TOOLS FROM STATISTICAL PHYSICS FOR SYSTEMS BIOLOGY AND FOR GENOMICS

by

ADEL DAYARIAN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Physics and Astronomy

Written under the direction of

Anirvan Sengupta

and approved by

____

____

____

____

New Brunswick, New Jersey

OCTOBER, 2010

ABSTRACT OF THE DISSERTATION

TOOLS FROM STATISTICAL PHYSICS FOR SYSTEMS BIOLOGY AND FOR GENOMICS

By ADEL DAYARIAN

Dissertation Director:
Anirvan Sengupta

My graduate studies involved three broad classes of problems, each of which are presented in different chapters of this thesis. The first two parts of my work were related to studying dynamics of biochemical networks. I studied a mean-field/stochastic model of epigenetic chromatin silencing in yeast. The model gives rise to different dynamical behaviors possible within the same molecular model and provides qualitative predictions that are being investigated experimentally. In another part of my work, I studied a model of segment polarity network in *Drosophila* and analyzed the parameter space of the system. I particularly studied the relation between the geometry of parameter space and the robustness of the network. I will show that, in addition to the volume, the geometry of this region has important consequences for the robustness and the fragility of a network. A major part of my PhD work involved applications of high-throughput sequencing technologies for extracting information at the genomic level. I present SOPRA, a new algorithm for exploiting the mate pair information for assembly of short reads. I have successfully applied SOPRA to real data and were able to assemble scaffolds of significant length with very few errors introduced in the process.

Acknowledgements

First, I would like to thank my collaborators, all of whom I found very pleasant to work with. The work on the robustness of biochemical networks (chapter 1) was done in collaboration with Madalena Chaves and Eduardo Sontag. The modeling of the epigenetic silencing (chapter 2) was first started by Mohammad Sedighi, the previous student of my adviser. The related experimental work was done by Viji Nagaraj. I should add that, since this project involved going back and forth between modeling (or reasoning) and experiments, many of the experiments had to be performed several times in different conditions. Our progress would not have been possible without Viji's patience. The work on developing SOPRA (chapter 3) was performed in collaboration with Todd Michael.

I would like to thank my adviser, Anirvan Sengupta, who not only directed me in all of the above projects, but also always encountered my questions and engaged me in discussions in an enthusiastic fashion.

I am indebted to my committee members, Eva Andrei, Andrew Baker and Gyan Bhanot. I always found them to have a positive attitude towards me, something very valuable, specially during the early phase of my research. I am very grateful to Gustavo Stolovitzky, who kindly accepted to be the external referee on my thesis defense committee.

At the end, I would like to acknowledge several useful discussions with Allan Haldane, Randall Kerstetter, Pankaj Mehta, Ariella Sasson, and David Sidote.

Dedication

To my mother.

Table of Contents

Chapter 1
Robustness of Biochemical Networks

The concept of robustness of regulatory networks has received much attention in the last decade. Robustness, in the context of biological networks, broadly indicates that the system remains viable under different perturbations. Defining robustness in a precise form is a challenging task, given that robustness to different kinds of perturbations, e.g., environmental variation, intrinsic fluctuations in chemical networks or changes due to mutations, might involve different features of an existing network [1, 2]. In this chapter, we are concerned with the robustness of functionality to changes in the kinetic parameters for a given network architecture.

Understanding the robustness of predictions of a biochemical network model to the choice of parameters is important for two reasons. One reason concerns fitting of biological data and making predictions. We need to know which combinations of parameters are strongly constrained by data and also which combinations seriously affect a particular prediction. The other reason has to do with understanding biochemical network evolution. If the functioning of the network requires fine-tuning in many parameters then mutations causing changes in regulatory interactions could quickly make the network dysfunctional. We expect naturally evolved networks to be somewhat robust to such perturbations.

One measure of robustness has been associated with the volume of the feasible region, namely the region in the parameter space in which the system is functional. Our point of view is that, in addition to volume, the geometry of this region has important consequences for the robustness and the fragility of a network. I will present an analysis of the segment polarity gene network to illustrate our approach. Our method provides a more complete measure of robustness to parameter variation. In addition, as a general modeling strategy, our approach is an interesting alternative to Boolean representation of biochemical networks.

1.1 Introduction

Many organisms (like humans) begin life with one cell and proliferate until there is a full functioning multicellular organism. An interesting question is: how it is possible to begin life with one cell but end up with many different cell types (muscle, blood, nerve, etc.)? There are two parts to this question. The first aspect of this question is related to how different cell types are robustly generated starting from one cell (zygote) and going through several rounds of cell divisions. The other part has to do with maintaining the pattern of cell fates, once it is generated. Such heritable locking of different cells into different fates without irreversible change in genetic information is called an *epigenetic* phenomenon.

In this chapter, we will analyze the segment polarity network in *Drosophila* (fruit fly), one of the best studied system for cell differentiation during development. In the next chapter, our focus will shift to studying a mechanism of epigenetic silencing. This section includes the basic biological background necessary to follow the subsequent materials. A more through presentation can be found in biology textbooks [3, 4].

1.1.1 Gene regulation

The genes encoded in the sequence of DNA contain the instructions necessary to build their associated proteins. This process is accomplished via an intermediate step called transcription. During transcription, the genetic information is used to produce a molecule called mRNA. In ribosomes, mRNA is used as a template to build protein. This last step is referred to as translation:

$$\text{DNA} \xrightarrow{\text{Transcription}} \text{mRNA} \xrightarrow{\text{Translation}} \text{Protein}$$

At each moment, in each cell, only a fraction of the genes are actively transcribed, and the rest are inactive or silenced. Generally speaking, the known mechanisms of heritable gene silencing fall into two categories:

- mechanisms involving transcription factor (TF) networks,
- mechanisms involving reversible modification of DNA or histone.

The second kind of mechanism usually affects a particular contiguous locus of the genome, and distinct loci can often be silenced independently. Such epigenetic mechanisms are the subject of the next chapter. In TF networks, genes from different loci could interact with each other through diffusible gene products. For example, genes have nearby sequence signatures which can bind regulatory proteins like TFs, and affect whether a gene is active or not. Different genes may have different TF binding sites and be regulated by different groups of TF proteins.

1.1.2 Fly development

The origin of cell differentiation in *Drosophila* is believed to start during egg formation. The developing egg is not homogeneous. Instead, it is maternally deposited with *bicoid* and *nanos* mRNA at the anterior and posterior parts of the embryo, respectively. After about 1 hour, translation of these mRNAs leads to protein concentrations with a gradient along the egg. Both bicoid and nanos are transcription factors, namely, each of them activates or represses a new set of genes. In this way, maternally deposited bicoid and nanos initiate a cascade of gene families, where each of the families initiates the expression of the next family (Figure 1.1). In addition, members of each family affect the expression of other members as well. At each step of the cascade, various members of the gene family are expressed inhomogeneously along the embryo. In other words, their expression forms a particular pattern. This pattern gets more and more complex and fine-tuned as the cascade progress.

The above procedure is an example involving one of the main concepts in development biology, the so called morphogens. It refers to the idea of having molecules that are unevenly distributed across a field of cells. The local concentration levels of these molecules are read by the cells, producing a specific phenotype.

The second family in the cascade, after the maternally deposited genes, is the *gap* genes (Figure 1.2). Mutation of gap genes produces large gaps, or missing parts, in the embryo body pattern. The next family is the *pair-ruled* genes, which are expressed in periodic bands. Pair-rule genes have complex promoters. A promoter is a combination of control points (TF binding sites) which decide the activity of a particular gene. The promoters of pair-ruled genes allow them to read out the coordinates from the completely aperiodic patterns set by Bicoid and the gap genes, and produce seven stripes of expression. For example, the second stripe of *even-skipped* (Figure 1.1) is activated by Bicoid and repressed by Kruppel and another gap gene product called Giant (Figure 1.2).

The next set of genes, arising from the initial queue of the pair-rule genes, is the *segment polarity* genes. As the fly develops, the expression of pair-rule genes disappears, making it necessary for the segment polarity system to hold on to its pattern. In the next part, I present the important segment polarity genes and the biological knowledge of how they interact with each other.

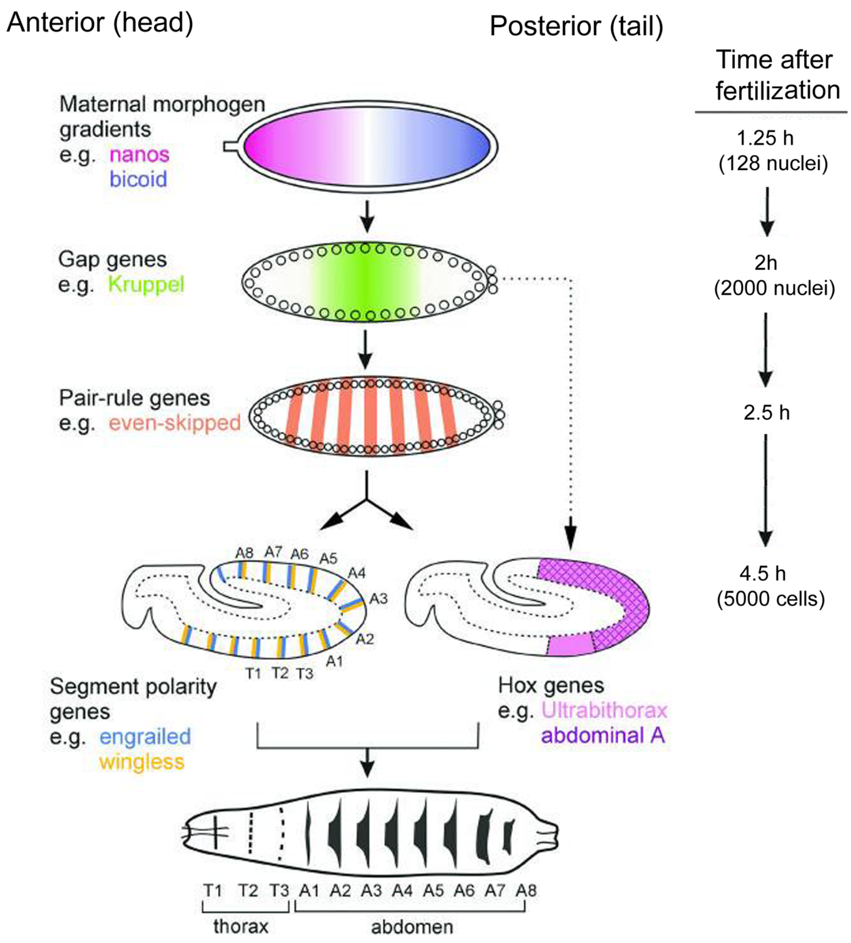

Figure 1.1: Pattern formation along the Anterior/Posterior axis of the *Drosophila* embryo. The inhomogeneously deposited maternal genes initiate a cascade of transiently expressed gene families. In each family, the spatial concentration of different genes results in a pattern which is more complex and refined compared to the pattern produced by the previous family. Before the expression of bicoid and nanos fade away, they activate the *gap* genes. Transiently expressed gap genes start the expression of *pair-rule* genes. The *segment polarity* genes and the *Hox* genes are activated by the pair-rule genes, but a subset of gap genes also influences directly the Hox genes. Both segment polarity and Hox genes affect the differentiation of each segment in the future larva. Adapted from [5].

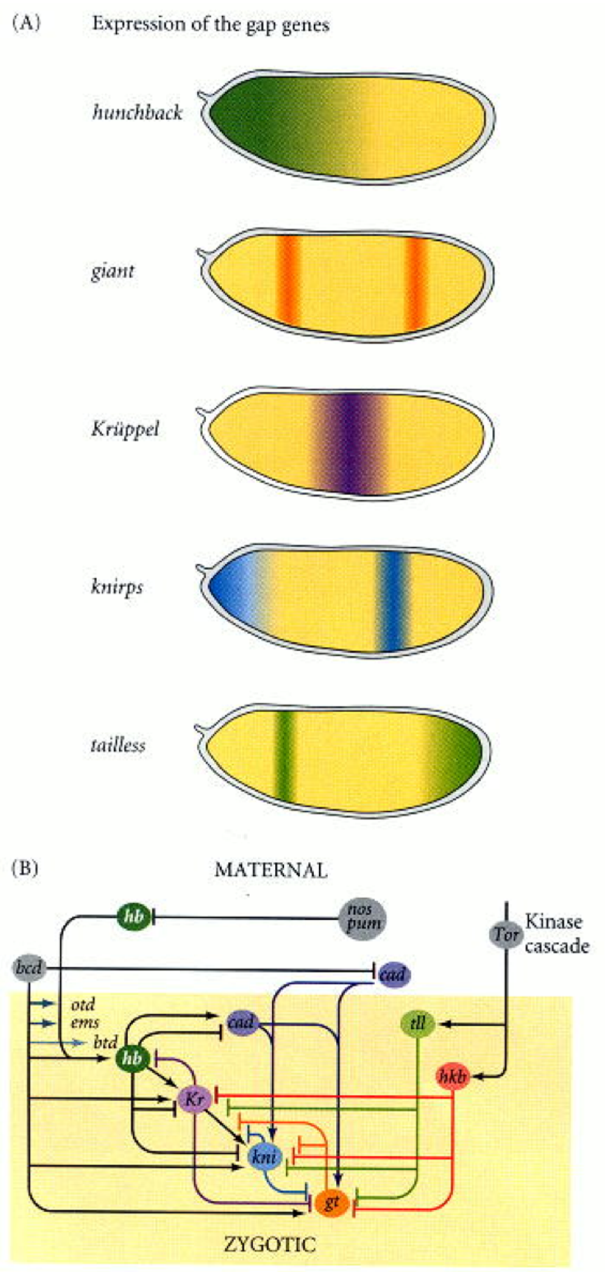

Figure 1.2: Gap genes (From [6].). A) The expression pattern along the embryo. B) The network of interaction among gap genes and the input from bicoid (bcd) and nanos (nos).

1.1.3 Segment polarity gene family

In the segment polarity pattern, genes are expressed periodically in 14 parasegments along the fly embryo. Each parasegment consists of four stripes of cells. Because of this periodicity, one could focus only on one parasegment (i.e. 4 cells). Figure 1.3 shows the gene expression pattern for two of the key components: *Wingless* (*WG*) and *Engrailed* (*EN*).

Wingless is a signaling molecule which diffuses to its neighboring cells and activate Engrailed. *EN*, itself a transcription factor, in turn triggers the production of another signaling molecule, *Hedgehog* (*HH*). *HH* gets secreted to the neighboring cell and maintains *WG* expression by stabilizing an activator of *wg*, called Cubitus interruptus (*CI*). In summary, it is known that *WG* and *EN* maintain the expression of each other through cell-to-cell communication. In this manner, we end up with a repeated four-cell pattern of (*WG*, *EN* and *HH*,none, none).

Although the above explanation superficially seems satisfactory, it leaves room for the following questions. Why is *EN* expressed only posterior to the *WG* expressing stripe? The anterior cell also receives *WG* signal but does not produce EN. Similarly, one could ask why *WG* is expressed only anterior to the *EN* expressing stripe. It is not clear why one should not get a pattern like (*WG*, *EN* and *HH*, *WG*, *EN* and *HH*). , which has a periodicity of two cells. Although, looking back, this problem is obvious, it was pointed out by von Dassow et al. [7], who made a mathematical model of the system. We will study their model in the next section. However, before that, we will give an overview of our point of view on the subject of robustness of biological networks.

1.1.4 Robustness

In modeling biological systems, it is common to be in a situation where much of the available data is of a qualitative nature. For example, one might only know

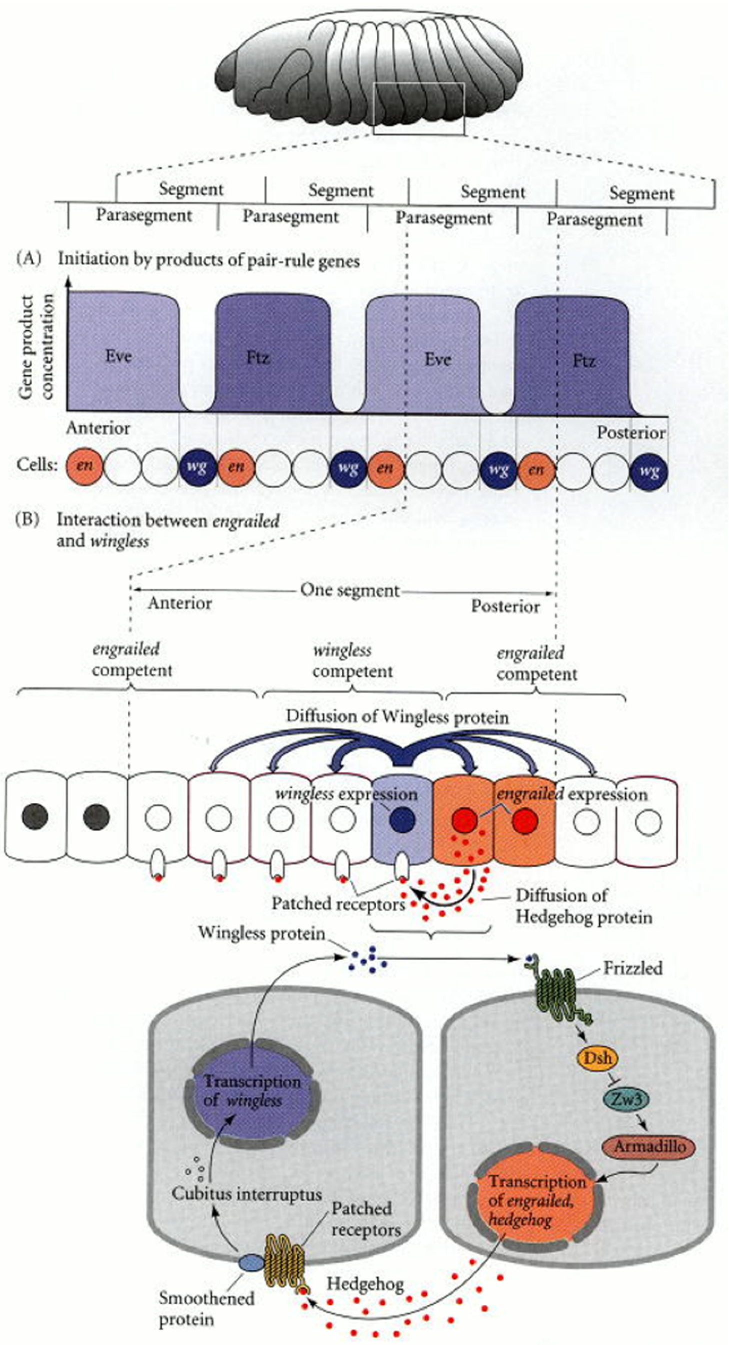

Figure 1.3: The segment polarity system. The segment polarity pattern emerges from such a cascade of events. Interactions amongst the segment polarity genes should maintain this pattern as the embryo grows through cell division. From [6].

that the expression of a certain gene is necessary for the expression/repression of another gene. However, the exact quantitative relations and the values of the parameters necessary to describe the associated dynamical system are not available. Still, the unavailability of the parameters or incompleteness of biological facts does not stop one from creating a quantitative model. Such a model will have a set of unknown parameters. One can associate a parameter space to the model where each point of this space corresponds to a set of values for the parameters.

