavior can be abstracted from the individual cellular behaviors and presented using static graphs and charts. As the experimental and modeling perspectives change from population average to single cell behavior, the focus changes from summary of events to the temporal sequence of events, which requires a more dynamic method of presentation. My work has included the visualization of the behavior of these complex systems as captured by the simulation of the models. The results are visualized as a series of DNA configurations showing the location of DNA bound factors. The series of configurations can be used to infer movement of components along the DNA, which can be used to create animations of the behavior of system components. Many of the dynamic behaviors of transcriptional regulation mechanisms are not currently taught as part of the undergraduate curriculum, therefore I produced an educational video to teach the dynamic and stochastic concepts.

The last aim proposed in our NSF grant was to merge the simulation and visualization into an interactive teaching tool to provide students, educators, and scientists with the ability to experiment, explore, and discover how different regulation mechanisms combine to achieve specific cellular responses. The pipeline described in Figure 5.4 was the first attempt to address this aim. Future work on the grant will focus on the teaching goals and user interface to provide an engaging game-like environment in which to learn how the cells control the stochastic behavior of components to achieve the desired results. The visualization and animations are an integral part of this teaching tool. The animations

highlight patterns of temporal behavior not captured in steady state models. Visualizations can draw attention to the emergent behavior of the complex system, such as the access to occluded DNA by the eviction of factors from an elongating transcriptional machinery.

### 6.3 Future Work

The limiting factor for making the models of transcriptional regulation is the lack of knowledge about the kinetics of all the DNA binding factors. My models can easily incorporate this new knowledge and can illustrate the type of information that is missing for individual components, including on and off rates, conformational changes in sequence affinity, and behavior of chromatin remodelers. As the knowledge of these kinetics continues to increase, my models will continue to advance towards biologically realistic and predictive models of transcriptional regulation.

The steady state models provide a good population averaged calculation for the competition of DNA binding. These models can be extended to capture some of the alternate states of dynamic behaviors of components. It may be possible to add additional biologically realistic states for chromatin remodeling factors by using a function that moves the nucleosome state probabilities along the DNA in the direction of the movement. The same may be possible for modeling the transcriptional machinery by a similar shift of probability along the DNA.

Chromatin remodelers are one of the major components missing in my current modeling framework. These transcription factors are responsible for the movement of nucleosomes along the DNA, which has a huge effect on the behavior of individual cells. Remodelers change the configuration landscape, activating or inhibiting transcription depending on the resulting configuration of factors bound to the DNA. They were not included in the modeling framework because their binding and kinetics are poorly understood. However, it is currently possible to model the different proposed behaviors and allow researchers to visualize the effects that alternate conditions have on the proposed behavior.

I have used a simple mechanism for describing and visualizing the individual interactions of components within the framework. Petri net descriptions of interactions, represented by



simple graphs, are used to generate the code that can be applied to any DNA sequence. The current method used by the modeling framework requires programming skills to manually convert the graph descriptions into code. I have shown preliminary work that describes the syntax and semantics of a scripting language to automate this process. With this language, anyone could create a graphical description of interactions and immediately build models using that interaction, removing the need for programming skills. By automating the conversion of graphic descriptions into computer understandable forms will allow the “Draw a graph, Build a model” paradigm to be used by non-programming researchers to enhance the modeling components.

The prototype for generating animations of the simulation results can be enhanced to provide a better representation of the component behavior. The crude prototype can be extended to provide better graphics and user interaction, thereby making it an invaluable tool in understanding the dynamics of complex systems.

So far I have only created the underlying tools needed to meet the goal of our third aim in the NSF grant. We still need to bring them all together into an interactive teaching tool that provides the ability to experiment, explore, and discover how different regulation mechanisms combine to achieve specific cellular responses. The application needs to support the scenario where a student is given a profile for how a gene’s expression changes over time and then asked to design a regulatory circuit to mimic that behavior. The student would create the regulatory sequence using a drag and drop interface to place regulatory elements (transcription factors, general transcription factors, nucleosome affinity, and RNA polymerase dynamics) into a promoter region of the gene. The student can watch animations of the molecule interaction simulations, examine summary plots of the results, and compare these to the profile desired. The student can experiment with different mechanism until the simulated results match the desired profile. As the students interact with the different mechanisms, they obtain a much deeper understanding of how these mechanisms work and interact.

## 6.4 Final Remark

I envision a time in the not too distant future where we can understand and predict the regulation of transcription for any sequence of DNA. This will allow researchers to predict the behavior of perturbations, such as sequence changes or drug interactions changing transcription factor concentrations, over the whole cellular system. This path towards personalized medicine will see the day when doctors will custom design drugs to recognize individual cells by their genome and environment. Tumor cells have a modified genome changing the cellular behavior that could be identified and individually marked for treatment. My work is a small step along this path. It takes a step towards understanding how to model the transcription behavior across large segments of DNA with all the variation possible in single cells.

## REFERENCES

- Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2007). *Molecular Biology of the Cell*.
- Andrews, A. J. and Luger, K. (2011). Nucleosome structure(s) and stability: variations on a theme. *Annual review of biophysics*, 40:99–117.
- Austin, S., Schwartz, R., and Placeway, P. (1991). The forward-backward search algorithm. [*Proceedings*] *ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*.
- Ay, A. and Arnosti, D. N. (2011). Mathematical modeling of gene expression: a guide for the perplexed biologist. *Critical reviews in biochemistry and molecular biology*, 46(2):137–51.
- Badis, G., Berger, M. F., Philippakis, A. A., Talukder, S., Gehrke, A. R., Jaeger, S. A., Chan, E. T., Metzler, G., Vedenko, A., Chen, X., Kuznetsov, H., Wang, C.-f., Coburn, D., Newburger, D. E., Morris, Q., Hughes, T. R., and Bulyk, M. L. (2009). Diversity and complexity in DNA recognition by transcription factors. *Science (New York, N.Y.)*, 324(5935):1720–3.
- Badis, G., Chan, E. T., van Bakel, H., Pena-Castillo, L., Tillo, D., Tsui, K., Carlson, C. D., Gossett, A. J., Hasinoff, M. J., Warren, C. L., Gebbia, M., Talukder, S., Yang, A., Mnaimneh, S., Terterov, D., Coburn, D., Li Yeo, A., Yeo, Z. X., Clarke, N. D., Lieb, J. D., Ansari, A. Z., Nislow, C., Hughes, T. R., Bakel, H. V., and Yeo, A. L. (2008). A library of yeast transcription factor motifs reveals a widespread function for Rsc3 in targeting nucleosome exclusion at promoters. *Molecular cell*, 32(6):878–87.
- Bai, L., Ondracka, A., and Cross, F. R. (2011). Multiple Sequence-Specific Factors Generate the Nucleosome-Depleted Region on CLN2 Promoter. *Molecular cell*, 42(4):465–76.
- Barnes, C. P., Silk, D., Sheng, X., and Stumpf, M. P. H. (2011). Bayesian design of synthetic biological systems. *Proceedings of the National Academy of Sciences of the United States of America*, 108(37):15190–5.
- Berg, O. G., Winter, R. B., and von Hippel, P. H. (1981). Diffusion-driven mechanisms of protein translocation on nucleic acids. 1. Models and theory. *Biochemistry*, 20(24):6929–48.
- Berger, S. L. (2002). Histone modifications in transcriptional regulation. *Current Opinion in Genetics & Development*, pages 142–148.
- Billington, J., Christensen, S. r., Hee, K. V., Kindler, E., Kummer, O., and Petrucci, L. (2003). The Petri Net Markup Language : Concepts , Technology , and Tools. pages 483–505.
- Black, A. J. and McKane, A. J. (2012). Stochastic formulation of ecological models and their applications. *Trends in ecology & evolution*, 27(6):337–45.
- Böhm, V., Hieb, A. R., Andrews, A. J., Gansen, A., Rocker, A., Tóth, K., Luger, K., and Langowski, J. (2011). Nucleosome accessibility governed by the dimer/tetramer interface. *Nucleic acids research*, 39(8):3093–102.
- Bradley, R. K., Li, X.-Y., Trapnell, C., Davidson, S., Pachter, L., Chu, H. C., Tonkin, L. a., Biggin, M. D., and Eisen, M. B. (2010). Binding site turnover produces pervasive quantitative changes in transcription factor binding between closely related *Drosophila* species. *PLoS biology*, 8(3):e1000343.

- Bryant, G. O. and Ptashne, M. (2003). Independent recruitment in vivo by Gal4 of two complexes required for transcription. *Molecular cell*, 11(5):1301–9.
- Bulyk, M. L., Huang, X., Choo, Y., and Church, G. M. (2001). Exploring the DNA-binding specificities of zinc fingers with DNA microarrays. *Proceedings of the National Academy of Sciences of the United States of America*, 98(13):7158–63.
- Bumgarner, S. L., Neuert, G., Voight, B. F., Symbor-Nagrabska, A., Grisafi, P., van Oudenaarden, A., and Fink, G. R. (2012). Single-cell analysis reveals that noncoding RNAs contribute to clonal heterogeneity by modulating transcription factor recruitment. *Molecular cell*, 45(4):470–82.
- Cantone, I., Marucci, L., Iorio, F., Ricci, M. A., Belcastro, V., Bansal, M., Santini, S., di Bernardo, M., di Bernardo, D., and Cosma, M. P. (2009). A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, 137(1):172–81.
- Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4):210–9.
- Colvin, J., Monine, M. I., Faeder, J. R., Hlavacek, W. S., Von Hoff, D. D., and Posner, R. G. (2009). Simulation of large-scale rule-based models. *Bioinformatics (Oxford, England)*, 25(7):910–7.
- Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z., and Players, F. (2010). Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–60.
- Coulon, A., Chow, C. C., Singer, R. H., and Larson, D. R. (2013). Eukaryotic transcriptional dynamics: from single molecules to cell populations. *Nature reviews. Genetics*, 14(8):572–84.
- Coulon, A., Gandrillon, O., and Beslon, G. (2010). On the spontaneous stochastic dynamics of a single gene: complexity of the molecular interplay at the promoter. *BMC systems biology*, 4:2.
- Darzacq, X., Shav-Tal, Y., de Turrís, V., Brody, Y., Shenoy, S. M., Phair, R. D., and Singer, R. H. (2007). In vivo dynamics of RNA polymerase II transcription. *Nature structural & molecular biology*, 14(9):796–806.
- Dion, M. F., Kaplan, T., Kim, M., Buratowski, S., Friedman, N., and Rando, O. J. (2007). Dynamics of replication-independent histone turnover in budding yeast. *Science (New York, N.Y.)*, 315(5817):1405–8.
- Dresch, J. M., Thompson, M. A., Arnosti, D. N., and Chiu, C. (2013). Two-Layer Mathematical Modeling of Gene Expression: Incorporating DNA-Level Information and System Dynamics. *SIAM Journal on Applied Mathematics*, 73(2):804–826.
- Drew, H. R. and Travers, a. a. (1985). DNA bending and its relation to nucleosome positioning. *Journal of molecular biology*, 186(4):773–90.
- Edwards, J. S. and Palsson, B. O. (2000). The Escherichia coli MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. *Proceedings of the National Academy of Sciences of the United States of America*, 97(10):5528–33.
- Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science (New York, N.Y.)*, 297(5584):1183–6.

