Variance in Initiation Factors Does Not Strongly Affect the Replication Profile of Budding Yeast DNA

by

Mike Chomitz

B.Sc., Trent University, 2012

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

in the
Department of Physics
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2015

Approval

Name:	Mike Chomitz
Degree:	Master of Science (Physics)
Title:	*Variance in Initiation Factors Does Not Strongly Affect the Replication Profile of Budding Yeast DNA*
Examining Committee:	**Dr. David Sivak** (chair) Assistant Professor

Dr. John Bechhoefer
Senior Supervisor
Professor _____

Dr. Malcolm Kennett
Supervisor
Associate Professor _____

Dr. Eldon Emberly
Internal Examiner
Associate Professor _____

Date Defended: August 19, 2015

Abstract

DNA replication starts at many sites (origins) throughout eukaryotic DNA. To fully understand the replication program in higher organisms, one needs to understand the behaviour of these origins. In *Saccharomyces cerevisiae* (budding yeast), the spatial organization of the origins is simple: The origins are confined to known specific sites on the genome. The temporal behaviour of origins in budding yeast is more complex: They fire stochastically with a broad distribution of firing times.

Several key proteins take part in the DNA replication process. The MCM2-7 hexamer in particular forms a helicase that unwinds DNA locally, allowing access for other proteins to replicate the separated DNA. Past analysis of the budding yeast replication program suggested the Multiple Initiator Model (MIM), which hypothesizes that the number of these MCMs loaded at an origin predicts the firing time of that origin. Part of the MIM formalism assumes that the number of loaded MCMs is large; in this case the relative fluctuations between cells will be small and are ignored. However, a recent experiment measuring the number of loaded MCMs has revealed that the number is low, and thus, cell-to-cell fluctuations may be larger than expected. The purpose of this thesis is to investigate the impact of large relative fluctuations in the number of MCMs on the MIM. To measure this effect, we built the "MIM simulator," a modular program that simulates the replication process. Although a naive argument suggests that the MIM should fail when the number of MCMs is low, the impact of these fluctuations is mitigated by the contributions from neighbouring origins. We conclude that inferences made with the MIM remain accurate in the case that the number of MCMs is lower than first assumed.

Keywords: DNA replication kinetics; multiple initiator model; phantom nuclei; Poisson process

Dedication

In loving memory of Peter Chomitz

Acknowledgements

Looking back on the past two years, I can think of many people that deserve my gratitude. Unfortunately, I don't think I can do them all justice. Thus, I extend first my thanks to those who I will neglect to mention by name in the following pages. If you think you helped me in some way, however minute, thank you.

A large share of my gratitude must go to Dr. John Bechhoefer, my senior supervisor during my research. John's patience and care in teaching is matched only by the astounding breadth and depth of his knowledge (a potent combination in a mentor). I would like to thank John for always striving to find a constructive and considerate way of telling me when my decisions were silly. Additionally, I would like to thank him for pushing me to work harder and faster, and to actively learn at conferences and seminars. In the last several weeks in particular, John maintained positive and constructive criticism to my rushed and somewhat panicked thesis drafts; I'm sure without his input this thesis wouldn't be finished nearly as well, or as quickly. I'm certain that John's strong influence on my approach to learning will positively affect my success in the future.

Next, I need to thank the two gentlemen who provided motivation, insight, and feedback to me from the beginning of my research to the end. Dr. Scott Yang, whose PhD work provided the backbone of my work helped me learn IGOR and showed me how to use and interpret the code he wrote to implement the MIM. Additionally, as my research progressed he provided key insight that helped develop my early work into what eventually became my thesis. It was Dr. Nick Rhind who provided the unpublished experimental data that motivated this research in the first place. I would like to express how grateful I am, not only for the early access to his work, but also for helping me learn the physical and biological processes I needed to know for my work, providing me with unpublished figures for use in my thesis, and for providing valuable feedback on my thesis drafts. Without the help and kindness of Drs. Yang and Rhind, this thesis would be radically different.

I would also like to extend my thanks to the wonderful people with whom I shared the lab. These people made it a great environment in which to work. I knew that if I ran into a problem I could talk with them about it and receive valuable insight. I hope that my future careers come with colleagues half as great as the people I got to work with at

SFU. Therefore, my thanks goes out to Momčilo, Matse, Paul, Leith, Leo, Jan, Laura, Lisa, Mathieu and Swapnil.

Of the rest of the Department of Physics at SFU and the SFU community, there are many who deserve my thanks, but a few should be highlighted. Stephen Flach and Joan Cookson both worked tirelessly to make my job easier (and did such a good job of it that I barely noticed) and for that I am very grateful. My good friends in the student body also deserve my thanks for both scientific and moral support, including (but not limited to) Rohan, James, Brendin, Phil, Sean, Lavisha, Natalie, Jeff, and Colin.

A very special "thank you" goes to Kelly Prodaniuk. In the last year she has become a tremendous part of my life and without her help I could not have accomplished nearly as much. In the last few weeks of writing especially, her support, from making sure I was eating well, to helping me overcome an injured arm so I could keep writing, has been invaluable. Her continued love and support is something I am going to enjoy, benefit from and reciprocate for a long, long time.

Finally, I have to thank my family. I have had the privilege of growing up in a very loving family. It has taken me years, but I now see how wonderful my family truly is and how much I have gained because of that. Thanks to my parents, Dave and Kathy, for creating a loving, stable environment and supporting me emotionally and financially. Thanks again for everything else, all the lessons big and small that I've learned from them. Without them, at best I'd be in more debt than I care to think about, and I have my doubts I'd be nearly as happy or successful. Thanks to my sisters, Grace and Natalie; we may not have always gotten along, but they are both lots more fun now that they don't yell at me. Thanks to Gramma Fay for teaching me that success isn't measured with dollar signs or grades, but with fun and friendship. Thanks to Gramma and Grampa Chomitz for teaching me about patience and the satisfaction of doing your job well.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The timely and accurate replication of DNA is critical for maintaining genetic integrity in cellular life. In simple cells ("prokaryotes"), the process used to replicate the genome (the "replication program") is well understood. Starting at a sequence-defined location (the "origin"), the double-stranded DNA (dsDNA) of the prokaryotic genome separates into two single-stranded DNA (ssDNA) segments. Complex biological machinery travels bidirectionally from the origin, separating the dsDNA into growing ssDNA segments, which are used as templates for the creation of two copies of the original genome. The machinery between separated and non-separated DNA ("forks") continues to propagate through the genome until it has been entirely separated and replicated. Prokaryotic organisms have such small genomes that replication can be completed using a single origin [1]. With this replication program an *E. coli* bacterium can replicate its entire genome in about 40 minutes[1].

More complex organisms ("eukaryotes") have genomes that are approximately 1000 times longer and have forks that propagate about 10 times slower than prokaryotes. For example, compare the human genome, about 3000 Mb [3] with mean fork speed 1.5 kb/min [4], to *E. coli*, about 4600 kb long with fork speed 1 kb/s [2]. The replication forks from a single origin would take nearly 4 years to completely replicate the entire human genome. In many human cells, DNA replicates in about 8 hours [5]. This is much less time than the four years that would be needed if there were only a single active origin of replication. Therefore, a single origin cannot be solely responsible for the replication of eukaryotic DNA. Eukaryotic DNA is thus replicated using a parallel process, with many origins along [6]. A considerable body of experimental evidence suggests that the timing (initiation) of origins is stochastic [7, 8, 9].

Using many origins in the replication program creates several non-obvious issues that must be addressed for the program to work effectively. One issue is the existence of separate replicated regions during the replication process. When the forks of two neighbouring replicated regions meet, the two coalesce into a single, larger replicated region. Another

[1] Under fast-growth conditions, the time to replicate the genome, $T = \frac{L}{2v} = \frac{4600 \text{ kb}}{2 \times 1 \text{ kb/s}} = 2300$ s. [2]

issue is the need to coordinate multiple origin-activation events that are driven by stochastic processes. The replication program describes the process by which the DNA is replicated using many origins of replication. These origins can coordinate their stochastic initiation times such that it is rare for any part of the DNA to be left unreplicated by the end of S phase [6].

1.1 The Cell Cycle

The cell cycle defines the steps taken during cellular reproduction and can be divided into four phases (see Fig. 1.1) [1, 5]: the first gap (G1) phase, the synthesis (S) phase, the second gap (G2) phase, and the mitosis (M) phase. The G1 phase of the cell cycle contains key processes that prepare the DNA for replication. During S phase (the second phase in the cell cycle) the DNA is replicated. The third phase of the cell cycle (G2) primarily acts as a buffer to ensure complete DNA replication. During the fourth phase of the cell cycle (M), the cell physically divides into two daughter cells.

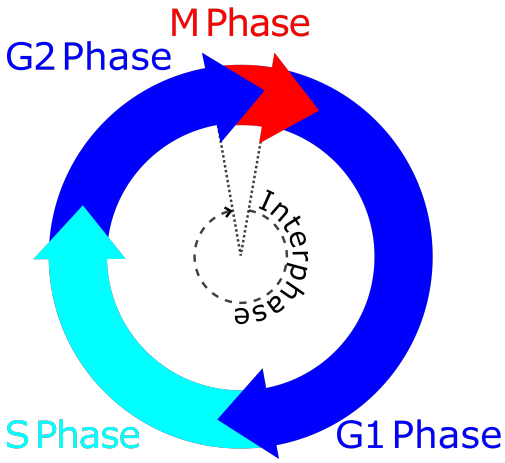

Figure 1.1: The cell cycle has four phases: Mitosis (M), when a mother cell separates into two daughter cells; the first Gap (G1), when the daughter cell undergoes growth and chemical preparation for DNA replication; Synthesis (S), when the DNA is replicated; and the second Gap (G2), which acts as a buffer to ensure complete replication before the M phase.

1.1.1 The G1 Phase

The G1 phase begins early in the life of a daughter cell, after the mother cell has divided in the preceding M phase. During this time, the cell grows and, more important, "licensing" is carried out to prepare for replication during S phase.

Licensing occurs at the origin recognition complex (ORC) [10], as seen in Fig. 1.2A. The ORC is made up of a single group of six proteins that bind to the DNA at an origin [11]. Two additional proteins (Cdc6 and Cdt1; left out of Fig. 1.2A) assist the ORC in recruiting minichromosome maintenance (MCM) 2-7 hexamer rings onto the DNA [12]. Loaded hexamers form pairs oriented away from each other [13]. Each such pair will later be referred to as "potential initiators," or just "initiators." After licensing, the resulting set of proteins associated with the origin is called the pre-replication complex (pre-RC). Licensing is suppressed during the S and G2 phases by cyclin-dependent kinases. This suppression effectively limits the cell cycle to a single replication event [1].

1.1.2 S Phase

The second phase in the cell cycle is S phase. The activation of cyclin dependent kinases (CDKs) suppresses licensing and marks the transition from the G1 phase to the S phase [1]. After licensing is completed in the G1 phase, the copying of the genome occurs during S phase. There are three main processes that happen during S phase: initiation, elongation, and coalescence, as shown in Fig. 1.2B.

An origin initiates (or "activates," or "fires") during S phase when five other proteins bind to each in a pair of MCM2-7 rings: Cdc45 and the tetrameric GINS complex[2] (only Cdc45 is shown in Fig. 1.2A). The total system of proteins is called the CMG complex (Cdc45, MCM2-7, GINS complex) and comprises a helicase that traverses the genome during S phase. As an origin fires, the pre-RC disassembles and the activated pair of helicases unwind, separating the double helix of the dsDNA into two complementary ssDNA chains [15].

After an origin has been initiated, there is a small region of ssDNA bounded on either side by the CMG complex helicases. The locations where the dsDNA is separated into two ssDNA chains are called replication forks (or just "forks"). Elongation is the process by which the replication forks, with the help of the biological machinery stored in the CMG complex [16], propagate bidirectionally from the origin, separating the dsDNA. As the forks propagate, DNA polymerases bind with the ssDNA between them. DNA polymerases use the ssDNA as a template and backbone for synthesizing dsDNA; essentially, it adds the missing half back onto the separated strand. DNA polymerase can only propagate in the $3'$ direction. This poses a problem: Because the two strands in dsDNA are oriented in opposing directions, the polymerase can smoothly traverse only one of the ssDNA chains (the "leading strand") at each fork. On the other strand (the "lagging strand"), the polymerase "stutters": It replicates a small region in the direction opposite to that of fork propagation until it hits a region that has already been replicated. The DNA polymerase then leapfrogs over and past the region it just replicated in the direction of fork propagation and repeats. These

[2] GINS is an abbreviation of the Japanese go-ichi-ni-san meaning five-one-two-three, which comes from the names of the four subunits of the complex, Sld5, Psf1, Psf2 and Psf3. [14]

small fragments are called Okizaki fragments. On the lagging strand, the Okizaki fragments are connected by DNA ligases [1, 5, 17].

Finally, when two forks meet, coalescence occurs: The helicases disassemble and the two regions of dsDNA are connected by DNA ligase [1].

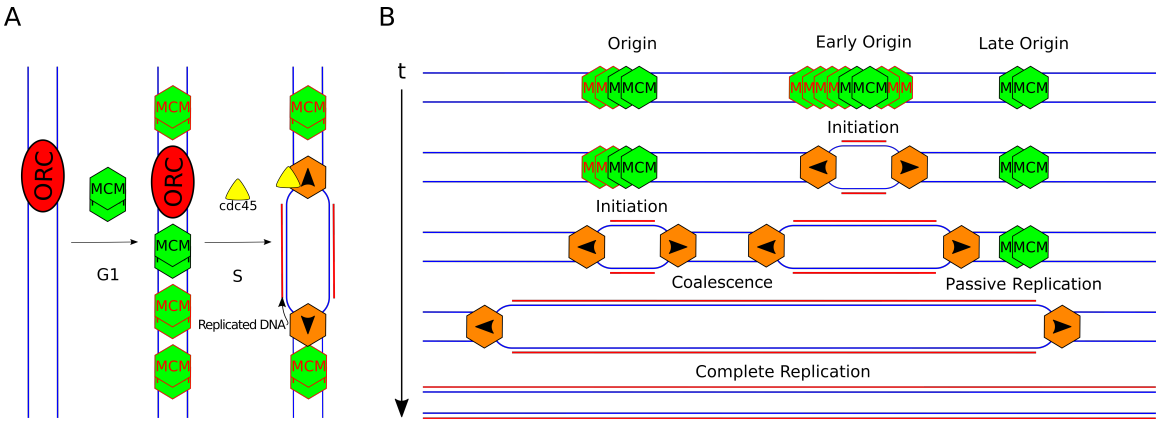

Figure 1.2: Simplified schematic of the G1 and S phases of the cell cycle, containing only those parts that are necessary to understand the model presented in Ch. 3.
A. In the G1 phase, origins are located and licensed when the ORC recruits pairs of MCM2-7 hexamers onto the dsDNA. During S phase, pairs of Mcm2-7 hexamers are activated by the Cdc45 protein. After activation, the resulting structures become replicative forks that traverse the DNA unwinding the dsDNA allowing the DNA to be replicated.
B. More detailed view of S phase. At the start of S phase, several origins are licensed along the genome (top of image). As time progresses (down), origins fire independently (initiate), and replicative forks propagate along the genome (elongation). It is common for some origins to be passively replicated; that is, they can be replicated by the replicative fork from a neighbouring origin before firing themselves. At the end of the S phase, two identical and complete sets of dsDNA will be present (bottom of image).

1.2 Origins of Replication

When prokaryotic DNA is replicated, a single origin suffices for a competent (i.e., timely and accurate) replication program. In this case, the origin is located at a sequence-specific site, and the firing time need only be early enough for complete replication. However, eukaryotic DNA requires many origins of replication for a competent replication program. The number of origins in the genome varies by species, from fewer than 800 in budding yeast [18] to about 100 000 in humans [17]. If multiple origins exist on the genome,

- *What determines the locations of the origins?*
- *What controls the timing of origin firing?*

1.2.1 Origin Locations

Depending on the organism, the factors determining origin locations vary considerably. In *Saccharomyces cerevisiae* (budding yeast), origins are tightly bound in sequences between 11 and 17 base pairs (bp) in length and are effectively localized [19]. In *Schizosaccharomyces pombe* (fission yeast), the origins are loosely associated with sequences between 100 and 200 bp long [17]. The region of potential licensing grows to about 200 kilobase pairs (kb) in the human genome [20]. In *Xenopus laevis* (African clawed frog) embryos, the origins are placed stochastically, with no sequence affinity at all [21].

1.2.2 Origin Firing Times

In favourable environments, prokaryotic organisms exhibit exponential growth, and their replication program can be quite complex. During exponential growth, the cell cycles overlap, and more than one S phase can be active simultaneously [22, 23]. However, this phenomenon is outside the scope of this thesis.

In eukaryotes, the need to initiate multiple origins leads to interesting timing dynamics. The origins do not all initiate simultaneously, with origin-initiation events occurring throughout S phase [10]. The mechanism that controls the relative timing of different origin initiation events is still a matter of some debate [19, 24, 25, 26] and is the topic of this thesis.

1.2.3 Budding Yeast

In this thesis, we focus on *S. cerevisiae* (budding yeast). Budding yeast is a useful model species because, unlike the other eukaryotic examples we discussed, the origins of *S. cerevisiae* are localized. In each cell cycle, origins of budding yeast may be licensed only in very narrow regions on the genome. These regions are defined by specific sequences in the genome, called autonomously replicating sequences (ARS elements), that have been identified and catalogued[3] [18]. The mean distance between origins is ≈ 20 kb, and the mean distance between origins is ≈ 15 kb. The difference between these two statistics is due to the long, exponential tail in the distribution of distances between origins that can be calculated from data in OriDB. Thus, the advantage of choosing budding yeast as the model species is that the potential stochasticity in origin locations has conveniently been removed from consideration.

Previous studies of the firing times of individual origins in budding yeast have found a correlation between the median firing times and the width of the firing-time distributions [19, 9]. Both studies measured the average firing time and the spread in firing times for each origin in budding yeast and discovered a correlation between them (see Fig. 2.4). Essentially, origins that tend to fire early have narrowly defined firing times, while those that tend to fire late have loosely defined firing times. This trend implies the existence of a mechanism

[3] An online database can be found at http://cerevisiae.oridb.org/

that strongly controls the firing time of origins at the start of S phase but loses its potency as S phase progresses.

One theory that explains this observation is the Multiple Initiator Model, which supposes that the number of MCM2-7 hexamers loaded on an origin will affect the timing width and median for that origin. The MIM will be discussed in detail below in Sec. 2.3.

1.3 Modelling Replication

To recap, DNA replication begins at origins which, in budding yeast, are localized spatially but whose firing times exhibit stochasticity. Once an origin has fired, replication forks traverse the DNA bidirectionally, enclosing a growing region of replicated DNA between them. When two regions of replicated DNA meet, they coalesce into a single, larger region.

This process can be mapped to a crystallization process in one dimension (Fig. 1.3): Crystallization starts when the crystal nucleates at nucleation sites, which map to origins of replication. From nucleation sites, the crystal grows bidirectionally, and the crystal domain is surrounded by boundaries that can be mapped to the replicative forks. Finally, when two crystal regions meet they coalesce into a larger region, which matches the coalescence of neighbouring regions of replicated DNA. This mapping of DNA replication to crystal growth is a formal mathematical one, not a physical one. It means that one can easily adapt well-developed stochastic models from crystal growth dynamics to describe DNA replication kinetics.

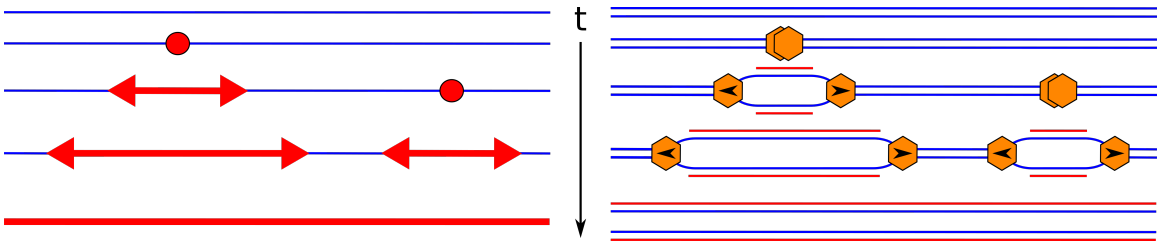

Figure 1.3: Comparison between the one-dimensional KJMA crystallization model and DNA replication. At left is the one-dimensional crystallization process, with round markers representing nucleation sites, and arrows representing crystal boundaries (pointing in the direction of propagation). Thick lines represent crystal regions, and thin lines liquid regions. Right: DNA replication process, as described in Fig. 1.2.