The first question one would want to ask is whether there is any set of parameters for which the model explains all the known data. Are the values of the parameters for those sets within a reasonable range? Investigation of such questions may lead to suggestion of new interactions necessary to explain the data. In this chapter, we will be involved with answering questions of this kind. Another way in which mathematical modeling could be useful is to provide qualitative predictions. An example would be to predict the effect of mutations or change in the concentrations of certain molecules in the environment. In turn, such predictions can guide us in planning more refined experiments. In fact, this is going to be the situation that we face in the next chapter, while studying a mechanism for *epigenetic silencing*.

As we mentioned, when the value of the parameters are unknown, one can look for the region in the parameter space for which the system is functional. We will call this region the *feasible region*. The motivation for the study presented in this chapter is to analyze the relation between the robustness of a biological network and the shape of its feasible region in the parameter space. Robustness, in the context of biological networks, broadly indicates that the system remains viable under different perturbations. One measure of robustness has been associated with the volume of the feasible region. For example, in an influential study of the *Drosophila* segment polarity network (SPN), robustness has been associated to the fractional volume of the region in parameter space associated with the wild type[1] gene expression pattern [7]. Our point of view is that the geometry of the feasible region contains additional information on essential aspects of robustness and the fragility of the network.

In the context of fitting biochemical kinetic models to time series data, investigators have looked at effects of small parametric perturbations on the quality of the fit. Sensitivity analysis [8, 9], namely considering the effect of changing parameters one at a time, is common practice by now. Brown and Sethna have looked at correlated changes of parameters and study how moving in different directions in parameter space affects the predictions [10]. Based on the eigenvalues and the eigenvectors of the Hessian of the cost function at the minimum, these authors and their collaborators find that, for many known biochemical networks, only a few directions in parameter space have stiff constraints, whereas the rest of the directions are ‘sloppy’ [11, 12]. In this chapter, we will consider the example of SPN and explicitly characterize the region in parameter space where the network could be functional. The anisotropy in the shape of this feasible region will become apparent from our analysis.

If a functional biological system is represented by a point in the feasible region of parameters, then a mutation causes the system to jump to a new point. If this new point also belongs to the feasible region, the system is robust with respect to that mutation. Otherwise the mutation is deleterious. If the jump in parameter space caused by a mutation is relatively large, then the result of successive mutations is to quickly probe different regions of the parameter space. In this case, robustness essentially depends on the volume of the feasible region. On the other hand, if the jumps in parameter space are relatively small, evolution of parameters due to successive mutations can be represented by a random walk in parameter space. The idea of representing evolution as a continuous random process has already been used in the adaptive landscape approach [13]. In this

[1] Wild type means the typical form of a species as it occurs in nature.

case, random walk exiting the feasible region in the parameter space corresponds to a deleterious mutation. The exit time distribution is very sensitive to the shape of the feasible region. Robustness to mutation is now related to the features of this distribution (e.g. half-life, asymptotic decay rate,..) [14] and therefore depends upon the shape and not just the volume of the feasible region.

If we want to choose a single measure for robustness, the inverse of the asymptotic decay rate is a good candidate [14]. This measure is sensitive to the geometry (both volume and shape) of the feasible region. For example, even if the total volume of the feasible region is relatively large, the existence of "narrow" directions will greatly affect the decay rate; or if the feasible region is constituted of several disconnected part, the decay rate will again be affected. In addition, this rate is independent of the initial conditions. Also, in the theoretical case where every mutation leads to a new uncorrelated point in parameter space, the inverse of the asymptotic decay rate is a simple function of the fractional volume of the feasible region.

In our study, we will estimate the half-life, a different but closely related measure of robustness. If the probability of remaining in the feasible region were given by a single exponential in time, these two measures of robustness would be proportional to each other. In practice, half-life depends partially on short time properties of the system and is initial condition dependent. On the other hand, measuring the asymptotic decay rate accurately for a high dimensional stochastic system needs more computational effort than estimating half-life.

Before we go on, let us explain what measure of distance we use when we talk about narrow or wide directions in the parameter space. If we consider the continuous random walk approximation to parameter evolution, then the short-time properties of diffusion set up a metric for the space of parameters. The metric tensor of this space is the inverse of the covariance matrix of infinitesimal displacements divided by the infinitesimal time interval. Once we have this metric, we could decide whether, from a generic point, the distance to reach the boundary in a certain direction is relatively small or large. This definition of distance is closely tied to the time the system typically takes to diffuse over a certain separation.

Once we characterize the feasible region in parameter space, we explore how the system fails as a result of such a random walk. For two alternative network models, we compare the exit time distributions. More importantly, we can see how, in a particular model, certain directions in the space of admissible parameter sets are narrower than others. These narrow directions are associated with the predominant modes of failure of the system in the random walk process. We end by speculating how these methods could be extended to generic biochemical network problems.

1.2 Mathematical Analysis of Segment Polarity Network

The segment polarity network (SPN) is part of a cascade of gene families responsible for generating the segmentation of the fruit fly embryo. Genes involved in initiating this pattern are transiently expressed, and interactions among the segment polarity genes should maintain and fine-tune this pattern as the embryo grows through cell division. Our goal in this section is to study how the interaction between segment polarity genes maintains their expression. Much of the information about this network comes from genetic analysis and is, therefore, of qualitative nature. In particular, we do not know many of the parameters necessary to describe this dynamical system. This is a common situation faced in modeling most biochemical networks.

In their work on modeling the SPN, von Dassow et al. [7] encountered the same problem. Their approach was to solve an ODE model of the network for random choices of parameters and then score the resultant expression patterns based on compatibility with the experimentally observed wild type pattern. If this score is found to be above a certain threshold, the given parameter combination is said to belong to the feasible region of the parameter space. Robustness of a particular architecture is then ascertained by the fractional volume of the feasible region, estimated from their simulation. Ingolia [15] looked at a set of criteria for bistability in particular submodules of the network and studied the extent to which these criteria describe this feasible region. In general, providing an approximate description of the structure of feasible region, even for a medium size biochemical network, remains an important challenge.

One could also get some insight into the functioning of the network by constructing a model where each gene or gene product is mostly ON or OFF. For example, in the context of this particular network, Boolean models have been employed to study dependence upon initial state or the effect of deletion of particular components [16]. Unfortunately, addressing the questions related to parameter dependence is not possible within the conventional Boolean framework. Therefore, we develop a new approximation, within which the treatment of our model shares the simplicity of Boolean analysis without sacrificing the possibility of exploring parameter dependence issues. This approximation enables us to explicitly characterize the feasible region in the parameter space of the model.

In the wild type segment polarity pattern, genes are expressed periodically in 14 parasegments along the fly embryo, and each parasegment consists of four stripes of cells. Because of this periodicity, one can focus only on one parasegment or in other words only on 4 cells. Figure 1.4A shows the wild type gene expression pattern for three key components of the SPN. For simplicity, each cell is assumed to have four faces, rather than six as in the original model [7]. When using abbreviated names for components of the network, we use uppercase letters to refer to proteins and lowercase letters for the corresponding mRNAs. As we mentioned in the previous section, Wingless (*WG)* is a signaling molecule known experimentally to activate Engrailed (*EN*) through cell-to-cell communication. *EN*, itself a transcription factor, in turn triggers the production of Hedgehog (*HH*), which is another signaling molecule. *HH* then gets secreted to the neighboring cell and maintains *WG* expression by stabilizing an activator of *wg*, called Cubitus interruptus (*CI*). Without *HH* signaling, *CI* gets proteolytically cleaved, leaving only its amino terminus (denoted by *CN*), which becomes a repressor of *wg*. In summary, experimentally it is known that *WG* and *EN* maintain the expression of each other through cell-to-cell communication. We represent the wild type expression pattern of these mRNA components as follows:

$$wg^{WT}_{1,2,3,4} = (0,1,0,0)\ ,\quad en^{WT}_{1,2,3,4} = (0,0,1,0)\ ,\quad hh^{WT}_{1,2,3,4} = (0,0,1,0)\ , \tag{1.1}$$

where the four entries of each of the vectors correspond to the gene expression in the four cells of a parasegment. The value ‘0’ means the gene is turned off and the value ‘1’ means it is maximally expressed.

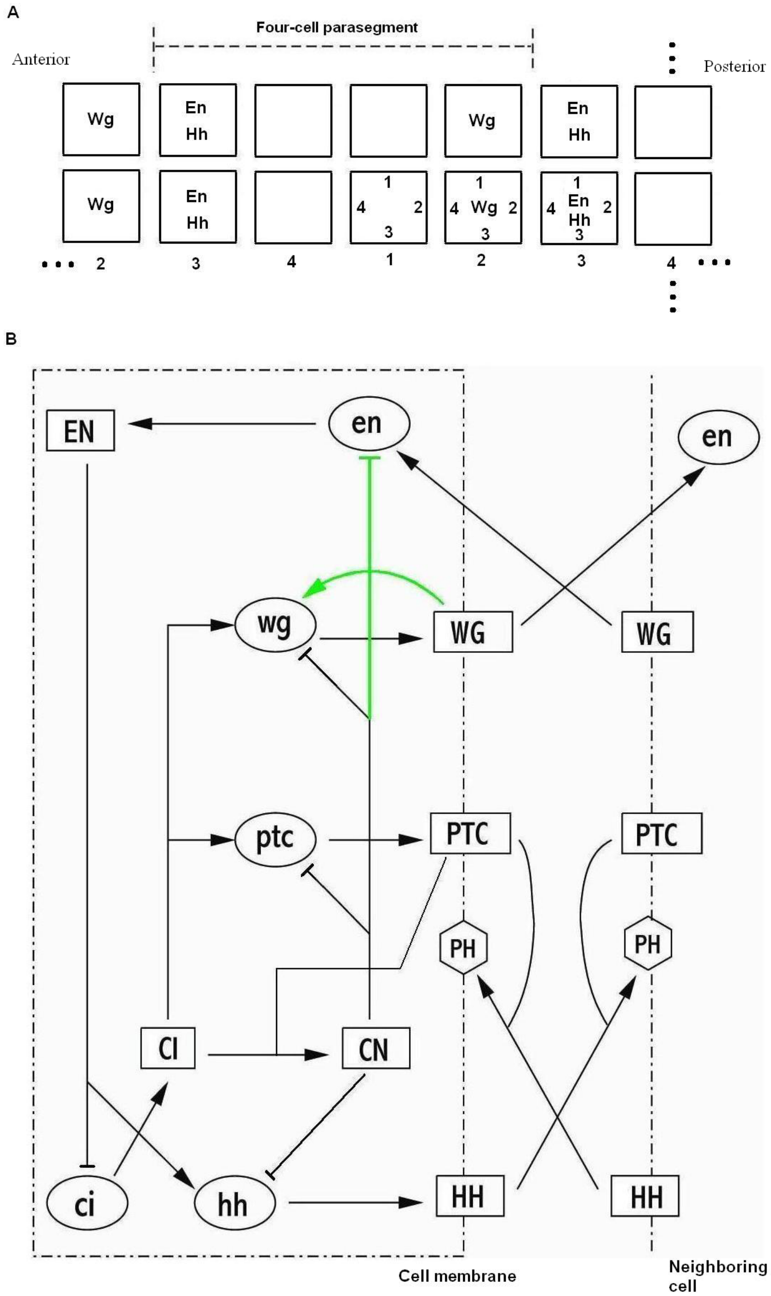

Figure 1.4: Expression pattern for key segment polarity genes and the interaction network. A) Four cells in a parasegment with periodic boundary conditions in both dimensions. Each cell is represented by a square. The convention for numbering cells and cell faces is shown. B) Interaction network used in reference [3]. Two green lines indicate interactions added by authors to achieve the target pattern. Black lines indicate interactions based on experimental data. The shape of the node indicates the corresponding component: Ellipses represent mRNAs; rectangles, proteins.

The above mentioned explanation has to have some missing pieces, as highlighted in the following questions. Why is *EN* expressed only posterior to the *WG* expressing stripe? The anterior cell also receives *WG* signal but does not produce *EN*. Similarly, one could ask why *WG* is expressed only anterior to the *EN* expressing stripe.

Figure 1.4B shows the interaction network used in reference [7]. The authors started only with interactions shown by black lines but were unable to reproduce the right pattern in their simulations. The best pattern the authors could achieve, using only black lines, was an alternative expression of *wg* and *en* in all cells. Therefore, authors decided to add two new interactions shown with green lines. With these links in place, they were able to find many parameter combinations to reproduce the target pattern. The wild type expression pattern for various components of the network is shown in Figure 1.5.

To explore the dependence of robustness of the network on its topology, Albert and Othmer [16][10] developed a Boolean model of the segment polarity network, a discrete logical model where each species has only two states (OFF or ON), but no kinetic parameters need to be defined. This Boolean model is amenable to various methods for systematic robustness analysis [17, 18, 19, 16]. Unfortunately, the ease of analysis comes at the cost of not being able to address questions related to the parameter dependence. We propose an approach which retains the information about kinetic parameters, but, at the same time, keeps part of the simplicity of a Boolean model by having most genes either in the fully ON or the fully OFF state. The detail of our treatment is presented in the following part.

1.2.1 Step function approach to the SPN model

Our strategy for analyzing the problem is as follow. First, we will solve the algebraic equations coming from the steady state conditions and write the steady state solutions in terms of the parameters. Since one of the steady state solutions

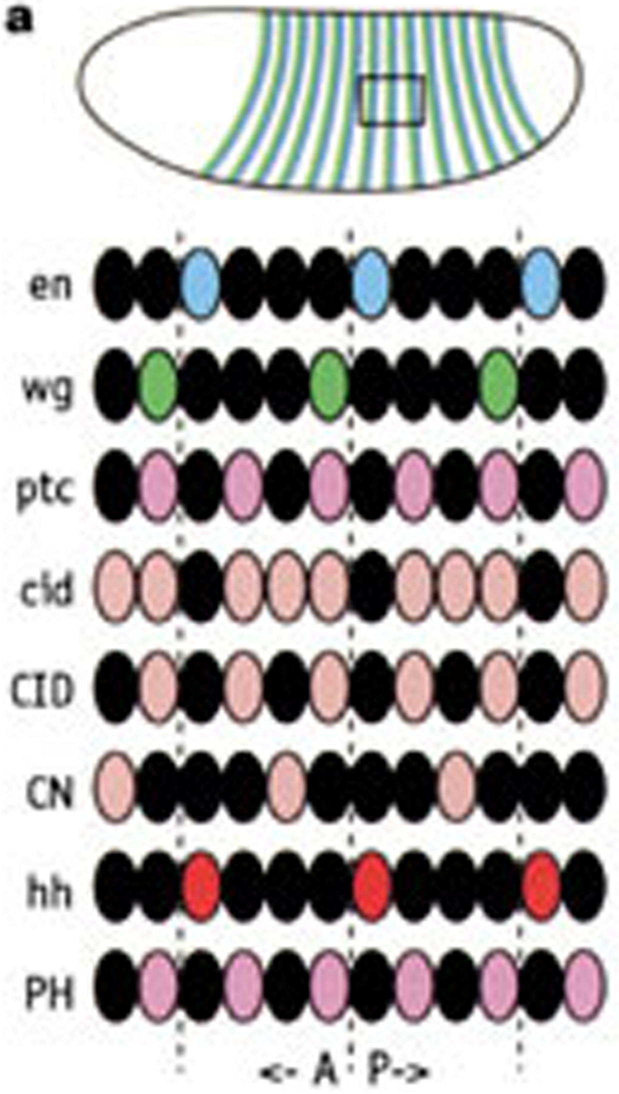

Figure 1.5: The wild type gene expression pattern for various components of the segment polarity network. If a gene is not expressed in a cell, the cell is in black. Adapted from [7].

should match the wild type pattern, one can look for the constraints on parameters that yield this pattern. This procedure provides a family of conditions defining regions of feasible parameters for the wild type steady state. One thing that we ignore is that although all of the parameters in the feasible region can maintain the desired pattern, we do not check whether the system can reach the wild type pattern from particular initial conditions.