- Epstein, J. M. (2008). Why Model? *Journal of Artificial Societies and Social Simulation*, 11(4).
- Erhard, F., Friedel, C. C., and Zimmer, R. (2008). FERN - a Java framework for stochastic simulation and evaluation of reaction networks. *BMC bioinformatics*, 9:356.
- Faeder, J. R. (2011). Toward a comprehensive language for biological systems. *BMC biology*, 9:68.
- Faeder, J. R., Blinov, M. L., Goldstein, B., and Hlavacek, W. S. (2005). Rule-based modeling of biochemical networks. *Complexity*, 10(4):22–41.
- Farnham, P. J. (2009). Insights from genomic profiling of transcription factors. *Nature reviews. Genetics*, 10(9):605–616.
- Field, Y., Kaplan, N., Fondufe-Mittendorf, Y., Moore, I. K., Sharon, E., Lubling, Y., Widom, J., and Segal, E. (2008). Distinct modes of regulation by chromatin encoded through nucleosome positioning signals. *PLoS computational biology*, 4(11):e1000216.
- Förster, J., Famili, I., Fu, P., Palsson, B. O., and Nielsen, J. (2003). Genome-scale reconstruction of the *Saccharomyces cerevisiae* metabolic network. *Genome research*, 13(2):244–53.
- Fox, P. and Hendler, J. (2011). Changing the equation on scientific data visualization. *Science (New York, N.Y.)*, 331(6018):705–8.
- Galburt, E. a., Grill, S. W., and Bustamante, C. (2009). Single molecule transcription elongation. *Methods*, 48(4):323–332.
- Gelfand, B., Mead, J., Bruning, A., Apostolopoulos, N., Tadigotla, V., Nagaraj, V., Sengupta, A. M., and Vershon, A. K. (2011). Regulated antisense transcription controls expression of cell-type-specific genes in yeast. *Molecular and cellular biology*, 31(8):1701–1709.
- Genrich, H., Kuffner, R., and Voss, K. (2001). Executable Petri net models for the analysis of metabolic pathways. *Int J STTT*, 3:394–404.
- Ghaemmaghani, S., Huh, W.-K., Bower, K., Howson, R. W., Belle, A., Dephoure, N., O’Shea, E. K., and Weissman, J. S. (2003). Global analysis of protein expression in yeast. *Nature*, 425(6959):737–41.
- Gibson, M. A. and Bruck, J. (2000). Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889.
- Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434.
- Gordân, R., Murphy, K. F., McCord, R. P., Zhu, C., Vedenko, A., and Bulyk, M. L. (2011). Curated collection of yeast transcription factor DNA binding specificity data reveals novel structural and gene regulatory insights. *Genome biology*, 12(12):R125.
- Gordon, D. B., Nekludova, L., McCallum, S., and Fraenkel, E. (2005). TAMO: a flexible, object-oriented framework for analyzing transcriptional regulation using DNA-sequence motifs. *Bioinformatics (Oxford, England)*, 21(14):3164–5.

- Greive, S. J., Dyer, B. a., Weitzel, S. E., Goodarzi, J. P., Main, L. J., and von Hippel, P. H. (2011). Fitting experimental transcription data with a comprehensive template-dependent modular kinetic model. *Biophysical journal*, 101(5):1166–74.
- Gullerova, M. and Proudfoot, N. J. (2010). Transcriptional interference and gene orientation in yeast: noncoding RNA connections. *Cold Spring Harbor symposia on quantitative biology*, 75:299–311.
- Hager, G. L., McNally, J. G., and Misteli, T. (2009). Transcription dynamics. *Molecular cell*, 35(6):741–53.
- Hahn, S. and Young, E. T. (2011). Transcriptional regulation in *saccharomyces cerevisiae*: Transcription factor regulation and function, mechanisms of initiation, and roles of activators and coactivators. *Genetics*, 189(3):705–736.
- Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J.-B., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, P. A., Takusagawa, K. T., Lander, E. S., Gifford, D. K., Fraenkel, E., and Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104.
- Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., and Guthke, R. (2009). Gene regulatory network inference: data integration in dynamic models-a review. *Bio Systems*, 96(1):86–103.
- Hesselberth, J. R., Chen, X., Zhang, Z., Sabo, P. J., Sandstrom, R., Reynolds, A. P., Thurman, R. E., Neph, S., Kuehn, M. S., Noble, W. S., Fields, S., and Stamatoyannopoulos, J. a. (2009). SUPPLEMENTARY: Global mapping of protein-DNA interactions in vivo by digital genomic footprinting. *Nature methods*, 6(4):283–9.
- Hongay, C. F., Grisafi, P. L., Galitski, T., and Fink, G. R. (2006). Antisense transcription controls cell fate in *Saccharomyces cerevisiae*. *Cell*, 127(4):735–45.
- Huang, D. W., Sherman, B. T., and Lempicki, R. a. (2009). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols*, 4(1):44–57.
- Hucka, M., Keating, S. M., Nov, N. L., Sahle, S., and Wilkinson, D. J. (2008). Systems Biology Markup Language ( SBML ) Level 2 : Structures and Facilities for Model Definitions ere California Institute of Technology , USA.
- Iec, I. S. O., Kindler, E., Petrucci, L., Umr, C., and Trèves, N. (2009). A primer on the Petri Net Markup Language and. (October).
- Jansen, A., van der Zande, E., Meert, W., Fink, G. R., and Verstrepen, K. J. (2012). Distal chromatin structure influences local nucleosome positions and gene expression. *Nucleic acids research*, 40(9):3870–85.
- Kaplan, N., Moore, I. K., Fondufe-Mittendorf, Y., Gossett, A. J., Tillo, D., Field, Y., LeProust, E. M., Hughes, T. R., Lieb, J. D., Widom, J., and Segal, E. (2009). The DNA-encoded nucleosome organization of a eukaryotic genome. *Nature*, 458(7236):362–366.
- Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature reviews. Molecular cell biology*, 9(10):770–80.
- Karplus, M. and McCammon, J. A. (2002). Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–52.

- Kaufmann, B. B. and van Oudenaarden, A. (2007). Stochastic gene expression: from single molecules to the proteome. *Current opinion in genetics & development*, 17(2):107–12.
- Kim, H. and Gelenbe, E. (2012). Stochastic gene expression modeling with Hill function for switch-like gene responses. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 9(4):973–9.
- Kireeva, M. L., Hancock, B., Cremona, G. H., Walter, W., Studitsky, V. M., and Kashlev, M. (2005). Nature of the nucleosomal barrier to RNA polymerase II. *Molecular cell*, 18(1):97–108.
- Kornberg, R. D. and Lorch, Y. (1999). Twenty-Five Years of the Nucleosome , Fundamental Particle of the Eukaryote Chromosome. *Cell*, 98:285–294.
- Kornienko, A. E., Guenzl, P. M., Barlow, D. P., and Pauler, F. M. (2013). Gene regulation by the act of long non-coding RNA transcription. *BMC biology*, 11(1):59.
- Kulaeva, O. I., Hsieh, F.-K., Chang, H.-W., Luse, D. S., and Studitsky, V. M. (2013). Mechanism of transcription through a nucleosome by RNA polymerase II. *Biochimica et biophysica acta*, 1829(1):76–83.
- Lander, A. D. (2010). The edges of understanding. *BMC biology*, 8:40.
- Larson, D. R., Zenklusen, D., Wu, B., Chao, J. a., and Singer, R. H. (2011). Real-time observation of transcription initiation and elongation on an endogenous yeast gene. *Science (New York, N.Y.)*, 332(6028):475–478.
- Lee, W., Tillo, D., Bray, N., Morse, R. H., Davis, R. W., Hughes, T. R., and Nislow, C. (2007). A high-resolution atlas of nucleosome occupancy in yeast. *Nature genetics*, 39(10):1235–44.
- Lei, J. (2011). A Modular, Qualitative Modeling of Regulatory Networks Using Petri Nets. *Modeling in Systems Biology*, 16:25.
- Levsky, J. M., Shenoy, S. M., Pezo, R. C., and Singer, R. H. (2002). Single-cell gene expression profiling. *Science (New York, N.Y.)*, 297(5582):836–40.
- Li, G., Levitus, M., Bustamante, C., and Widom, J. (2005). Rapid spontaneous accessibility of nucleosomal DNA. *Nature structural & molecular biology*, 12(1):46–53.
- Lickwar, C. R., Mueller, F., Hanlon, S. E., McNally, J. G., and Lieb, J. D. (2012a). Genome-wide proteinDNA binding dynamics suggest a molecular clutch for transcription factor function. *Nature*, 484(7393):251–255.
- Lickwar, C. R., Mueller, F., Hanlon, S. E., McNally, J. G., and Lieb, J. D. (2012b). SUPPLEMENTARY: Genome-wide protein-DNA binding dynamics suggest a molecular clutch for transcription factor function. *Nature*, 484(7393):251–5.
- Lieb, J. D., Liu, X., Botstein, D., and Brown, P. O. (2001). Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nature genetics*, 28(4):327–34.
- Lowary, P. T. and Widom, J. (1998). New DNA sequence rules for high affinity binding to histone octamer and sequence-directed nucleosome positioning. *Journal of molecular biology*, 276(1):19–42.