Early in the 20$^{\text{th}}$ century, Kolmogorov [27], Johnson and Mehl [28], and Avrami [29, 30, 31] independently developed a stochastic model to describe crystallization growth in three dimensions. Since its inception, the "KJMA model" has been used for many studies that range from phase transition kinetics [32] to Rényi's car-parking problem in one dimension [33]. In 2002, J. Herrick *et al.* introduced a KJMA-like model of DNA replication to analyze experiments on *X. laevis* embryo extracts [34]. In 2005, the KJMA-like

model was expanded, and a formalism which can be used to infer the replication program of a genome given a set of parameters that describe the speed of replication forks and the origins' locations in time and space [35, 36].

One of the quantities that can be inferred using the KJMA formalism is the replicated fraction, $f(x,t)$. The replicated fraction can be interpreted as the probability that the genome at position x has been replicated by a time t after the start of S phase. The replicated fraction is an important quantity that will be discussed more in Sec. 2.1.

1.4 Experiments Measuring Replication

The theory describing DNA replication has been driven by experimental results. In this section, we ask, *How can DNA replication be observed experimentally?* There are several techniques to help answer this question, including flow cytometry [37], DNA combing [7], microarray [38, 39], and sequencing experiments [9, 37]. When designing an experimental procedure to measure DNA replication, there are two main considerations to take into account: the temporal scope and the spatial scope of the desired measurement.

There are two experimental approaches to study the time course of replication. The first is to synchronize the cell cycle. A common way to do this is to arrest the cell cycle [40]. Arresting usually entails using a chemical bath to stop the cells from progressing from one phase to another (for experiments related to DNA replication, this is generally just prior to entering the S phase). After the cell cycle has been arrested, another chemical bath can be used to force the entire population of cells to enter the S phase synchronously. The main drawbacks to this approach are that it is difficult to arrest many species of eukaryotic organisms and that, although arresting will stop a cell from moving through the cell cycle, it will not stop a cell from growing. Because arrested cells continue to grow and acquire resources without replicating, cells that have been arrested may not remain synchronized [40]. This loss of synchronicity may explain the observation that uncertainty in measurements from arrested cell populations grows with time (discussed in Sec. 3.2.1). The second approach is to forego arresting the cell cycle and pull samples from an asynchronous population. The main drawback to this method is that it either creates time-averaged data or it relies on multiple techniques to infer the timing information. For this research, the first strategy was investigated.

Spatially, approaches range between two extremes: perfect resolution down to the scale of individual base pairs and no spatial information. At one extreme, some experiments infer quantities that are averaged over the entire genome. For example, a common flow-cytometry technique called fluorescence-activated cell sorting (FACS) [37, 41] measures only the total amount of DNA in the cell. Techniques such as FACS provide only limited information but are simple and fast. At the other extreme, techniques such as DNA sequencing and DNA

microarrays can spatially resolve windows as small as 1 kb[4] [37]. Experiments with this level of resolution are quite complex but provide tremendous insight toward understanding DNA replication. For this research, highly resolved DNA sequencing data were investigated.

Because of their impact on the research presented in this thesis, microarray and sequencing experiments will be discussed in more detail. Both experiments were designed to maximize spatial resolution, and both can use either a synchronous or an asynchronous population.

1.4.1 Microarray Experiments

Microarray experiments are high-throughput experiments that count the genome of entire populations of cells simultaneously [38]. They start with a microarray chip[5] and a population of cells. The population of cells is allowed to grow in an isotopically dense medium, then arrested at G1. The cells are then transferred to an isotopically light medium and allowed to replicate. This way, replicated DNA is lighter than DNA that has not replicated. The DNA of the population is fragmented, separated by mass, and hybridized with the chip. Then, the replicated fraction is given by the relative intensities of the two sets of hybridization data. Depending on the temporal scope of the experiment, the measured replicated fraction can be time-averaged, $f(x)$, or in the case of an arrested population, it can be the replicated fraction for a specific time, t_i after the start of S phase, $f(x, t = t_i)$.

Because this technique measures entire populations, microarray experiments do not provide information about cell-to-cell variability. More importantly, microarrays suffer from artifacts that can be challenging to overcome. For example, in 2008, McCune *et al.* measured replication fractions that spanned only 80% of the possible values [39]. In his PhD thesis, Yang discusses possible sources of this artifact such as poor discrimination between replicated DNA and unreplicated DNA [42].

1.4.2 Sequencing Experiments

A sequencing experiment determines the precise sequence of base pairs contained in the input segment of DNA [41]. To measure the replicated fraction using sequencing, one must start with with the fully mapped genome of the organism in question. The DNA from a population of cells to be measured is harvested and broken into segments about 50 bp long [9]. Each segment is sequenced and matched to the previously mapped genome. By doing this for a number of sequences such that the total length of sequenced genome is many times greater than the total length of the genome in question, one can be reasonably certain that each base-pair has been measured equally. Therefore, the normalized histogram of reads over the genome then provides the replicated fraction: Regions that have not replicated are

[4] Sequencing experiments can measure at a resolution of a single base pair [41], but when measuring DNA replication, the data are histogrammed in bins of 1 kb.

[5] actually, many chips.

measured approximately once, and regions that have replicated are measured approximately twice.

Except for the actual process of measuring how many of each segment of DNA are present in the sample, sequencing experiments and microarray experiments are very similar. The process of arresting cells, or not, is the same for both, as is the broad analysis of the output. However, sequencing experiments do not require any clever data-processing to remove artifacts such as those present in microarray experiments. Recent advances lowering the cost of sequencing have seen a transition from microarray experiments to sequencing experiments for measuring DNA replication [43].

1.5 Reading This Thesis

Here, we give a brief outline of the thesis.

Chapter 2 - Motivation. We summarize the mathematical details of the KJMA-like model that describes DNA replication and describe its use to calculate the replicated fraction from a theoretical model. We then discuss the development of the Sigmoidal Model and the correlation between the firing time width and the firing time median of an origin that it revealed. This correlation led to the creation of the Multiple Initiator Model (MIM), which we describe. Finally, we introduce the recent experimental work done in N. Rhind's lab that measured loaded MCM. These experiments measured lower number of MCMs than assumed by the MIM. With this chapter, we motivate our work in measuring the effect of small numbers of MCMs on the MIM.

Chapter 3 - Methods. We discuss the MIM simulator program, which was used in our analysis of the Multiple Initiator Model. We start by outlining the simulation process in detail and motivate the assumptions and choices we made. Our discussion of the MIM simulator transitions into an analysis of noise; our simulations produce data similar to experiment, including the level of noise. We estimate the noise in current cutting-edge sequencing data, and compare that to the noise in our simulation. Because the noise in experiment exceeds the noise in our simulated data, we introduce two methods of increasing the noise in our simulations: One method is based on experimental limitations, and the other method involves adding Gaussian noise.

Chapter 4 - Results. We outline the four investigations we undertook in our exploration of the effect of small numbers of MCMs on the MIM. We started with preliminary single-origin measurements comparing the analytical MIM to simulated data as a quick investigation. These simple measurements indicated that there is a large effect and motivated the topic of this thesis. Next, we redesigned our methodology and performed more single-origin measurements. Third, we simulated Chromosome I of budding yeast. In this investigation, we measured how the fit from MIM changes when it is forced to have high or low numbers of MCMs. The results from our third investigation contradicted our findings

from the first two. To resolve the contradiction, we simulated artificial genomes containing two, three and four origins. In these multiple-origin simulations, we observe two trends that contribute such that inferences made with the MIM grow in accuracy as the number of origins increases.

Chapter 5 - Conclusions. We discuss the implications of our results and suggest new directions for this research.

Chapter 2

Motivation

Previous work on quantitative modelling of DNA replication has investigated the timing of origin initiation [19, 26, 9, 44, 45]. In 2010, Yang *et al.* developed the "Sigmoidal Model," which uses three parameters per origin (position, median firing time, and spread in firing time) to describe the replication program of budding yeast [19]. After fitting these parameters to microarray data, the authors observed a correlation between the median firing time and spread in firing time. This result (discussed in detail in Sec. 2.2) was confirmed by Hawkins *et al.*, who used a similar model to analyze sequencing data in 2013 [9].

The work of Yang *et al.* led to the development of a second analytical model, "the Multiple Initiator Model" (MIM) [19], which we present in Sec. 2.3. The MIM proposes a biological hypothesis that explains the observed correlation between median firing time and spread in firing time. The benefit of the MIM over the Sigmoidal Model is that the MIM uses only two parameters per origin to define the replicated fraction, effectively removing one-third of the parameters from the model.

However, recent work performed in N. Rhind's laboratory[1] [46] has shown that one part of the scenario assumed in the MIM may not be biologically realistic (Sec. 2.4). The purpose of this thesis is to explore the impact of these new experimental data on the MIM. This chapter will expand on the above story.

2.1 Replicated Fraction

In Secs. 1.3 and 1.4, we saw that the replicated fraction, f, can be calculated from theoretical models and inferred from experiments. The replicated fraction as a function of time and space, $f(x,t)$, can be interpreted two ways: as describing either a single cell or a population of cells. In the single-cell case, $f(x,t)$ is interpreted as the probability that the sequence at position x in the genome has replicated by a time t after the start of S phase. For a population of cells, $f(x,t)$ represents the fraction of cells in the population that have replicated

[1] The Rhind laboratory is at the University of Massachusetts Medical School in Worcester MA, USA.

at position x by a time t after the start of S phase. Although the two interpretations of f might seem equivalent, we will see in Sec. 2.3 that they are subtly different. Both definitions lead to a function that has values ranging from zero (no replication has occurred), to one (replication has certainly occurred).

2.1.1 Qualities of the Replicated Fraction

Before we describe in detail the KJMA-like model of DNA replication, we will build some valuable intuition. In DNA sequencing experiments, the replicated fraction is measured spatially in windows about 1 kb wide and temporally in steps of 5 minutes [9]. Since the budding-yeast genome has about 600 origins of replication and is about 12 Mb long, origins are, on average, spaced every 20 kb [18, 47]. The distribution of inter-origin distances that can be calculated from the data in the origin database[2] [18] shows that $\approx 10\%$ of origins are within 2 kb of each other. Thus, a spatial resolution of 1 kb is narrow enough to uniquely identify and observe the majority of origins individually. However, there are generally no more than ten time points measured experimentally [9, 37, 39]. (Indeed, the data analyzed in Sec. 3.2.1 have only six.) Fortunately, this amount of temporal data is enough to infer the important features of the replication program.

Figure 2.1 shows an example set of replicated fraction data. The data come from measurements done on Chromosome IV of budding yeast by Hawkins *et al.* [9]. We will now provide a simple explanation of the experimental processes that produced the data in Fig. 2.1. Two populations of cells were synchronized by arresting the cells and releasing them into S phase. FACS was applied to one population to generate the average replicated fraction over the entire genome at six times throughout S phase. The DNA of the other population was fragmented and sequenced, which provided a relative measure of where the genome had been replicated (more sequences from a region means that region has replicated in more cells within the population). Finally, the sequencing data was normalized such that the average replicated fraction over the genome was equal to that observed with FACS. The reader may notice a few features in the data in Fig. 2.1: there are gaps in the spatial data; the replicated fraction ranges lower than zero and higher than one; and some regions of the genome replicate faster than others.

The gaps in Fig. 2.1 exist because of a limitation of the sequencing experiment used to gather this data. Sequencing experiments match short sequences of DNA to the fully mapped genome (Sec. 1.4.2). The budding yeast genome contains repeated patterns, defined here as sequences longer than 50 bp that appear more than once [48]. When a sequence of DNA extracted from one of these patterns is measured, it is not counted because it cannot be uniquely located.

[2] http://cerevisiae.oridb.org/

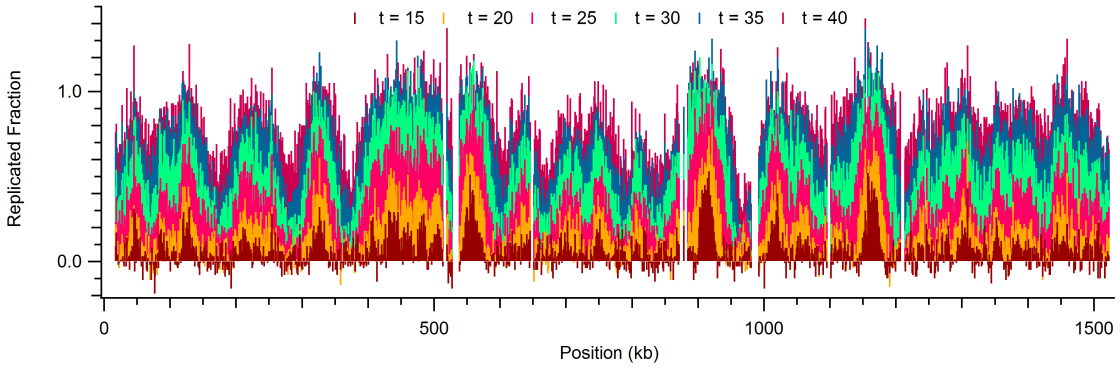

Figure 2.1: Example graph of replicated fraction. Data from chromosome IV of budding yeast, as measured by Hawkins *et al.* ([9] supplementary data). x-axis represents the spatial organization of the genome as if it had been stretched out straight. y-axis is the replicated fraction. Six time points.

The replicated fraction in Fig. 2.1 has a range that goes below zero and above one. This surprising feature results from two assumptions: first, that the measured sequences were evenly distributed spatially; and, second, that all cells have the same average replicated fraction at the time measured, $f(t = t_i)$. In [9], Hawkins *et al.* extracted and measured about 10^7 sequences. However, because of counting statistics and because sequences may be unevenly distributed, there will be regions of the genome that are sequenced more and regions that are sequenced less. Additionally, Hawkins *et al.* normalized the measured replicated fraction by setting the average replicated fraction, $f(t = t_i)$, equal to the replicated fraction measured using FACS on the bulk sample. This normalization assumes that the measured cells have the same average replicated fraction as the population measured with FACS. The error in regions that have been over-extracted or under-extracted can be exaggerated by normalizing incorrectly, leading to values of f greater than one or less than zero.

The most important observation is that some regions of the genome start replicating much earlier than others. This can be seen in the peaks in Fig. 2.1; for example, at $x \approx 910$. Because replication starts at an origin and propagates outward, peaks in the replicated fraction imply early replication and, hence, the presence of origins. Additionally, early origins should create stronger peaks, and late origins should create weaker peaks. Finally, neighbouring origins that initiate at similar times may not lead to distinct peaks in $f(x, t)$.

2.1.2 Calculating Replicated Fraction from the KJMA Formalism

The replication program is defined by the origins through their spatial and temporal organization. The speed at which the replicative forks propagate also plays a role in determining

the replicative program. Based on the work of Jun *et al.* [35], here we outline estimation of the replicated fraction from data describing the origins of replication and the replicative forks.

We define the rate of initiation, $I(x,t)$, to be the number of origins initiated per time per genome length at an unreplicated position x, and time t after the start of S phase. Of course, initiation can happen only at origins of replication. In budding yeast, origins are localized at known locations [18], labeled x_i. Therefore, we define the rate of initiation at origin i to be $I_i(x,t) = \delta(x - x_i)I_i(t)$, where $\delta(x)$ is the Dirac δ function. Finally, we define the rate of initiation to be $I(x,t) = \sum_i I_i(x,t)$.

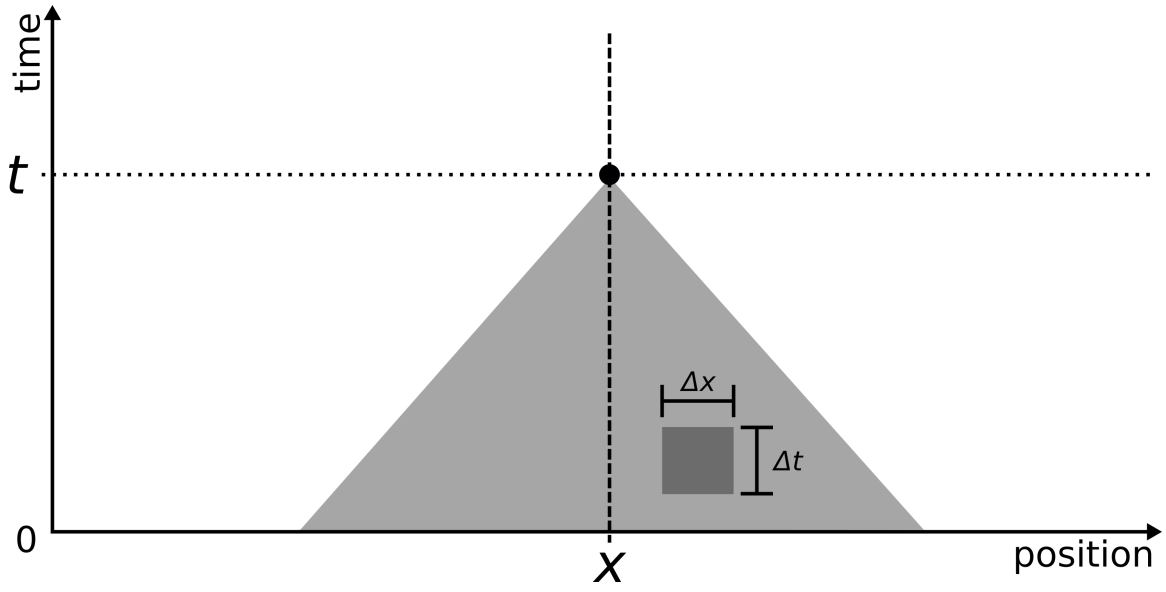

Figure 2.2: KJMA approach to calculating $f(x,t)$. In order for the point at (x,t) to not have been replicated, there cannot be any initiation events within the shaded triangle.

Given the function $I(x,t)$, we can infer the replicated fraction, $f(x,t)$, at a position x a time, t, after the start of S phase:

$$f\left(x,t\right) = 1 - \prod_{\Delta} \left[1 - I\left(x',t'\right)\Delta x'\Delta t'\right] \ , \tag{2.1}$$

where the product is over intervals $\Delta x'\Delta t'$ lying within the "past triangle" shown in Fig. 2.2. In words, Eq. 2.1 says that the probability that the genome at position x has been replicated is one minus the probability that no origin has fired long enough in the past to have a replication fork pass over position x. In the limit $\Delta x \to 0$ and $\Delta t \to 0$, Eq. 2.1 becomes

$$f\left(x,t\right) = 1 - \exp\left[-\iint_{\Delta} dx'dt' I\left(x',t'\right)\right] \ . \tag{2.2}$$

Now, it is possible to define a new quantity, $g(\Delta x_p, t)$, that is a local measure of origin firing:

$$g\left(\Delta x_p, t\right) = \int_{x_p}^{x_{p+1}} dx \sum_i \delta\left(x - x_i\right) \int_o^t dt' I_i\left(t'\right) \tag{2.3}$$

over the region $\Delta x_p \equiv [x_p, x_{p+1})$ of a genome of length, L, discretized into M segments;

$$\Delta x = \frac{L}{M} \qquad x_p = p\left(\Delta x\right) \qquad p = 0, 1, 2, \ldots, M-1 \ . \tag{2.4}$$

$g(\Delta x_p, t) = 0$ if there are no origins enclosed in Δx_p because initiation will only occur at an origin. Thus, we replace the double integral in Eq. 2.2 by the function $g(\Delta x_p, t)$ and arrive at

$$f\left(x, t\right) = 1 - \exp\left[-\sum_{p=0}^{M-1} g\left(\Delta x_p, t - \frac{|x - x_p|}{v}\right)\right] \ , \tag{2.5}$$

where v is the speed of replication forks, Δx_p is the p^{th} interval, x_p is the p^{th} position, and $|x - x_p| / v$ is the time at the edge of the past triangle in Fig. 2.2.

Recognizing that $g(\Delta x_p, t)$ represents the initiation rate of budding yeast, we can constrain it to better describe the biological system: First, we constrain g such that replication cannot happen before the start of S phase; $g(\Delta x_p, t < 0) = 0$. Second, we constrain the initiation rate to be non-negative. Because of the definition in Eq. 2.3, this constrains g as well: $\frac{d}{dt} g(\Delta x_p, t) \geq 0$. Thus, as a consequence of the first two constraints, $g(\Delta x_p, t) \geq 0$.

Finally, we derive the cumulative initiation probability, $\Phi(x_p, t)$, from $g(\Delta x_p, t)$ using a calculation similar to that used for a Poisson process [49]:

$$\Phi\left(x_p, t\right) = 1 - e^{-g(\Delta x_p, t)} \ . \tag{2.6}$$

The cumulative initiation distribution is an important quantity that will be revisited below, in Sec. 2.3. Note that $\Phi(x_p, t)$ is a general function that can be defined throughout the genome, but in the case of budding yeast is nonzero only for intervals $[x_p, x_{p+1})$ that contain an origin.

2.2 The Sigmoidal Model

The sigmoidal model is a phenomenological approach to characterizing each origin. Developed by S. Yang as part of his PhD thesis, this model assumes that the functional form of $I_i(t)$ is a sigmoidal function that has a range from zero to one and that is defined by three parameters for each origin, i, on the genome [19, 42].