In our analysis, we used the fact that many of the differential equations in the model involve terms of the Hill form:

$$\phi(\chi, \kappa, \nu) = \frac{\chi^\nu}{\kappa^\nu + \chi^\nu} \ ,$$

where χ is the concentration of some species, κ is the dissociation constant and ν is the Hill coefficient. The steepness of the Hill function is characterized by the Hill coefficient ν. As χ increases from zero and passes the threshold κ, the function ϕ has a transition from OFF to ON state. For moderately large Hill coefficient, this transition becomes quite steep, and ϕ is practically insensitive to the actual value of ν. In the model presented in reference [7], ν is indeed found to be often quite large, between 5 to 10 [20]. Any such term may thus be replaced by a step function with two levels:

$$\phi(\chi, \kappa, \nu) \to \theta(\chi - \kappa) = \begin{cases} 0 & \text{if} \quad \chi - \kappa < 0 \ , \\ 1 & \text{if} \quad \chi - \kappa > 0 \ . \end{cases}$$

Using this, the steady state gene expression is characterized by the following equations:

$$wg_i = \frac{\alpha_{CIwg}\theta(\mathrm{CI}_i - \kappa_{CIwg})\theta(\kappa_{CNwg} - \mathrm{CN}_i) + \alpha_{\mathrm{WG}wg}\theta(\mathrm{IWG}_i - \kappa_{\mathrm{WG}wg})}{1 + \alpha_{CIwg}\theta(\mathrm{CI}_i - \kappa_{CIwg})\theta(\kappa_{CNwg} - \mathrm{CN}_i) + \alpha_{\mathrm{WG}wg}\theta(\mathrm{IWG}_i - \kappa_{\mathrm{WG}wg})} \tag{1.2}$$

$$\mathrm{IWG}_i = \frac{H_{\mathrm{IWG}} r_{\mathrm{endo}}}{1 + H_{\mathrm{IWG}} r_{\mathrm{exo}}}\,\mathrm{EWG}_{i,T} + \frac{1}{1 + H_{\mathrm{IWG}} r_{\mathrm{exo}}}\,wg_i \tag{1.3}$$

$$\begin{aligned}\frac{\mathrm{EWG}_{i,j}}{H_{\mathrm{WG}}} &= \frac{1}{4} r_{\mathrm{exo}}\mathrm{IWG}_i - r_{\mathrm{endo}}\mathrm{EWG}_{i,j} + r_M(\mathrm{EWG}_{n,j+2} - \mathrm{EWG}_{i,j}) + \\ &\quad r_{LM}(\mathrm{EWG}_{i,lr} - 2\mathrm{EWG}_{i,j})\end{aligned} \tag{1.4}$$

$$en_i = \theta(\mathrm{EWG}_i - \kappa_{\mathrm{WG}en})\,\theta(\kappa_{CNen} - \mathrm{CN}_i) \tag{1.5}$$

$$\mathrm{EN}_i = en_i \tag{1.6}$$

$$hh_i = \theta(\mathrm{EN}_i - \kappa_{\mathrm{EN}hh})\,\theta(\kappa_{CNhh} - \mathrm{CN}_i) \tag{1.7}$$

$$\begin{aligned}\mathrm{HH}_{i,j} &= \frac{1}{4} hh_i - \kappa_{PTCHH} H_{HH} [\mathrm{PTC}]_0 \mathrm{PTC}_{n,j+2}\mathrm{HH}_{i,j} + \\ &\quad r_{LMHH} H_{HH}(\mathrm{HH}_{i,lr} - 2\mathrm{HH}_{i,j})\end{aligned} \tag{1.8}$$

$$ptc_i = \theta(\mathrm{CI}_i - \kappa_{CIptc})\,\theta(\kappa_{CNptc} - \mathrm{CN}_i) \tag{1.9}$$

$$\begin{aligned}\mathrm{PTC}_{i,j} &= \frac{1}{4} ptc_i - \kappa_{PTCHH} H_{PTC} [\mathrm{HH}]_0 \mathrm{HH}_{n,j+2}\mathrm{PTC}_{i,j} + \\ &\quad r_{LMPTC} H_{PTC}(\mathrm{PTC}_{i,lr} - 2\mathrm{PTC}_{i,j})\end{aligned} \tag{1.10}$$

$$\frac{\mathrm{PH}_{i,j}}{H_{PH}} = \kappa_{PTCHH} [\mathrm{HH}]_0 \mathrm{HH}_{n,j+2}\mathrm{PTC}_{i,j} \tag{1.11}$$

$$ci_i \;=\; \theta(\kappa_{\mathrm{EN}ci} - \mathrm{EN}_i) \tag{1.12}$$

$$\mathrm{CI}_i \;=\; \frac{\theta(\kappa_{\mathrm{EN}ci} - \mathrm{EN}_i)}{1 + H_{CI} C_{CI}\, \theta(\mathrm{PTC}_{i,\mathrm{T}} - \kappa_{PTC\,CI})} \tag{1.13}$$

$$\mathrm{CN}_i \;=\; \frac{H_{CI} C_{CI}}{1 + H_{CI} C_{CI}}\, \theta(\kappa_{\mathrm{EN}ci} - \mathrm{EN}_i)\, \theta(\mathrm{PTC}_{i,\mathrm{T}} - \kappa_{PTC\,CI}) \tag{1.14}$$

Here we use the same notation as in [7]. X_i , $i = 1, 2, 3, 4$, denotes the total concentration of the protein species X in cell i, with lower case x_i referring to the concentration of the corresponding mRNA molecules. In addition, for three of the components involved in cell-to-cell communication, namely, *external Wingless* (*EWG*), *Patched* (*PTC*) and *HH*, the concentration on each of the four cell faces could be different. For any of these components, the concentration in cell i at face j is denoted by $X_{i,j}$, $i = 1, 2, 3, 4,$, $j = 1, 2, 3, 4$. For these three species, the sum of the concentrations over all four faces of cell i is denoted by $X_{i,T}$. The adjacent cell face to face j of cell i is shown by $X_{i,lr}$. The opposite cell face to face j of cell i is shown by $X_{n,j+2}$.

Also, κ_{XY} denotes the dissociation constant for species Y corresponding to the binding that regulates the species X. The range for κ_{XY} is chosen to be between zero and one. The equations are in normalized form, meaning that the concentrations of the components have been scaled so that the maximal steady state level is one.

1.2.2 Study of the two new interactions

The structure of this particular network allows one to draw several interesting conclusions immediately. For example, the steady state levels for *HH* and *PTC* are completely determined once one specifies the mRNA levels of *en*, *hh* and *ptc* (this does not depend on the high Hill coefficient approximation). Assuming that *en* and *hh* are expressed only in the cell 3, which is the case in the wild type pattern, it can be shown that $ptc_2 = ptc_4$, and $\text{PTC}_{2,T} = \text{PTC}_{4,T}$. The reason is as follows. If $ptc_2 > ptc_4$, cell 2 ends up producing more PTC, part of which get bound to HH diffusing over from cell 3. However, the symmetric nature of the diffusion leads to more PTC in cell 2 than in cell 4: $PTC_{2,T} > PTC_{4,T}$. A higher level of PTC results in a higher rate of proteolysis of *CI*. Therefore, in the steady state, CI_i is a decreasing function of PTC_i and CN_i is an increasing function of PTC_i. This means that (given *en* is not present in cells 2 and 4, and therefore has no repressive effect on *ci* production):

$$CI_2 < CI_4 \quad , \quad CN_2 > CN_4 \ . \tag{1.15}$$

However, *CI* is an activator and *CN* is a repressor of *ptc*, which together with Equation 1.15 implies $ptc_2 < ptc_4$, which contradicts the assumption we started with. Of course, we could have started with $ptc_2 < ptc_4$ and again ended up with contradiction (for the formal proof see [21]). This argument shows that the concentration levels of *ptc*, *PTC*, *CI*, *CN*, and *PH* are exactly the same in cells 2 and 4:

$$ptc_2 = ptc_4 \ , \ PTC_2 = PTC_4 \ , \ CI_2 = CI_4 \ , \ CN_2 = CN_4 \ , \ PH_2 = PH_4 \ . \tag{1.16}$$

This observation will turn out to be quite significant for the following reason. The *wg* level in a cell is controlled by the *CI*-*CN* pathway and the postulated feedback [7] from *internal Wingless* (*IWG*). Since cells 2 and 4 do not differ when it comes to *CI* and *CN* levels, any difference in the *WG* expression has to be attributed to the *wg* autoregulation.

In order to analyze the *wg* sector, we note that, in this model, the *EWG* and *IWG* levels are uniquely determined by a set of linear equations once the *wg* levels are given. After solving these linear equations, using the periodic boundary conditions and the fact that *wg* is produced only in cell 2, we find that:

$$EWG_1 = EWG_3 < EWG_2 \ . \tag{1.17}$$

This result is not surprising because the distribution of *WG* is determined by a symmetric diffusion process from the source in cell 2, the only *wg* producing cell in each parasegment. Therefore, we expect cells 1 and 3 to have identical amounts of *WG* signaling. It turns out that *EWG* at the source, cell 2, is higher than that of the flanking cells (for the formal proof see [21]). These observations have important consequences for the regulation of *en*, as explained below.

Since *en* is expressed in cell 3, we have:

$$EWG_3 > \kappa_{EWGen} \ . \tag{1.18}$$

This, together with Equation 1.17, implies:

$$EWG_1 \ , \ EWG_2 \ > \kappa_{EWGen} \ . \tag{1.19}$$

Had the *en* production been solely controlled by *EWG*, the model would have implied that if EWG_3 is high enough to activate *en* in cell 3, *en* will be also activated in cells 1 and 2. This is why, in reference [7], adding repression of *en* by *CN* was necessary to achieve the wild type expression pattern. The two new links introduced in reference [7] (green lines in Figure 1.4B) give rise to two positive feedback loops. The *wg* autoactivation gives rise to bistability, allowing cells 2 and 4 to have distinct levels of *wg* expression. The other loop (En $\dashv$ ci $\rightarrow$ *CI* $\rightarrow$ *CN* $\dashv$ *en* $\rightarrow$ EN), generated by adding repression of *en* by *CN*, is required to prevent *en* from being expressed in cells 1 and 2. This also requires *CN* to be expressed in those cells. The bistability of the *EN*-*CI*-*CN* system allows cells 1 and 3 to have different *en* level even when the external *WG* signal is the same for both of them.

We should note that autoactivation as a way for maintaining the *WG* expression is problematic in the following sense. In the model described above, *wg* is always activated via autoactivation and the preexisting *CI*-*CN* pathway never contributes to the pattern. This is in contrast with the experimental data which suggest that *HH* signaling from the neighboring cell plays a crucial role in maintaining the *wg* expression. The fact that model [7] does not depend upon *HH* signaling for maintaining the expression of *wg* manifests itself when cell division is considered. In this model, both daughters of a cell in the wg-expressing stripe are able to retain the *wg* ON state through autoactivation. This causes the stripe to grow wider and wider over cell divisions. However, in wild type fly, the wg-expressing stripe should remain one cell wide. The daughter which is further from the en-expressing stripe, and therefore not exposed to *HH* signaling, loses *wg* expression. This means that one stripe of *WG* is left after each division.

Ingolia [15] has also noticed that in this model, *IWG* level must always be above κ_{wgwg} (the autoactivation threshold) in the cell that expresses *wg*. When we removed the *CI*-*CN* cycle for activation of *wg* from the simulation performed in reference [7], the fraction of 'good solutions' increased by a factor of 3. This suggests that most of the time the *CI*-*CN* pathway is either not contributing to *WG* expression or it leads to misexpression of *WG* in cell 4.

The model is too dependent on the bistability of the two sub-networks with positive feedback for maintaining four cell expression patterns. One could avoid this problem by making some of the four cells special, either by inclusion of other genes in the network or by explicitly breaking the symmetry via introducing different gene expression rates from cell to cell for some of the genes already in the model.

The major candidate for inclusion in the model is the *Sloppy-paired* protein (*SLP*) as has already been suggested by others [22, 16, 15]. *SLP* is only present in cells 1 and 2: $SLP^{WT}_{1,2,3,4} = (1, 1, 0, 0)$. It is a necessary (but not sufficient) factor for activation of *wg* and it also represses en. In the presence of *SLP*, the reason *en* is not expressed in cell 1 despite *WG* signaling is that it is being repressed by *SLP*. Also, despite *HH* signaling, *wg* is not produced in cell 4 because *SLP* is not present there. With *SLP* added, the two new interactions introduced in [7] are not necessary anymore, and also *WG* expression will depend on the *CI-CN* pathway.

In later sections, we will analyze the effect of including *SLP*. We keep *SLP* as an external factor, meaning the expression pattern of *SLP* is given. However, it can easily be incorporated into the network. If *WG* activates *SLP*, a positive feedback loop is formed which allows for bistability: both *WG* and *SLP* can be ON or both can be OFF. On the other hand, if *EN* represses *SLP*, another positive feedback loop is formed which again allows for bistability: *SLP* can be ON and *en* OFF or vice versa. A model with explicitly different rates of production of *ptc* and *ci* from cell to cell has been presented in [[21].

1.2.3 Characterizing the feasible region

Here, we consider two particular cases:

I) The regulatory network used by von Dassow et al. [7]. This network is shown in Figure 1.4B. We will refer to this case as von Dassow et al. model.

II) The regulatory network including *Sloppy-paired* protein, but without the two positive feedback links introduced in [7]. This network is shown in Figure 1.6. We will refer to this case as *SLP* model.

We can explicitly write down the conditions characterizing the feasible region for these two models. The results are presented in Tables 1.1 and 1.2. The derivation of these conditions is presented below. However, one can skip to the next section, since, the details of the derivation will not be necessary.

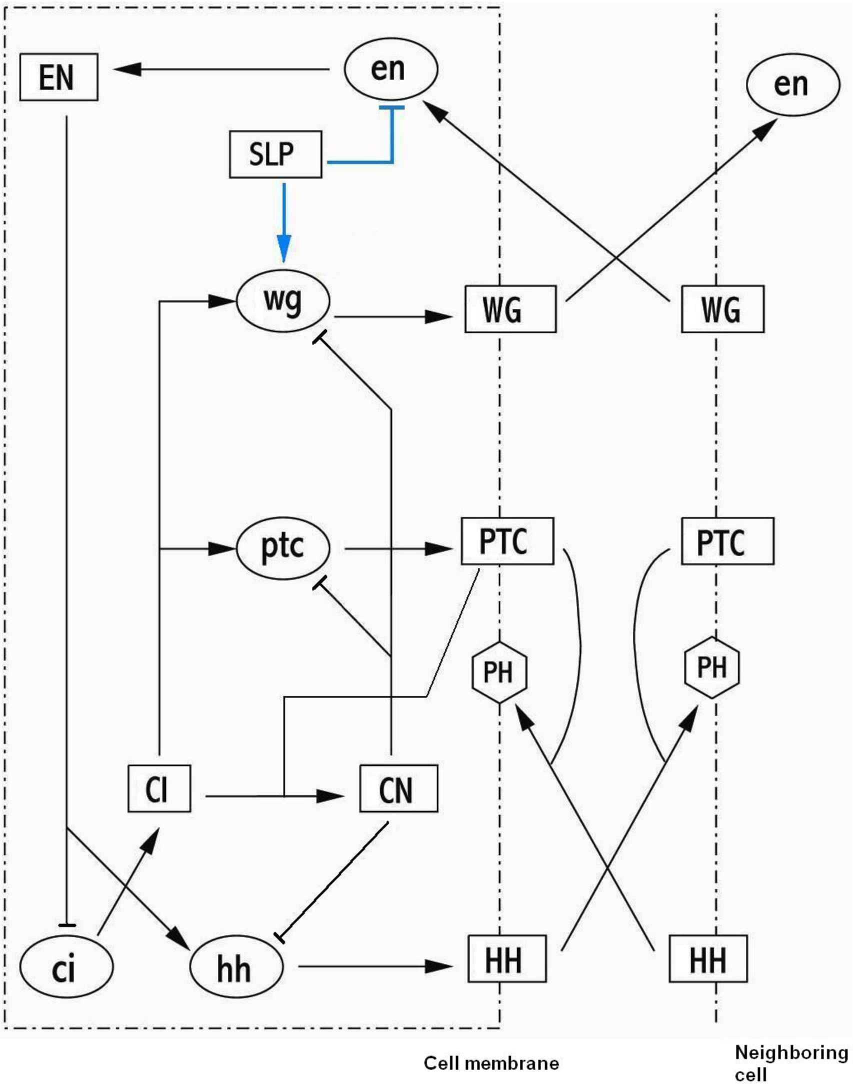

Figure 1.6: Segment polarity regulatory network including sloppy-paired protein. In this model, the possibility of *Wg* autoactivation and en repression by *CN* is not included.

$0 < \kappa_{PTCCI} < \mathrm{PTC}_1^m \, , \, \mathrm{PTC}_2^m \, , \, \mathrm{PTC}_4^m$	(1)
$1 > \kappa_{CIwg} > 1 - Z_c \quad \text{or} \quad 0 < \kappa_{CNwg} < Z_c$ $Z_c := \min(1 - \kappa_{CIptc} \, , \kappa_{CNptc} \, , \frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}})$	(2)
$0 < \kappa_{EWGen} < \mathrm{EWG}_3$	(3)
$0 < \kappa_{CNen} < Z_c$	(4)
$\max\{\mathrm{IWG}_{1,3,4}\} < \kappa_{\mathrm{WG}wg} < \mathrm{IWG}_2$	(5)

Table 1.1: Conditions characterizing the feasible region for the regulatory network used by von Dassow and collaborators. This network, shown in Figure 1.4B, includes two positive feedback loops achieved by adding *WG* autoactivation and en repression by *CN*.

Having explicitly characterized the feasible region, we could easily estimate its volume by randomly choosing points in parameter space and checking whether they satisfy the appropriate conditions. In addition, we are able to explore the geometry of the feasible region by following random walks starting from random points in this space. As we discussed in the introduction, the fate of random walks, especially where they exit the feasible region, teaches us a lot about the relative vulnerability of different constraints. This will be the subject of the next section.

$\mathrm{PTC}_2^m = \mathrm{PTC}_4^m < \kappa_{PTCCI} < \mathrm{PTC}_1^m$	(1)
$(1 > \kappa_{CIwg} > 1 - Z_c \quad \text{and} \quad 0 < \kappa_{CNwg} < 1)$ or $(1 > \kappa_{CIwg} > 0 \quad \text{and} \quad 0 < \kappa_{CNwg} < Z_c)$ $Z_c := \min(1 - \kappa_{CIptc} \ , \kappa_{CNptc} \ , \frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}})$	(2)
$\mathrm{EWG}_4 < \kappa_{EWGen} < \mathrm{EWG}_3$	(3)

Table 1.2: Conditions characterizing the feasible. region for the regulatory network including *Sloppy-paired* protein. In this network, shown in Figure 1.6, the two links of *WG* autoactivation and en repression by *CN* are absent.