- Lublinter, S., Keren, L., and Segal, E. (2013). Sequence features of yeast and human core promoters that are predictive of maximal promoter activity. *Nucleic acids research*, 41(11):5569–5581.
- Lublinter, S. and Segal, E. (2009). Modeling interactions between adjacent nucleosomes improves genome-wide predictions of nucleosome occupancy. *Bioinformatics (Oxford, England)*, 25(12):i348–55.
- Luger, K., Dechassa, M. L., and Tremethick, D. J. (2012). New insights into nucleosome and chromatin structure: an ordered state or a disordered affair? *Nature reviews. Molecular cell biology*, 13(7):436–47.
- MacIsaac, K. D., Wang, T., Gordon, D. B., Gifford, D. K., Stormo, G. D., and Fraenkel, E. (2006). An improved map of conserved regulatory sites for *Saccharomyces cerevisiae*. *BMC bioinformatics*, 7:113.
- Mack, A. H., Schlingman, D. J., Ilagan, R. P., Regan, L., and Mochrie, S. G. J. (2012). Kinetics and thermodynamics of phenotype: unwinding and rewinding the nucleosome. *Journal of molecular biology*, 423(5):687–701.
- Mäkelä, J., Lloyd-Price, J., Yli-Harja, O., and Ribeiro, A. S. (2011). Stochastic sequence-level model of coupled transcription and translation in prokaryotes. *BMC bioinformatics*, 12(1):121.
- Mirny, L. A. (2010). Nucleosome-mediated cooperativity between transcription factors. *Proceedings of the National Academy of Sciences of the United States of America*, 107(52):22534–22539.
- Morozov, A. V., Fortney, K., Gaykalova, D. a., Studitsky, V. M., Widom, J., and Siggia, E. D. (2009). Using DNA mechanics to predict in vitro nucleosome positions and formation energies. *Nucleic acids research*, 37(14):4707–22.
- Mura, I. and Csikász-Nagy, A. (2008). Stochastic Petri Net extension of a yeast cell cycle model. *Journal of theoretical biology*, 254(4):850–60.
- Ovacik, M. and Androulakis, I. (2008). On the Potential for Integrating Gene Expression and Metabolic Flux Data. *Current Bioinformatics*, 3(3):142–148.
- Pahle, J. (2009). Biochemical simulations: stochastic, approximate stochastic and hybrid approaches. *Briefings in bioinformatics*, 10(1):53–64.
- Palmer, A. C., Egan, J. B., and Shearwin, K. E. (2011). Transcriptional interference by RNA polymerase pausing and dislodgement of transcription factors. *Transcription*, 2(1):9–14.
- Parmar, J. J., Marko, J. F., and Padinhateeri, R. (2014). Nucleosome positioning and kinetics near transcription-start-site barriers are controlled by interplay between active remodeling and DNA sequence. *Nucleic acids research*, 42(1):128–36.
- Pelechano, V., Chávez, S., and Pérez-Ortín, J. E. (2010). A complete set of nascent transcription rates for yeast genes. *PloS one*, 5(11):e15442.
- Prescott, E. M. and Proudfoot, N. J. (2002). Transcriptional collision between convergent genes in budding yeast. *Proceedings of the National Academy of Sciences of the United States of America*, 99(13):8796–8801.



- Qi, Y., Rolfe, A., MacIsaac, K. D., Gerber, G. K., Pokholok, D., Zeitlinger, J., Danford, T., Dowell, R. D., Fraenkel, E., Jaakkola, T. S., Young, R. a., and Gifford, D. K. (2006). High-resolution computational models of genome binding events. *Nature biotechnology*, 24(8):963–70.
- Ramsey, S., Orrell, D., and Bolouri, H. (2005). Dizzy: stochastic simulation of large-scale genetic regulatory networks. *Journal of bioinformatics and computational biology*, 3(2):415–36.
- Rando, O. J. and Winston, F. (2012). Chromatin and transcription in yeast. *Genetics*, 190(2):351–87.
- Raveh-Sadka, T., Levo, M., Shabi, U., Shany, B., Keren, L., Lotan-Pompan, M., Zeevi, D., Sharon, E., Weinberger, A., and Segal, E. (2012). Manipulating nucleosome disfavoring sequences allows fine-tune regulation of gene expression in yeast. *Nature genetics*, (May).
- Ribeiro, A., Zhu, R., and Kauffman, S. a. (2006). A general modeling strategy for gene regulatory networks with stochastic dynamics. *Journal of computational biology : a journal of computational molecular cell biology*, 13(9):1630–9.
- Ribeiro, A. S. (2010). Stochastic and delayed stochastic models of gene expression and regulation. *Mathematical biosciences*, 223(1):1–11.
- Ribeiro, A. S., Smolander, O.-P., Rajala, T., Häkkinen, A., and Yli-Harja, O. (2009). Delayed stochastic model of transcription at the single nucleotide level. *Journal of computational biology*, 16(4):539–53.
- Rinaldi, A. (2012). More than meets the eye. Modern experimental techniques require increasingly sophisticated approaches to data visualization. *EMBO reports*, 13(10):895–9.
- Roussel, M. R. and Zhu, R. (2006). Stochastic kinetics description of a simple transcription model. *Bulletin of mathematical biology*, 68(7):1681–713.
- Ruths, D., Muller, M., Tseng, J.-T., Nakhleh, L., and Ram, P. T. (2008). The signaling petri net-based simulator: a non-parametric strategy for characterizing the dynamics of cell-specific signaling networks. *PLoS computational biology*, 4(2):e1000005.
- Sanchez, A., Choubey, S., and Kondev, J. (2013). Stochastic models of transcription: From single molecules to single cells. *Methods*, 62(1):13–25.
- Sanchez, A., Garcia, H. G., Jones, D., Phillips, R., and Kondev, J. (2011). Effect of promoter architecture on the cell-to-cell variability in gene expression. *PLoS computational biology*, 7(3):e1001100.
- Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, 18(20):6097–100.
- Schwabe, A., Dobrzyski, M., Rybakova, K., Verschure, P., and Bruggeman, F. J. (2011). Origins of stochastic intracellular processes and consequences for cell-to-cell variability and cellular survival strategies. *Methods in enzymology*, 500:597–625.
- Schwabish, M. A. and Struhl, K. (2004). Evidence for eviction and rapid deposition of histones upon transcriptional elongation by RNA polymerase II. *Molecular and cellular biology*, 24(23):10111–7.

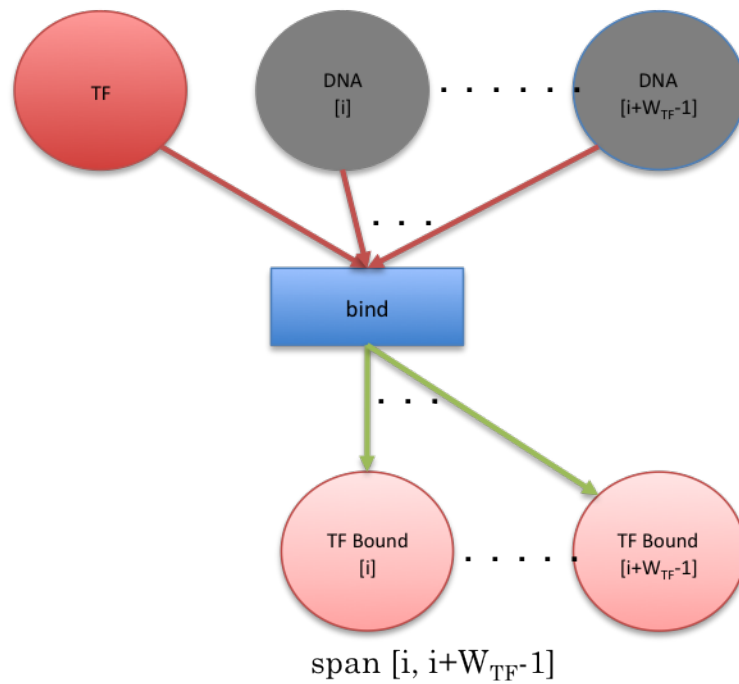
- Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thå ström, A., Field, Y., Moore, I. K., Wang, J.-P. Z., and Widom, J. (2006). A genomic code for nucleosome positioning. *Nature*, 442(7104):772–8.
- Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U., and Gaul, U. (2008). Predicting expression patterns from regulatory sequence in *Drosophila* segmentation. *Nature*, 451(7178):535–40.
- Segal, E. and Widom, J. (2009a). From DNA sequence to transcriptional behaviour: a quantitative approach. *Nature reviews. Genetics*, 10(7):443–56.
- Segal, E. and Widom, J. (2009b). Poly(dA:dT) tracts: major determinants of nucleosome organization. *Current opinion in structural biology*, 19(1):65–71.
- Sneddon, M. W., Faeder, J. R., and Emonet, T. (2011). Efficient modeling, simulation and coarse-graining of biological complexity with Nfsim. *Nature methods*, 8(2):177–183.
- Sneppen, K., Dodd, I. B., Shearwin, K. E., Palmer, A. C., Schubert, R. A., Callen, B. P., and Egan, J. B. (2005). A mathematical model for transcriptional interference by RNA polymerase traffic in *Escherichia coli*. *Journal of molecular biology*, 346(2):399–409.
- Stormo, G. D. (2000). DNA binding sites: representation and discovery. 16(1):16–23.
- Stormo, G. D. (2013). Modeling the specificity of protein-DNA interactions. *Cold Spring Harbor symposia on quantitative biology*, 1(2):115–130.
- Struhl, K. and Segal, E. (2013). Determinants of nucleosome positioning. *Nature structural & molecular biology*, 20(3):267–73.
- Takahashi, K., Kaizu, K., Hu, B., and Tomita, M. (2004). A multi-algorithm, multi-timescale method for cell simulation. *Bioinformatics (Oxford, England)*, 20(4):538–46.
- Taniguchi, Y., Choi, P. J., Li, G.-W., Chen, H., Babu, M., Hearn, J., Emili, A., and Xie, X. S. (2010). Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells. *Science (New York, N.Y.)*, 329(5991):533–538.
- Teixeira, M. C., Monteiro, P., Jain, P., Tenreiro, S., Fernandes, A. R., Mira, N. P., Alenquer, M., Freitas, A. T., Oliveira, A. L., and Sá-Correia, I. (2006). The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*. *Nucleic acids research*, 34:D446–D451.
- Tenazinha, N. and Vinga, S. (2011). A survey on methods for modeling and analyzing integrated biological networks. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 8(4):943–58.
- Tillo, D. and Hughes, T. R. (2009). G+C content dominates intrinsic nucleosome occupancy. *BMC bioinformatics*, 10:442.
- To, T.-L. and Maheshri, N. (2010). Noise can induce bimodality in positive transcriptional feedback loops without bistability. *Science (New York, N.Y.)*, 327(5969):1142–5.
- Tolić-Nørrelykke, S. F., Engh, A. M., Landick, R., and Gelles, J. (2004). Diversity in the rates of transcript elongation by single RNA polymerase molecules. *The Journal of biological chemistry*, 279(5):3292–9.