Figure 2.3 shows the preliminary observations that motivated the sigmoidal model. First, the replicated fraction[3] at an origin, $f(x = x_i, t)$, was extracted from experimental data of the entire genome, $f(x, t)$ (Fig. 2.3A). The figure shows the analysis of microarray data [39]. Second, a sigmoidal curve was fit to $f(x = x_i, t)$ (Fig. 2.3B). This sigmoidal curve is parameterized by the median replication time, t_{rep}, and by the spread of replication times, t_{width}:

$$f(t) = \frac{1}{1 + \left(\frac{t_{\text{rep}}}{t}\right)^r} \ , \tag{2.7}$$

where t_{width} is defined by

$$t_{\text{width}} = \left(3^{1/r} - 3^{-1/r}\right) t_{\text{rep}} \ . \tag{2.8}$$

One can see in Fig. 2.3C the correlation between t_{rep} and t_{width}. However, this approach ignores interactions between neighbouring origins and their effects on $f(x = x_i, t)$; these parameters may not describe intrinsic properties of the origins. Thus, the correlation may not be due to a biological process controlling the origins, but because of coincidental interactions due to their relative positions and firing times.

To better analyze the data, Yang developed the Sigmoidal Model, an analytical method for quickly calculating the replicated fraction over the whole genome, $f(x, t)$. The model calculates $f(x, t)$ from a set of origins defined by three parameters (x_i, $t_i^{(1/2)}$, and $t_i^{(\text{w})}$). The parameters $t_{1/2}$ and t_{w}, which define intrinsic properties of the origins, are analogous to t_{rep} and t_{width} respectively. (t_{rep} and t_{width} are inferred directly from replicated fraction data and do not take into account overlapping replication regions, whereas $t_{1/2}$ and t_{w} do take into account the overlap in replicated regions. This difference can be seen by compairing the spread of data points in Figs. 2.3C and 2.4.) Thus, $t_{1/2}$ and t_{w} are defined by Eqs. 2.7 and 2.8 as well [42]. With this, the entire set of experimental data was used to characterize every origin simultaneously (not illustrated).

Figure 2.4 graphs the intrinsic $t_{1/2}$ vs t_{w} calculated from fitting the Sigmoidal Model to microarray data. Notice that the strong correlation between timing width and median observed in the crude analysis of Fig. 2.3C is present in the intrinsic parameters as well. This correlation means that early origins have narrowly defined firing times, while late origins have loosely defined firing times. An implication is that there is a mechanism that controls origin firing time that is strong at the start of S phase but weakens as S phase progresses [42]. This observation suggested the Multiple Initiator Model (MIM).

2.3 The Multiple Initiator Model

The MIM, in its simplest form, assumes that each origin has a given number of potential initiators that may be initiated during S phase [42]. If each of these potential initiators

[3] Yang used the term "replication fraction" in his thesis.

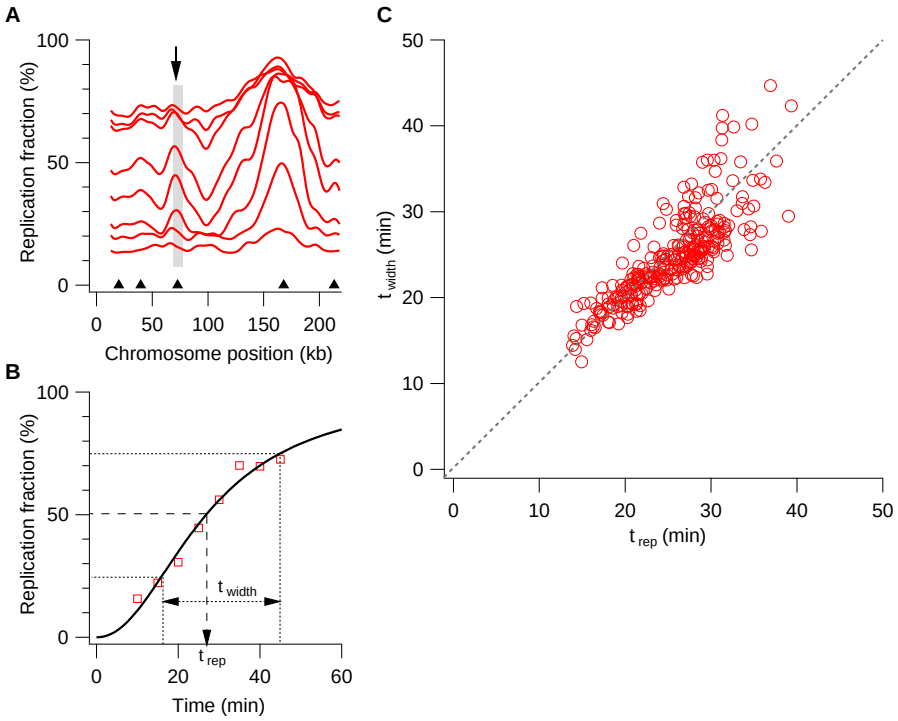

Figure 2.3: Schematic of the initial analysis of budding yeast data that led to the creation of the Sigmoidal Model.
A. Sample replicated fraction: smoothed data from microarray measurements of Chromosome I (solid lines) [39]. The black triangles indicate the locations of previously identified origins [50]. Data from the replicated fraction at a single origin (grey region) at a time were analyzed.
B. Equation 2.7 fitted to the extracted replicated fraction at an origin. The fit function has parameters, $t_{\rm rep}$ and $t_{\rm width}$, which are shown.
C. Scatter plot of origin parameters from fitting the replicated fraction at every origin reveals a correlation. The dashed line shows $t_{\rm width} = t_{\rm rep}$. Figure reproduced with permission from S. Yang [42].

has equal opportunity to fire, then origins with large numbers of initiators should tend to fire earlier than origins with few initiators. Effectively, origins with more initiators loaded will tend to fire earlier in S phase than origins with fewer pairs. However, it is important to note that other factors, such as chromatin structure (the three-dimensional organization of the genome), can affect the relative firing times of origins [51]. For example, because of chromatin structure, some regions of the genome are less accessible and the proteins that make up the replication machine may be impeded, slowing the loading of MCMs during G1 phase and delaying their activation during S phase.

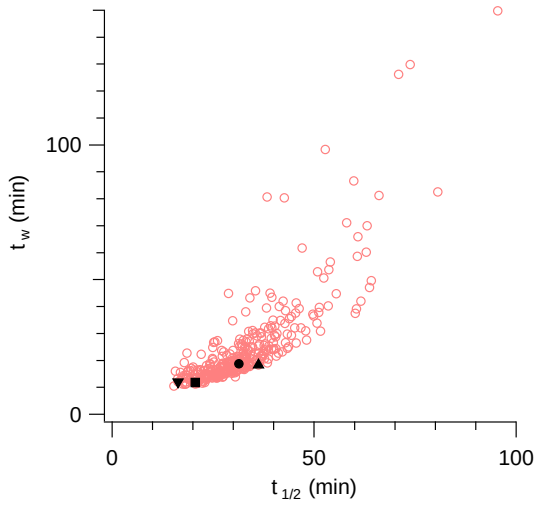

Figure 2.4: Scatter plot of origin parameters from fitting the replicated fraction of the entire genome with the Sigmoidal Model reveals a correlation. Solid points are specific origins identified for discussion in Yang's thesis. Figure reproduced with permission from S. Yang [42].

2.3.1 MIM Basics

One hypothesis for the biological mechanism that makes up a potential initiator focuses on the MCM2-7 hexamer pairs loaded at each origin. During licensing, the ORC can load MCM2-7 hexamers in excess [52]. A simple hypothesis (that will be discussed in Sec. 3.1.1) is that initiators are loaded as a Poisson process: MCM2-7 hexamers are loaded at an origin individually with some probability determined by the affinity of that origin. We define the average number of initiators loaded at the i^{th} origin to be n_i, and, the actual number of initiators to be N_i (thus, N_i is a random number that differs each cell cycle and $n_i = \langle N_i \rangle$). We assume that N_i is Poisson distributed, an assumption we motivate in Sec. 3.1.1.

During S phase, the initiators are activated by the addition of Cdc45 and the GINS complex. The MIM assumes that each initiator has the same cumulative probability of firing as time progresses through S phase, given by

$$\Phi_0(t) = \frac{1}{1 + \left(\frac{t^*_{1/2}}{t}\right)^{r^*}} \ , \tag{2.9}$$

where $t^*_{1/2}$ is the median firing-time for a single initiator and where r^* sets the width of the distribution. These variables are global, defining the behaviour of every initiator on the genome. From this assumption, the cumulative probability that an origin with N loaded initiators has fired is

$$\Phi_{\text{eff}}(t, N) = 1 - [1 - \Phi_0(t)]^N \ . \tag{2.10}$$

From Eq. 2.10, the replicated fraction can be inferred from a set of global parameters (fork velocity, time, $t^*_{1/2}$, r^*, and two parameters defining the noise in the experimental data) and two parameters per origin (its position on the genome, and the number of initiators it loads). We start by calculating the effective cumulative firing time distribution, $\Phi_i^{(\text{eff})}(x, t, N_i)$ for each origin, i. Next, we invert Eq. 2.6,

$$\ln\left(1 - \Phi_i^{(\text{eff})}(x_i, t, N_i)\right) = -g(\Delta x_p, t) \ , \tag{2.11}$$

and sum over every origin in the genome,

$$\sum_{\text{all origins } i} \left[\ln\left(1 - \Phi_i^{(\text{eff})}(x, t, N_i)\right)\right] = -\sum_{p=0}^{M-1} g(\Delta x_p, t) \ , \tag{2.12}$$

to calculate the initiation rate for the entire genome. Finally, we can replace the exponent in Eq. 2.5 with the left-hand side of Eq. 2.12. Using this process, Yang fit the parameters listed above to microarray data as part of his PhD research [42]. (When Yang's research was undertaken, sequencing experiments were not yet common.)

Equation 2.10 calculates the effective cumulative probability distribution of an origin with N loaded initiators. This is a property of a single cell, with a single value for N. In Sec. 2.1, we mentioned that there is a subtle difference between the single-cell interpretation and the population interpretation of the replicated fraction. If the number of loaded initiators at an origin does not change between cell cycles (i.e. $N = n$), then the two interpretations are equivalent and Eq. 2.10 applies to large cell populations as effectively as a single cell. However, if N varies among cell cycles, the two interpretations diverge. Equation 2.10 then becomes

$$\Phi_{\text{eff}}(t, n) = 1 - \langle[1 - \Phi_0(t)]^N\rangle \ , \tag{2.13}$$

where $\langle\cdots\rangle$ denotes the ensemble average over $P_n(N)$.

2.3.2 Accounting for variability in N

Given the many factors that affect the ability of the ORC to load MCM2-7 initiators onto DNA [52], it is reasonable to assume that the number of initiators will vary over cell cycles. Thus, we need a way to evaluate Eq. 2.13 to calculate $\Phi_{\text{eff}}(t, n)$.

In this thesis, we assume that initiators are loaded as a Poisson process. This means in a large population of cells, if a particular origin has a mean number of initiators, n, the standard deviation of N will be $\sqrt{n}$ [53]. Therefore, as n grows, the relative fluctuations within the population shrinks as $n^{-1/2}$. Thus, for large-enough n, we can neglect fluctuations in n and Eq. 2.10 becomes accurate. Indeed, in his thesis, Yang assumed n was large and, therefore, that $P(N_i) = \delta(N_i - n_i)$ was accurate [42]. However, recent experimental evidence

suggests that, typically, n ranges between one and five, which is not as large as curvefits based on the MIM assume.

2.4 Experimental Measurement of the Number of Initiators

The MIM makes a strong hypothesis about the physical mechanism controlling origin firing times during S phase. In particular, its predictions about the relative number of MCM pairs at a given origin can be checked experimentally. Here, we describe recent experiments performed by Das *et al.* that constitute a first attempt to estimate relative and absolute numbers of loaded MCM pairs in budding yeast [46]. The work was unpublished at the time the thesis was written and was kindly made available by N. Rhind.

Das *et al.* made several measurements to test the MIM. First, they measured the relative number of initiators loaded at each origin; according to the MIM, origins that fire earlier should have more initiators than those that fire later. The relative number of initiators indicates which origins have more initiators, but cannot measure exactly how many initiators there are (i.e., it can say that origin a has twice as many initiators as origin b but cannot differentiate between $n_a = 2$, $n_b = 1$ and $n_a = 200$, $n_b = 100$). Second, Das *et al.* measured the effect of reducing the number of initiators at an origin; according to the MIM, reducing the number of loaded initiators should delay the mean firing time of that origin. Third, they measured the average absolute number of initiators loaded on a particular early origin; according to the MIM, an early-firing origin should have many initiators loaded, on average.

2.4.1 Relative Number of Initiators

To measure the relative number of initiators, Das *et al.* used ChIP-seq in a population of G1-arrested cells. ChIP-seq (Chromatin immunoprecipitation followed by sequencing) is a technique that profiles genome-wide DNA-binding proteins, histone modifications or nucleosomes [54]. Das *et al.* used ChIP-seq to measure the number of MCMs bound to budding yeast DNA. In this case, Das *et al.* prepared the experiment such that the output provided a measure of the relative number of MCM proteins loaded throughout the genome. Figure 2.5A shows the relative number of MCM proteins in Chromosome X of budding yeast. The peaks align with origins identified in OriDB [18]. Figure 2.5B shows the relative number of MCM2-7 hexamers loaded at an origin vs. the n value predicted from the MIM in 2010 [19]. After ignoring origins that are believed to fire late due to their location in chromatin structure (blue origins in Fig. 2.5B) [55], Das *et al.* observed a correlation between the number of initiators and the theoretical parameter n calculated by the MIM. This measurement confirms the first prediction made by the MIM, that the number of initiators loaded correlates with origin firing times.

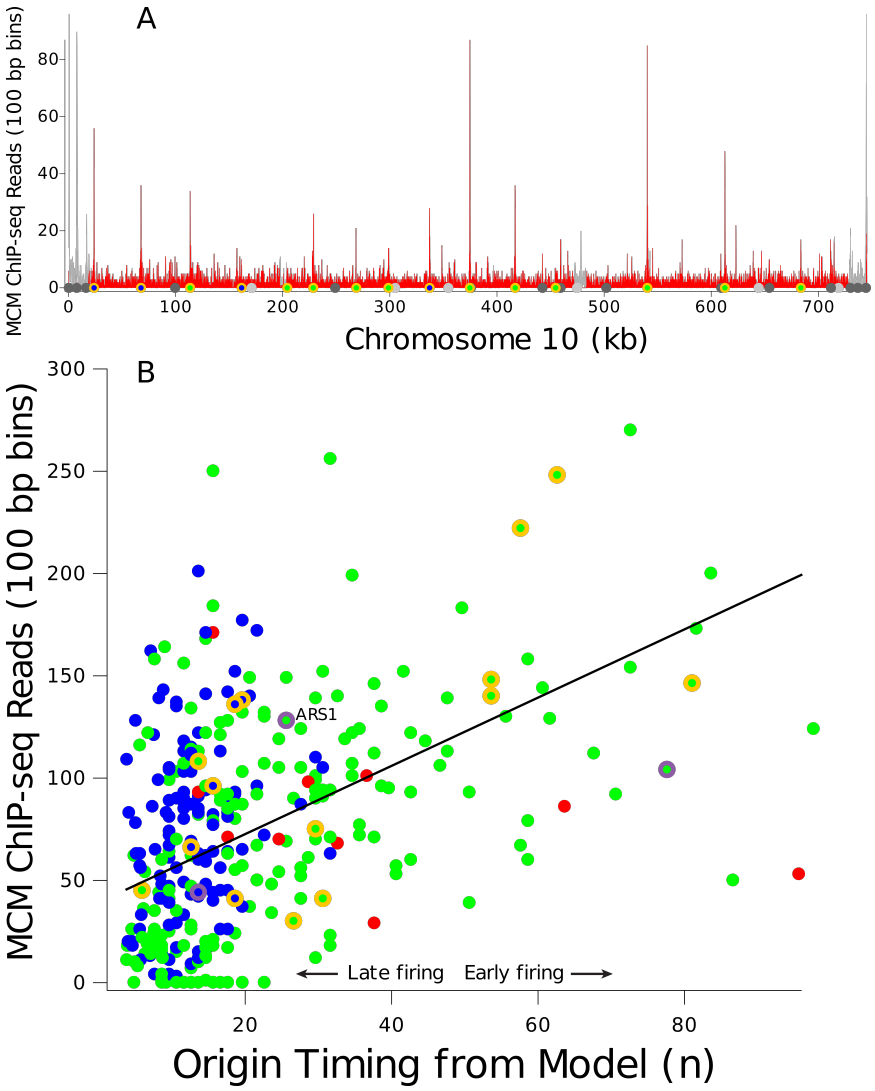

Figure 2.5: Results from ChIP-seq experiments measuring the relative number of loaded MCMs throughout the budding yeast genome.
A. ChIP-seq measurements for Chromosome X from G1 arrested wild-type cells. Red histogram shows uniquely located reads, grey is multiply-located reads. Circles along x-axis show the locations of identified origins.
B. Scatter plot of ChIP-seq data at origins over the entire genome vs their n values calculated from the MIM. The firing times of blue and red origins are believed to be affected by chromatin structure [51]; green are not. Some origins are labeled with double circles. These labels refer to other parts of the data presented by Das *et al.*, but not presented here. ARS1 origin is labeled with text. Line represents the best linear fit to green dots ($r = 0.56$, $r^2 \approx 0.31$). Figure reproduced with permission from N. Rhind [46].

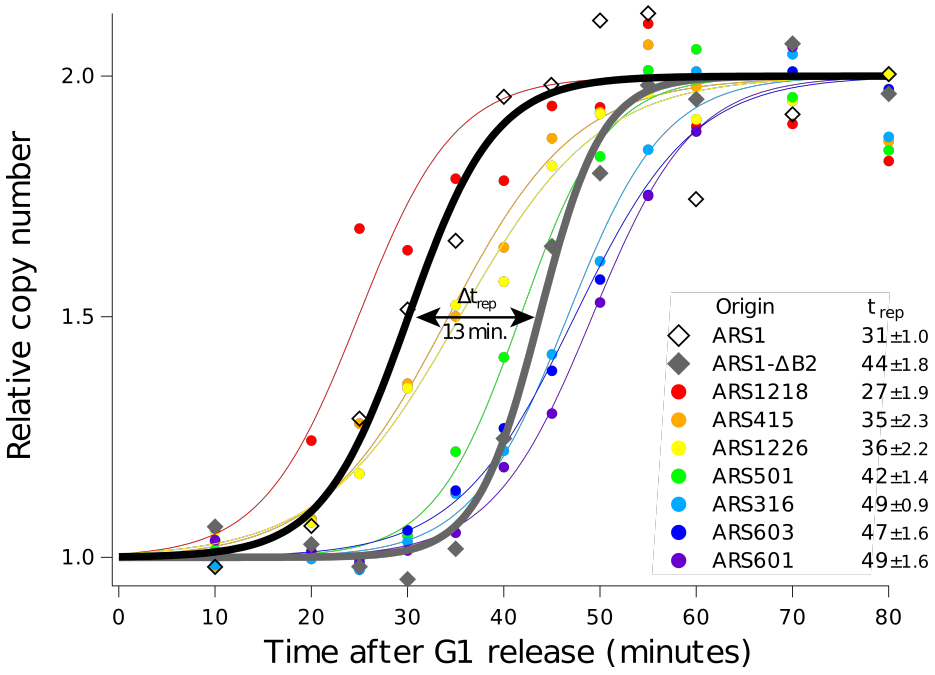

Figure 2.6: Replication profiles of various origins. Many of the origins are outside the scope of this discussion. ARS1 and ARS1ΔB2 are shown as the thick black and thick grey curves respectively. The removal of the B2 element of ARS1 causes that origin to fire, on average, 13 minutes later in S phase. Note that, on the y-axis, f = [Relative copy number] − 1. Figure reproduced with permission from N. Rhind [46].

2.4.2 Suppressing the Loading of Initiators

To evaluate the effect of suppressing the loading of initiators on the replication program, Das *et al.* measured the replication profile of ARS1 and the ARS1-ΔB2 mutant. The ARS1 origin is known to be early firing [18] and should therefore have a relatively high number of initiators. Because the B2 element of ARS1 takes part in the recruitment of Mcm2p, the ARS1-ΔB2 mutant (which has the B2 element removed) reduces MCM2-7 loading [56]. The expectation, based on the MIM, is that the mutant ARS1-ΔB2 will have a later mean firing time because of the reduced number of initiators loaded. By measuring the replicated fraction of cells with both wild-type and mutant ARS1 origins independently, Das *et al.* saw a marked (13 minute) delay in the average replication timing of ARS1 caused by the ΔB2 mutation (Fig. 2.6).