Derivation of the conditions

Here we analyze two particular cases: I) The regulatory network used by von Dassow et al. [7] which we refer to as the von Dassow et al. model (Figure 1.6B). II) The regulatory network including *Sloppy-paired* protein, but without the two positive feedback links introduced in [3]. We will refer to this case as the *SLP* model (Figure 1.6).

We first focus on case I. This network is characterized by Equations 1.2 - 1.14. The wild type expression pattern for *wg*, *en* and *hh* is given in Equation 1.1. Since *en* is only expressed in cell 3, ci and *ptc* are expressed in all cells except cell 3:

$$ci_{1,2,3,4}^{WT} = (1, 1, 0, 1) \ , \ \ ptc_{1,2,3,4}^{WT} = (T_1, T_2, 0, T_4) \ . \tag{1.20}$$

This is because in the absence of *EN*, *ci* is basally expressed which also leads to production of *ptc*. We will allow T_i to take values between zero and one. The reason for the special, non-Boolean, treatment of *ptc* has to do with capturing the effect of the negative feedback loop in the *CI*-*CN*-*PTC* sector properly. This negative feedback loop leads to lower *ptc* level in cell 1 than in cells 2 and 4, as we shall see. The *ptc* level in cells 2 and 4 turn out to be comparable ($T_2 = T_4$). This is also the experimentally observed expression pattern of *ptc* [23].

How could we ever get such an intermediate values in our approach? First, from Equations 1.13 and 1.14, in the cells where *en* is not expressed and therefore *ci* is not repressed, namely in cells 1, 2 and 4, we have $CI + CN = 1 \Rightarrow CI = 1 - CN$ (this does not depend on the high Hill coefficient approximation). Since *ptc* is regulated by *CI*-*CN*, we could draw one nullcline expressing *ptc* concentration as a function of *CN*. This curve is represented by the green graph in Figure 1.7. We will call it the *ptc*-nullcline. Here it is assumed that the negative feedback on *ptc* coming from repression by *CN* is active. This means that *CN* and *ptc* are not expressed maximally. For *ptc* to be expressed, the activation by CI requires . In addition, we need *CN* to be smaller than κ_{CNptc} to avoid repression of *ptc* by *CN*. Thus, for values of *CN* smaller than the threshold of $\min(1 - \kappa_{CIptc}\ , \kappa_{CNptc})$, *ptc* is fully expressed. As *CN* passes this point, the value of *ptc* will drop sharply. In the high Hill coefficient limit, *ptc* will abruptly fall to zero.

On the other hand, *CN* production itself is dependent upon *PTC* protein. *PTC* is a monotonically increasing function of *ptc* and a decreasing function of *HH* signaling. Therefore, for a fixed value of *HH* level, we can also look at the concentration of *CN* as a function of ptc. This provides us with the *CN*-nullcline which depends upon the HH signaling strength. If we think of *CN* as a function of *ptc* level, the transition in *CN* from low level to its highest value happens at a particular *ptc* threshold, where the *PTC* level is just enough to start producing *CN*. If the cell is exposed to more *HH* signaling, sequestering away a larger fraction of total Patched protein, one needs more *ptc* to reach this threshold. The blue and the red graphs in Figure 1.7 show the *CN*-nullclines for relatively higher and lower values of *HH* signaling levels, respectively.

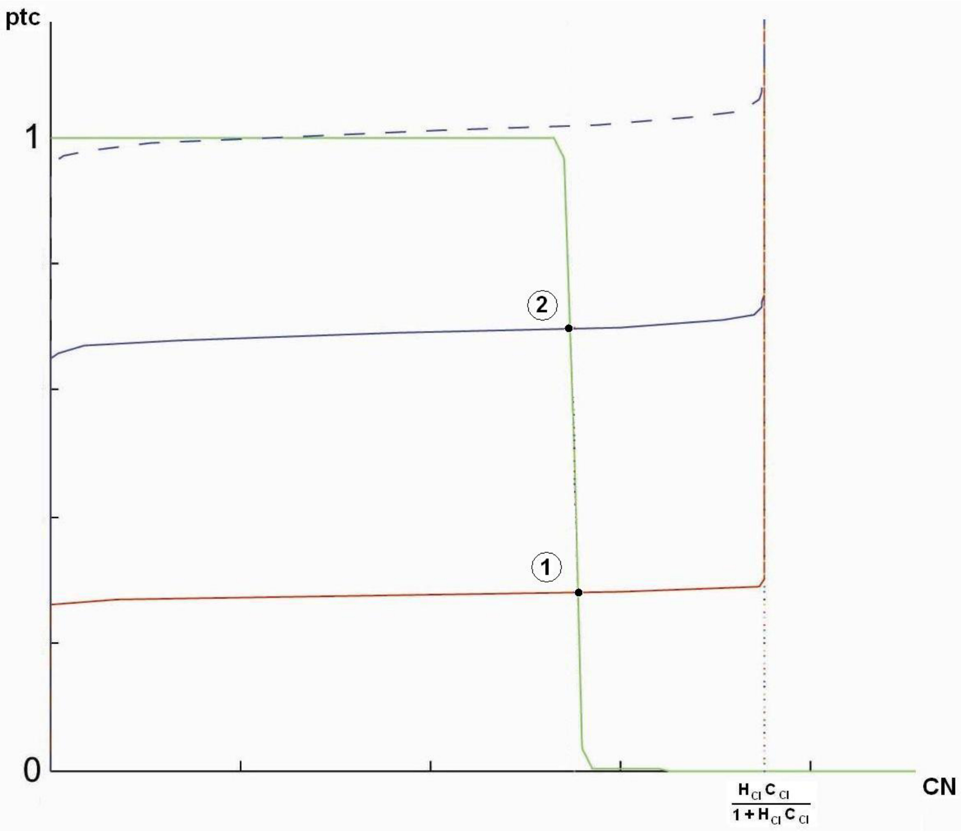

Figure 1.7: The nullclines for *ptc* and CN. The green curve shows the *ptc*-nullcline. In the high Hill coefficient limit, the ptc value drops sharply from one to zero as CN passes the threshold of $\min(1-\kappa_{CIptc}\ , \kappa_{CNptc})$. Blue and red curves show the CN-nullclines for relatively higher and lower values of HH signaling levels, respectively. Intersection points 1 and 2 determine CI, CN and *ptc* in cell 1 and 2/4, respectively. Here it is assumed that the negative feedback on ptc coming from repression by CN is active. Therefore, ptc and CN are not maximally expressed. Dashed blue line shows the CN-nullcline for a fine-tuned set of parameters.

Because cell 1 receives less external *HH* signaling than cells 2 and 4, generally the red curve could be associated to cell 1 and the blue one to cells 2 and 4. The intersection points 1 and 2 determine *CI*, *CN* and *ptc* levels in cell 1 and 2/4, respectively. As we see, the *ptc* value could indeed be higher in cell 2 than in cell 1. However, *CN* concentration seems to be comparable in those cells. This is an artifact of our model where Hill coefficients are very large, which causes the transition from high to low in concentration value to happen in a very narrow range. The only way to have CN_2 to be non-zero but different from CN_1 is to be in the situation where the *CN*-nullcline for cell 2 is like the dashed blue line in Figure 1.7. In this case, the *ptc* threshold for *CN* production in cell 2 is fine-tuned to be very close to maximal *ptc* level. In a model with small Hill coefficients in the *CI*-*CN*-*PTC* sector, we would get $\mathrm{CN}_1 > \mathrm{CN}_2$ and $ptc_1 < ptc_2$ without such fine-tuning. We will come back to this point later.

We should point out that, in this study, we places down the conditions only on the expression levels of key components *en*, *wg* and *hh* as specified in Equation 1.1. The reason, other than simplicity of analysis, is that we believe that segment formation lays much stronger constraints on the key components. It is not clear to us that the *CI*-*CN*-*PTC* negative feedback has an extremely important role in the segment formation stage of development. The study of von Dassow et al. [3] also uses an scoring function which rewards wild type levels only for these key components

Having specified the requirements of functionality, let us now analyze what conditions are laid on the parameters of the model. Table 1.1 shows the set of inequalities characterizing the feasible region in the parameter space. Here we present the arguments leading to these conditions. The presence of *EN* in cell 3 requires the *WG* signaling for this cell to be above the activation threshold for *en*. This requirement is condition 3 in Table 1.1 (recall that κ_{XY} can take values only between zero and one). Also, in this cell, *EN* will shut off the expression of *ci*

(Equation 1.12) which is necessary for the production of *CI*, *ptc*, *PTC* and *PH*. Therefore, none of those components are expressed in cell 3. In cells 2 and 4, the expression level of these components has been shown to be the same (Equation 1.16). Therefore, we only need to focus on the expression of these components in cells 1 and 2.

Let PTC_i^m be the *PTC* level corresponding to the maximal *ptc* mRNA ($ptc = 1$) in cell i. If the threshold to produce *CN* is above PTC_i^m, then cell i would not produce *CN*. As was pointed out before, the presence of *CN* in cells 1 and 2 is essential to repress *en* in those cells. These facts together necessitate condition 1 in Table 1.1.

What would the *CN* level in cells 1 and 2 be when condition 1 is satisfied? As one sees from Figure 1.8A, there are two possibilities depending upon whether $\min(1-\kappa_{CIptc}\ ,\kappa_{CNptc})$ is smaller or larger than $\frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}}$. The case corresponding to the *ptc*-nullcline in solid green has been discussed before. This is the case where *ptc* levels are affected by the negative feedback, and the *CN* level is equal to $\min(1-\kappa_{CIptc}\ ,\kappa_{CNptc})$, which is less than its maximal possible value of $\frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}}$. When the *ptc*-nullcline is like the dashed green line in Figure 1.8, *CN* levels in both cell 1 and cell 2 are equal to the maximal amount of $\frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}}$, which is lower than $\min(1-\kappa_{CIptc}\ ,\kappa_{CNptc})$. In this case, the negative feedback is not active and *ptc* is maximally expressed ($ptc = 1$). We conclude that the *CN* level is given by $\min(1-\kappa_{CIptc}\ ,\kappa_{CNptc}\ ,\frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}})$ which we call Z_c. We will now discuss the conditions to be satisfied by Z_c for proper expression pattern of *en* and *wg*.

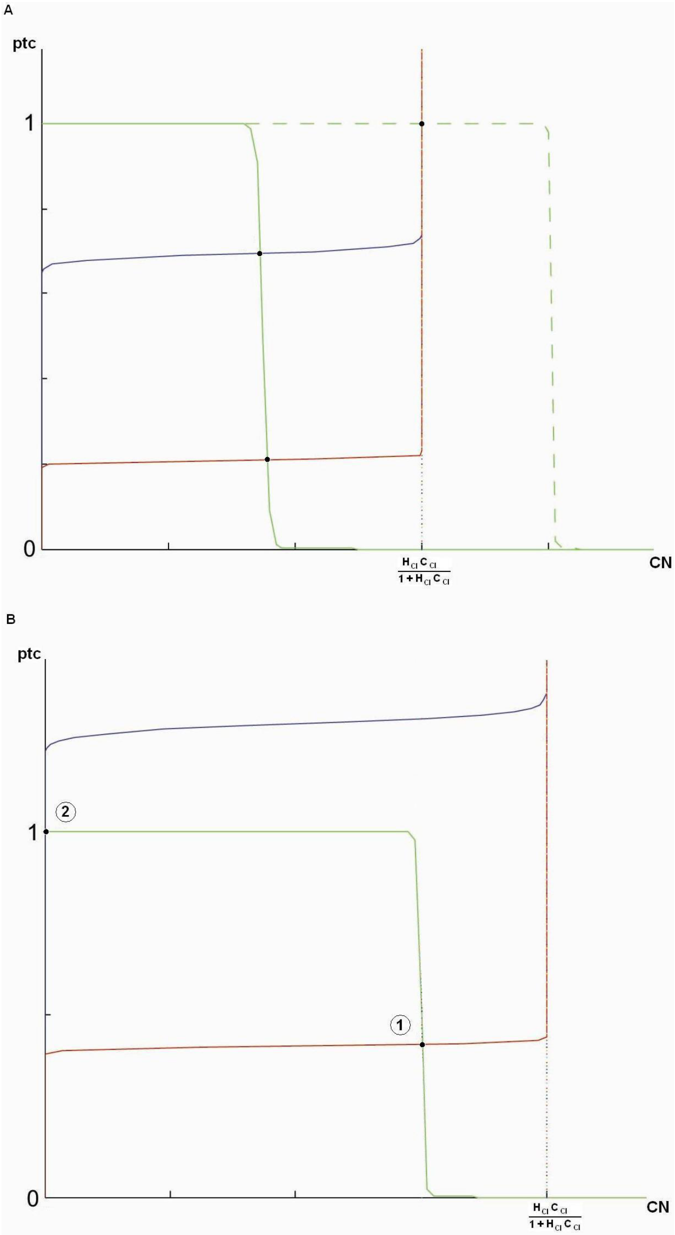

Figure 1.8: The nullclines for *ptc* and CN. A) Blue and red curves show the CN-nullclines for relatively higher and lower HH signaling levels, respectively. The green curve shows the *ptc*-nullcline when $\min(1-\kappa_{CIptc}\ ,\kappa_{CNptc}) < \frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}}$. In this case, the negative feedback on ptc coming from repression by CN is active. Therefore, ptc and CN are not maximally expressed. The dashed green curve shows the other case where $\min(1-\kappa_{CIptc}\ ,\kappa_{CNptc}) > \frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}}$. In this case, both CN and ptc are maximally expressed. This means that the negative feedback on ptc is inactive. B) The green curve shows the ptc-nullcline. Blue and red curves show the CN-nullclines for relatively higher and lower HH signaling levels, respectively. The blue curve shows the situation where HH signaling is strong enough so that the ptc concentration needed to produce CN is higher than the maximal possible value for ptc, namely, one. Therefore, CN will not be produced in the corresponding cell. In the high Hill coefficient approximation, this is the only way that we can have CN level in cell 2 (intersection point 2) be different from that in cell 1 (intersection point 1).

The *en* repression in cells 1 and 2 gives rise to condition 4 in Table 1.1. The fact that the*CI-CN* pathway should not activate *wg* in cell 4 is guaranteed by condition 2 in Table 1.1. Consequently, *WG* in cell 2 has no contribution from the *CI-CN* pathway (remember that cells 2 and 4 have the same *CI* and *CN* levels) and is solely produced by the autoactivation term. The autoactivation should only operate in cell 2 and nowhere else. This is condition 4 in Table 1.1.

von Dassow and Odell analyzed randomly generated solutions for the segment polarity model in reference [7] and plotted the marginal distribution of parameters (see Figure 6 of [20]). We can relate their results to the constraints presented in Table 1.1. From condition 1, we expect κ_{PTCCI} to have a tendency towards lower values. From condition 2, we expect κ_{CNwg} to have a tendency towards lower values and κ_{CIwg} for higher values. Also, in order to have higher values for Z_c, we expect κ_{CIptc} to have a tendency towards lower values and κ_{CNptc} for higher values. From conditions 3 and 4, we expect κ_{EWGen} and κ_{CNen} to have tendencies towards lower values. From condition 5, we expect $\kappa_{\mathrm{WG}wg}$ to have a tendency towards intermediate values. These expectations agree qualitatively with the results presented in Figure 6 of [20].

From Figure 6 of reference [20], we see that many of the parameters are uniformly distributed. One should note that a uniform distribution for a certain parameter could arise from two different scenarios. It could be the case that changing that parameter in a wide range of values does not influence the final outcome of the network. The other possibility is that the effect of changing that particular parameter could be compensated by changes in other parameters in such a way that for each value of the parameter, there is roughly equal number of solutions.

Now, let us contrast these sets of conditions to the one obtained for the *SLP* model. Table 1.2 shows the conditions defining the feasible region for this case. For this regulatory network (Figure 1.6), instead of Equations 1.2 and 1.5, we have:

$$wg_i \;=\; \theta(slp_i - \kappa_{\text{slp}wg})\;\theta(\text{CI}_i - \kappa_{CIwg})\theta(\kappa_{CNwg} - \text{CN}_i)\ , \tag{1.21}$$

$$en_i \;=\; (\kappa_{\text{slp}en} - slp_i)\;\theta(\text{EWG}_i - \kappa_{\text{WG}en})\ . \tag{1.22}$$

The rest of the equations are the same as before (Equations 1.3, 1.4 and 1.6 - 1.14). Since *SLP* is present only in cells 1 and 2, *wg* has the possibility to be expressed only in those two cells. The decisive factor is *CN* levels in cells 1 and 2 (remember that, in these cells, $CI = 1 - CN$). In the wild type pattern, *wg* is expressed only in cell 2 and this means that *CN* levels cannot be the same in cells 1 and 2. The only way to have less *CN* in cell 2 compared to cell 1 is to have $\text{PTC}_2^m \;\leq\; \kappa_{PTCCI} \;\leq\; \text{PTC}_1^m$. The condition $\text{PTC}_2^m \;\leq\; \kappa_{PTCCI}$ corresponds to the plateau in the *CN*-nullcline for cell 2 being higher than or equal to the maximal *ptc* level (blue graph in Figure 1.8B). When it is higher, CN_2 is zero and when it is fine-tuned to be equal, CN_2 is between 0 and 1. If we had $\text{PTC}_1^m \;\leq\; \kappa_{PTCCI}$, given that $\text{PTC}_2^m \;\leq\; \text{PTC}_1^m$, we would have $\text{CN}_1 = \text{CN}_2 = 0$. This is inconsistent with our requirement that CN_1 and CN_2 be different. Therefore, we have $\kappa_{PTCCI} \;\leq\; \text{PTC}_1^m$.