- Venters, B. J., Wachi, S., Mavrich, T. N., Andersen, B. E., Jena, P., Sinnamon, A. J., Jain, P., Roller, N. S., Jiang, C., Hemeryck-Walsh, C., and Pugh, B. F. (2011). A comprehensive genomic binding map of gene and chromatin regulatory proteins in *Saccharomyces*. *Molecular cell*, 41(4):480–92.
- Wasson, T. and Hartemink, A. J. (2009). An ensemble model of competitive multi-factor binding of the genome. *Genome research*, 19(11):2101–12.
- Wieman, C. E., Adams, W. K., and Perkins, K. K. (2008). PHYSICS. PhET: simulations that enhance learning. *Science (New York, N.Y.)*, 322(5902):682–3.
- Workman, J. L. (2006). Nucleosome displacement in transcription. *Genes & development*, 20(15):2009–17.
- Zeevi, D., Sharon, E., Lotan-Pompan, M., Lubling, Y., Shipony, Z., Raveh-Sadka, T., Keren, L., Levo, M., Weinberger, A., and Segal, E. (2011). Compensation for differences in gene copy number among yeast ribosomal proteins is encoded within their promoters. *Genome research*, 21(12):2114–28.
- Zhu, R., Ribeiro, A. S., Salahub, D., and Kauffman, S. a. (2007). Studying genetic regulatory networks at the molecular level: delayed reaction stochastic models. *Journal of theoretical biology*, 246(4):725–45.

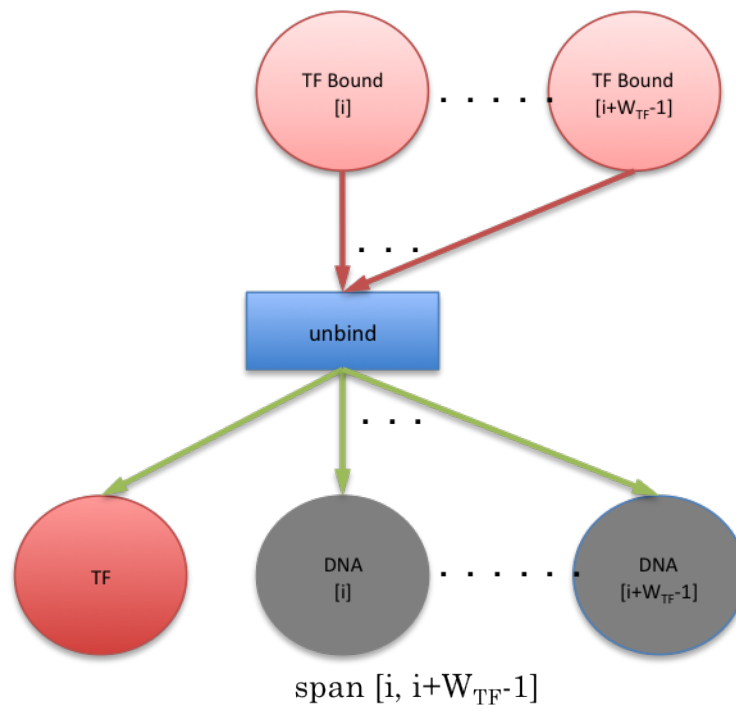
## APPENDIX A

### PETRI NET GRAPHS

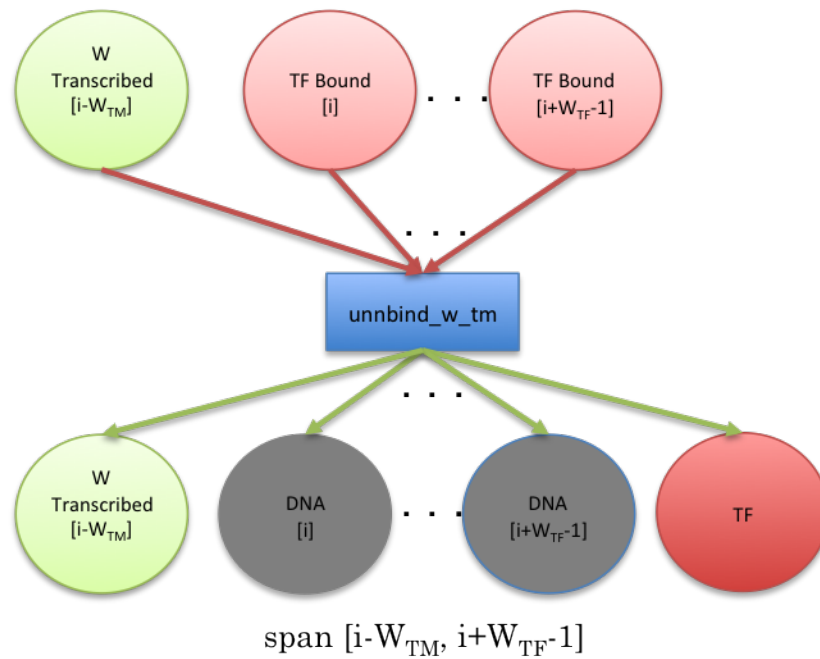
The behavior of the components can be captured in a graphical format known as Petri nets. These descriptions are event based and are easily converted into the biochemical rules needed by the simulation engine. Below are abstract descriptions of interactions, in graphical form, between components of my framework. The Petri Net graphical descriptions represent the component states as colored ovals and actions as blue rectangles. Each component is shown in different colors: unbound DNA nucleotides in gray, transcription factors in red, histones and nucleosomes in blue, and transcriptional machinery in green. DNA nucleotides bound to other components are shown in a pale color of the bound component. Each interaction consumes and creates components. The preconditions for an action are listed as molecules in a given state with arrows into the action. Arrows from the action to the new states of the molecules indicate the post conditions. Multiple consecutive positions bound by a single molecule are represented with a shorthand notation showing only the first and last position. Many components bind to a fixed length of nucleotides which is typically specific to that component, represented by  $W_{component}$  (e.g.  $W_{TF}$  for transcription factors,  $W_{TM}$  for the transcriptional machinery, and  $W_N$  for nucleosomes). Often a component is only required to exist for the rule to be activated and is not consumed. This could be represented by a bi-directional arrow, but here I list the component in both the preconditions above the action and the post conditions below. In each case, I also provide the span, or total number of nucleotides influenced by the application of this rule.



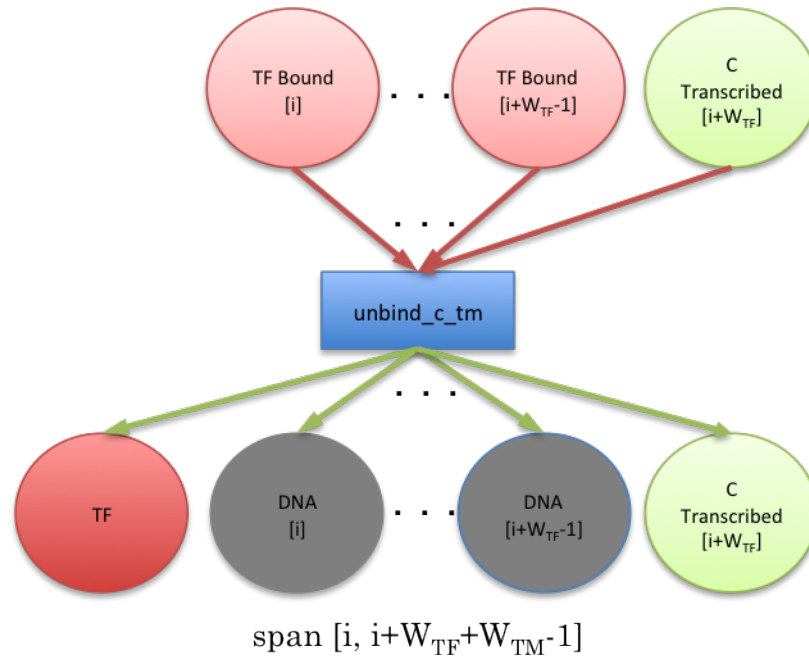
**Figure A.1: Transcription Factor Bind.** Each TF binds to consecutive nucleotides for a length matching the area occluded by the TF. Consumes a molecule of the TF and unbound DNA for each occluded position. Produces molecule of TF bound DNA for each position.