2.4.3 Number of Initiators Loaded

Das *et al.* engineered several special plasmids and used them to measure the average number of initiators loaded at an early firing origin. A plasmid is a small loop of dsDNA that is separate from the genome and that is replicated independently [1]. On each plasmid was

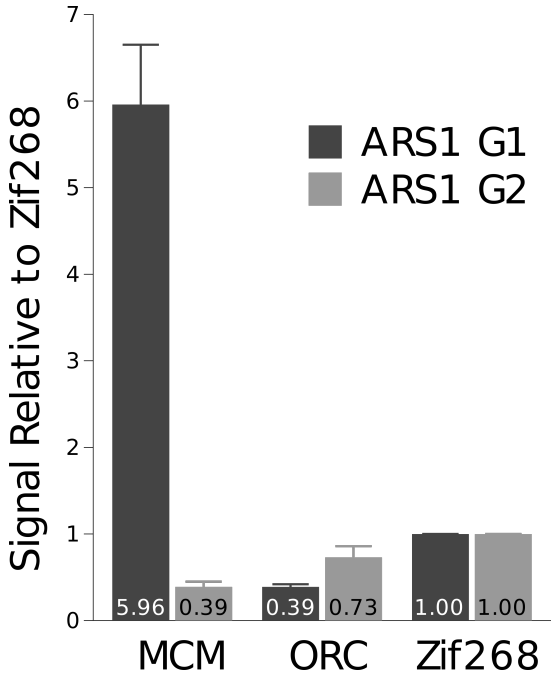

Figure 2.7: Quantization of data from a western blot experiment that measured the amounts of MCM, ORC and Zif268 present in populations of G1 arrested plasmids and G2 arrested plasmids. The left-most column shows that, on average, there are about 3 initiators loaded at ARS1 during the G1 phase. Figure reproduced with permission from N. Rhind [46].

engineered one of a selection of origins (plasmids were separated into populations such that each plasmid in a given population contained the same origin). One of the six proteins in the ORC and one of the six proteins in the MCM2-7 hexamer were tagged, so that their relative average numbers could be measured with western blotting. In order to calculate the absolute average numbers of MCM2-7 hexamers and ORCs, Das *et al.* normalized the measurements using a zinc-finger protein. The normalization was possible because Zif268, the so-called "zinc-finger protein," binds to a specific 10-bp sequence of DNA with sub-nanomolar affinity [57]. By including a single instance of the Zif268 binding sequence in the plasmid, Das *et al.* concluded that each plasmid had exactly one Zif268 protein bound to it. From a population of G1 phase arrested cells, the engineered plasmids were extracted and the relative number of ORCs, MCM2-7s and zinc-finger proteins measured and normalized such that the average number of zinc-finger proteins was one.

The results of Das *et al.* for the ARS1 origin are particularly instructive. Figure 2.7 shows the average number of MCM2-7 hexamers (one initiator is a pair of these) loaded on ARS1, and the average number of loaded ORCs during G1 arrest and G2 arrest. The conclusion is that there are $n \approx 3$ MCM pairs on the ARS1 origin during G1 arrest. It is important to restate that the MIM does not report absolute n but rather gives relative n. In other words, the value for n reported by the MIM is proportional to the number of unique

chances an origin has to fire. Thus, the fact that the number measured by Das *et al.* does not match that predicted by the MIM is not immediately troublesome. However, the small number of initiators, means that cell-to-cell variability can be important. The question we set out to answer with this research is, thus, *How does cell-to-cell variability in the initiation factor affect the replication program?*

Chapter 3

Methods

In this chapter, we outline the tools we used in our investigation of the impact of variability in the initiation factor on the MIM. The primary tool that we developed for our investigation is the MIM simulator, a Monte Carlo program that simulates DNA replication. A great deal of effort went into the details of the simulation program to ensure it efficiently produces meaningful results: We ensured the randomly generated numbers were distributed properly. We adopted the phantom-nuclei algorithm, an efficient way to simulate the replicated fraction [35]. We rewrote key parts of our code in a low-level programming language, C++, to increase performance. We simulated measurements consistent with current experimental results [9].

Except when noted, all computations were performed in IGOR Pro Version 6.3.6.4.

3.1 The MIM Simulator

The MIM simulator takes as inputs a set of parameters nearly identical to those defined by the MIM. There are four global inputs: the elapsed time since the start of S phase t_{sim}, the speed of replicative forks v, the median firing time $t_{1/2}$, and r, which defines the width of the cumulative firing time distribution. There are also two local parameters per origin: the position x_i and the average number of initiators n_i. This set of parameters is not identical to those outlined in Sec. 2.3 because, as we will describe in Sec. 3.2, noise was not treated the same way. The simulator uses these parameters to generate the replicated fraction, $f(x, t = t_{\text{sim}})$, over the entire genome. The simulation does this over several sets of parameters for which only t_{sim} changes by steps of 5 minutes, thereby efficiently creating data comparable to those from sequencing experiments.

The MIM simulator has three modules (see Fig. 3.1): The preparation module sets the randomly distributed parameters. The phantom-nuclei module uses those parameters to calculate $f(x, t = t_{\text{sim}})$. The housekeeping module tracks progress, calls the preparation and phantom-nuclei modules, and analyzes the results. Note that while the preparation and

the phantom-nuclei modules both simulate only a single cell at a time, the housekeeping module loops over many cells to find the average behaviour of a population. These three modules will be discussed in more detail below.

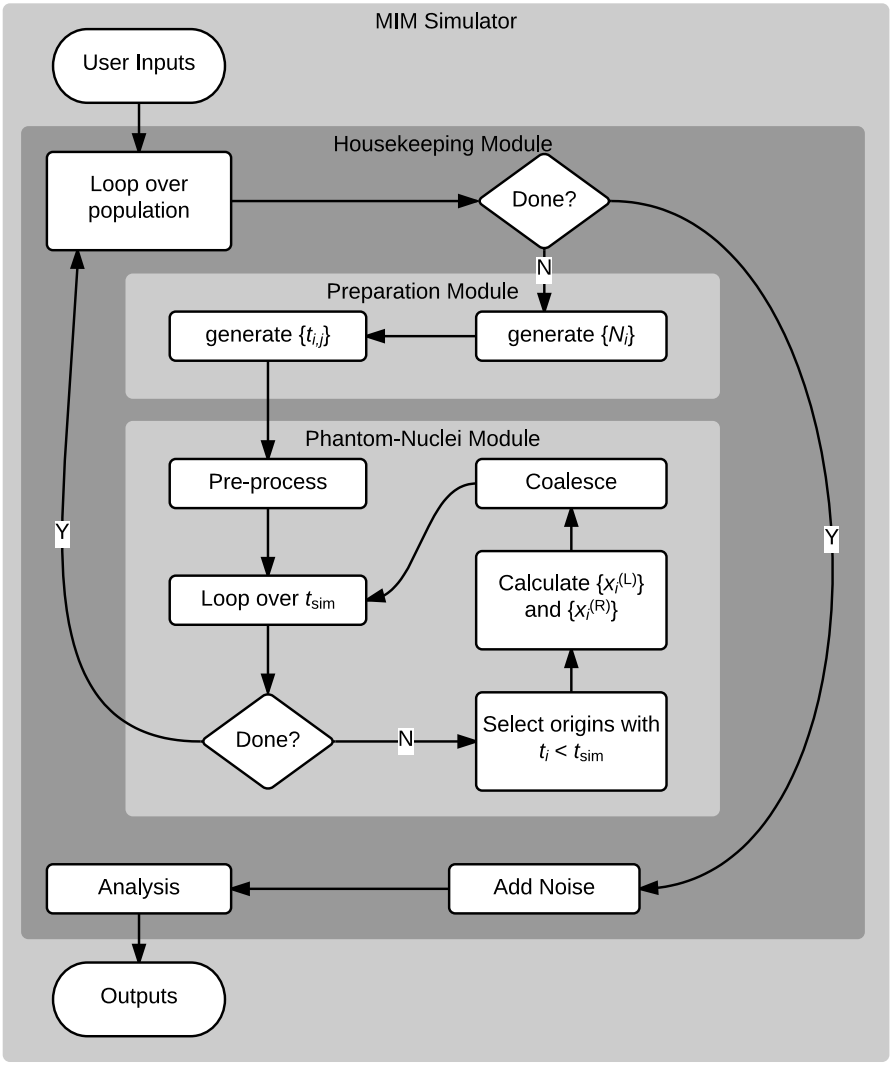

Figure 3.1: Flow chart illustrating the MIM simulator structure. The program contains three modules: The housekeeping module loops over every cell in the population being simulated, calls the preparation and phantom nuclei modules, adds noise to the results and performs analysis. The preparation module generates two sets of random data, the number of initiators at each origin and the firing times of each initiator. The phantom-nuclei module pre-processes the data passed to it, and calculates the replicated fraction for each time step using the phantom nuclei algorithm, which itself is broken into three steps.

3.1.1 The Preparation Module

The preparation module is a Monte Carlo program, one whose output depends on random numbers [58]. For the preparation module of the MIM simulator, we need two sets of random numbers: First, the program requires a set of absolute numbers of initiators, $\{N_i\}$, for all origins i. Second, the program requires a set of firing times, $\{t_{i,j}\}$, for each initiator j loaded at each origin i. For both sets, we took care to ensure that the generated values were properly distributed to match MIM theory (Sec. 2.3). The preparation module is analogous to the licensing process undertaken in the G1 phase of the cell cycle (Sec. 1.1.1).

The first task of the preparation module is to randomly generate $\{N_i\}$, the set of absolute numbers of initiators at all origins i for the cell being simulated. Thus, the first choice we made in creating the simulator was how the values for N_i should be distributed, given their average n_i. In Sec. 2.3.1, we mentioned that a simple hypothesis is that initiators are loaded onto an origin as a Poisson process; if this is the case, the number of initiators should be Poisson distributed. This simple model, which assumes initiators are loaded at a constant rate and loading events are not correlated, is easily implemented. However, we are unaware of any experiments that have measured the distribution of the number of initiators over different cell cycles.

In discussion with collaborators, another hypothetical distribution was considered. Several studies have shown that histone modification[1] is correlated with origin locations [59, 60, 61]; origins tend to be in open, easily accessible regions of the DNA We hypothesis that the accessibility also affects the rate of initiator loading, and that the first initiator at an origin may load much faster than additional initiators. One way to implement qualitatively this idea would be to enforce a minimum probability for loading at least one MCM2-7 pair. In an extreme case, this probability would be one.

We chose to use the Poisson distribution for setting the number of initiators at an origin for two reasons: First, without experimental evidence to motivate the selection of a complex model, the simple model is preferred. Second, since we are testing the efficacy of the MIM, which assumes constant n, the Poisson distribution represents a worst-case scenario: It includes the possibility of loading zero initiators, and having zero initiators leads to the largest perturbation from the assumption made in the MIM. Therefore, the preparation module selects the number of initiators at origin i from a Poisson distribution defined by the average n_i.

The second task of the preparation module is to assign a firing time to each initiator on the genome. This is different from assigning a firing time to each origin: If there are k origins, then the number of initiators is given by $\sum_{i=0}^{k} N_i = K$. Therefore, K randomly generated firing times are required. The MIM dictates the desired firing time distribution of an initiator, which we derive from the cumulative firing time probability shown in Eq. 2.9

[1] Histones are small proteins that combine to form large, octameric structures. These structures play a role in organizing DNA into its three-dimensional structure in the cell [1].

(and again in Eq. 3.1).

$$\Phi_0(t) = \frac{1}{1 + \left(\frac{t^*_{1/2}}{t}\right)^{r^*}} \ , \tag{3.1}$$

where $t^*_{1/2}$ and r^* are global parameters defining, for a single initiator, the median firing time and the spread in firing times, respectively. Recognizing that Φ_0 goes from zero (when $t = 0$), to one (when $t \to \infty$), we can use inverse transform sampling [62] to randomly generate firing times that reproduce the desired cumulative firing time probability. If we generate u, a uniformly distributed number between zero and one we can transform that to be distributed as desired with

$$F(u) = \frac{t^*_{1/2}}{\left(\frac{1}{u} - 1\right)^{\frac{1}{r^*}}} \ , \tag{3.2}$$

where $F(u)$ is the firing time. This will produce random numbers that exhibit a cumulative probability distribution given by Eq. 3.1. A histogram of 10^5 samples from the transformation coincided satisfactorily with the cumulative fire-time distribution, implying that the method is sound. After all K firing times are generated, the time of the first-to-fire initiator at each origin is kept because each origin can only fire once; thus, the firing time of the origin i is given by the firing time of the earliest initiator $\min\{t_j\}_i$.

3.1.2 The Phantom-Nuclei module

Based on work done by S. Jun *et al.*, the phantom-nuclei algorithm we used in the simulation is a powerful tool for calculating replicative data from a set of parameters describing the origins of replication in the KJMA formalism [35]. Figure 3.2 illustrates the key features of the phantom-nuclei method. There are three major steps in our phantom nuclei module: pre-processing the parameters, simulating replication, and compiling the replicated fraction. In taking these three steps, the phantom nuclei module quickly calculates the regions on the genome of a single cell which have been replicated. These steps have been separated for the sake of clarity; however, there is some overlap between them in our implementation to increase performance.

The strength of the phantom nuclei algorithm is that it pre-processes the origin data it receives. To reduce the amount of work needed to fully simulate the replication process, the program removes origins that are passively replicated ("phantom" nuclei) from the simulation. As we mentioned above, we designed the simulator to loop through many values of t_{sim}. The algorithm starts by calculating the state of replication at the highest value for t_{sim}, $t^{(\text{max})}_{\text{sim}}$. We start at $t^{(\text{max})}_{\text{sim}}$ because that is when every meaningful event will have occurred: origins have fired or not, and every passively replicated origin can be identified.

When pre-processing, the program calculates the positions $\{x_i^{(\text{L})}\}$ of the left forks and $\{x_i^{(\text{R})}\}$ of the right forks originating from all origins i. Calculating these positions is done

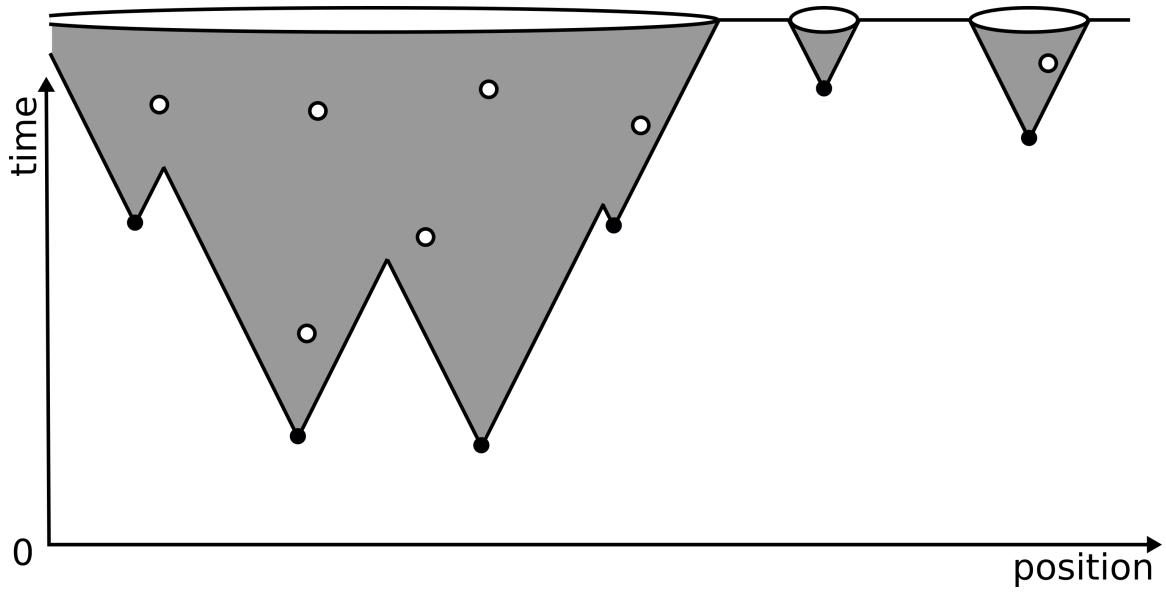

Figure 3.2: Schematic of the Phantom Nuclei algorithm. Only the active origins (Black circles) are considered during simulations. Open circles correspond to passively replicated origins ("phantom" nuclei). The algorithm outputs the replicated fraction, which is one in replicated regions (ellipses at top) or zero in unreplicated regions (black line at top).

via simple kinematics:

$$x_i^{\binom{\mathrm{R}}{\mathrm{L}}} = x_i \pm v \times (t_{\mathrm{sim}} - t_i) \ , \tag{3.3}$$

where the right fork is given by the sum and the left fork by the difference, and where the bracketed term calculates the time since the origin fired. As a part of pre-processing, any origins for which $t_i > t_{\mathrm{sim}}^{(\mathrm{max})}$ are immediately removed from the simulation, as they will not contribute to the replicated fraction. Once the algorithm calculates the set of fork locations, the forks from each pair of neighbouring origins are analyzed to determine which origins are passively replicated. Any phantom nuclei are removed from the simulation (open circles in Fig. 3.2). Pre-processing is finished when only active origins (black circles in Fig. 3.2) are left in the simulation. Pre-processing is computationally expensive, but for complex genomes will dramatically decrease the calculations needed for the second step, and the number of calculations needed for this process on simple genomes is small.

The second step of the phantom nuclei algorithm is taken at every time step. During the simulation step, the algorithm performs three major calculations: First, it selects which origins will fire by comparing their firing times to the current value of t_{sim}; only origins with $t_i < t_{\mathrm{sim}}$ will fire. Second, using Eq. 3.3, the algorithm calculates two sets of fork positions ($\{x_i^{(\mathrm{L})}\}$ and $\{x_i^{(\mathrm{R})}\}$) from the origins selected in the first step. These two sets of fork data are used to define replicated regions on the genome. Third, it analyzes the replicated regions defined by the two sets of fork data, and identifies where replicated regions overlap (i.e.,

coalescence has occurred). Any overlapping regions are combined. This step is analogous to the S phase of the cell cycle, including initiation (selecting cells that fire before t_{sim}), elongation (calculating fork positions), and coalescence (combining overlapping regions), as described in Sec. 1.1.2.

Immediately after any overlapping replicated regions are coalesced, the algorithm compiles the replicated fraction. Therefore, the replicated fraction is compiled at every time step in the simulation. To compile the replicated fraction, the algorithm simply loops through the replicated regions defined by $\{x_i^{(\mathrm{L})}\}$ and $\{x_i^{(\mathrm{R})}\}$ and sets the replicated fraction for the cell to one inside those regions and zero outside. Although this process may sound straightforward, we were unable to do it without nested loops, which significantly slowed the simulation process when coded in the IGOR Pro language. For this reason, this step was written both in IGOR and in C++. When we simulated large data sets early in our work (Sec. 4.1.1), we called the C++ function as an external program. Using this external function increased performance 8 fold.

3.1.3 The Housekeeping Module

The simulation described above calculates the replicated fraction on the entire genome of a single cell. However, we are investigating sequencing data that are acquired by averaging over a large population. Therefore, the housekeeping module is designed to loop over a population calling the preparation and phantom nuclei modules for each cell. The resulting data are then averaged.

A powerful feature of the housekeeping module is that it can be easily modified to analyze and alter the simulated replicated fraction f_{sim}. In Ch. 4, we discuss how we used the housing module to alter f_{sim} by adding noise to best recapture the noise observed in experiment. Additionally, we discuss two different analyses of f_{sim}: We fit the MIM parameters to f_{sim}, and calculated the difference between f_{sim} for Chromosome I and experimental measurements of Chromosome I.

3.1.4 Qualities of the MIM Simulator

The MIM simulator is a powerful tool for generating the replicated fraction of a population of cells with known $\{n_i\}$. The Monte Carlo process used in the MIM Simulator calculates the replicated fraction as the average of a population of cells. Therefore, the larger the population, the better the averaging and the more confident we are in the data. It may seem simple to use the MIM simulator instead of the analytical MIM shown in Sec. 2.3. However, simulating the replicated fraction to the accuracy needed for a fit takes many thousands of single-cell measurements to average over, and this is computationally expensive. By contrast, a single calculation with the analytical MIM will produce the desired fit function.