For our discussion, we will ignore the fine-tuned cases, leaving us with condition 1 in Table 1.2. This means $\text{CN}_2 = 0$ and $\text{CN}_1 = \min(1 - \kappa_{CIptc}\,, \kappa_{CNptc}\,, \frac{H_{CI}C_{CI}}{1+H_{CI}C_{CI}})$, which we again call Z_c. Conditions 2 in Table 1.2 guarantees the absence of *wg* in cell 1. The fact that external *WG* signaling has to be strong enough in cell 3 to activate *en* but has to be weak enough in cell 4 not to produce *en* is coded in condition 3 of Table 1.2.

1.3 Random Walk in the Feasible Region

We explore the feasible region by following random walks starting from random points. Whenever one of the random trajectories hits a boundary and exits the feasible region, we terminate the walk and keep track of the inequality that was violated. This process can be viewed as a simulation of parameter evolution due to mutations in a fitness landscape that looks like a plateau. The points in the feasible region have a constant high fitness, and the rest of the points have zero fitness.

1.3.1 Calculation of half-life

To get an estimate for the fractional volume of feasible region in the parameter space, we randomly chose 10^6 parameter combinations and check if they satisfy the conditions given in Table 1.1 and 1.2 for the corresponding model. We perform the random walk by first selecting a random point, P^0, from the set of admissible parameters and follow successive random perturbations ($\overrightarrow{P}^k = \overrightarrow{P}^{k-1} + d\overrightarrow{P}^k$, $k = 1, 2, ...$). Each component of $d\overrightarrow{P}^k$ is selected from an independent Gaussian distribution with standard deviation of 2×10^6 . We follow this random walk until it hits a boundary and exits the space. This happens when one of the inequalities which characterize the feasible region is violated. Whenever the random walk exits the region, we record the time as well as the condition which was violated and therefore caused the exit. The parameter ranges were similar to those used in [7], except that we facilitated the transport processes for hh and PTC. We simulated the random walk for 30,000 runs. The result of the simulation is presented in Figure 1.9.

The graphs in Figure 1.9A show the probability of survival as a function of time for both models. This is the probability that the random walk has not exited the feasible region in the first t steps. From the graph, we can easily measure

$T_{1/2}$, defined as the time for which there is a 50% chance that the system has already suffered a deleterious mutation. As we discussed in the introduction, this number is a possible indicator of robustness.

1.3.2 Main modes of network failure

Figure 1.9B shows the histogram of violated conditions. The number below each bin indicates the corresponding condition in Table 1.1 and 1.2. The lead cause of failure in the von Dassow et al. model is the constraint on $\kappa_{\mathrm{WG}wg}$ whereas in the *SLP* model it is the constraints on κ_{EWGen}. Higher vulnerability of the *SLP* model with respect to the constraint on κ_{EWGen} can be understood by comparing condition 3 in Table 1.1 and the corresponding condition in Table 1.2. In the *SLP* model there is a lower bound on κ_{EWGen} coming from the fact that κ_{EWGen} should be greater than EWG_4 to prevent activation of *en* in cell 4. However in the von Dassow et al model, *en* is being repressed by *CN* and therefore there is no lower limit on κ_{EWGen}.

One might raise the question of whether including repression of *en* by *CN* in the *SLP* model changes the constraints on κ_{EWGen}. In the high Hill coefficient limit, adding this interaction does not change the conditions in Table 1.2. To see this, note that as was mentioned before, requiring *CI* and *CN* levels to be different in cells 1 and 2 forces us to have $\mathrm{CN}_2 = \mathrm{CN}_4 = 0$. In cell 4, *CN* is not expressed, and in cells 1 and 2, *en* is already being repressed by *SLP*. Therefore, adding the possibility of *en* repression by *CN* does not change any of the constraints.

If we consider the case where Hill coefficients in the *CI-CN-PTC* sector are small, the transition from high to low in concentration value for the *ptc*-nullcline and *CN*-nullcline would not be sharp. Instead, the transition would happen over a wide range. This means that we would get a non-zero value for CN_4. In that case, adding repression of *en* by *CN* can indeed help in maintaining the wild type pattern, thereby increasing the robustness of the model.

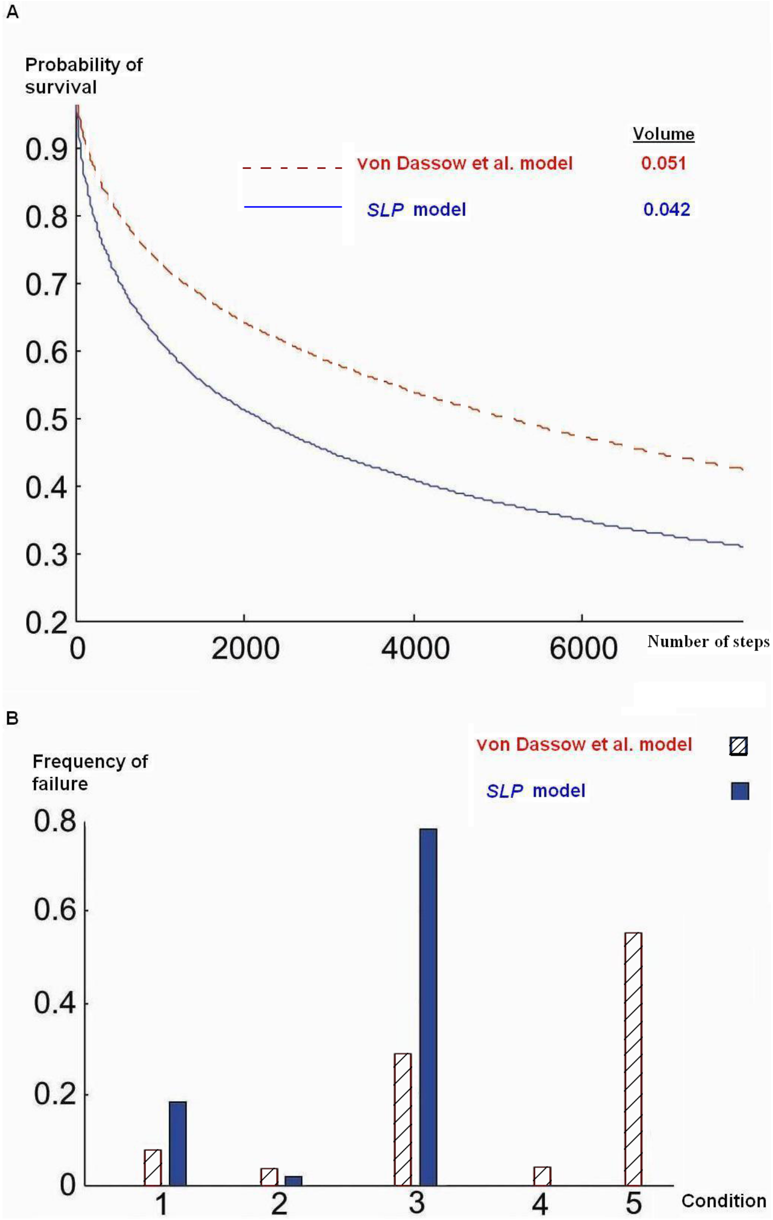

Figure 1.9: Random walk in the space of admissible parameters. We choose a random point from the admissible parameter set and follow a random walk until it hits a boundary after t steps. A) The red and the blue graphs represent the probability of survival as a function of time for von Dassow et al. and *SLP* models, respectively. These graphs results from 30,000 runs of random walks. The results given for volume are based on the fraction of feasible parameter combinations found in 1,000,000 randomly chosen combinations. B) Histogram of violated conditions for the random walk in A. The number above each bin indicates the corresponding condition in Table 1.1 and 1.2.

The parameters κ_{CIwg}, κ_{CNwg} and $\kappa_{\mathrm{WG}wg}$ are related to alternative routes controlling *wg* expression. The first two parameters play an important role in deciding *WG* expression in the *SLP* model, while this role is played by $\kappa_{\mathrm{WG}wg}$ in the von Dassow et al model. Comparison of the frequency of failure for conditions 2 and 5 in the histogram in Figure 1.9B suggests that controlling *wg* via the *CI-CN* pathway in the presence of *SLP* is the more robust way of achieving the target gene expression pattern for *wg*.

What about adding the *WG* autoactivation to the *SLP* model? If one just cares about producing the right four cell pattern for *en*, *hh* and *wg*, then this addition could give rise to more solutions. However, as we discussed before, not having *wg* production to be sensitive to *HH* signaling from the neighboring cell is problematic and gives rise to wide stripes of *wg* expression under cell division. If we constrain the model so that *wg* is sensitive to *HH* signaling via *CI-CN* pathway, we find that adding *wg* autoactivation to a functional solution in the *SLP* model often leads to misexpression of *wg* in cell 1 or cell 3, thereby shrinking the feasible region in parameter space.

1.4 Discussion

Our results imply that the lack of robustness is not only dependent upon the size of the feasible region, but also upon the existence of critical directions along which this region is globally very narrow. We found relatively few constraints on the parameters given that we have specified the gene expression patterns for *en*, *hh* and *wg* in each of the four cells. Much has been said about the relation between the topology of the network and robustness. In practice, we found that it is not only the structure of the network but also the nature of the wild type expression pattern which plays an important role in the ultimate simplicity of the constraints that dictate robustness. For example, the fact that only one cell is expressing *en* and *hh* and that *wg* had no direct effect on the *CI-CN-PTC* sector allowed us to draw several conclusions about certain variables being the same in cell 2 and cell 4. If one stares only at the network structure, *wg* indeed has an effect on the *CI-CN-PTC* sector via its effect on *en*. However, specifying the *en* expression pattern hides the influence of *wg* and helps us disentangle the constraints. The role of *wg* shows up only when one insists upon self-consistency, namely, the *wg* expression pattern is going to lead to the target *en* expression pattern. Simplicity of the final constraints is not a result of some obvious modularity in the network itself but some combination of the network structure as well as of the sparseness of the expression pattern. We cannot be sure that this is a general feature of robust genetic networks. A broader study which takes into account the role of the wild type pattern on the robustness of a network would be a welcome deviation from discussions centered purely on network architecture.

We noted that capturing the *CI-CN-PTC* negative feedback in the Boolean model is difficult. For example, in the Boolean model constructed by Albert and Othmer [16], they are forced into a situation where *ptc* mRNA is OFF but *PTC* protein is ON. This is achieved because of an exception made in *PTC* production rule, namely, *PTC* can continue to be in the ON state even if there is no *ptc*. Of course, this implausible rule results in a distribution of *ptc* and *ci* products which mimics the wild type pattern. For example cell 1 has less *ptc* but more *CN* compared to cell 2. In our model, we partially capture the effect of the feedback. We can indeed get the *ptc* levels to vary between cell 1 and cell 2. Unfortunately, we saw that in the high Hill coefficient model, producing different *CN* levels requires fine-tuning of the parameters. Therefore, we understand why von Dassow et al. find that setting the Hill coefficients in the *CI-CN-PTC* sector to be small enhances their chance of finding good solutions [20].

The present approach shows that, in addition to volume, the topology and geometry of the feasible region have important consequences for the robustness of a system. Of special interest is the structure of the boundary in the parameter space that separates between functional and non-functional systems. In the models studied here, it was possible to describe this boundary explicitly as a collection of constraints. For a generic biochemical network model with a scoring function it may not be feasible to explicitly write down the boundary surface corresponding to the threshold of functionality. However, one could generate a sampling of the boundary surface by following random walks in the parameter space until they hit the boundary of the functional region (decided by a threshold score). Instead of what we did in this study, we could slightly alter our strategy and let the walk be reflected off the boundary. In that process the same walk would hit many neighboring points on the boundary surface. If one generates a large enough sample of boundary points, one could use methods like manifold learning [24, 25] to approximately reconstruct the boundary.

Contrast this method to boundary reconstruction from uncorrelated random sampling. One could generate many points some of which are inside the region and many others which are outside. Indeed, many machine learning techniques for classification involve learning decision boundaries from such data. However, when the good region has a very small fractional volume and many of the randomly sampled points outside this region are far from the decision boundary, most of the sampled points have very little impact on boundary reconstruction. The uncorrelated nature of the sampling is useful for getting a good estimate for the fractional volume, but makes the process of mapping the geometry inefficient. It would be better to take advantage of one good solution to generate other good ones for the purpose of exploring local geometry.

To summarize, our analysis of the segment polarity network provides us with insights regarding the constraints that are crucial for functioning of the system. We showed how the system is particularly vulnerable to parametric perturbations in certain directions in the parameter space. We believe that the ideas developed here could be applied to other regulatory networks, to explore how the shape of feasible region in the parameter space contributes to its robustness. Hill terms appear often in models of biochemical networks. A simpler model, obtained by replacing these terms with step function, could be useful, because such a model enjoys some of the simplicity of the Boolean networks, while retaining many of the parameters of the original model.

Chapter 2
Epigenetic Chromatin Silencing

In a multicellular organism, there are many different cell types (e.g. muscle, blood, nerve, etc.) despite the fact that all the cells have the same DNA. Once it is generated, this pattern of cell fates has to be maintained through cell division. Such heritable locking of different cells into different fates without irreversible change in genetic information is called *epigenetic* phenomenon. Building a mathematical model of epigenetic chromatin silencing based on current biological knowledge, and exploring the consequences and predictions of the model is the subject of this chapter.

2.1 Biological Background

In this section, we will go over some basic biological background relevant to the material discussed in this chapter. A more through presentation can be found in biology textbooks [3, 4].

2.1.1 DNA packaging and gene activity

Organisms are divided into two groups: eukaryotes and prokaryotes, depending, respectively, whether or not their cells contain a distinct nucleus compartment. Many unicellular organisms, like bacteria, belong to prokaryotes. Typical multicellular organisms happen to be eukaryotes. DNA of eukaryotes is not a free polymer. Instead, it is packaged into a certain structure. The basic unit of this packaging is nucleosome, which is composed of 147 bp of DNA wrapped around a

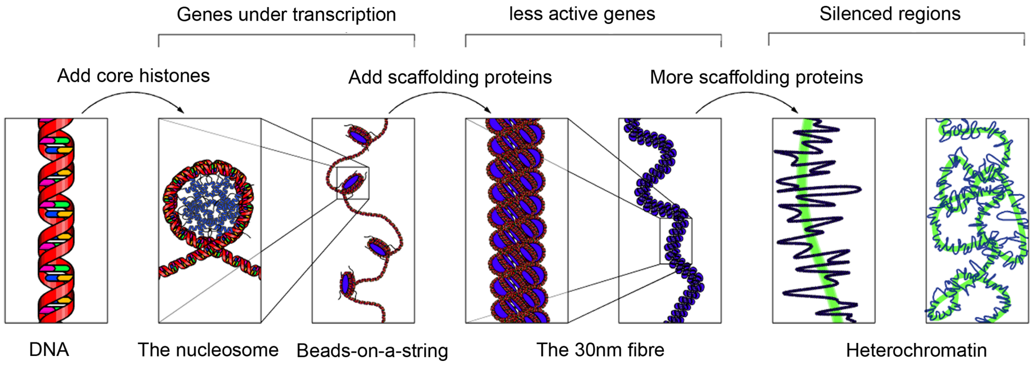

Figure 2.1: Several levels of DNA packaging. The basic unit of packaging is nucleosome, composed of a protein complex made of histones with 147 base pair of DNA wrapped around it. Histones have tails which can be chemically modified. The modification of tails correlates with the formation of higher level structures. Therefore, depending on the local modifications, the degree of packaging varies for different regions of DNA. Genes located in heterochromatin, i.e. the more condensed regions, are systematically silenced. In contrast, genes located in euchromatin can be active. Adapted from wikipedia.com

core complex formed from eight proteins called *histones* (Figure 2.1). The diameter of the histone core is around 11 nm. In between two consecutive nucleosomes, there is some unwrapped DNA named linker DNA typically smaller than 100 bp. The mixture of DNA and structural proteins is called *chromatin.* Higher order organisms have more than one DNA molecules, each of which is called a chromosome (e.g. humans have 24 chromosomes).

The array of nucleosomes can fold and have more levels of packaging. This packaging requires association of specialized proteins with the nucleosomes. Specifically, histone cores have long tails sticking out which can get chemically modified and affect the degree of compactification. This local modification of histone tails plays a central role in the mechanism of epigenetic silencing that we will discuss.

Each gene encodes the instruction necessary to build its associated protein. All the cells within a multicellular organism have the same DNA. However, in each cell, depending on the cell type, certain proteins are never expressed and associated regions of DNA are systematically silenced. That is why, even though all the cells of a multicellular organism have the same genetic instruction, they can have different types.

The systematically silenced regions correspond to highly condensed and packed areas of DNA called *heterochromatin*. In contrast, the other parts of DNA called *euchromatin* are lightly condensed and are often under active transcription. In order for the cell type to be preserved in cell division, the pattern of heterochromatin and euchromatin regions has to be inherited to daughter cells. In fact, for a region to be qualified as epigenetically silenced, by definition, the pattern has to be inheritable to daughter cells during cell divisions.

As a side note, it should be mentioned that genes located in euchromatin are not necessarily always active. It is possible that such genes get silenced for certain amount of time through the mechanisms involving transcription factors. This type of gene silencing, which works at the level of individual gene (as opposed to a longer region of DNA) and can change by time, is not the focus of this chapter.

The first indication for the existence of systematically silenced regions which are inheritable during cell division came from the phenomenon of position effect variegation, explained in the next subsection. Another example of epigenetic silencing is the *HML* and *HMR* Loci in budding yeast, which I will explain below as well.

2.1.2 Position effect variegation

Position effect variegation is a consequence of the fact that, for eukaryotes, not only the average activity of a gene, but also the variability of the expression depends on gene's position along the genome. Typically, the boundary between heterochromatin and euchromatin region is not fixed and can move one way or the other. However, the boundary is quite stable and only occasionally displaces significantly. Imagine a gene is located in the euchromatin region, however, close to the boundary with heterochromatin. This gene gets transcribed and produces certain protein which has some observable effect. For example, in the case of fruit fly and the so-called *white* gene, the expression of the gene causes the eye to become red. The daughters of this cell will also be red. In this manner, after several rounds of division, we a get a patch of cells in red color.