**Figure A.2: Transcription Factor Unbind.** Consumes molecules of TF bound DNA for each position. Produces a molecule of the TF and unbound DNA for each position

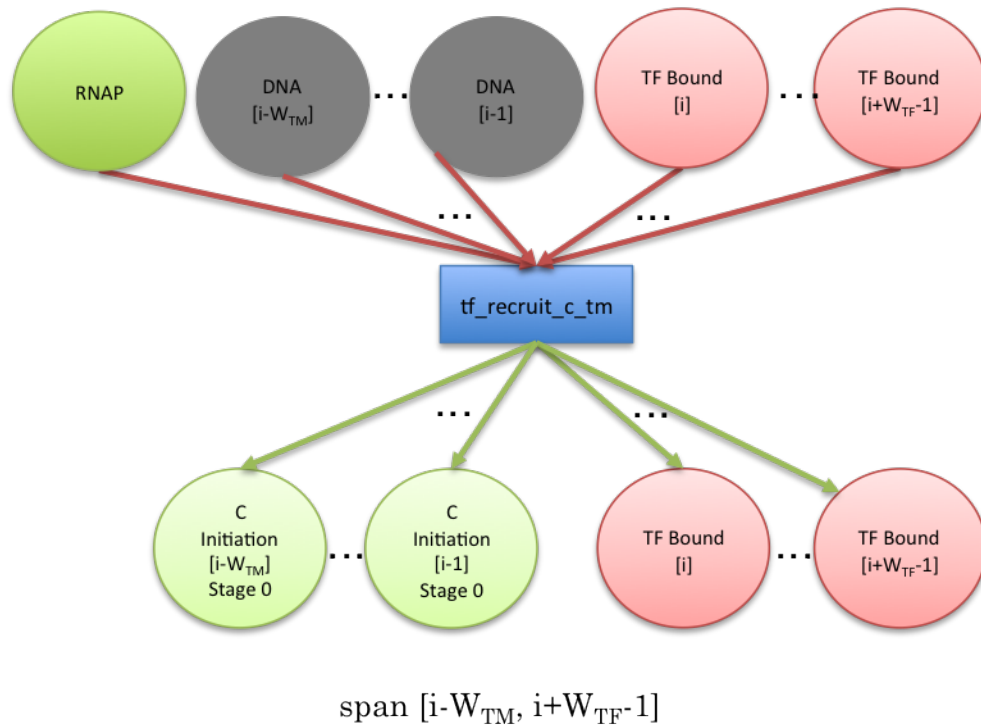


**Figure A.3: Transcription Factor Unbind by Watson strand Transcriptional Machinery.** When the TM is adjacent to bound TF, the TM presumably actively evicts the TF. I need to increase the probability of the TF unbinding by adding an additional interaction rule to complement the normal unbinding rate. Here the TM bound remains bound while the DNA is vacated.

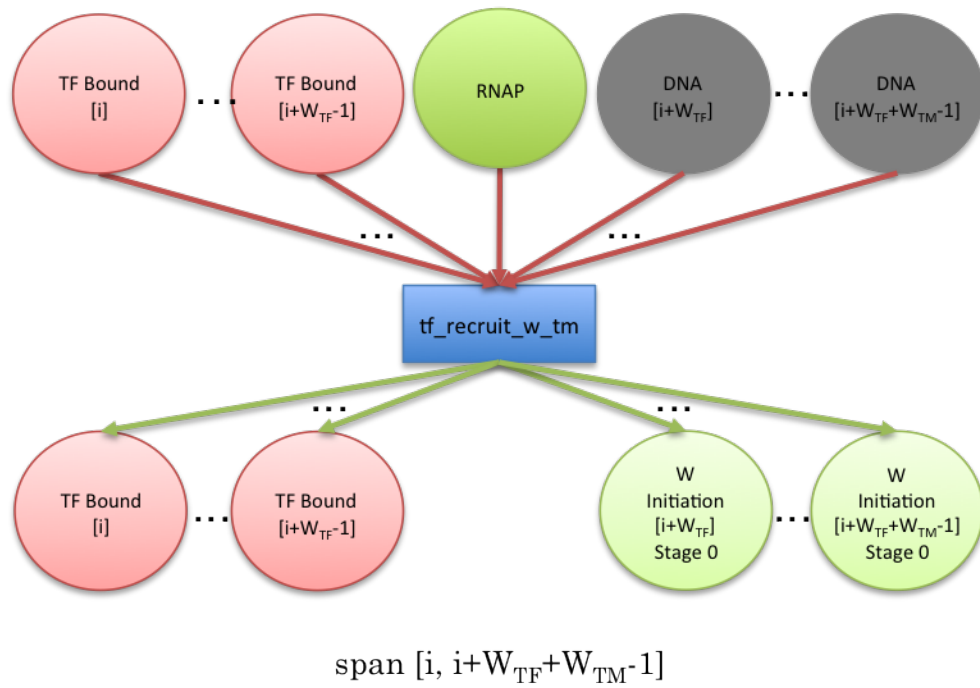


**Figure A.4: Transcription Factor Unbind by Crick strand Transcriptional Machinery.** When the TM is adjacent to bound TF, the TM presumably actively evicts the TF. I need to increase the probability of the TF unbinding by adding an additional interaction rule to complement the normal unbinding rate. Here the TM remains bound while the DNA is vacated. Because the TM is unidirectional in its movement, I need to have a rule for each edge of the TF being adjacent to the appropriate direction of TM. This abstraction represents the TM moving along the Crick strand (decreasing position) and adjacent to the right edge.

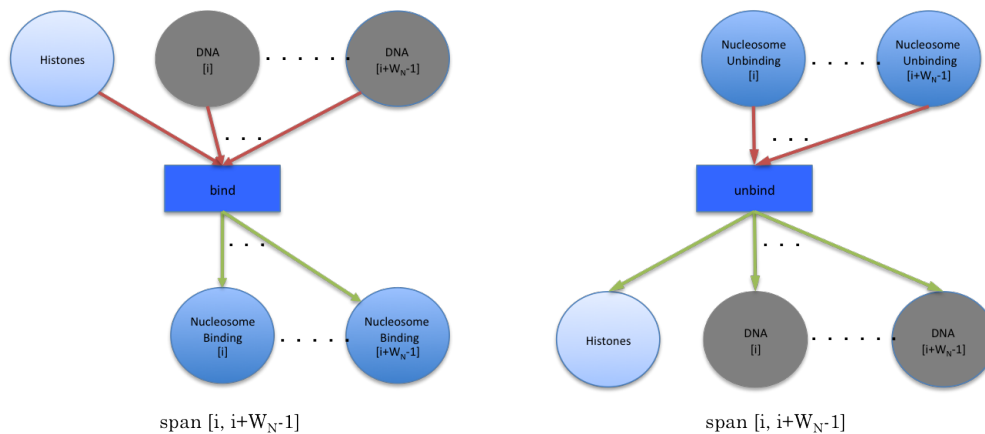




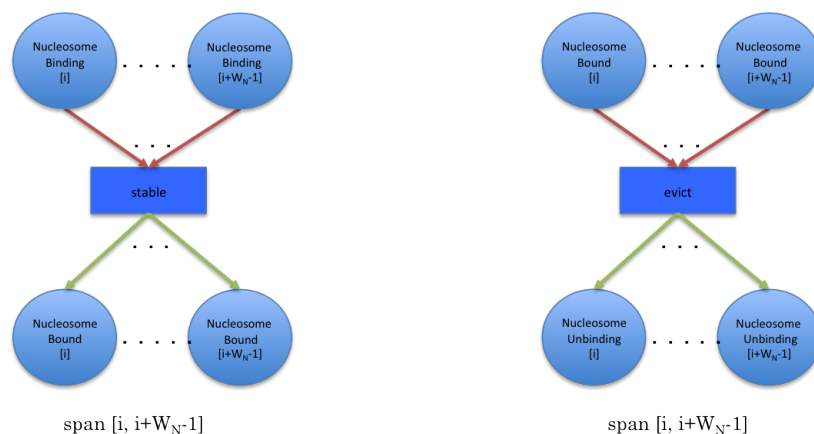
**Figure A.5: Recruit Crick strand Transcriptional Machinery Upstream.** Many TFs are known to recruit the TM to locations near the bound TF. This abstraction represents the recruitment of TM at the immediately adjacent position. my implementation allows for the user to define the offset from the TF position for the TM binding.



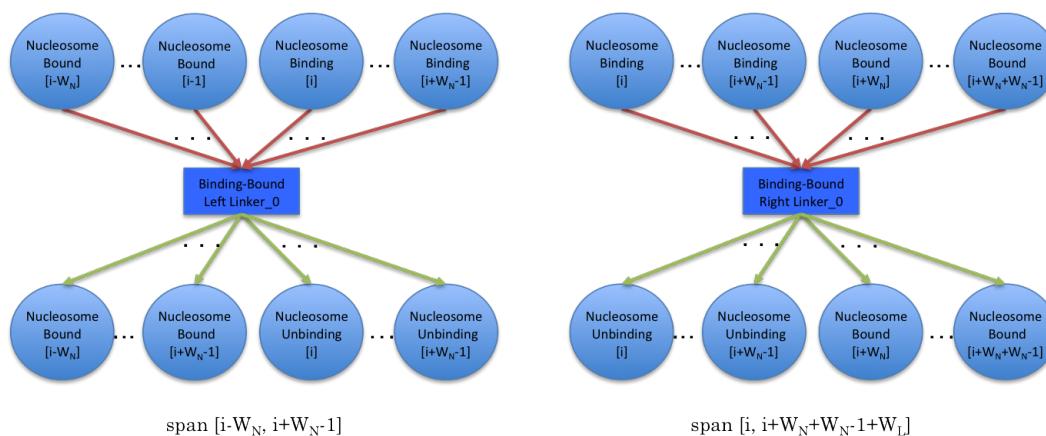
**Figure A.6: Recruit Watson strand Transcriptional Machinery Downstream.** All the TM abstractions must be specified for both strands. This is the complementary abstraction to Figure A.5.