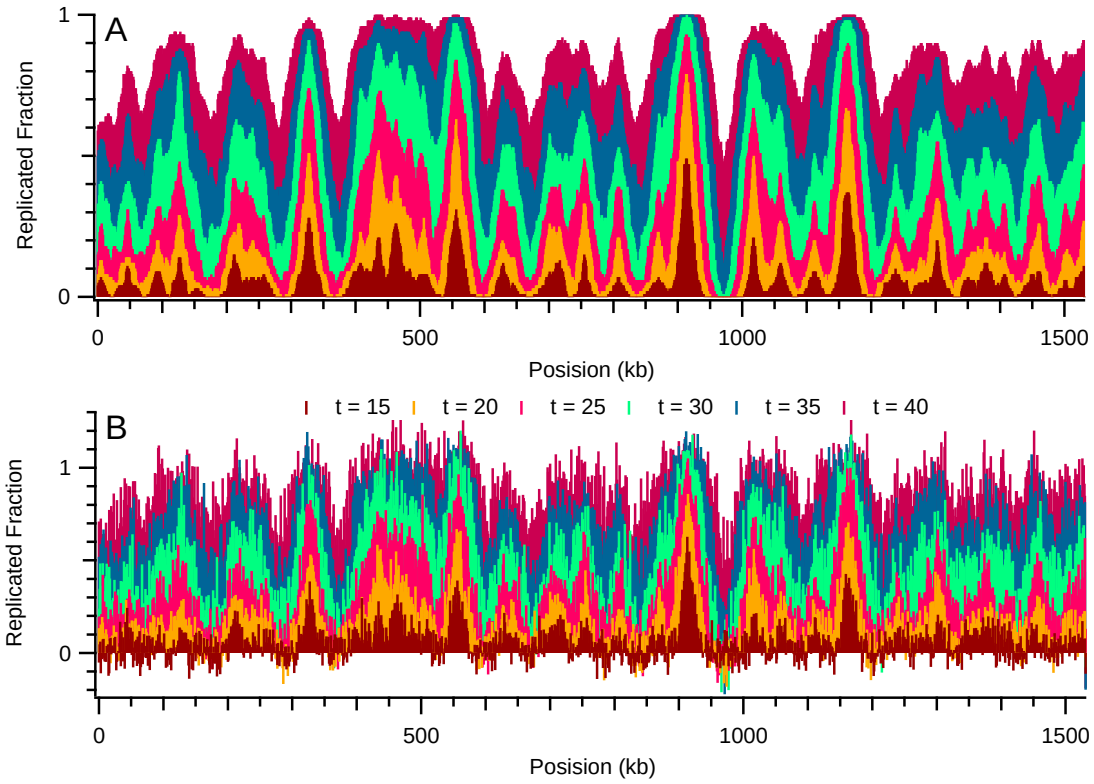

Figure 3.3: Example output of Chromosome IV from the MIM simulator. x-axis is the position in the genome. y-axis is the replicated fraction.
A. Data averaged over 100 cells, no artificial noise.
B. Data averaged over 100 cells, Gaussian noise added according to the procedure outlined in Sec. 3.2.3. Parameters for the simulation were taken from [19] supplementary data.

Thus, the MIM simulator is not a good replacement for the analytical MIM; rather, it is a tool to measure the efficacy of the analytical MIM in the small-n regime.

One of the strengths of our program is its modular structure: It is simple to change the probability distribution of $\{N_i\}$ (currently Poisson distributed) or $\{t_j\}_i$ (currently distributed as described above). Additionally, doing new analysis is simply a matter of creating a new function that the housekeeping module can call.

Figure 3.3 shows two examples of the replicated fraction of Chromosome IV generated by the MIM Simulator.[2] Both simulations were over a population of 100 cells. Figure 3.3A shows data output from the program as described so far. Below, we describe how and why we generated the noisy data presented in Fig. 3.3B

[2] Figure 2.1 shows the replicated fraction of the same chromosome measured with DNA sequencing [9].

3.2 Analyzing Noise in the Data

For our initial investigations, we chose to do simulations of many simple artificial cells. However, as we discuss in Sec. 4.1.2, this was not a good choice because this method does not scale well to complex cells. Because of this problem, we created simulations that were limited in population size and accuracy to reflect current experimental standards. We expected that limiting the population size would increase the noise enough to make a good comparison with experiment. However, as we show below, the experiment has noise beyond that due to finite sample size.

3.2.1 Estimating Experimental Noise

Here, we analyze data from a sequencing experiment investigating the replicated fraction of budding yeast performed by Hawkins *et al.* in 2013 [9]. In their experiment, Hawkins *et al.* used DNA sequencing to calculate the replicated fraction of two strains of budding yeast: wild-type budding yeast and a mutant with three origins of replication removed. Figure 2.1 shows their results for Chromosome IV of the wild-type genome. Notice that the noise in the experiment leads to replication fraction estimates that lie outside the possible range between zero and one. Our goal was to create simulated replicated fraction data that closely resembles data from sequencing experiments. To do this, we need to include noise in our data commensurate with that seen experimentally and must therefore estimate experimental noise.

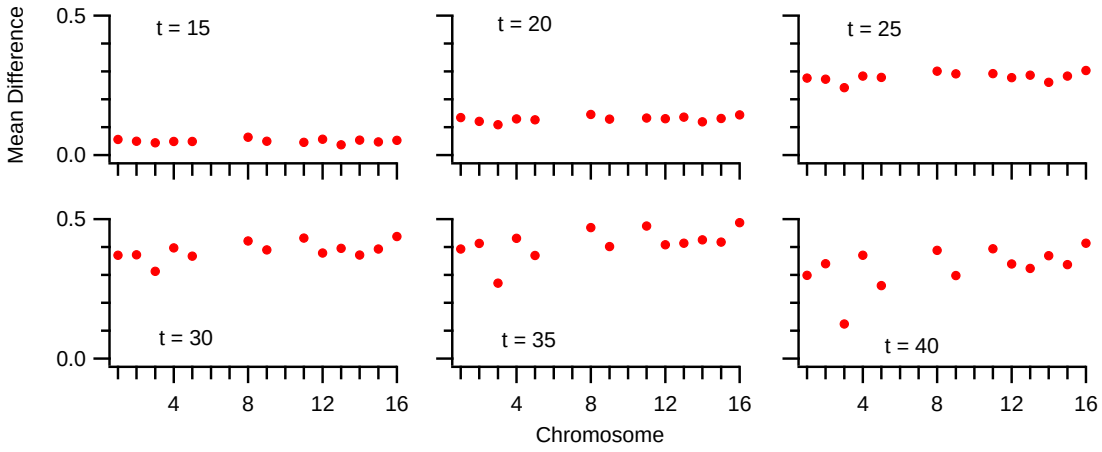

Figure 3.4: Mean point-by-point difference between wild-type and mutant replicated fractions for each chromosome at each time step. Each set of axis is for a different time after the start of S phase (labeled). y-axis shows the mean difference. x-axis is the chromosome label. Note that no data are shown for Chromosomes VI, VII, and X, because they were not analyzed due to their mutations. Data derived from [9] supplementary data.

Following the process used by Yang *et al.* ([19], supplementary material), we analyzed the experimental data to estimate the uncertainty in the measured replicated fraction. Ideally, we would estimate the noise distribution for each data point by analyzing data from an experiment that has been repeated many times. Unfortunately, Hawkins *et al.* did not publish any repetitions of their data set. Therefore, we worked with two measurements we assume to be in close agreement: the wild-type budding yeast and the mutant budding yeast measurements reported in [9]. Since the mutation removed only three origins, we assumed that the replication profiles between the wild-type and mutant measurements would be the same, except on the chromosomes with missing origins (Chromosomes VI, VII, and X). Thus, we compared the remaining 13 of the total 16 budding yeast chromosomes. To estimate the distribution of fluctuations, we considered how the differences between the experiments, calculated point-by-point, were distributed. Figure 3.4 shows the mean difference for each chromosome (x-axis) at each time step (separate axis, labeled). Since the differences vary in time, they are analyzed at each time step separately. Within each time point the fluctuations are much more stable, except for a downward trend in Chromosome III. Thus, in addition to the three chromosomes that were mutated, Chromosome III was removed from our analysis.

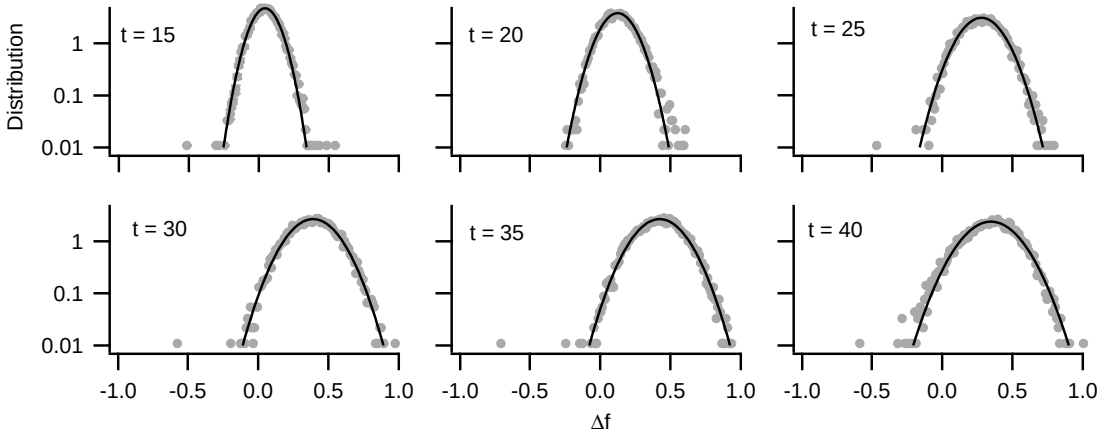

Figure 3.5: Histograms of the point-by-point difference between wild-type and mutant data (Δf) and Gaussian fits. Each set of axis is for a different time after the start of S phase (labeled). y-axis shows the normalized distribution. x-axis shows difference. Grey circles are calculated from experiment [9]. Black lines show the best Gaussian fit. Note that data from Chromosomes III, VI, VII, and X have been excluded (see text). Data derived from supplementary data from [9].

After removing the data from the four chromosomes mentioned, we compiled histograms for the six time steps measured. These histograms (shown in Fig. 3.5) estimate the probability distribution between the two noisy measurements. To properly duplicate the noise of a single experiment, we need the distribution of a single noisy measurement. From

elementary properties of the variance, two independent random variables A and B have $\mathrm{Var}[A - B] = \mathrm{Var}[A] + \mathrm{Var}[B]$. If the two measurements are equally noisy, the standard deviation of the differences is $\sqrt{2}$ times larger than the standard deviation of a single measurement. Our estimates of the noise are shown as open circles in Fig. 3.6.

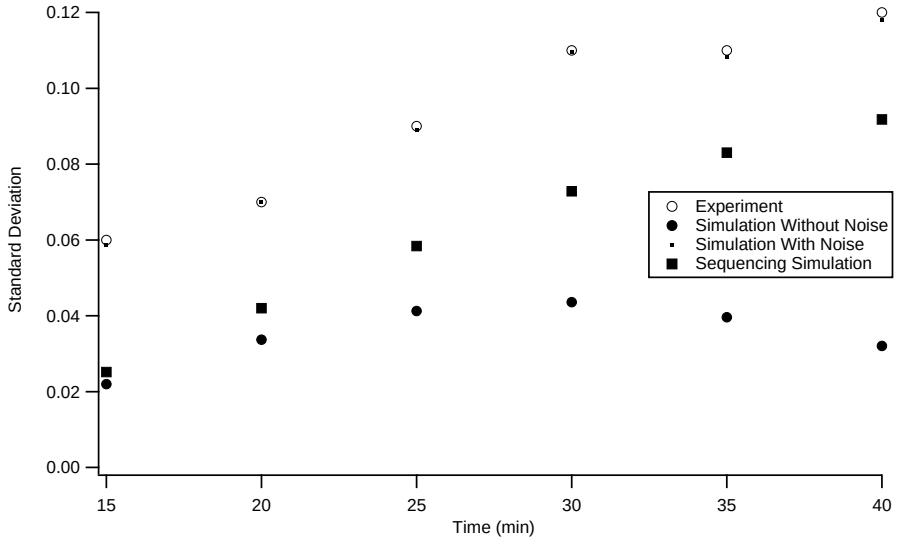

Figure 3.6: Scatter plot of estimated σ vs time since the start of S phase for experimental data and simulation data. Open circles show experimental estimates. Black circles show simulation estimates. Crosses show calculated values for σ_{add} (Eq. 3.4) Squares show estimates from sequencing simulations. Dots show estimates from simulations with added Gaussian noise.

There are three features of the histograms in Fig. 3.5 to note. First, unlike the microarray data that Yang *et al.* analyzed, the histograms extracted from sequencing data are Gaussian distributed. This implies that the data from sequencing experiments are better suited to analysis with the MIM, since the MIM is fit to experimental data assuming Gaussian-distributed noise [19]. Second, the standard deviation evolves as time progresses. This is expected: Early in the replication program and late in the replication program, many of the cells will be mostly unreplicated and mostly replicated respectively. Therefore, we expect that the noise will be diminished at early time and late time. Third, the mean of the Gaussian fits evolve dramatically as time progresses. We believe this is due to a global systematic error in the data, potentially the reported time since the start of S phase, or a possible global effect of the mutation that removes origins from Chromosomes VI, VII, and X.

In addition to measuring the distribution of the point-by-point differences in experimental data, we measured the correlation length. Figure 3.7C shows the autocorrelation of the differences after the mean difference had been subtracted. We observe two features in the autocorrelation function. There is a delta function at $\Delta x = 0$, implying that the noise is

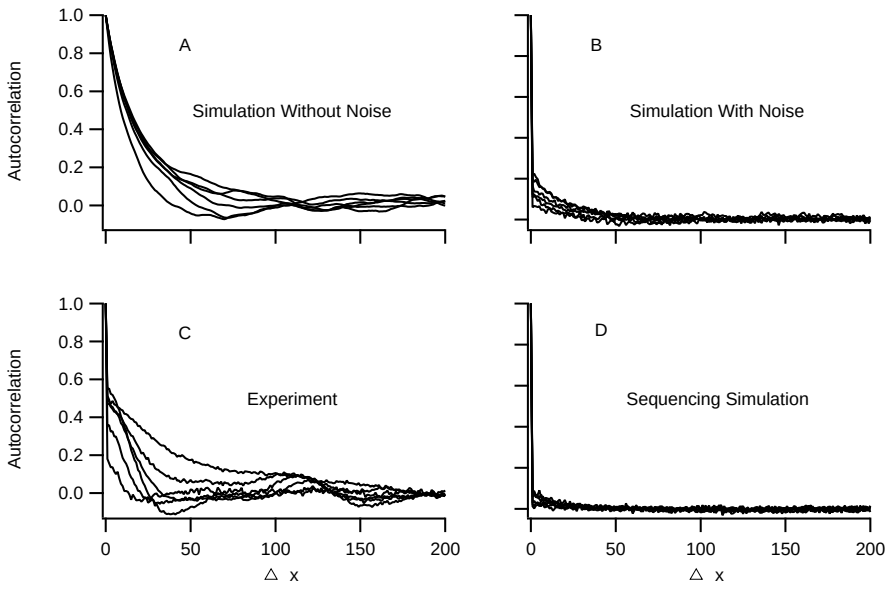

Figure 3.7: Autocorrelation of experimental data and data from three simulations of the full budding yeast genome. Graphs show data from each measurement over time (total 6). No trend over time was observed. **A.** Autocorrelation of simulated data of a population of 100 cells. **B.** Same as **A**, with artificial noise added as described in Sec. 3.2.3. **C.** Autocorrelation of experimental data over the full genome (calculated from [9] supplementary data). **D.** Autocorrelation of simulated data of a population of 1000 cells, taking only one tenth of the genome per cell (100-fold coverage).

uncorrelated at each point. However, there is also a non-negligible tail, implying long-range order in the experimental data. We expect some amount of long-range order due to the mechanics of the replication process: Replication propagates through space in a predictable way (via replication forks). Therefore, $f(x,t)$ directly influences $f(x+\Delta x, t+\Delta t)$, which leads to long-range order.

3.2.2 Estimating Simulation Noise

Now that we have estimated noise level in current sequencing experiments, we would like to use those data to ensure our simulated replicated fraction has noise commensurate with experimental data. Two steps were taken to make this happen: the first based on experimental procedures, the second by artificially adding Gaussian noise.

The first step taken to make our simulated data similar to experimental data was to limit the size of the population of simulated cells. As we mentioned above, the Monte Carlo program operates by taking the average of many cells, which is very similar to sequencing experimental techniques. In their experiment, Hawkins *et al.* extracted 10–25 million 50 bp sequences [9]. Over the genome of 12 Mb, that is equivalent to 50-to 100-fold coverage per

base. Therefore, we limited our simulations to a population of 100 cells to get equivalent coverage. Because we do not know the source of the evolution of the mean difference (shown in Fig. 3.5) and because we want to keep this program relatively simple, we did not attempt to account for this evolution or the possible desynchronization of cells over time.

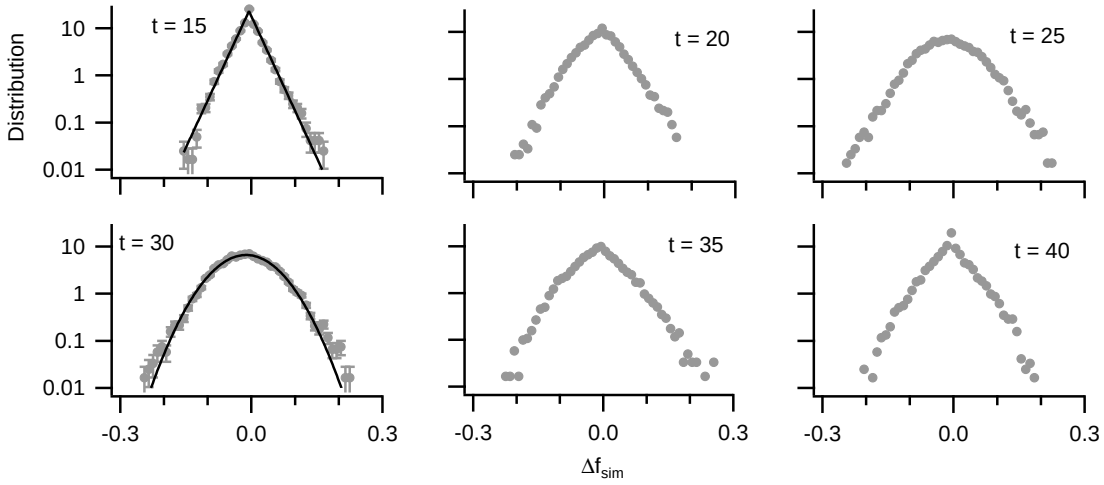

Figure 3.8: Histograms of the point-by-point difference between two sets of simulated data (Δf_{sim}). Each set of axis is for a different time after the start of S phase (labeled). y-axis shows the normalized distribution. x-axis shows difference. Grey circles are calculated from experiment. Black line at $t = 15$ shows the best Laplace distribution fit. Black line at $t = 30$ shows the best Gaussian fit.

We generated two simulated replicated fraction functions over the entire genome from 100-cell populations (parameters were set using results from the MIM [19]). These two functions were used to estimate the noise in the simulation, σ_{sim}, using the same process as outlined in Sec. 3.2.1. Figure 3.8 shows the distributions of the difference between the two sets of simulated data at each time step. There are two noteworthy observations: First, because we chose a simple approach that assumes perfect synchronicity and timing, the mean difference between the two simulations is zero. Second, there is an evolution in the noise from near-Laplace distributed at early time, to near Gaussian, and back to near-Laplace distributed. The Laplace distribution, $P(x) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$, governs the difference between two independent identically distributed exponential random variables [63]. Figure 3.6 shows our estimation of the noise in simulation.

Is the noise in simulations distributed differently from the noise in experiment? To address this concern, we qualitatively investigated a possible source of the noise. We know that the greatest uncertainty in replicated fraction will coincide with the presence of forks of replication: While regions that replicate early and regions that replicate late will simulate a replicated fraction of primarily ones and primarily zero respectively, regions that are in

the process of replicating will return both values. Therefore, we measured the average number of replicated regions across the genome over simulation time (Fig. 3.9). Except when a fork has hit the end of the chromosome, the number of forks is twice the number of replicated regions. We observed a peak in the number of replicated regions, and hence forks, at 30 minutes after the start of S phase. This time coincides with the time at which the distribution of the simulated noise is most Gaussian (Fig. 3.8). Thus, noise in simulation is Laplace distributed when the number of forks is small, and adding more forks makes the distribution more Gaussian. This may be explained by the fact that the time between initiations of early-firing origins will be exponentially distributed (because the exponential distribution describes the time between events that are Poisson distributed [64]), leading to differences that are Laplace distributed. However, when many origins are active, the central limit theorem tells us that the differences will tend toward a Gaussian distribution [65].

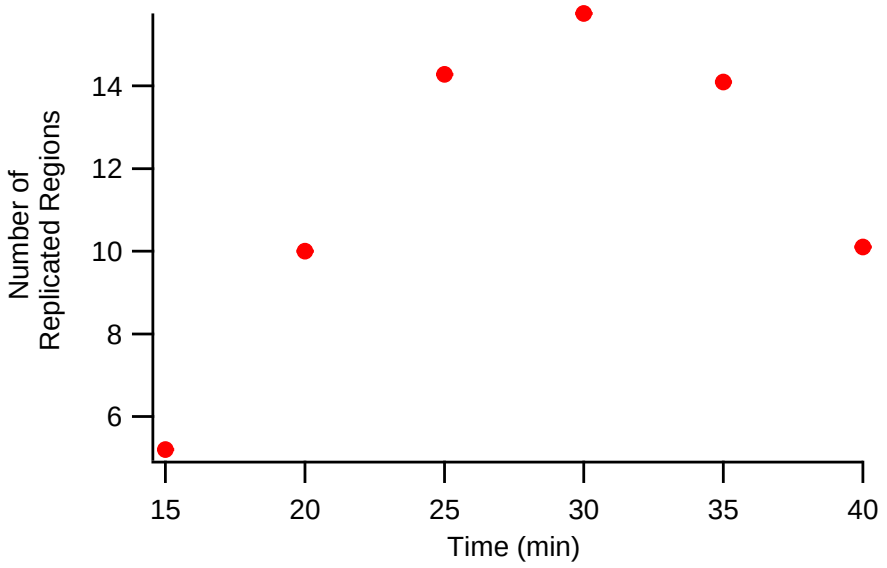

Figure 3.9: Histogram of the number of replicated regions per cell in the simulation. x-axis shows t_{sim} in minutes.