Once after several cell cycle, in one of the red cells, the heterochromatin region may spread into the euchromatin and cover the aforementioned *white* gene. Therefore, the gene gets silenced which causes the cell to become white. Since, the newly formed heterochromatin boundary is stable under cell cycle, the progenies of this white cell will be white as well. In this way, we get a patch of white cells within the bigger red patch. After several rounds of division, one of the white cells may go back to the red state, i.e. the heterochromatin region shrinks and the *white* gene becomes active again. The overall effect of the above phenomenon is that we get several patches of red and white color in the fly eye. Each patch has been created because of a switching event caused by the displacement of the boundary between heterochromatin and euchromatin.

2.1.3 Silencing in budding yeast

Budding yeast (*Saccharomyces Cerevisiae*) can be found in two forms: haploid and diploid. Haploid cells have only one set of 16 chromosomes. Diploids, on the other hand, have two sets of 16 chromosomes, i.e. 32 chromosomes or 16 pairs. Each pair of chromosomes have the same set of genes. The copies of a gene on the two chromosomes can be exactly the same or can be two different versions of the same gene.

Figure 2.2 shows different states of yeast cells. In normal conditions, diploid cells divide (via budding) into two diploid cells. However, in starvation condition, a diploid cell divides into four haploid cells (referred to as sporulation). Haploid cells exist in two types, **a** and α, which can be considered as two opposite sex

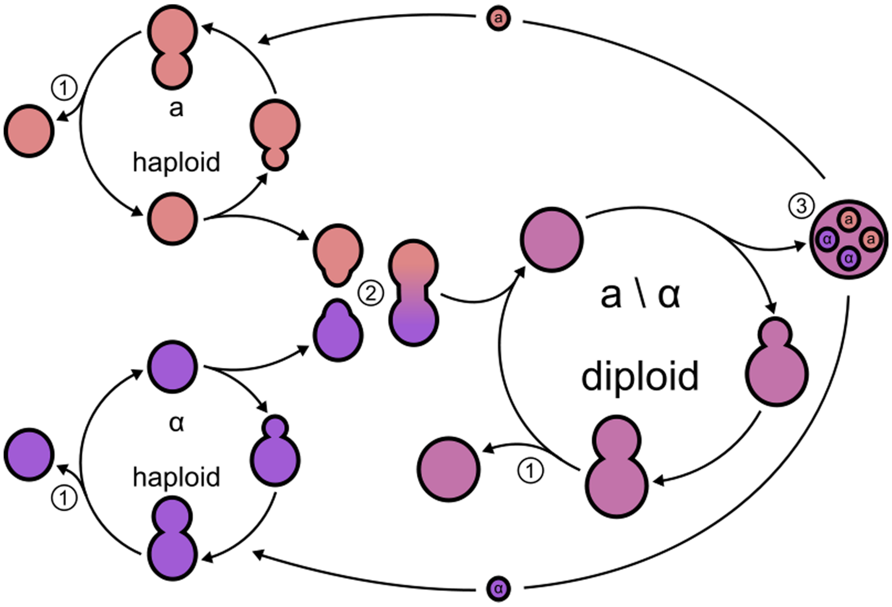

Figure 2.2: Life cycles and different states of budding yeast. 1: budding of a cell into two cells; 2: Mating of two haploid cell into one diploid cells; 3: Sporulation or division of a diploid cell into four haploids. From wikipedia.com

types. Two haploid cells of opposite cell type (**a** and α) can fuse together and form a diploid cell. This fusion is called *mating*.

Haploid cells of a particular type can always divide (via budding) to form more haploid cells of the same type. A haploid cell can also switch its type during cell division, namely, an α cell can switch to an **a** cell or vice versa. These cells can they mate with other haploid and form diploids. The process of mating type switching allows even an isolated haploid budding yeast to give rise to a colony of mostly diploid cells.

The cell type and type switching during cell division in haploids are associated with three regions on chromosome III. The cell type is determined by the genes contained in the *MAT* locus (mating type locus). For α type cells, $\alpha 1$ and $\alpha 2$ genes are at *MAT* locus. Instead **a** type cells have **a1** and **a2** genes. The *MAT* locus is always active. There are two other regions on the chromosome called

HML and *HMR* (hidden mating type loci), which are always silenced. *HML* contains a copy of the α genes, whereas, *HMR* contains the **a** genes. These two loci save a silenced copy of genetic information for both mating types. However, the particular set of the genes at *MAT* locus determines the mating type. During the budding of a new haploid cell, one of the genes from *HML* or *HMR* loci gets copied to the *MAT* locus, which is the reason why the mating type can change.

If the silencing at *HML* and *HMR* gets disrupted, the haploid cell will express both **a**-specific and α-specific genes. This leads to the production of **a1**-$\alpha 2$ heterodimer complex which can be found in diploids as well. This complex represses the transcription of haploid-specific genes (e.g. genes necessary for pheromone production). Such haploid cells are unable to mate with other haploids. In particular, they are resistant to pheromone of the opposite type. This defective behavior can be used in experiments to detect any disruption in the repression of *HML* and *HMR* loci.

In addition to *HML* and *HMR* loci, there are other parts of the genome which are silenced. One example is telomeric silencing [26, 27]. Telomeres are the regions located at the two ends of each chromosome. Telomeric silencing is not specific to yeast, rather, it is the case for all eukaryotic genomes. In the case of yeast, the mechanism and the proteins involved in the silencing of both telomeres and cell type related regions are similar. Below, I present the biological model of silencing for these regions.

2.1.4 A mechanism for silencing: nucleation & spreading

Different models have been proposed for silencing in different organisms and even for different regions of the genome in one organism [28]. However, there is some similarity between these mechanisms [29]. In general, whether a region of chromosome is in the heterochromatin or euchromatin state depends on the type of modification of histone proteins in the nucleosomes of the corresponding region.

Here, we will discuss one of the silencing models which applies to *HML*, *HMR* and telomeric silencing in yeast.

In this system, nucleosomes in silenced regions are bound by three proteins: Sir2p, Sir3p and Sir4p. These proteins form a complex named *Silenced Information Regulator* (SIR) complex. Also, in silenced regions, acetyl group from particular lysines (K) in histone tails are removed. Histone acetylation is normally associated with transcriptionally active regions. One of the main sites of acetylation is H4K16, a lysine at position 16 on the amino tail of histone H4.

It is believed that the silencing originates from a nucleation center which recruits histone modifying enzymes, specifically certain histone deacetylases. These enzymes modify neighboring histone tails to create a binding site for the SIR complex. Sir2p, a member of the complex, is a histone deacetylase which modifies the neighboring histones and provides more binding sites for SIR complex. In this manner, several rounds of histone modifications and SIR binding results in the spreading of the silenced region. There are some other proteins which work in an opposing way to the silencing propagation. Particularly, Sas2, a histones acetyl transferase, attaches acetyl groups to certain lysine in histone tails and prevents SIR complex binding [30, 31].

Although the silencing nucleation step for telomeric and *HML*/*HMR* regions involve somewhat different sets of proteins, the spreading step seems to be similar [32]. Another difference is that for *HML*/*HMR* loci, the nucleation center exists on both end of the silenced regions, whereas, for telomeres, there is only one nucleation site.

One immediate question is what stops the spreading of the silenced region? There are two possible scenarios. One is that there are some explicit boundary elements (e.g. strong gene promoters) stopping the propagation [34, 35]. The other possibility is that, because of finite supply of SIR complex, a stationary state between silenced and unsilenied region is reached. I will explain what I

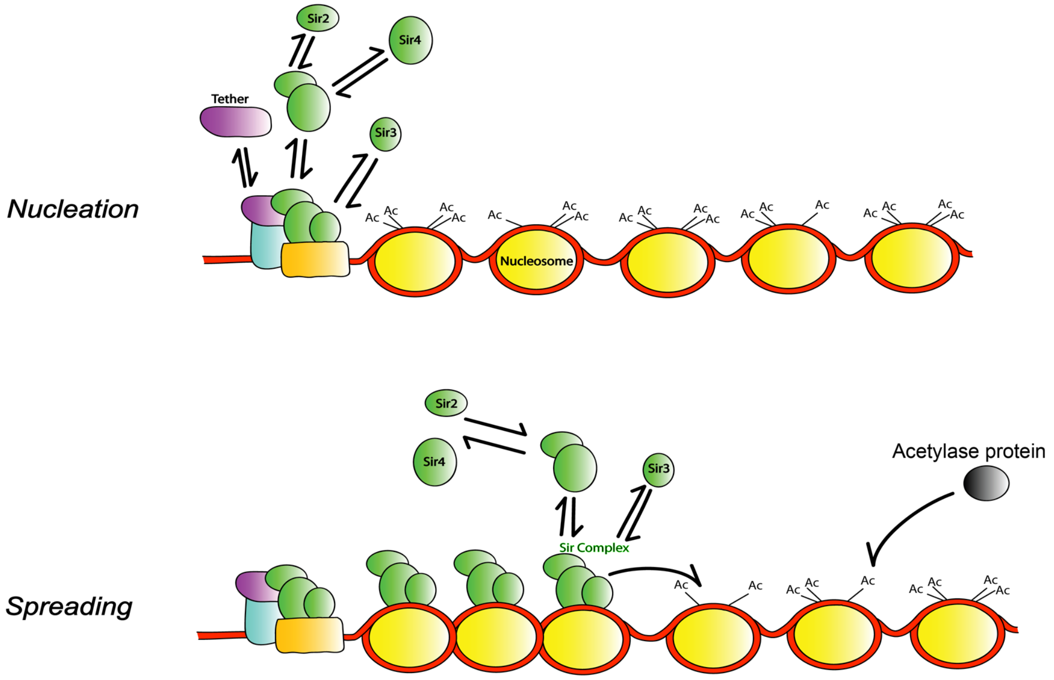

Figure 2.3: Biological model of silenced chromatin domain formation in yeast. The nucleation site initiates the process by recruiting specific histone modifying enzymes, which then modify neighboring histones. The modified histones allow binding of Sir complex. This complex, in turn, modifies the neighboring histones and provides more binding sites for Sir complex. Consecutive rounds of modication and binding result in the stepwise spreading of silencing complexes along the chromosome. From [33].

mean by this last sentence in the following section, once a mathematical model for the above mechanism and the corresponding bifurcation diagram is presented. Before getting there, let me give an overview of some experimental knowledge about silencing in yeast.

2.1.5 Experimental observations

As we mentioned, the silencing in the *HML* and *HMR* loci initiates from a nucleation center. A protein named Sir1 seems to play a role in the nucleation step. In one of the early experiments done in 1989 by L. Pillus and J. Rine [36], it was found that in *sir1* mutants (where the nucleation effect is defective if not absent), a population of yeast cells is divided into two distinguishable groups. In one group, composed of around 20% of the population, *HML* and *HMR* loci are silenced similar to normal cells. In the other group, those loci are active and cells would not mate like a normal haploids. Both of the epigenetic states (silenced vs active) are quite stable and are inherited most of the time during cell division. In fact, it was observed that switching from active to silenced state occurs approximately once in every 250 cell divisions, or with the small probability of $4 * 10^{-3}$. This observation suggests that the system can be thought of as being in a bistable regime, where two stable states can exist under the same conditions.

Another experimental fact comes from the mutants of *SAS2*, the gene encoding a protein responsible for acetylation of histones. In the current biological picture, the histone acetylation prevents SIR complex from binding to the histones [30, 31]. In other words, Sas2 activity is essential in preventing the silenced region from spreading into the active region. Therefore, one may expect that over-expression of Sas2 should cause the silenced region to shrink. This effect has been indeed observed. On the other hand, one may also expect that in Sas2 mutant (where acetylation is defective or absent) some active regions should turn silenced. Contrary to this expectation, it was found that the deletion of Sas2 decreases the level of silencing, rather than improving it [37]. In the absence of deacetylation activity, cells lose the bistability at mating-type loci and demonstrates an intermediate state which is neither silenced nor fully active [38]. This intermediate state can be considered as a promiscuous silenced state where SIR proteins can be bound at random places along the DNA.

We have performed some experiments on different yeast mutants as well. Our experiments were motivated by qualitative predictions of our mathematical model of silencing in yeast. We will study this model and the experimental results in the subsequent sections.

2.2 Mathematical Model of the Silencing Mechanism

I present a stochastic model for the process of silencing introduced in the previous section. I give a mean field formulation to describe the state of the system and analyze the conditions under which it becomes bistable, allowing different epigenetic states. Many of the materials presented in this section have received a similar treatment in [33].

2.2.1 Stochastic equation

One can think of a chromosome as a 1 dimensional lattice, where each site corresponds to a nucleosome (or a histon core complex). Each site i can be in one of four possible states:

- Bound by silenced proteins with probability S_i
- Not bound by any proteins with probability E_i
- Bound by one acetyl group with probability A_i
- Bound by two acetyl groups with probability AA_i

Figure 2.4 shows these possible states and the transition rates between them. The rate of SIR binding, which is a function of the concentration of ambient SIR proteins, is denoted by ρ. Free Sir2p, Sir3p and Sir4p proteins in the environment do not form SIR complex. Instead, they form the complex when they are attached to a nucleosome[1]. In the case where each protein is in low abundance, ρ is proportional to the product of the three concentrations for Sir2p, Sir3p and Sir4p. For our analysis, we will not need to know the exact form of dependence of ρ. We will just keep it as an effective parameter, monotonically increase with the concentration of each of SIR proteins.

The histone acetylation rate, caused by Sas2 activity, is represented by α. The rate at which SIR complex fall off the nucleosomes is shown by η. Also, the basal rate at which acetyl group falls off the nucleosomes is denoted by λ. The deacetylation rate increases if adjacent sites are in the silenced state. This increase is given by the term $\Gamma_{ij}S_j$, where Γ_{ij} is a function of $|i-j|$ and drops significantly as this separation increases. All the above parameters may be position and/or time dependent. However, for the sake of brevity, this dependence is not explicitly written.

We have included a double acetylation state. One justification could be that in each nucleosome, there are two lysine tails (H4k16) which host the main binding sites for acetyl group. However, there is no evidance that the chemical process of acetyl binding to these tails involves cooperativity. For example, it could be that one of them can be bound by an acetyl group whereas the other one is bound by SIR complex. The reason we insist our model to have a double acetylation state is to get bistability (see below). One reasonable scenario is that the incorporation of cooperativity (via inclusion of double acetylated state) originates from the fact that some other players and degrees of freedom (e.g. certain methylation marks)

[1] This sentence is not intended to imply anything on the exact order of various proteins attachment and multimerization

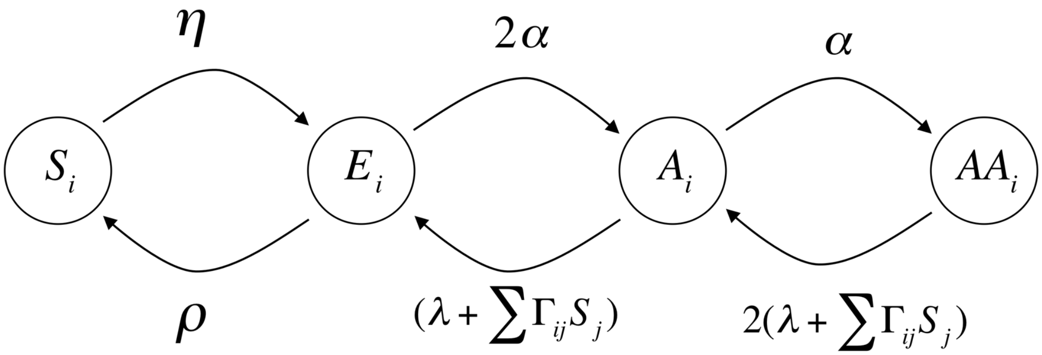
η
2α
α
S_i
E_i
A_i
AA_i
ρ
$(\lambda + \sum \Gamma_{ij} S_j)$
$2(\lambda + \sum \Gamma_{ij} S_j)$

Figure 2.4: Four possible states for each nucleosome. Chromosome is modeled as a 1 dimensional lattice, where each site corresponds to one nucleosome (see Figure 2.3). Site i can be either in silenced (S_i), unbound (E_i), monoacetylated (A_i) or double-acetylated (AA_i) state. The deacetylation rate depends on the silencing state of neighboring nucleosomes (the term $\Gamma_{ij}S_j$).

are not included in this model.

$$\begin{aligned}
\frac{dS_i}{dt} &= \rho E_i - \eta S_i \\
\frac{dE_i}{dt} &= (\eta S_i - \rho E_i) + ((\lambda + \Gamma_{ij}S_j)A_i - 2\alpha E_i) \\
\frac{dA_i}{dt} &= 2\alpha\, E_i + 2(\lambda + \Gamma_{ij}S_j)AA_i - (\alpha + \lambda + \Gamma_{ij}S_j)A_i \\
\frac{dAA_i}{dt} &= \alpha A_i - 2(\lambda + \Gamma_{ij}S_j)AA_i
\end{aligned} \tag{2.1}$$

2.2.2 Uniform solutions

We consider uniform steady state solutions for the set of equations 2.1, namely, we drop the subscript i and put the left hand side of the equations equal to zero.