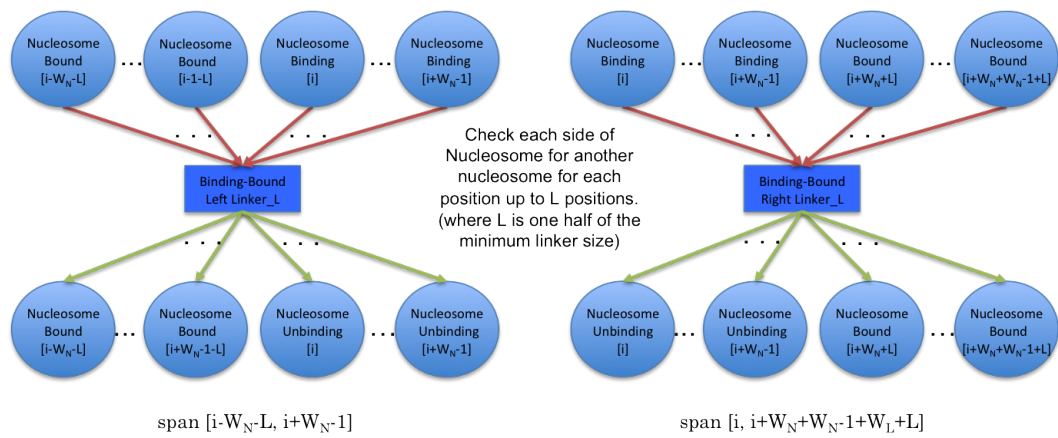
**Figure A.7: Nucleosome Binding and Unbinding.** Nucleosome formation is similar to the TF binding. A molecule of the histones binds to region of unbound DNA to produce a region of DNA forming a nucleosome. The length of the region is constant at 147 nucleotides. The nucleosome formation is a multistate transition: Unbound Binding Nucleosome Unbinding Unbound. This figure represents the first and last of these states. See Figure A.8 for the other two transitions.



**Figure A.8: Nucleosome Stabilization and Eviction.** Nucleosome formation is similar to the TF binding. A molecule of the histones binds to region of unbound DNA to produce a region of DNA forming a nucleosome. The length of the region is constant at 147 nucleotides. The nucleosome formation is a multistate transition: Unbound Binding Nucleosome Unbinding Unbound. This figure represents the middle two state transitions. See Figure A.7 for the other two transitions.



**Figure A.9: Nucleosome Linker Maintenance: Bound - Binding.** Nucleosomes have an inherent spacing that is maintained between adjacent nucleosomes. This is due to physical and thermodynamic limitations. To ensure the linker spacing is maintained, I describe rule abstractions to increase the probability of nucleosome formation being aborted when another nucleosome is at an adjacent position. My implementation reduces the rate of the unbinding as the distance between the nucleosomes increases.



**Figure A.10: Nucleosome Linker Maintenance: Bound - Binding with Linker.**

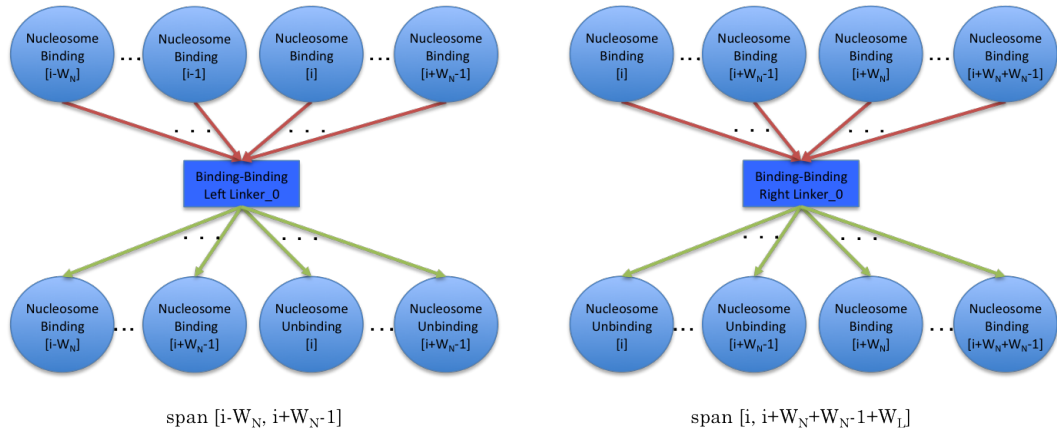


Figure A.11: Nucleosome Linker Maintenance: Binding - Binding.

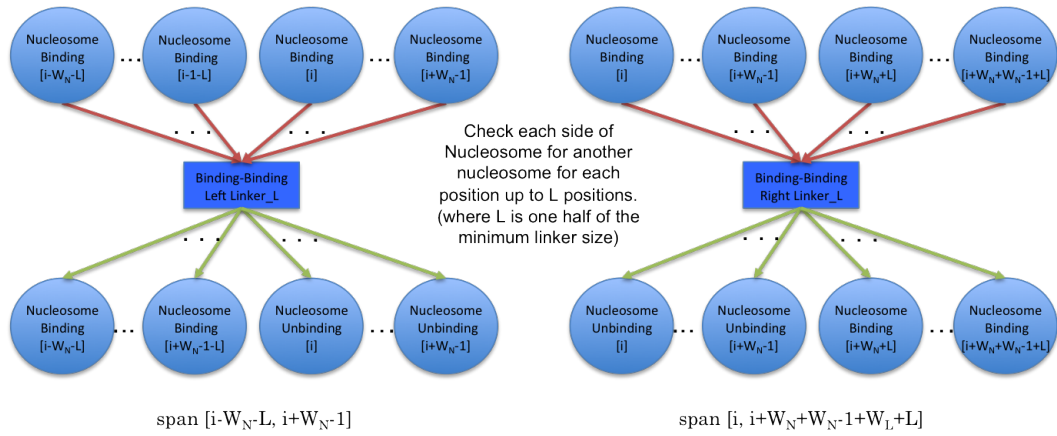


Figure A.12: Nucleosome Linker Maintenance: Binding - Binding with Linker.

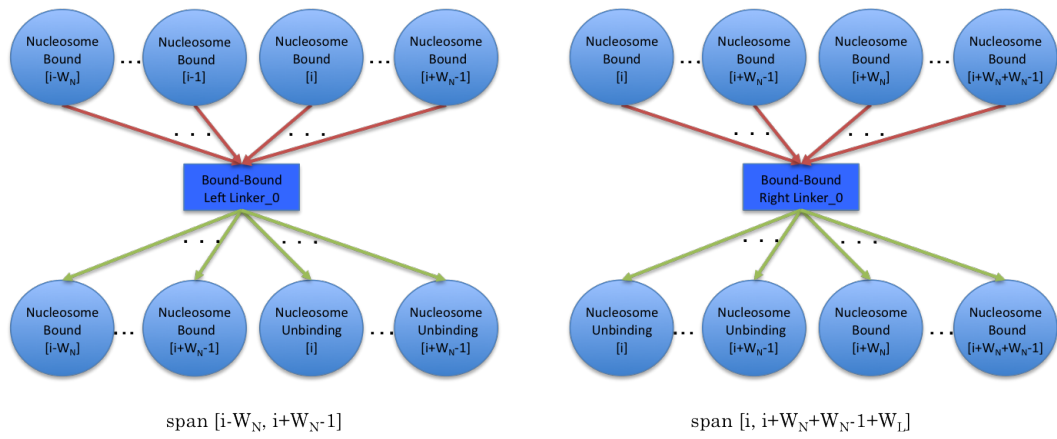


Figure A.13: Nucleosome Linker Maintenance: Bound - Bound.