3.2.3 Adding Gaussian Noise to the MIM Simulator

We changed two features of the noise in simulation to better produce noise commensurate with experiments. First, as shown in Fig. 3.6, the statistical noise that arises from the random sampling of the Monte Carlo process is not large enough to match the noise we estimated for the experiment. Second, the experimental data has noise that is uncorrelated (shown by the peak at $\Delta x = 0$ in Fig. 3.7C) at each point that is lacking in the simulation over a population of 100 cells (Fig. 3.7A). Therefore, we added extra uncorrelated noise to the simulated data to match the levels we found in Sec. 3.2.1. To add the noise, we used

our estimate of the uncertainty from the simulation, σ_{sim}, then calculated the amount of Gaussian noise we had to add, σ_{add}, such that the resulting uncertainty matched the desired values:

$$\sigma_{\text{add}} = \sqrt{{\sigma_t}^2 - {\sigma_{\text{sim}}}^2} \ , \tag{3.4}$$

where σ_t is the experimental noise calculated for the simulated time t from experimental data (Sec. 3.2.1). The resulting values of σ_{add} were added to our simulations in the housekeeping module, after the phantom-nuclei module was finished.

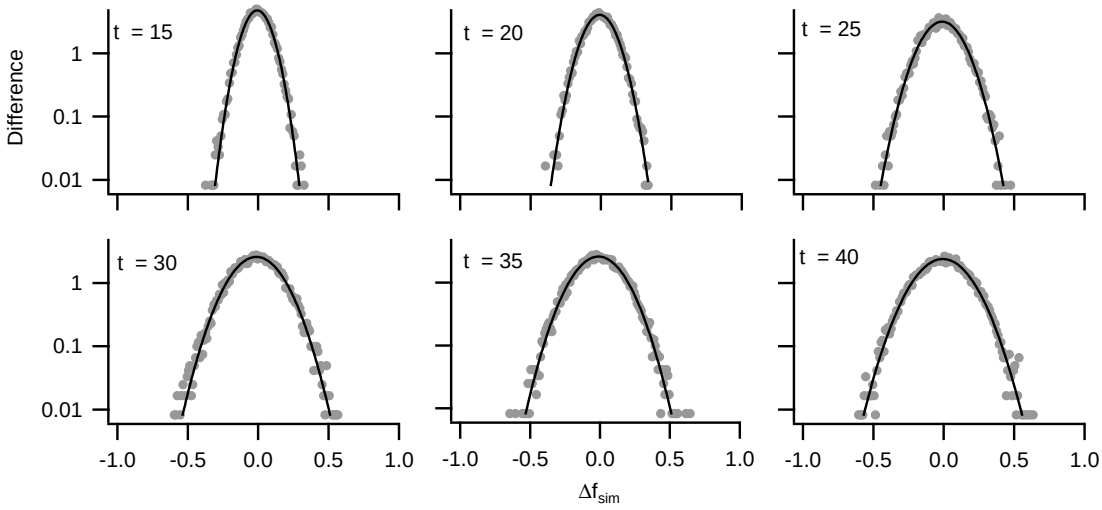

Figure 3.10: Histograms of the point-by-point difference between two sets of simulated data with artificially added Gaussian noise (Δf_{sim}). Each set of axis is for a different time after the start of S phase (labeled). y-axis shows the normalized distribution. x-axis shows difference. Grey circles are calculated from experiment. Black lines show the best Gaussian distribution fit.

Figure 3.7B shows the autocorrelation function for the simulated data with artificial noise. With the addition of the noise, we have acquired the delta function at $x = 0$ but lost much of the long-range order. To understand the effect that creates the long-range order in experiment, and potentially improve our simulation, we tried a second approach to creating noise. This approach, called the "sequencing simulation," simulates 1000 cells, records only one tenth of the data, and does not add any artificial noise. With this method, the simulation is closer to sequencing experiments which sample 50 bp sequences from an effectively infinite population. Analysis of the point-by-point difference shows a similar evolution from Laplace-distributed noise to Gaussian (figure not shown). The estimated standard deviations, shown as black squares in Fig. 3.6, are closer to the experimental estimates than our initial simulation but do not coincide. However, as shown in Fig. 3.7D, the correlation length is shorter in this case than in the simulation with added noise.

In the end, we chose to artificially add noise to a simulation of 100 cells. Adding artificial noise has three main benefits: First, it effectively increases the uncertainty in the simulated data to match that seen experimentally. Second, by adding artificial noise, the distribution of differences between simulations is much closer to the distribution of differences between experiments, with both being approximately Gaussian (compare Figs. 3.5 and 3.10). Third, it is about 10 times faster than simulating 1000 cells and takes one tenth of the data. The estimated standard deviations from the simulations with artificial noise (shown as dots in Fig. 3.6) are within one percent of those estimated from experimental data.

To better include noise in simulations, the two features discussed above need to be addressed. We believe a better understanding of the experimental procedure and its sources of error would help with both of these. Given that the analysis above shows adding Gaussian noise makes the simulation noise match experimental noise much more closely, we believe the presented method is an effective first approach to incorporating noise in the MIM simulator.

Chapter 4

Results

In this chapter, we outline our investigations into the effect of small n on the Multiple Initiator Model. Using the MIM simulator described in Ch. 3, we performed four major investigations.

Our preliminary investigation was a single-origin comparison between the analytical MIM and the MIM Simulator. We defined a parameter that measures the difference between the replicated fraction from simulation and from the MIM. In Sec. 4.1.1, we use this "difference parameter" to show that the fluctuations in absolute numbers of initiators, N_i, due to small average numbers of initiators, n_i, does create a disagreement between the analytical MIM and the MIM simulator proportional to n^{-1}. The presence of this error and its large tail motivated further study into how small n affects the MIM.

In our second investigation, we developed a new metric for measuring the error in the MIM at low n. The difference parameter defined in Sec. 4.1.1 does not scale to more than one origin. Section 4.1.2 outlines the new method which consists of simulating the replicated fraction for a single origin of fixed n, followed by using the MIM to find the value of n that best fits the simulated data. The results of this investigation show that our first approach, while qualitatively in agreement, may overestimate the difference between the MIM and the simulation.

Third, we progressed to simulating and fitting the more complex Chromosome I. We started by fitting parameters with the MIM to data from DNA sequencing [9]. We fit two sets of parameters by fixing $t_{1/2}$ as high and low, forcing the MIM to produce small and high n respectively. Using the parameters from the fits, we then simulated the replication of Chromosome I. By calculating the root-mean-squared difference between simulated data and experimental data, we showed that the two simulations are effectively indistinguishable from each other. Additionally, analysis of the fit parameters shows that the small-n values are proportional roughly to the large-n values.

The surprising results from our third investigation motivated additional work that we use to argue why the MIM is inaccurate for a single origin but produces good chromosome-

wide results. We expanded our single-origin analysis to a genome with two origins and more and show that multiple origins interact to reduce the inaccuracy in inferences made by the MIM.

4.1 Single-Origin Investigations

Our early work consisted of single-origin simulations intended to motivate further study. In 2014, we received the results discussed in Sec. 2.4 from N. Rhind's lab [46]. As mentioned, these results called into question the assumption that the average number of initiators loaded on origin i, n_i, is large enough to ignore variations in the number of initiators loaded on origin i during a particular cell cycle, $N_i^{(j)}$. (Recall that when the MIM was developed, it was assumed that, due to relatively small fluctuations, $P(N_i) = \delta(N_i - n_i)$ was an effective distribution.) Therefore, our first goal was to discover whether or not simulations of small n agreed with the MIM predictions for small n. In this naive investigation, discussed further below, we developed the "difference parameter," a metric to measure the difference between the simulated and predicted $f(x, t, n)$ of a single origin. The investigation shows that the difference parameter decreases as n^{-1} and motivated the deeper research presented in this thesis.

The single-origin investigations that followed our preliminary work consisted of simulating $f(n)$ and fitting the MIM parameters to the result. Thus, our metric changed from the difference parameter to a comparison between the simulated n_{sim} and the fitted n_{fit}. With these investigations, we refined the simulation program to run more efficiently and to create data similar to that measured in sequencing experiments. We show that our new metric reveals a qualitatively similar behaviour for the MIM; the difference between n_{sim} and n_{fit} goes approximately as $n^{-1/2}$. We attribute the change in the rate to the change in how the metric is defined.

4.1.1 The Difference Parameter

When starting this project, we performed a quick investigation into the difference between $f_{\text{sim}}(n)$ and $f_{\text{MIM}}(n)$ for a single origin. To generate $f_{\text{sim}}(n)$, the MIM simulator calculated the average replicated fraction from $\approx 10^6$ sequences, producing data with very little statistical error (in contrast to the noisy data shown in Sec 3.2). The global parameters used for the simulation were set equal to those measured previously by fitting the MIM to microarray measurements of budding yeast [19]. The difference parameter was then defined to be

$$DP = \max_x \frac{\Delta P(x)}{P} \ , \tag{4.1}$$

where $\Delta P(x)$ is the difference between $f_{\text{sim}}(n)$ and $f_{\text{MIM}}(n)$ (illustrated in Fig. 4.1), and P is the peak value of $f_{\text{MIM}}(n)$.

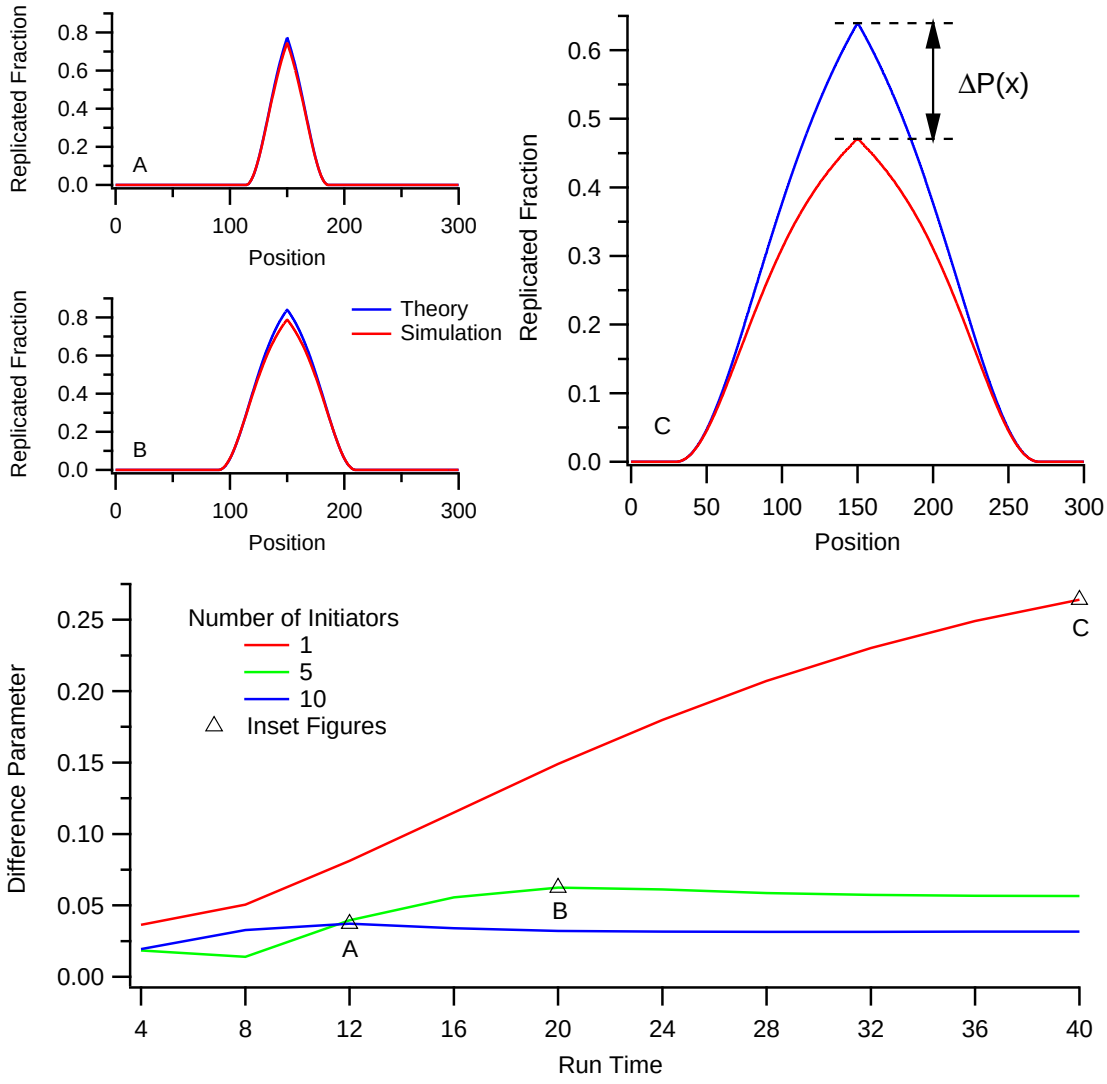

Figure 4.1: Schematic of the difference parameter calculations. Coloured lines represent the difference parameter for different values of n. Triangles show the parameter values in the corresponding inset graphs. **A.** Replicated fraction simulation and theory curves of an artificial genome with a single origin at [position]= 50 kb for $n = 10$; $DP = 0.037$. **B.** Same as **A** for $n = 5$; $DP = 0.062$. **C.** Same as **A** for $n = 1$; $DP = 0.264$. Also illustrated in **C** is the value $\Delta P(x)$ at the peak. Note that $\Delta P(x)$ is defined over the entire domain.

Figure 4.1 shows the analysis process of the difference parameter for a single origin. First, $f_{\text{sim}}(n)$ and $f_{\text{MIM}}(n)$ were calculated over several time steps and n ranging from 1 to 128. Example replicated fractions are shown in the insets of Fig. 4.1. From these data, we calculated $DP(n, t)$, shown in the main graph of Fig. 4.1. Note the value for DP "saturates"

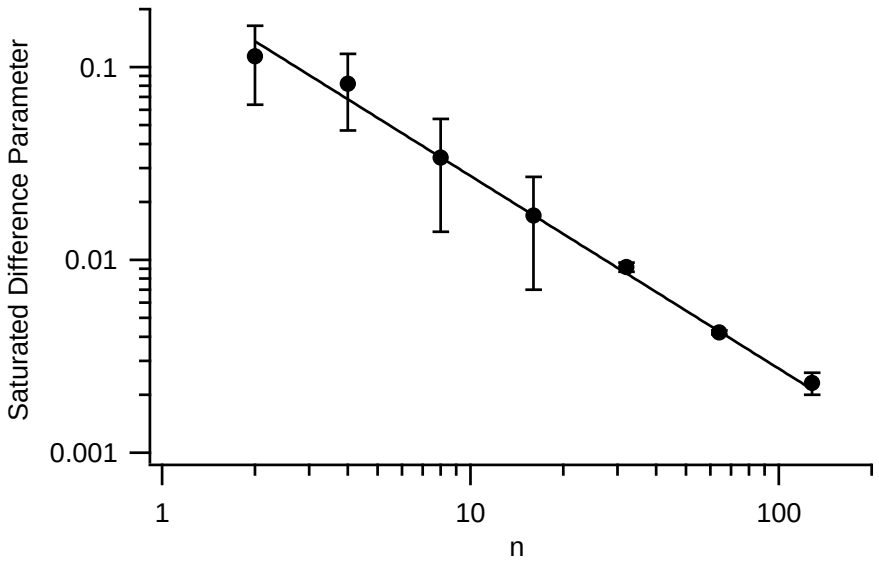

Figure 4.2: Saturated difference parameter vs. n. The large uncertainty in low n arises because the simulation time is too short to accurately measure the saturated difference parameter. The line is proportional to n^{-1} ([Saturated Difference Parameter] $\approx \frac{0.3}{n}$).

as t increases; we call this value the "saturated difference parameter." Figure 4.2 shows[1] our measurements of the saturated difference parameter as a function of n. The large noise for low values of n is due to the difference parameter not saturating before the end of the simulation. This effect can be reduced by increasing the simulation time, however we felt the data presented was strong enough motivation to move forward with our research. For this reason, we did not acquire more precise data for the saturated difference parameter. This initial investigation showed the saturated difference parameter decreases as n^{-1}.

From our initial investigation, we concluded that the difference parameter grows quickly with decreasing n. Therefore, we suspected that the MIM will not produce accurate results in the case that n is small. These results motivated the in-depth research into the effect of small n on the MIM that follows.

4.1.2 Biased Fits

The definition of the difference parameter does not scale to more than one origin. The saturated difference parameter is measurable only when the theory curve has a peak value near one. Additionally, the difference parameter is defined to be a single value for the whole simulated genome; therefore, we cannot infer anything about more than one origin. Thus,

[1] The reader may notice a change in the n values displayed in the graphics. Fig. 4.1, the saturated difference parameter as a function of n, is illustrative but contains old data; the data are accurate, but were not analyzed further. After producing that graph we used a slightly different set of parameters in the simulation and changed the range of n simulated when producing Fig. 4.2.

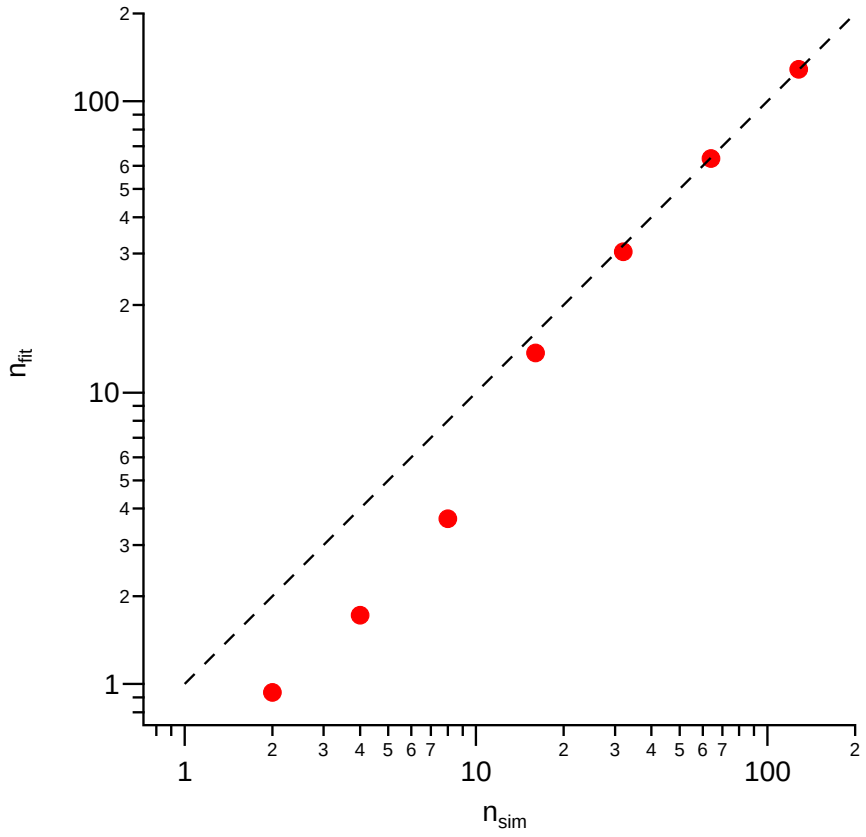

Figure 4.3: Scatter plot of n_{sim} vs. n_{fit} for simulations of a large population. Red circles show the data. Dashed line shows unity.

we developed a second investigation that measures the bias in parameters inferred with the MIM from a simulation of small n.

In this investigation, we first calculated $f_{\text{sim}}(n)$ and then fit the parameters of the MIM to the result. Thus, we have two parameters: n_{sim}, the value for n input to the simulation, and n_{fit}, the value for n that results from the MIM fit. Initially, we performed these measurements using simulations of large populations of sequences (about 10^6 and more 100 kb sequences). Figure 4.3 shows the preliminary results from this investigation on large-population simulations. The graph is a scatter plot of n_{sim} vs. n_{fit}, the dashed line shows unity. As we expected, the bias is relatively large for low n_{sim}, but decreases as n_{sim} grows. However, even using a C++ module to increase performance, these simulations were slow and were far more precise than the current experimental standard (Sec. 3.2). Therefore, we limited our simulations as described in the previous chapter; in this way, we simulated data comparable to those generated experimentally.