In this section, we analyze the system with constant parameters. Let us define $\gamma = \Sigma_j \Gamma_{ij}$. This quantity is independent of i, hence, the drop of the subscript. Using the above notation, the uniform solutions of the set of equations 2.1 has to satisfy:

$$\begin{aligned} 0 &= \rho E - \eta S \\ 0 &= (\eta S - \rho E) + ((\lambda + \gamma S)A_i - 2\alpha E) \\ 0 &= 2\alpha\, E + 2(\lambda + \gamma S)AA - (\alpha + \lambda + \gamma S)A \\ 0 &= \alpha A - 2(\lambda + \gamma S)AA \end{aligned} \tag{2.2}$$

By eliminating the variables in the above equations, we find:

$$E = \frac{\eta}{\rho}\, S; \qquad A = \frac{2\,\alpha\,\eta}{\rho\,(\lambda + \gamma S)}\, S; \qquad AA = \frac{\alpha^2\,\eta}{\rho\,(\lambda + \gamma S)^2}\, S\ . \tag{2.3}$$

Let us also redefine the parameters as follow:

$$\rho = \frac{\rho}{\eta}; \qquad \alpha = \frac{\alpha}{\lambda}; \qquad \gamma = \frac{\gamma}{\eta}\ . \tag{2.4}$$

Since sum of the probabilities has to be 1, we have:

$$S\left(1 + \frac{1}{\rho} + \frac{2\,\alpha}{\rho\,(1 + \gamma S)} + \frac{\alpha^2}{\rho\,(1 + \gamma S)^2}\right) = 1\ ;$$

which we can rewrite as:

$$S = \frac{\rho\,(1 + \gamma S)^2}{[(1 + \rho)\,(1 + \gamma S)^2 + 2\alpha(1 + \gamma S) + \alpha^2]}\ . \tag{2.5}$$

Figure 2.5 shows the graph of left hand and right hand side of the above equation, as a function of S, for a few different combination of the parameters. The above equation is of degree 3 and can have maximum of three real roots. For certain set of parameters, there is only one real solution (Figure 2.5A-B). This situation is referred to as monostable. For relatively small values of α (or high values of γ or ρ), this solution happens at high S (Figure 2.5A), whereas, for relatively high values of α, the solution is at low S (Figure 2.5B). There is also a regime of parameters where there are three real solutions (Figure 2.5C). The middle solution is unstable, whereas, the other two solutions at low and high values of S are stable. We will denote these two stable solutions by S_l and S_h, respectively (See Appendix A for more detail). When the parameters allow us to have two stable solutions, we are in the bistable regime. As we play with the parameters, for example by increasing/decreasing γ, two of the three solutions merge together (Figure 2.5D). At this point, the graphs are tangent to each other. If we continue increasing/decreasing γ, we fall in the one solution regime, as in Figure 2.5A and B.

For each set of parameters α, ρ and γ, we would like to be able to characterize how many solutions exist. The bifurcation diagram, helps to visualize this characterization.

Bifurcation diagram

As we mentioned, by changing the parameters continuously, we can switch between the two regimes of monostability and bistability. At the transition between these two state, the two curves in Figure 2.5 touch each other at a point. This is the point where two of the solutions merge and disappear or a degenerate solution appears and eventually give rise to two solutions, depending on the direction that we change the parameters. At this point, not only the equation 2.5 is satisfied, but also the derivative of both sides with respect to S should be equal. Let us

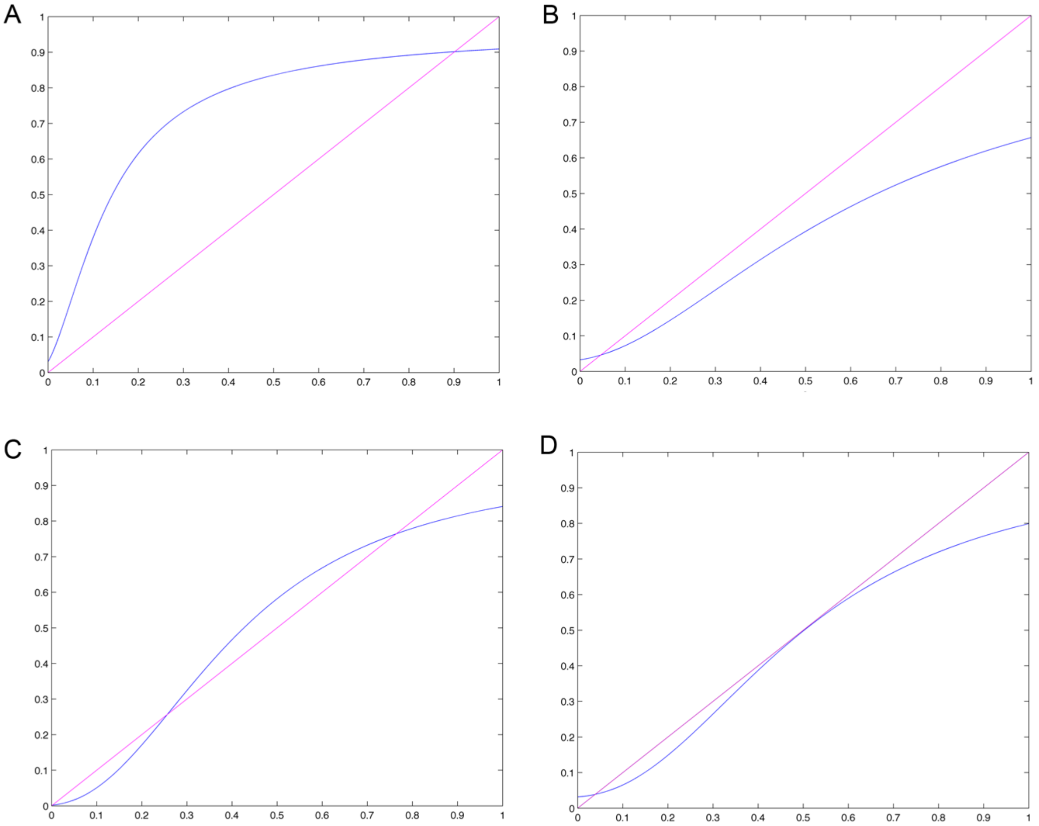

Figure 2.5: The intersections of nullcline curves. Graphs shows the left hand side (magenta) and the right hand side (blue) of equation 2.5, for different sets of parameters α, ρ and γ.

rewrite equation 2.5 as:

$$\rho\ (1+\gamma S)^2 = S\left[(1+\rho)\ (1+\gamma S)^2 + 2\alpha(1+\gamma S) + \alpha^2\right]\ . \tag{2.6}$$

Putting the derivative of both sides equal, and using equation 2.6, we get:

$$2\rho\ \gamma S\ (1+\gamma S) = \rho\ (1+\gamma S)^2 + S^2\left[2\gamma(1+\rho)\ (1+\gamma S) + 2\alpha\gamma\right]\ . \tag{2.7}$$

This implies for the transition point:

$$\alpha = (1+\gamma S)\left[\frac{\rho(1-S)}{S} - 1 - \frac{\rho\ (1+\gamma S)}{2\gamma\ S^2}\right]\ . \tag{2.8}$$

After replacing α in 2.6 by the above equation, and dividing both sides by $S(1+\gamma S)^2$, we get:

$$\frac{\rho(1-S)}{S} = \left[\frac{\rho(1-S)}{S} - \frac{\rho\ (1+\gamma S)}{2\gamma\ S^2}\right]^2 \Longrightarrow \sqrt{\frac{\rho(1-S)}{S}} = \frac{\rho(1-S)}{S} - \frac{\rho\ (1+\gamma S)}{2\gamma\ S^2}\ .$$

The reason we take the positive root is because $\alpha > 0$; therefore, the term in the bracket in equation 2.8 is positive. We can solve the above equation for γ:

$$\gamma = \left(S - 2S^2 - \sqrt{\frac{4(1-S)S^3}{\rho}}\right)^{-1}\ . \tag{2.9}$$

We can replace the above equation in 2.8 to get:

$$\alpha = \frac{\left(2(1-S) - \sqrt{4\rho^{-1}(1-S)S}\right)\left(\sqrt{\rho S(1-S)} - S\right)}{S\left(1 - 2S - \sqrt{4\rho^{-1}(1-S)S}\right)}\ . \tag{2.10}$$

In Equations 2.9 and 2.10, for each value of ρ, S can take any value between 0 and 1, as long as both α and γ are positive real numbers.

There is one more case that we did not mention and is not shown in 2.5. It is possible to have a situation where all three solutions merge together. This case is similar to 2.5D, with the difference that the two curves intersect only at one point. To be in such a situation, in addition to Equations 2.6 and 2.7, the second derivative of both sides of the Equation 2.6 with respect to S have to be equal. Putting the second derivatives equal, and using Equation 2.6 and equation 2.7, we get:

$$S_C = \frac{\gamma\rho - 2\alpha - 2(1+\rho)}{3\gamma(1+\rho)} \ . \tag{2.11}$$

The subscript C is meant to indicate the critical point. Note that the parameters in the above equation have to satisfy Equations 2.6 and 2.7 as well.

Equations 2.9 and 2.10 are the consequence of the two conditions 2.6 and 2.7 that we have imposed. Instead of solving for α and γ, we could have chosen to solve for α and ρ, or ρ and γ. If we solve for ρ and γ in equations 2.6 and 2.7, we find:

$$\gamma = \frac{(\alpha - 2(1+\alpha S))}{2S} \pm \sqrt{\left[\frac{(\alpha - 2(1+\alpha S))}{2S}\right]^2 - \frac{(1+\alpha)}{S^2}} \ , \tag{2.12}$$

$$\rho = \frac{4(1-S)S^3}{(S - 2S^2 - \gamma^{-1})^2} \ , \tag{2.13}$$

where γ in the second equation has to be replaced from the first one.

Equations 2.9 and 2.10 can be used to draw a plane in the three dimensional ρ - α - γ coordinates (note that S can be replaced by any value between 0 and 1, as long as parameters remain positive real numbers). In fact, Equations 2.12 and 2.13 gives us exactly the same plane in the 3-dimensional coordinates. This plane separate the the two regimes of monostability and bistability. It is convenient to draw the intersections of this plane with, for example, the constant ρ or the constant α surface. To get the former one, we should keep ρ in Equations 2.9 and 2.10 constant. Instead, for the later case, we should keep α in Equations 2.12 and 2.13 constant. In the next section, we find it convenient to work with Equations

2.12 and 2.13. However, for now, we stick with Equations 2.9 and 2.10.

Figure 2.6 shows the bifurcation diagram in the α - γ plane (constant ρ). The correspondence between this diagram and Figure 2.5 is as follow. The monostable silenced region corresponds to Figure 2.5A; monostable active to Figure 2.5B; Bistable to Figure 2.5C; the magenta to Figure 2.5D. At the cusp is the critical point, where three solutions merge together (Equation 2.11). Figure 2.7 show how this curve moves as one increases ρ.

One might ask, why the critical point defined by the condition used to get Equation 2.11 is actually located at the cusp of the magenta curve in Figure 2.6 and not at any other point along the magenta curve? From Equations 2.9 and 2.10, we have:

$$\gamma = \gamma(S) \quad \& \quad \alpha = \alpha(S) \ , \tag{2.14}$$

where the dependence on ρ has not been written. The cusp is located at the point where

$$\frac{\partial \gamma}{\partial S} = \frac{\partial \alpha}{\partial S} = 0 \ . \tag{2.15}$$

There one has to show that these equalities are satisfied at the critical point. Let us rewrite Equation 2.5 as:

$$f(\gamma, \alpha, S) = 0 \ , \tag{2.16}$$

where f is a function and we have not written the dependence on ρ. The condition for the first derivative to be zero implies:

$$\frac{\partial f(\gamma, \alpha, S)}{\partial S} = 0 \ . \tag{2.17}$$

Equations 2.16 and 2.17 result in2.14. We can rewrite 2.16 and 2.17 as:

$$f(\gamma(S), \alpha(S), S) \ = \ 0 \ \Rightarrow \frac{\partial f}{\partial \gamma} \frac{\partial \gamma}{\partial S} + \frac{\partial f}{\partial \alpha} \frac{\partial \alpha}{\partial S} = 0 \ , \tag{2.18}$$

$$\frac{\partial f(\gamma(S), \alpha(S), S)}{\partial S} \ = \ 0 \ \Rightarrow \frac{\partial^2 f}{\partial \gamma \partial S} \frac{\partial \gamma}{\partial S} + \frac{\partial^2 f}{\partial \alpha \partial S} \frac{\partial \alpha}{\partial S} + \frac{\partial^2 f}{\partial S^2} = 0 \ . \tag{2.19}$$

In the first equation, we have used the fact that $\frac{\partial f}{\partial S}$ is always zero on the bifurcation line. In addition, at the critical point, $\frac{\partial^2 f}{\partial S^2}$ is also zero. Therefore, the above equations become:

$$\frac{\partial f}{\partial \gamma} \frac{\partial \gamma}{\partial S} + \frac{\partial f}{\partial \alpha} \frac{\partial \alpha}{\partial S} = 0 \ , \tag{2.20}$$

$$\frac{\partial^2 f}{\partial \gamma \partial S} \frac{\partial \gamma}{\partial S} + \frac{\partial^2 f}{\partial \alpha \partial S} \frac{\partial \alpha}{\partial S} = 0 \ . \tag{2.21}$$

We see that the condition 2.15 satisfies the above requirement for the critical point (one can satisfy himself that the determinant of the coefficients is not zero, e.g. $\frac{\partial f}{\partial \alpha}$, $\frac{\partial^2 f}{\partial \alpha \partial S} > 0$). This implies that the critical point is indeed at the cusp of the bifurcation curve in Figure 2.6.

In the bistable regime, there are two stable solutions. Given that we are dealing with an stochastic system, we should really call these metastable solutions, in the sense that for a real system with noise, there is a possibility of switching between the two state. We would like to know, for each part of the bistable regime, which of the two solutions are more stable. Another interesting question is whether it is possible to have different parts of the lattice to be in different states (silenced vs active) and these states can have a stable boundary. The subject of the next chapter is addressing such issues.

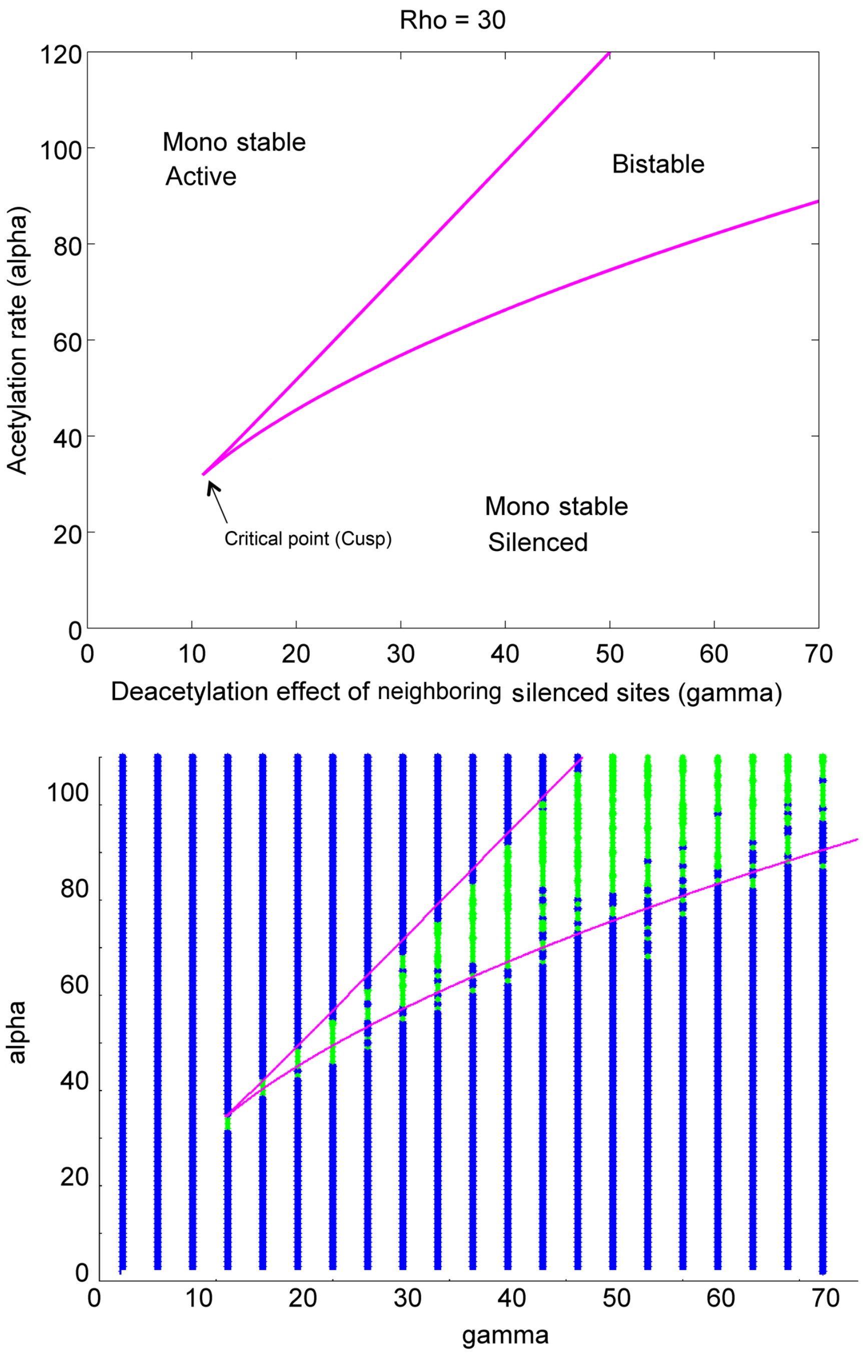

Figure 2.6: Bifurcation diagram in the ρ constant surface. The magenta line is obtained using Equations 2.9 and 2.10. The blue and green points in the lower panel shows the result of stochastic simulation where the system is simulated using two initial conditions (high and low S). Monostable and bistable regimes can be differentiated based on whether the two initial conditions lead to one (blue) or two (green) different states.