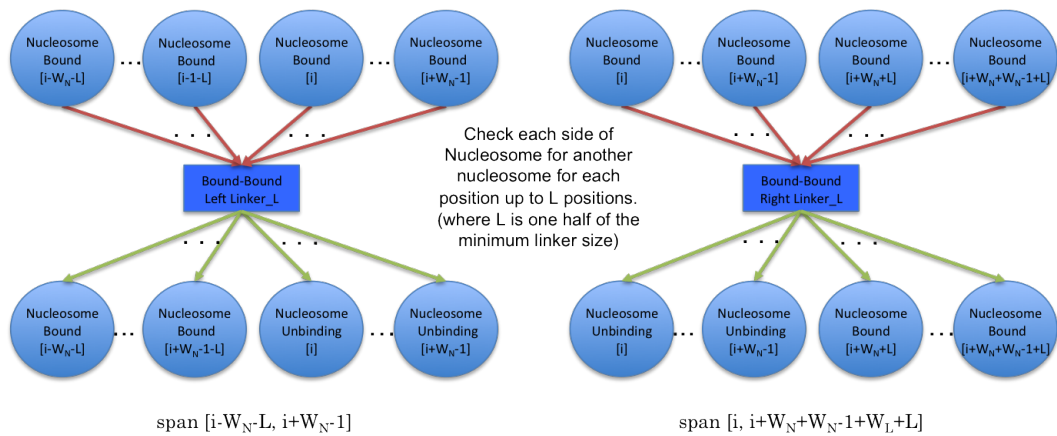
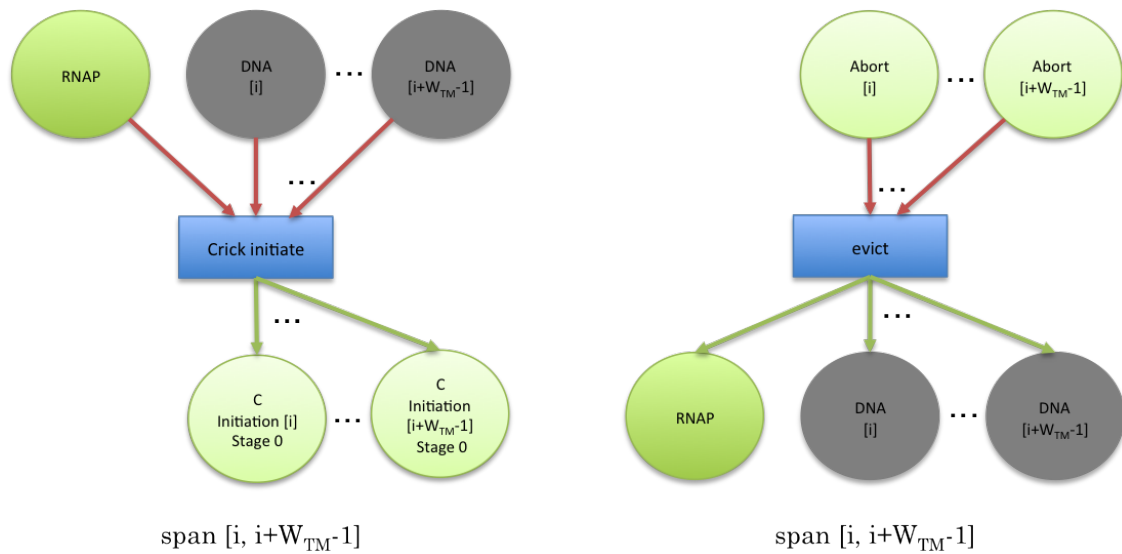
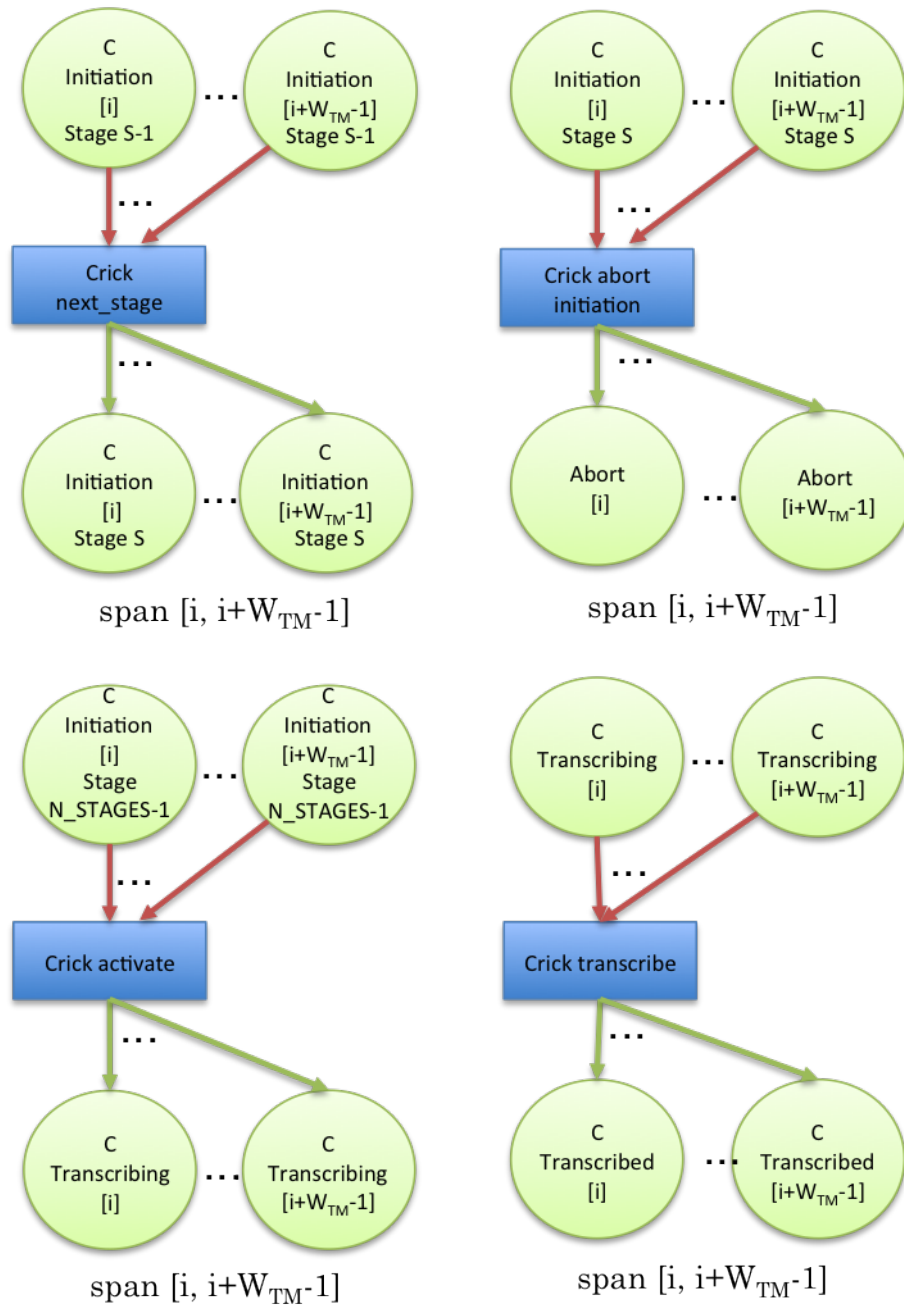


Figure A.14: Nucleosome Linker Maintenance: Bound - Bound with Linker.

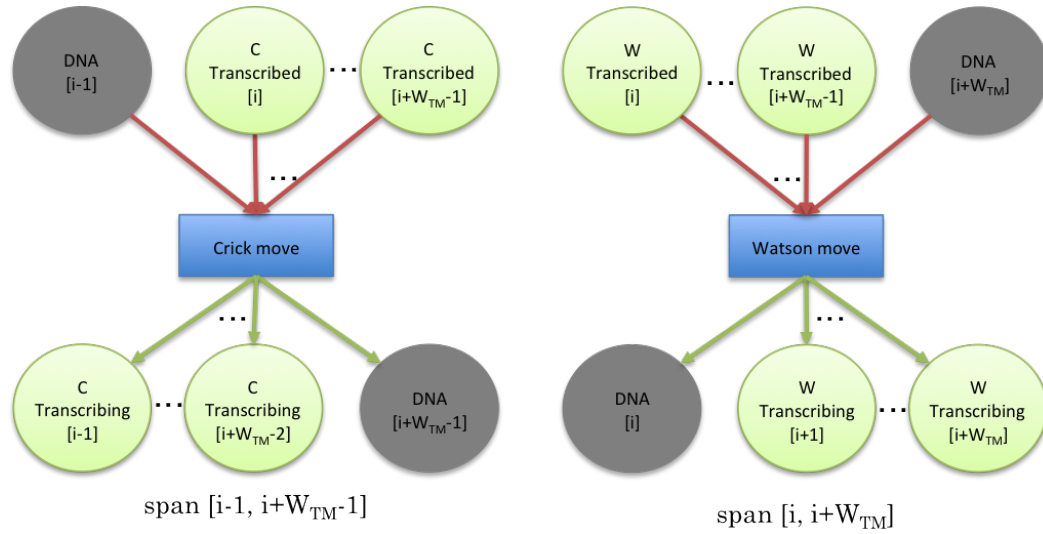


**Figure A.15: Transcriptional Machinery Initiation and Eviction.** The transcriptional machinery is similar to other DNA binding factors. The binding transitions a molecule of the TM and a region of unbound DNA to a TM bound state. When the TM is vacating the DNA, the inverse transition is applied. Initiation is a stochastic signal that can be represented in my framework by using recruiting TF (such as TBP). The TM is capable of binding with any available DNA, however, the kinetics of this behavior is not well understood and my implementation requires explicit initiation positions. (There is a duplicate graph for the initiation of TM on the Watson strand)