In Sec. 3.2, we outlined the process used to generate noisy data. We used the MIM Simulator to generate $f_{\text{sim}}(n_{\text{sim}})$ with noise and used the MIM to fit the parameters to

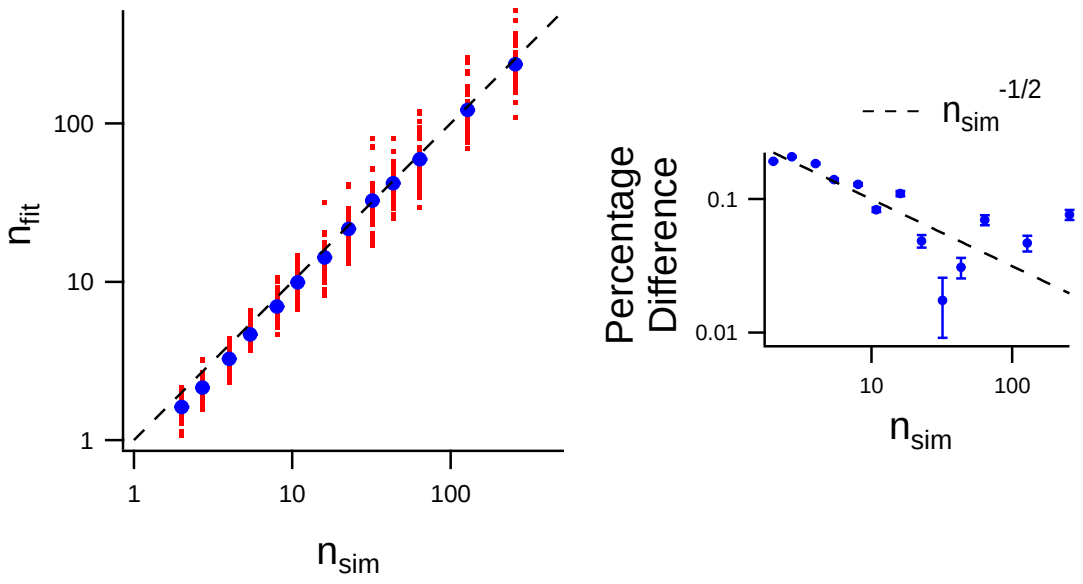

Figure 4.4: Scatter plot of n_{sim} vs. n_{fit} for noisy simulations of a single origin. Red dots show data from 50 simulations at each value n_{sim}. Blue circles with error bars show the mean and standard deviation of the mean. Dashed line shows unity. **Inset.** Scatter plot of the percent difference ($[n_{\text{sim}} - n_{\text{fit}}]/n_{\text{sim}}$) vs. n_{sim}. [Dashed line] $= 0.315/\sqrt{n_{\text{sim}}}$.

that data. Figure 4.4 is a scatter plot of n_{sim} vs. the resulting n_{fit}. In this case, because of the increased noise in the simulated data, we performed this procedure fifty times per n_{sim} value (red dots). The blue circles show the mean for each value of n_{sim}. Here, we see that the bias is largest for small n_{sim}, and decreases as n_{sim} grows, in agreement with our earlier investigation. In the inset to Fig. 4.4, we show the percent difference given by $[n_{\text{sim}} - n_{\text{fit}}]/n_{\text{sim}}$. The dashed line shows $0.315/\sqrt{n_{\text{sim}}}$, but that trend is presented only for comparison with the difference parameter.

The data shown in Fig. 4.4, which comes from fitting the MIM to artificially noisy simulated data, shows the behaviour we expect: The MIM works poorly when n is small, and better when n increases. With these results assuring us the program is sound, we continued our research exploring the same process on genomes with multiple origins.

4.2 Simulations of Chromosome I

Our results from single-origin simulations indicate that the MIM does not perform well in the small-n regime for a single origin. However, in eukaryotes, origins are not alone: as we discussed in Ch. 1, replication in eukaryotes starts at many origins. In this section, we investigate the replicated fraction of Chromosome I and do not observe the same reduction in accuracy. Using the same process as our single-origin investigation, we simulated the

Position (kb)	38.3	72.6	124.2	155.6	174	216
Small n	1.65	1.93	2.3	2.5	5.7	1.5
Large n	10.2	12	14	16	36	9

Table 4.1: High and low fitted values for n on Chromosome I. Top row shows the fixed positions of the six fitted origins. Center row shows the values of n when $t_{1/2} = 40$. Bottom row shows n from a fit with $t_{1/2} = 90$. Small n vs Large n is plotted in Fig. 4.7

replicated fraction of Chromosome I of budding yeast and then used the MIM to infer n_{fit} values at each origin. For this investigation, we used the same simulation method as described above for the noisy single-origin analysis, except that the genome size and origin parameters were chosen to represent Chromosome I.

The origin parameters were fit with the MIM to the replicated fraction for wild-type budding yeast reported by Hawkins *et al.* [9]. To test the effect of small n, the fitted parameter $t_{1/2}$ was fixed at a high value (90 minutes) to produce high values for n and at a low value (40 minutes) to produce low values for n. To be sure that any effects we observed were due only to the fitted values of n, the origin positions were fitted once then held constant for the second fit. The resulting values for n at each origin can be seen in Tab. 4.1.

We simulated the two sets of parameters that resulted from the high-n and low-n fits. The resulting replicated fractions are shown in Fig. 4.5. To quantify the quality of the two fits, we calculated the root mean squared difference Δf_{rms} between each replicated fraction and the experimental data. We simulated each set of parameters fifty times and averaged Δf_{rms} at each time step over the fifty simulations. Figures 4.6A and B show $\Delta f_{\text{rms}}(x,t)$ for high and low n respectively and Fig. 4.6C shows the average value for each time step, $\Delta f_{\text{rms}}(t)$. Surprisingly, and in contrast to the results presented so far, these data imply that the quality of the MIM fits is nearly identical[2] for large and small n.

Further, we were interested in the relationship between the large-n values from the fit and the small-n values from the fit. The MIM does not claim that n is the absolute number of initiators on an origin; rather, the fitted parameter should be proportional to the absolute number of initiators. If this is true, the values for low-n and high-n should be linearly related. Our suspicion, based on the results of our single-origin investigation, was that the relationship would not be linear but go approximately as $n^{-1/2}$. However, our results from fitting to Chromosome I seem to contradict this suspicion. Indeed, Fig. 4.7A shows that the relationship is linear: $n_{\text{small}} \propto n_{\text{large}}$.

Using the same technique as outlined in Sec.4.1.2, we used the MIM to calculate n_{fit} values for each origin that we simulated on Chromosome I. For the six origins we simulated,

[2] We observe a trend for high-n simulation to be slightly more accurate than low-n simulations; however, the error bars show the standard deviation of the mean of 50 simulations. Thus, the trend is well within the noise of a single measurement.

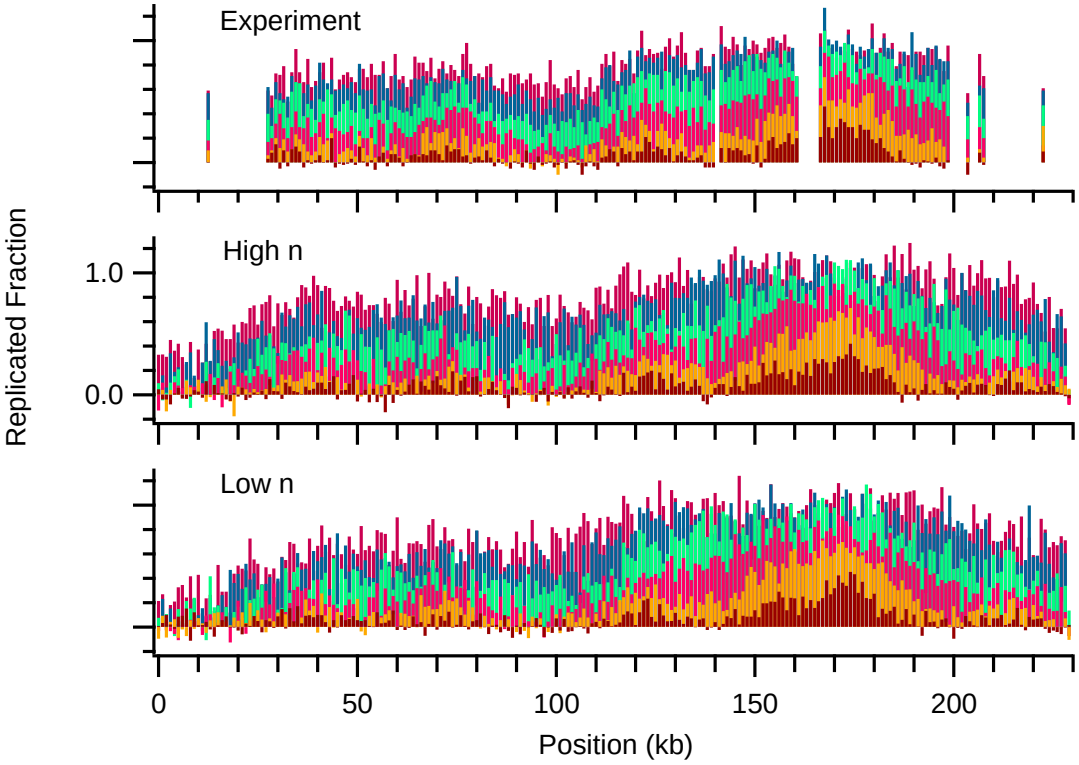

Figure 4.5: The replicated fraction of Chromosome I from experimental data ([9] supplementary data), and simulations with high n and low n.

we calculated the percent difference between n_{fit} and n_{sim} (shown in Figs. 4.8 B and D). The trend we observe in the single origin case of decreasing percentage difference with increasing n is no longer present. Our measurements of Chromosome I indicate that the MIM is approximately equally effective for both high-n and low-n.

We suspected that the spatial organization of the origins in Chromosome I played a role in the observed equality in the fit. Therefore, Fig. 4.7B shows the percentage difference as a function of location within the genome. We do not observe any meaningful pattern in this plot.

4.3 Neighbouring Origins Reduce the effect of Small n

The results shown in Sec. 4.2 appear to contradict the results from the previous, single-origin investigations. What is different between the single-origin simulations and the simulations of Chromosome I? The immediate answer is that the number of origins has changed from one to six. Perhaps then, contributions from multiple origins combine to reduce the effect of fluctuations. Here, we show our investigations of genomes with more than one origin.

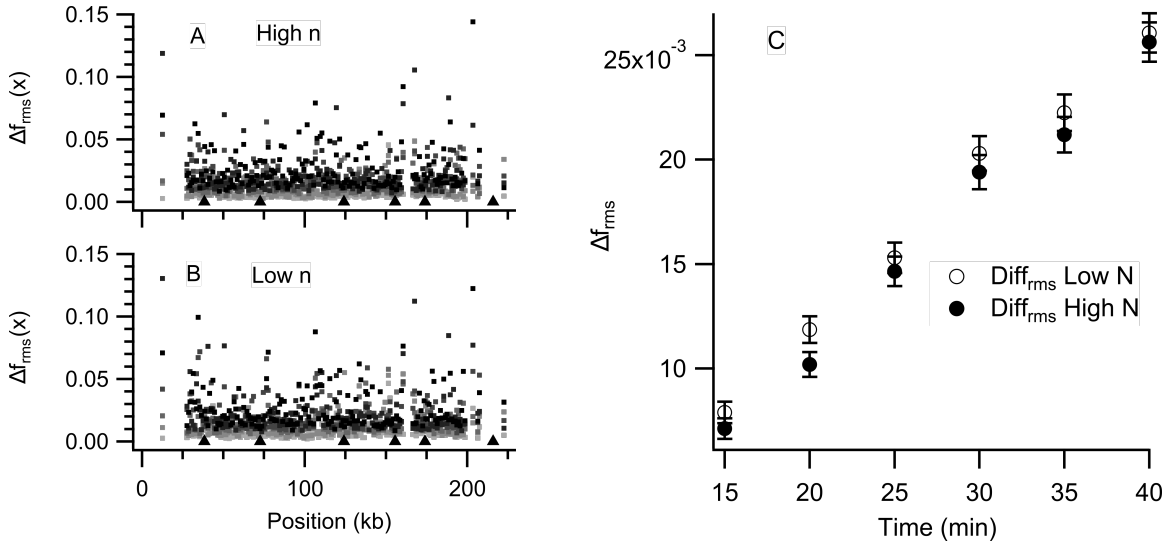

Figure 4.6: Analysis of Δf_{rms}. **A.** Plot of $\Delta f_{\mathrm{rms}}(x)$ for six time-steps between high n simulations and experimental data. The gradient goes from light grey (15 min) to black (40 min). Triangles show the positions of origins. **B.** Same as **A** between low n simulations and experimental data. **C.** Δf_{rms} averaged over the genome vs time since the start of S phase.

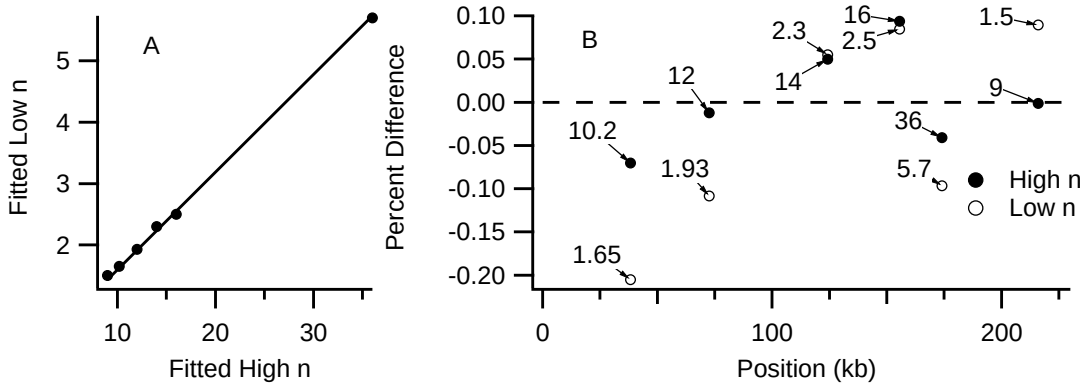

Figure 4.7: **A.** Scatter plot of Low-n fit values vs. high-n fit values. The line shows the best linear fit, $n_{\mathrm{low}} \approx 0.16 \times n_{\mathrm{high}}$. **B.** Scatter plot of the percentage difference shown in Figs. 4.8B and D vs the positions of the origins. Labels show n_{sim}.

4.3.1 Two-Origin Investigation

To test our hypothesis that contributions from multiple origins combine to reduce the effect of fluctuations on the MIM in the small-n regime, we expanded our single-origin investigation to a genome with two origins. With the addition of a second origin, there is a new consideration: By what distance should the two origins be separated? The obvious maximum distance the origins should be separated is $2vt_{\mathrm{sim}}^{(\mathrm{max})}$ (v is the speed of propagation

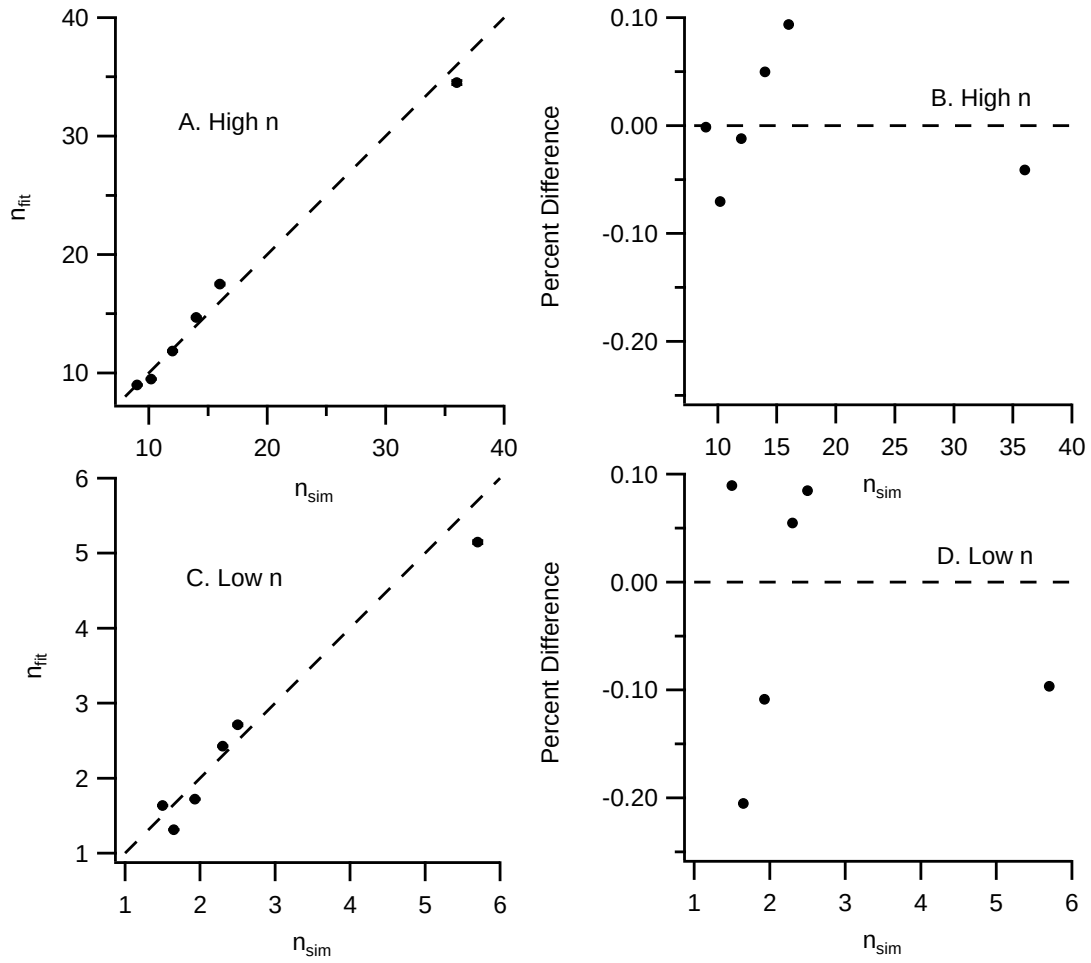

Figure 4.8: n_{fit} vs. n_{sim} and percentage difference for high n and low n simulations of Chromosome I. **A.** Scatter plot of n_{fit} vs. n_{sim} for high n. Dashed line is unity. **B.** Scatter plot of the percentage difference between n_{fit} and n_{sim} vs. n_{sim} for high n. Dashed line is zero. **C - D.** Same as **A** and **B** respectively for low n.

of replication forks) because if they are any further apart, their replicated regions will never overlap. For our simulations, that is 180 kb. However, it is very rare for an origin to fire at the start of S phase, so placing the two origins 180 kb apart is too far. We looked to the fitted locations of origins on Chromosome I shown in Tab. 4.1 acquired by fitting the parameters of the MIM to sequencing data [9] as a guide. Here, we see the two closest origins are 18.4 kb separated, and the two farthest origins are 54.2 kb separated. Therefore, we simulated two origins with equal n_{sim} spaced both 18.4 kb apart and 52.4 kb apart.

Figure 4.9 shows the percent difference between n_{sim} and n_{fit} resulting from these simulations for n ranging from 2 to 64 compared to single-origins simulations. Contrary to our hypothesis, these results show that two origins are fitted less accurately than single origins.

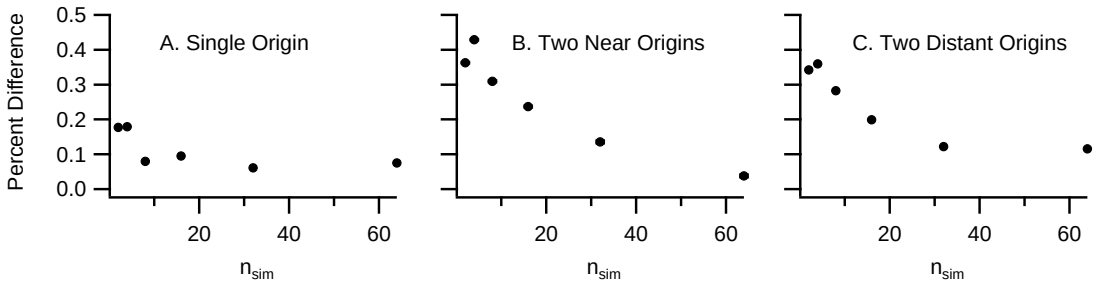

Figure 4.9: Scatter plots of single-and two-origin percent difference between n_{fit} and n_{sim}. **A.** Percent difference vs. n_{sim} for a single origin. **B.** Same as **A** for two near origins (18.4 kb separation) **C.** Same as **A** for two distant origins (54.2 kb separation)

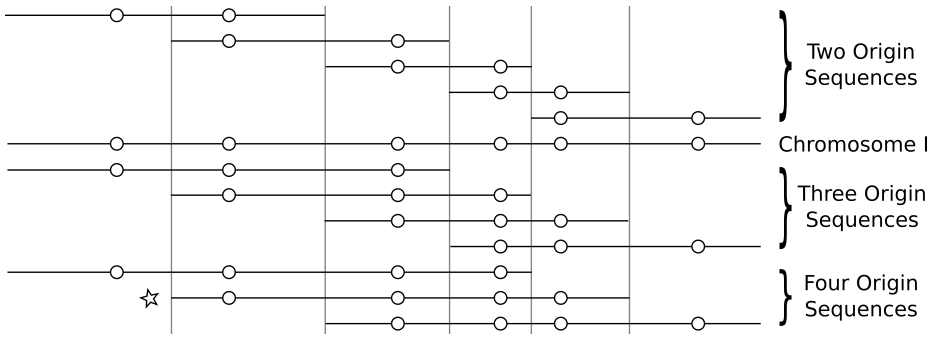

Figure 4.10: Sketch of sub-sequences of Chromosome I used in simulations. Vertical lines denote chosen edge points for sub-sequences. Circles represent origins. Horizontal lines represent sub-sequences (number of origins labeled and the full chromosome (labeled). Star indicates the sub-sequence highlighted in Fig. 4.11.