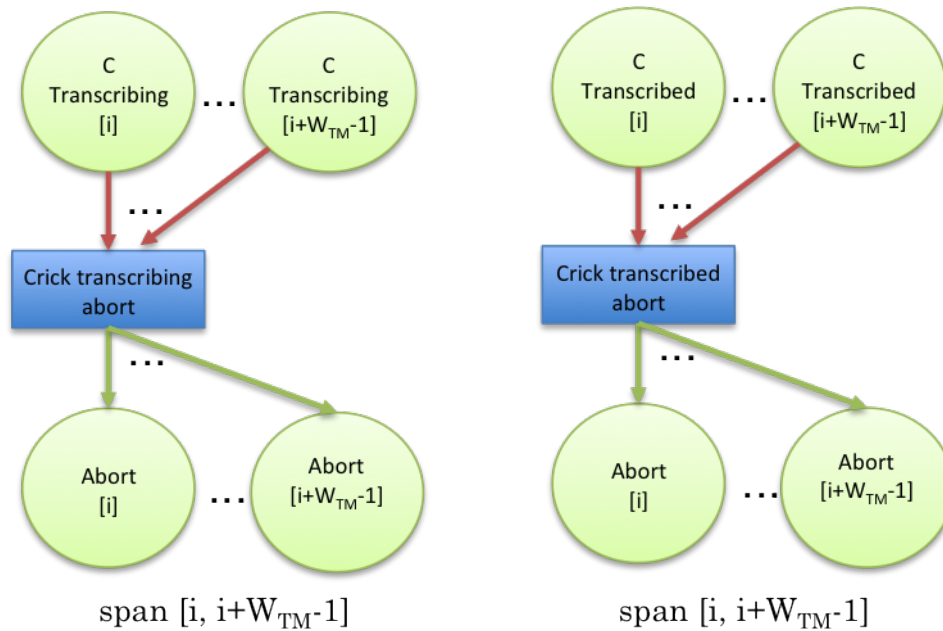




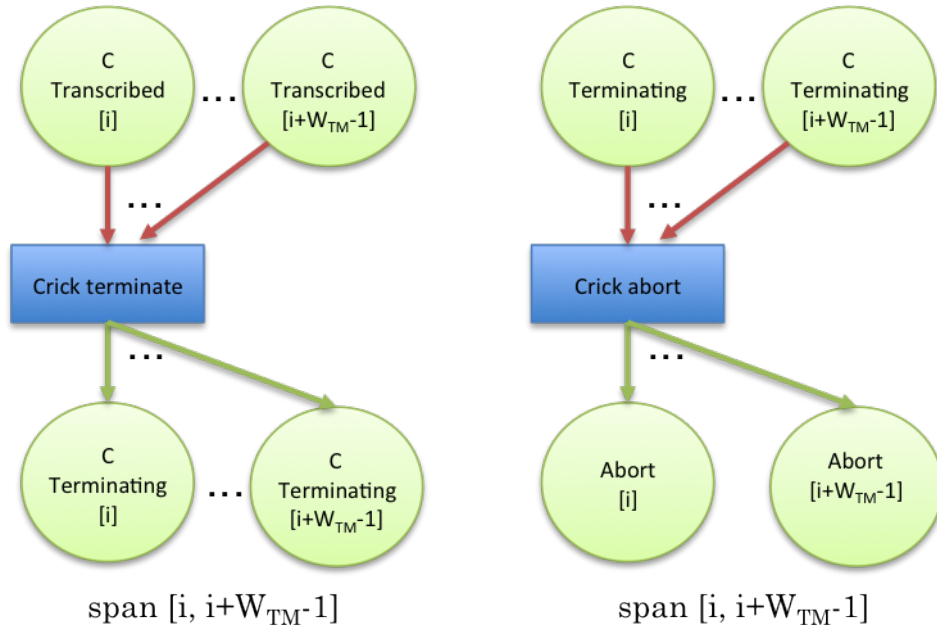
**Figure A.16: Transcriptional Machinery Initiation stages and transition to Elongation.** Each TM moves through the TM states: Unbound Initiating (multiple consecutive stages to represent the time required to build and activate a TM) Transcribing Transcribed Transcribing at next position Transcribed Terminating Abort (vacating). At each of the states it is possible for spontaneous aborting and eviction of the TM. (There are duplicate graphs for the TM on the Watson strand)



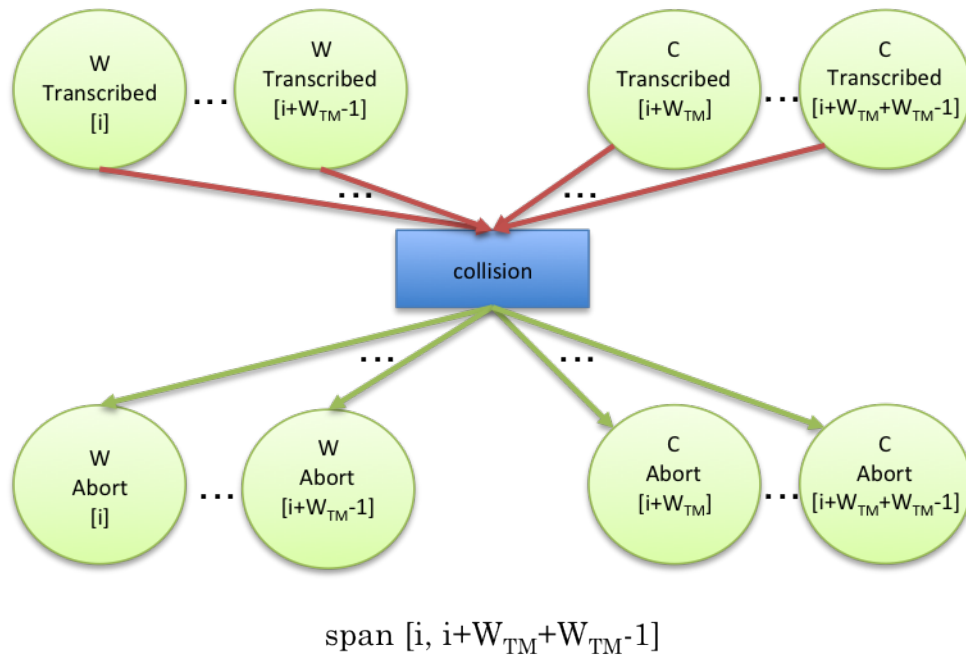
**Figure A.17: TM movement one position.** This abstraction represents the rate of transcription. When the nucleotides are grouped together, this may become a significant period of time between transitions. Moving (advancing) Transcriptional Machinery. Once the DNA position has been transcribed, the TM can move along the DNA. This abstraction represents the movement along the Watson strand (increasing position). When the position is transcribed and the adjacent DNA position is available, the TM can move to the next position and being transcribing. The previously occupied position is now available for other interactions.



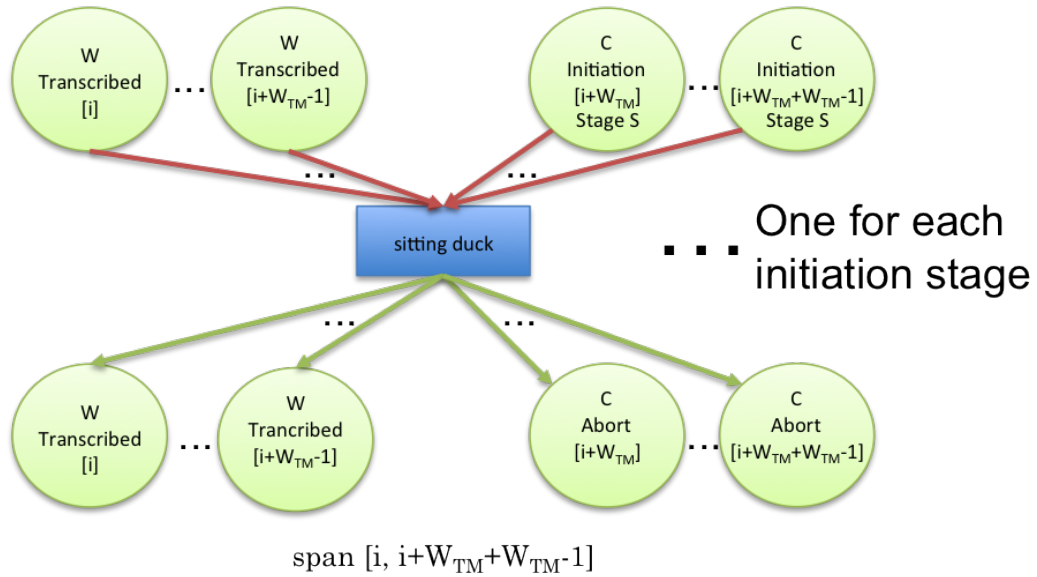
**Figure A.18: Transcriptional Machinery aborting from transcribing or transcribed.** At each of the states it is possible for spontaneous aborting and eviction of the TM. (There are duplicate graphs for the TM on the Watson strand)



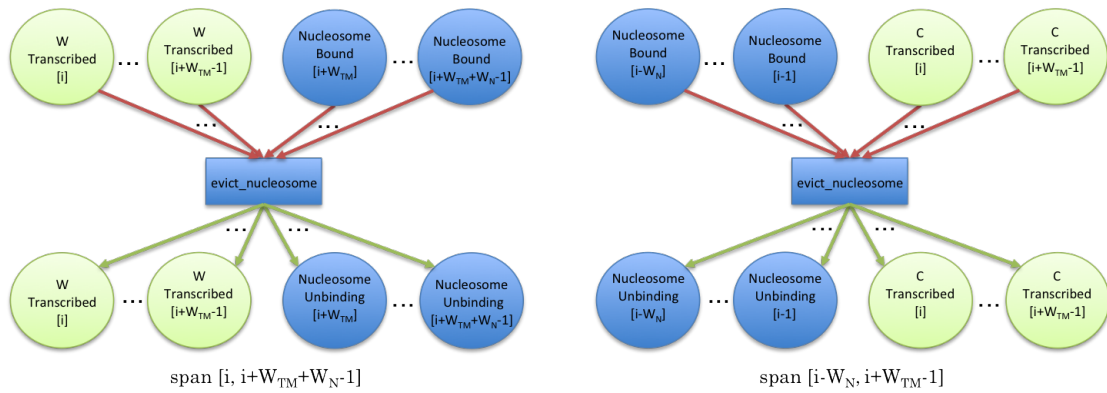
**Figure A.19: Terminating and eviction of Transcriptional Machinery.** The termination of the TM can be signaled by a sequence of DNA. This is a stochastic signal that can be represented in my framework. However, the kinetics of this behavior is not well understood and my implementation requires explicit termination positions. (There are duplicate graphs for the TM on the Watson strand)



**Figure A.20: Transcriptional Interference Collision of two elongating Transcriptional Machinery.** When two TM transcribing in opposite directions collide, my implementation aborts both TM.



**Figure A.21: Transcriptional Interference Collision of elongating Transcriptional Machinery and initiating Transcriptional Machinery.** Proudfoot and colleagues (Sneppen et al., 2005) defined the collision of an elongating TM with an initiating TM as sitting duck. This abstraction encapsulates the behavior of the elongating TM causing the eviction of an initiating TM. Because I model the initiation as multiple stages, I must define the abstraction for each of the initiation stages.



**Figure A.22: Elongating Transcriptional Machinery evicting a Nucleosome.** Similarly to the TM eviction of a TF, the TM also actively evicts a nucleosome.