4.3.2 Multiple-Origin Investigation

The results so far have been surprisingly contradictory: In the single-origin case, the MIM fits get progressively worse as n decreases. In the many-origin case of Chromosome I, the MIM fits are equally accurate for both high n and low n. In the two-origin case, the MIM fits are less accurate than the single-origin case for low n.

To explore the transition from the inaccurate 2-origin system to the accurate 6-origin case, we simulated sub-sequences of Chromosome I containing 2 origins, 3 origins, and 4 origins. Figure 4.10 illustrates how the sub-sequences were selected with divisions occurring directly in the middle[3] of neighbouring origins. There may be better selection criteria for where the endpoints of sub-sequences should fall; for example, the fifth origin has higher n than both of its neighbours; therefore, it will have a larger region of influence than they. However, as the results will show, in addition to its simplicity, this method is effective.

[3] Because of a calculation error, the second division was 4 kb off the midpoint. This should not have a strong impact on the results.

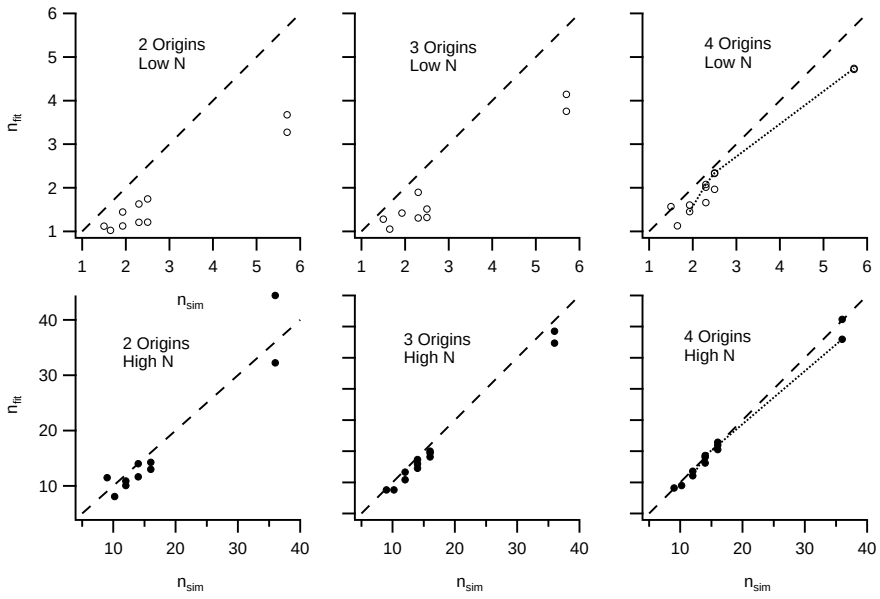

Figure 4.11: Scatter Plots of n_{fit} vs. n_{sim} for two, three and four origins for high n and low n. Open circles show low-n data. Black circles show high-n data. Dotted lines in right-most graphs show a single sub-sequence of four origins. Dashed lines show unity.

We simulated the sub-sequences of Chromosome I using the n values shown in Tab. 4.1. Figure 4.11 shows a scatter plot of n_{fit} vs. n_{sim} for two-origin (left), three-origin (middle), and four-origin (right) sub-sequences (shown in Fig. 4.10) of Chromosome I in the high-n (black circles) and low-n (open circles) regimes. The two dotted lines show the four origins on a single sub-sequence labeled with a star in Fig. 4.10. In this data, we observe two trends: First, as the number of origins in the sequence increases, inferences made with the MIM grow in accuracy. While between high n and low n, the accuracy is still better in the high-n regime, we suspect that as n grows the difference will become negligible. Second, as illustrated by the dotted lines in Fig. 4.11, we observe that origins in the center of the sequence (surrounded by other origins) are inferred more accurately than origins on the edges that have a single neighbour only. These two trends together make inferences made with the MIM accurate for all values of n on chromosomes with many origins.

Chapter 5

Conclusions

In this thesis, we developed and used the MIM simulator to explore the effect of variation in initiation factors on the accuracy of the analytical MIM. We presented several investigations ranging in complexity from simulations of a single origin to simulations of Chromosome I of budding yeast. From our investigations, we concluded that the inferences made with the MIM remain accurate even when the number of initiators is low.

Our research was motivated by a recent experiment measuring low numbers of loaded initiators which contradicts the assumption made in the analytical MIM. Naive interpretations of the MIM suggest that the MIM should fail when the number of initiators is low. We started our investigations by analyzing simple single-origin genomes. The results of these single-origin studies confirmed our suspicions: Inferences made with the MIM become less accurate as the number of initiators decreases. In contrast, simulations of Chromosome I of budding yeast showed that these inferences are accurate when the number of initiators is low. To understand this contradiction, we simulated sequences with multiple origins. From these simulations, we observed two trends that explain the transition from inaccuracy for single origins to accuracy for several origins. First, the overall accuracy of MIM inferences increases with the number of origins. Second, the accuracy is greater for origins in the middle of a cluster of origins than origins at the edges (i.e., origins with two neighbours are handled better than origins with only one). Combined, these two observations mean that inferences made with the MIM on genomes with many origins are equally accurate for any number of initiators.

We conclude that inferences made with the MIM are accurate for small numbers of initiators as long as there are many origins. We have provided a qualitative analysis of multiple-origin simulations that shows that as the number of origins increases, so too does the accuracy. The positive outcome of this research is that we now have increased confidence in the MIM approach to analyzing DNA replication data from experiments.

5.1 Future Considerations

5.1.1 Quantitative Analysis

It is apparent that the next step in this research is to perform a quantitative analysis of how the contribution of multiple origins acts to mitigate the inaccuracy in inferences made by the MIM due to fluctuations in initiation factors. Such research could explore two features of our results: first, a quantification of the accuracy as a function of number of origins and number of initiators; second, an exploration of how origin spacing and relative initiation factors between origins in a sequence affect inferences made by the MIM. A successful investigation along these lines would help future researchers quantify the accuracy of their measurements made with the MIM for a given genome.

5.1.2 Analysis of Other Organisms

The research presented here was based on the model organism *S. cerevisiae*. This was a deliberate choice: because the origins of replication in *S. cerevisiae* are confined to known locations, the complexity of the replication process is significantly reduced. However, as we discussed in Ch. 1, *S. cerevisiae* is a special case, and it is far more common for origins to be located diffusely in a region. Therefore, research expanding the MIM to address stochastically located origins would dramatically increase the number of organisms it can analyze.

5.1.3 Toward a Biological Research Tool

We believe that in the long-term the culmination of this research will be the development of a tool for biological research. If the studies described above of multiple origins and stochastically located origins are successful, the MIM could form the basis of a research tool for DNA replication. This tool would be used by researchers performing studies of DNA replication to quickly make inferences about the number of initiators loaded on the DNA.

5.1.4 Comments About the Simulator

In our investigations, we created a modular program that simulates DNA replication. This simulation makes use of the KJMA framework discussed in Sec. 2.1.2. As we mentioned in our discussion of the KJMA, it is a mathematical framework that has a broad range of applications. Thus, the MIM simulator can address problems described by the KJMA in one dimension with the addition or replacement of modules within the program. Therefore, there are many new and peripherally related avenues of research that can be investigated using our MIM simulator.

Bibliography

[1] H. Lodish, A. Berk, C. A. Kaiser, M. Krieger, M. P. Scott, A. Bretscher, H. Ploegh, and P. Matsudairu. *Molecular Cell Biology* (W.H. Freeman and Company, New York, NY), 6th ed. (2008).

[2] C. Zimmer. *Microcosm: E. Coli and the New Science of Life* (Pantheon Books) (2008).

[3] N. E. Morton. Parameters of the human genome. *Proc. Natl. Acad. Sci. USA* **88**:7474–7476 (1991).

[4] C. Conti, B. Saccà, J. Herrick, C. Lalou, Y. Pommier, and A. Bensimon. Replication fork velocities at adjacents replication origins are coordinately modified during DNA replication in human cells. *Mol. Biol. Cell* **18**(8):3059–3067 (2007).

[5] G. M. Cooper and R. E. Hausman. *The Cell: A Molecular Approach* (Sinauer Associates Inc.), 6th ed. (2013).

[6] J. Herrick and A. Bensimon. Global regulation of genome duplication in eukaryotes: An overview from the epifluorescence microscope. *Chromosoma* **117**(3):243–260 (2008).

[7] D. M. Czajkowsky, J. Liu, J. L. Hamlin, and Z. Shao. DNA combing reveals intrinsic temporal disorder in the replication of yeast chromosome VI. *J. Mol. Biol.* **375**(1):12–19 (2008).

[8] P. K. Patel, B. Arcangioli, S. P. Baker, A. Bensimon, and N. Rhind. DNA replication origins fire stochastically in fission yeast. *Mol. Biol. Cell* **17**:308–316 (2006).

[9] M. Hawkins, R. Retkute, C. A. Müller, N. Saner, T. U. Tanaka, A. P. S. de Moura, and C. A. Nieduszynski. High-resolution replication profiles define the stochastic nature of genome replication initiation and termination. *Cell Reports* **5**(4):1132–1141 (2013).

[10] A. Dutta and S. P. Bell. Initiation of DNA replication in eukaryotic cells. *Annu. Rev. Cell. Dev. Biol.* **13**:293–332 (1997).

[11] S. P. Bell and B. Stillman. ATP-dependent recognition of eukaryotic origins of DNA replication by a multiprotein complex. *Nature* **357**:128–134 (1992).

[12] S. J. Aves. *DNA Replication: Methods And Protocols*, Chap. DNA replication initiation, 3–18 (Humana Press) (2009).

[13] C. Evrin, P. Clarke, J. Zech, R. Lurz, J. Sun, S. Uhle, H. Li, B. Stillman, and C. Speck. A double-hexameric MCM2-7 complex is loaded onto origin DNA during licensing of eukaryotic DNA replication. *Proc. Natl. Acad. Sci.* **106**(48):20 240–20 245 (2009).

[14] S. A. MacNeill. Structure and function of the GINS complex, a key component of the eukaryotic replisome. *Biochem. J.* **425**(3):489–500 (2010).

[15] K. Kamada. The GINS complex: Structure and function. In *The Eukaryotic Replisome: A Guide to Protein Structure and Function*, *Subcellular Biochemistry*, vol. 62 (editor S. MacNeill), 135–156 (Springer Netherlands) (2012).

[16] J. T. P. Yeeles, T. D. Deegan, A. Janska, A. Early, and J. F. X. Diffley. Regulated eukaryotic DNA replication origin firing with purified proteins. *Nature* **519**:431–435 (2015).

[17] H.-P. Nasheuer, R. Smith, C. Bauerschmidt, F. Grosse, and K. Weisshart. Initiation of Eukaryotic DNA Replication: Regulation And Mechanisms. 41–94 (Academic Press) (2002).

[18] C. A. Nieduszynski, S. Hiraga, P. Ak, C. J. Benham, and A. D. Donaldson. OriDB: A DNA replication origin database. *Nucleic Acids Res.* **53**:D40–D46 (2007).

[19] S. C.-H. Yang, N. Rhind, and J. Bechhoefer. Modeling genome-wide replication kinetics reveals a mechanism for regulation of replication timing. *Mol. Syst. Biol.* **6**:404 (2010).

[20] M. L. DePamphilis. Replication origins in metazoan chromosomes: Fact or fiction? *Bioessays* **21**(1):5–16 (1999).

[21] O. Hyrien and M. Méchali. Chromosomal replication initiates and terminates at random sequences but at regular intervals in the ribosomal DNA of *Xenopus* early embryos. *EMBO* **12**:4511–4520 (1993).

[22] R. Pugatch. Greedy scheduling of cellular self-replication leads to optimal doubling times with a log-Frechet distribution. *Proc. Natl. Acad. Sci.* **112**(8):2611–2616 (2015).

[23] S. Cooper and C. E. Helmstetter. Chromosome replication and the division cycle of *Escherichia coli*. *J. Mol. Biol.* **31**(3):519–540 (1968).

[24] J. Bechhoefer and N. Rhind. Replication timing and its emergence from stochastic processes. *Trends in Genetics* **28**(8):374–381 (2012).

[25] C. A. Nieduszynski and A. de Moura. Mathematical modeling of genome replication. *Phys. Rev. E* **86**:031 916 (2012).

[26] A. P. S. de Moura, R. Retkute, M. Hawkinds, and C. A. Nieduszynski. Mathematical modelling of whole chromosome replication. *Nuc. Acids Res.* **38**:5623–5633 (2010).

[27] A. N. Kolmogorov. On the statistical theory of chrystallization in metals. *Izv. Akad. Nauk SSSR. Ser. Fiz.* **1**:355–359 (1937).

[28] W. A. Johnson and R. F. Mehl. Reaction kinetics in processes of nucleation and growth. *Trans. AIME* **135**:416–442 (1939).

[29] M. Avrami. Kinetics of phase change. I General theory. *J. Chem. Phys* **7**:1103 (1939).

[30] M. Avrami. Kinetics of phase change. II Transformation-time relations for random distribution nuclei. *J. Chem. Phys* **8**:212 (1940).

[31] M. Avrami. Kinetics of phase change. III Granulation, phase change, and microstructure. *J. Chem. Phys* **9**:177 (1941).

[32] J. W. Christian. *The Theory of Transformations in Metals and Alloys* (Pergamon Press, New York) (1981).

[33] A. Rényi. On a one-dimensional problem concerning random space filling. *Publ. Math. Inst. Hung. Acad. Sci* **3**(109-127):30–36 (1958).

[34] J. Herrick, S. Jun, J. Bechhoefer, and A. Bensimon. Kinetic model of DNA replication in eukaryotic organisms. *J. Mol. Biol.* **320**(4):741–750 (2002).

[35] S. Jun, H. Zhang, and J. Bechhoefer. Nucleation and growth in one dimension. I. The generalized Kolmogorov-Johnson-Mehl-Avrami model. *Phys. Rev. E* **71**:011 908 (2005).

[36] S. Jun and J. Bechhoefer. Nucleation and growth in one dimension. II. Application to DNA replication kinetics. *Phys. Rev. E* **71**:011 909 (2005).

[37] C. A. Müller, M. Hawkins, R. Retkute, S. Malla, R. Wilson, M. J. Blythe, R. Nakato, M. Komata, K. Shirahige, A. P. S. de Moura, and C. A. Nieduszynski. The dynamics of genome replication using deep sequencing. *Nucleic Acids Res.* **42**(1):e3 (2013).

[38] J. D. Hoheisel. Microarray technology: Beyond transcript profiling and genotype analysis. *Nat. Rev. Genet.* **7**:200–210 (2006).

[39] H. J. McCune, L. S. Danielson, G. M. Alvino, D. Collingwood, J. J. Delrow, W. L. Fangman, B. J. Brewer, and M. K. Raghuraman. The temporal program of chromosome replication: Genomewide replication in clb5Δ saccharomyces cerevisiae. *Genetics* **180**(4):1833–1847 (2008).

[40] B. Futcher. Cell cycle synchronization. *Methods in Cell Science* **21**:79–86 (1999).

[41] L. T. C. Fraça, E. Carrilho, and T. B. L. Kist. A review of DNA sequencing techniques. *Q. Rev. Biophys.* **35**(2):169 (2002).

[42] S. C.-H. Yang. *Modelling the DNA Replication Program in Eukaryotes*. Ph.D. thesis, Simon Fraser University (2012).

[43] J. Shendure. The beginning of the end for microarrays? *Nat. Methods* **5**:585–587 (2008).

[44] A. Goldar, M.-C. Marsolier-Kergoat, and O. Hyrien. Universal temporal profile of replication origin activation in eukaryotes. *PLoS ONE* **4**:e5899 (2009).

[45] O. M. Aparicio. Location, location, location: it's all in the timing for replication origins. *Genes Dev.* **27**(2):117–128 (2013).

[46] S. P. Das, T. Borrman, S. C.-H. Yang, V. W. T. Liu, J. Bechhoefer, and N. Rhind. *Private correspondence* (2014).

[47] A. Goffeau, B. G. Barrell, H. Bussey, R. W. Davis, B. Dujon, H. Feldmann, F. Galibert, J. D. Hoheisel, C. Jacq, M. Johnston, E. J. Louis, H. W. Mewes, Y. Murakami, P. Philippsen, H. Tettelin, and S. G. Oliver. Life with 6000 genes. *Science* **274**:546–567 (1996).

[48] P. Marinangeli, D. Angelozzi, M. Ciani, F. Clementi, and I. Mannazzu. Minisatallites in *Saccharomyces cerevisiae* genes encoding cell wall proteins: a new way towards wine strain characterisation. *FEMS Yeast Res.* **4**:427–435 (2004).

[49] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek. *Spikes: Exploring the Neural Code* (The MIT press, Cambridge, MA) (1999).

[50] G. M. Alvino, D. Collingwod, J. M. Murphy, J. Delrow, B. J. Brewer, and M. K. Raghuraman. Replication in hydroxyurea: it's a matter of time. *Mol. Cell Biol.* **27**(18):6396–6406 (2007).

[51] N. Rhind and D. M. Gilbert. DNA replication timing. *Cold Spring Harb. Perspect. Biol.* **5**:a010 132 (2013).

[52] J. L. Bowers, J. C. Randell, S. Chen, and S. P. Bell. ATP hydrolysis by ORC catalyzes reiterative mcm2-7 assembly at a defined origin of replication. *Mol. Cell* **16**(6):967–978 (2004).

[53] R. Cowan. *Stochastic Processes: Modelling and Simulation*, Chap. 4. Stochastic models for DNA replication, 137–166 (Elsevier, Boston, MA) (2003).

[54] P. J. Park. ChIP-seq: advantages and challenges of a maturing technology. *Nat. Rev. Genet.* **10**:669–680 (2009).

[55] H.-Y. Lian, E. D. Robertson, S.-I. Hiraga, G. M. Alvino, D. Collingwood, H. J. McCune, A. Sridhar, B. J. Brewer, M. K. Raghuraman, and A. D. Donaldson. The effect of Ku on telomere replication time is mediated by telomere length but is independent of histone tail acetylation. *Mol. Biol. Cell* **22**(10):1753–1765 (2011).

[56] L. Zou and B. Stillman. Assembly of a complex containing Cdc45p, replication protein A, and Mcm2p at replication origins controlled by S-phase cyclin-dependent kinases and Cdc7p-Dbf4p kinase. *Mol. Cell Biol.* **20**(9):3086–3096 (2000).

[57] M. Elrod-Erickson and C. O. Pabo. Binding studies with mutants of Zif268. Contribution of individual side chains to binding affinity and specificity in the Zif268 zinc finger-DNA complex. *J. Biol. Chem.* **274**(27):19 281–19 285 (1999).

[58] N. J. Giordano and H. Nakanishi. *Computational Physics* (Pearson Education Inc., Upper Saddle River, NJ), 2nd ed. (2006).

[59] M. Vogelauer, L. Rubbi, I. Lucas, B. J. Brewere, and M. Grustein. Histone acetylation regulates the time of replication origin firing. *Mol. Cell* **10**:1223–1233 (2002).

[60] F. Chiani, F. D. Felice, and G. Camilloni. Sir2 modifies histone H4-K16 acetylation and affects superhilicity in the ARS region of plasmid chromatin in Saccharomyces cerevisiae. *Nuc. Acids Res.* **34**(19):5426–5437 (2006).

[61] H. K. MacAlpine, R. Gordan, S. K. Powell, A. J. Hartemink, and D. M. MacAlpine. *Drosophila* ORC localizes to open chromatin and marks sites of cohesin complex loading. *Genome Res.* **20**(2):201–211 (2010).

[62] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge University Press), 3rd ed. (2007).

[63] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. 55 (Courier Corporation) (1964).

[64] J. C. B. Cooper. The Poisson and exponential distr. In *Mathematical Spectrum*, vol. 37 (editor D. W. Sharpe) (Applied Probability Trust) (2005).

[65] J. A. Rice. *Mathematical Statistics and Data Analysis* (Duxbury Press), 3rd ed. (2006).