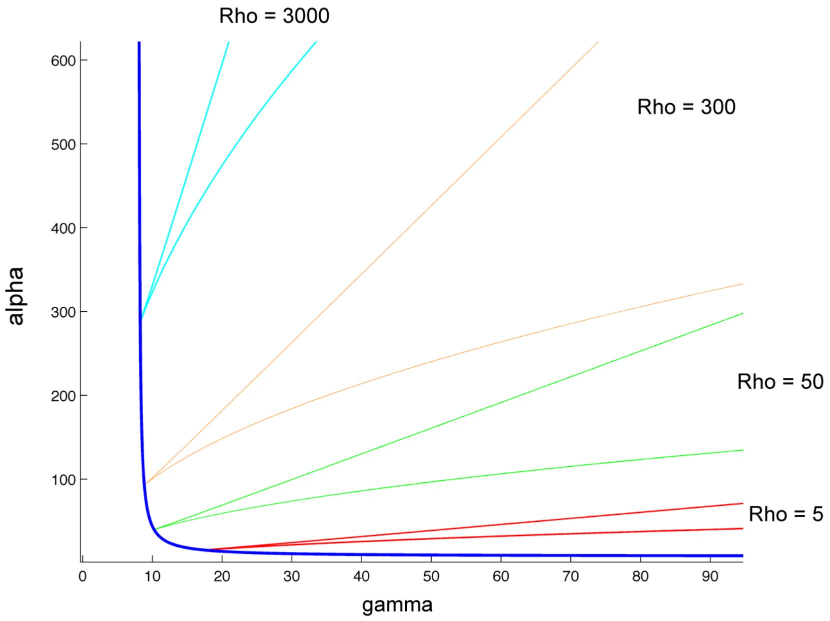

Figure 2.7: Position of the cusp point as a function of gamma (blue curve).

2.2.3 Non-uniform solutions, continuum limit

Since we are dealing with an spatially extended system, in addition to uniform solutions, we would like to explore the possibility of having non-uniform spatial solutions (i.e. coexistence of different domains) for the parameter sets located in the bistable regime. In particular, we will be interested in the dynamics of the front between silenced and active domains.

For a system with N nucleosomes, there are 4^N possible distinct states. We cannot directly solve the time-independent solutions of the stochastic system given by the set of equations 2.1. Therefore, we will resort to the continuum limit approximation[2]. In this limit, we have:

$$\begin{aligned}
\frac{dS(x)}{dt} &= \rho E(x) - \eta S(x) \ , \qquad (2.22)\\
\frac{dE(x)}{dt} &= \eta S(x) - \rho E(x) - 2\alpha E(x) + \left(\lambda + \int \Gamma(x-y)S(y)dy\right) A(x) \ ,\\
\frac{dA(x)}{dt} &= 2\alpha \ E(x) + 2\left(\lambda + \int \Gamma(x-y)S(y)dy\right) AA(x) -\\
& \left(\alpha + \lambda + \int \Gamma(x-y)S(y)dy\right) A(x) \ ,\\
\frac{dAA(x)}{dt} &= \alpha A(x) - 2\left(\lambda + \int \Gamma(x-y)S(y)dy\right) AA(x) \ .
\end{aligned}$$

We can Taylor expand $S(y)$ around x. Since $\Gamma(x-y)$ falls sharply as $|x-y|$ increases, we will only keep up to the second order in the exapnsion:

$$\begin{aligned}
\int \Gamma(x-y)S(y)dy &= \int \Gamma(x-y)\left(S(x) + (y-x)\frac{dS(X)}{dx} + \frac{(y-x)^2}{2}\frac{d^2S(X)}{dx^2} + ...\right) dy\\
&\simeq \gamma S(X) + \gamma_2 \frac{d^2S(X)}{dx^2} \ . \qquad (2.23)
\end{aligned}$$

The second term in the Taylor expansion disappears since $\Gamma(x-y)$ is symmetric. We have also defined:

$$\gamma = \int \Gamma(x-y)dy \quad \& \quad \gamma_2 = \int \Gamma(x-y)\frac{(y-x)^2}{2}dy \qquad (2.24)$$

Replacing 2.23 in the set of equation 2.22 and simplifying the equation we arrive at:

$$\gamma_2 \frac{d^2S(X)}{dx^2} = -1 - \gamma S(X) + \alpha \ \frac{S(X) + \sqrt{\rho S(X)(1-S(X))}}{\rho(1-S(X)) - S(X)} \ . \qquad (2.25)$$

[2] Note that, given our system, there is not really a parameter which gives the continuum limit as it converge to some limit.

If we define:

$$V(S) = +S + \frac{\gamma}{2}S^2 - \alpha \int^S \frac{S' + \sqrt{\rho S'(1-S')}}{\rho(1-S') - S'} dS' \ . \tag{2.26}$$

then Equation 2.25 can be written in the following form:

$$\gamma_2 \frac{d^2 S(X)}{dx^2} = -\frac{dV(S)}{dS} \ . \tag{2.27}$$

The similarity between 2.27 and the formula for the motion of a particle in a potential field in classical mechanics is clear.

For the two uniform stable solutions of Equations 2.22, S_l and S_h, the right hand side of Equations 2.25 and 2.25 is zero. At those points, the potential V, defined in Equation 2.26, is flat $(dV(S_l)/dS = dV(S_h)/dS = 0)$. Using this equation, we can numerically calculate the value of V, for the points between the two stable solutions. Figure 2.8 shows the result of numerical integration for different combination of parameters within the bistable regime.

Coexistence of different domains

We are looking for a solution which starts from one of the stable solutions (e.g. S_l) and ends in the other solution (e.g. S_h). From biology point of view, this case is of special interest. As we mentioned in the previous section, heterochromatin and euchromatin domains can occupy close by regions along the DNA without clear boundary element stopping them from invading into each other. An example would be the region around the boundary of telomeres.

From our experience in classical mechanics with equations in the form of 2.27, we now that the necessary condition is (Figure 2.8B):

$$V(S_l) = V(S_h) \ . \tag{2.28}$$

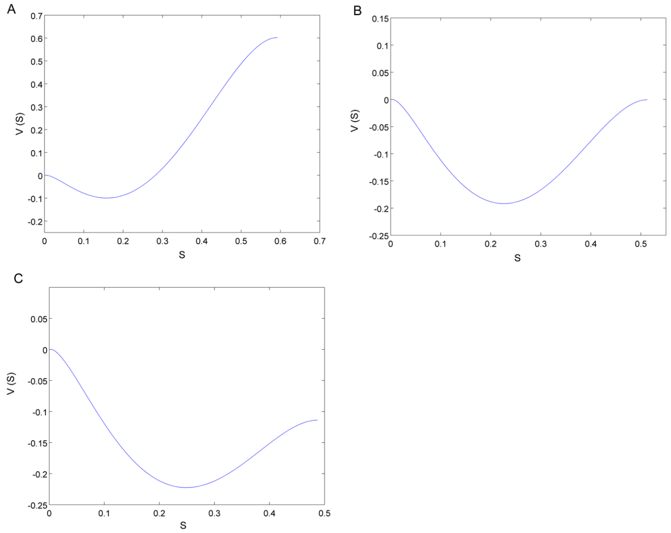

Figure 2.8: Potential $V(S)$ for different combination of parameters in the bistable regime. At each point in the bistable regime, there are two stable solutions for Equation 2.22, S_l and S_h. The graph is only drawn for values of S between these two solutions. Note that, we are able to use this potential only to describe zero-velocity fronts, and not for the general traveling solution.

We would like to characterize the points in the bistable regime which satisfy the above condition. It turns out that in the bifurcation diagram in the α - γ plane (constant ρ), this conditions define a line that we will call *zero-velocity* line. Figure 2.9 show the zero velocity line. This line starts from the critical point and divides the bistable regime into two sections. In the lower part, close to the monostable silenced regime, the S_h solution is more stable than the S_l one. This means, if we start from a non-uniform solution, the domain associated to the S_h solution invades into the active domain. The opposite happens in the upper section. In summary, the coexistence of different domains is possible only if we are around the zero-velocity line. Otherwise, in region I or II of Figure 2.9, the front between two domains is unstable and moves in the direction of the favorite state.

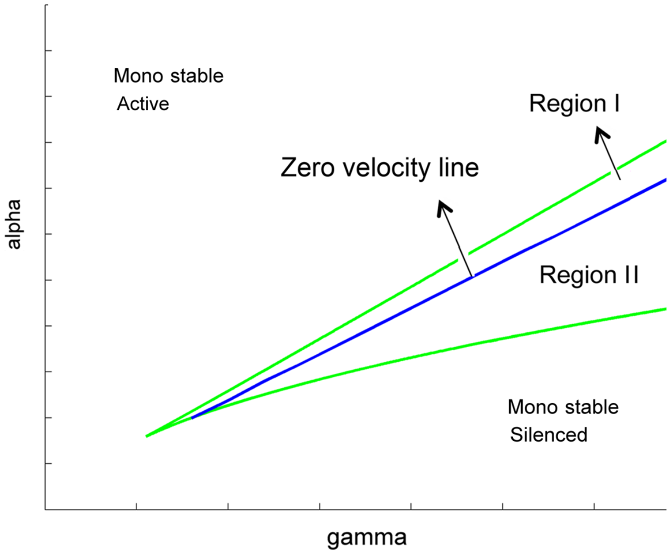

Figure 2.9: Zero-velocity line subdividing the bistability region. The correspondence with Figure 2.8 is as follow: the case shown in Figure 2.8B is located on the zero-velocity line; the one in Figure 2.8A is located in Region II; and finally the case in Figure 2.8C is in Region I. The slope of the blue line has been manually reduced slightly so that Region I will be more clear.

The effect of lattice discreteness on domain boundary

In the above discussion on coexistence of different domains, we considered a continuum system. One might wonder how our results would change if we had, instead, studied a discrete lattice model. To get insight into this, we simulated the stochastic system. As one may have expected, in the discrete version, the zero-velocity line broadens into a band of propagation failure [39, 33]. In the stochastic version of the model, within this band, the boundary seems to fluctuate without any noticeable drift. In addition, even for very large values of the parameters (α, γ and ρ), the time scale of fluctuation in the boundary position is quite slow. One of our future plans is to have a theoretical estimate on the relation between boundary fluctuation and the parameters of the system.

2.2.4 Where is real system located on the bifurcation diagram?

Let us go back and see whether the model of stepwise spreading of silencing introduced in the previous section fits into our mathematical description. First, we will assume all the parameters are constant.

In Region I of Figure 2.9, we do not expect the silenced domain to spread from the nucleation center. Instead, this domain should be localized around the nucleation center. In contrast, in Region II, the silenced domain spreads out from the nucleation center. Although this behavior is similar to the stepwise spreading model, however, it requires an explicit boundary element to stop it from taking over the whole active domain.

In some heterochromatin (silenced) part of the DNA, e.g. telomeres, there does not seem to be an explicit boundary element stopping the spread of silenced domain. For example, by over expressing the SIR complex, the silenced domain invades into the active one to some extent and then stops again [40]. This implies that, instead of being fixed by some element, the boundary between the two domains is, in principle, dynamic. At first glance, this behavior seems to indicate that the system is actually on the zero-velocity line in Figure 2.9, which, in turn, implies that there is an stable dynamic boundary between the two domains.

Assuming the system is on the zero-velocity line raises two concerns. The first one is that being on this line requires fine tuning of the parameter. The other issue is that if one of the parameters changes, e.g. ρ increases because of over-expression of SIR complex, the system moves away from the zero-velocity line. This will cause one domain to invade the other domain. However, in reality, this invasion happens only to certain extent and the boundary stabilizes at a new place. So far, our mathematical description does not seem to capture this behavior.

In the above discussion, we assumed that all the parameters are constant. In particular, the available ambient concentrations of Sir complexes, reflected in ρ, was held constant. It turns out that by relaxing this assumption, not only our mathematical description captures the experimental observation (stable dynamic boundary between two domains), but also leads to some interesting prediction which we are exploring experimentally. In the next section, we will get into the detail of this subject.

2.3 Consequences of Finite Supply of Sir Proteins

One of the assumption was that the available ambient concentrations of Sir complexes, reflected in ρ, was constant. Instead, one can consider the case where the total number of SIR complexes, which is the sum of the complexes in the ambient and the ones bound to the nucleosomes, is fixed. In other words, there is a finite supply of SIR complexes:

$$\rho\ v + \sum_i S_i = S_{tot} = \text{constant}\ . \tag{2.29}$$

Here, v is proportional to the volume of the system. This equation means that whenever a SIR complex gets bound to the nucleosome, the ambient concentration of available complexes drops. Therefore, ρ is now a self-adjusting parameter, as opposed to being constant. We will see that there will be two implications from this assumption: boundary stabilization and coupling of different silenced regions on the genome. Before going forward, let us look at the bifurcation diagram from another angle.

As we mentioned, the bifurcation diagram is a surface in the three dimensional space formed by α, γ and ρ axis. For the convenience, so far we have chosen to look at the intersection of this surface with the constant ρ plane (formed by α - γ axis). For our discussion in this section, we change this choice and switch

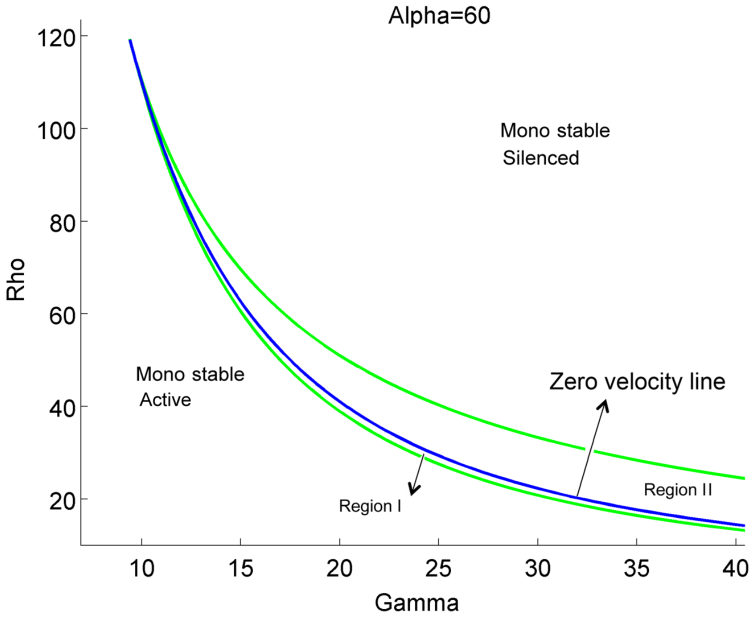

Figure 2.10: The bifurcation diagram in the ρ - γ plane (constant α). The correspondence between different regions of this graph and Figure 2.9 is shown on the picture

to constant α plane. The graph is shown in Figure 2.10. This diagram can be sketched using Equations 2.12 and 2.13.

2.3.1 Boundary stabilization without requirement for fine-tuning

Consider a system located in Region II of Figure 2.10 and assume there exist a small silenced domain or silencing has been initiated from a nucleation center. Being the favorite solution in Region II, this silenced domain invades into the active one. However, as silencing is spreading and SIR complexes get bound to the chromosome, the available SIR proteins in the environment reduces, namely, ρ drops. This means, on Figure 2.10, the system moves vertically downward and approaches the zero-velocity line. In this way, the system automatically goes on the zero-velocity line and the two silenced and active domains will have a stable boundary between them.

The same would have happened if we had started with a system in Region I with some sites in the silenced domain. This time, the silenced domain would shrink and the system moves upward in Figure 2.10 until it reaches the zero-velocity line. In the sense of the above discussion, the constraint 2.29 acts as a negative feedback on the perturbation to the system.

In conclusion, if the system has an stable free boundary between silenced and active domain, then it has to be on the zero-velocity line. The only way a system can be not on this line is if all the sites are in one domain. If a system is stably in Region I (Region II) of Figure 2.10, then all the sites will be in S_h state (S_l state).

For a given system with certain length (L), S_{tot}, v and γ, we want to determine where in the bifurcation diagram it is located. If the system is in the bistable regime, there are two possible states, $S_l(\rho, \alpha, \gamma)$ and $S_h(\rho, \alpha, \gamma)$. In the silenced monostable or active monostable region, there is only one possible solution, $S_m(\rho, \alpha, \gamma)$. Note that for fixed α and γ, these solutions are monotonically increasing function of ρ.

Consider a particular value of γ. Assume this value is high enough so that, for certain range of ρ, the system is in the bistable regime. In other words, γ is greater than $\gamma_{critical\ point}$ (for Figure 2.10 this value is around 10). Lets consider the following function.

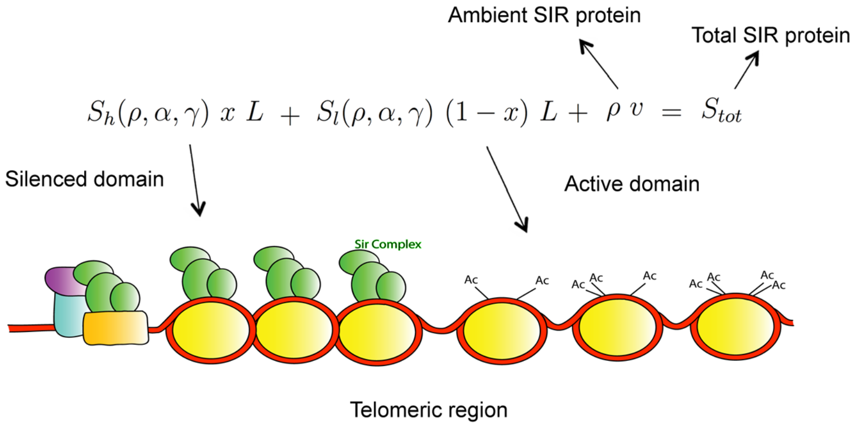
Ambient SIR protein
Total SIR protein
$S_h(\rho, \alpha, \gamma)\ x\ L\ +\ S_l(\rho, \alpha, \gamma)\ (1-x)\ L +\ \rho\ v\ =\ S_{tot}$
Silenced domain
Active domain
Sir Complex
Ac
Telomeric region

Figure 2.11: Constraint imposed by finite supply of SIR proteins. Adapted from [33].

$$\phi(\rho,\alpha,\gamma,x) = \begin{cases} \phi_1 = S_m(\rho,\alpha,\gamma)\ L + \rho\ v & \text{if} \quad \rho \text{ in active monostable region },\\ \phi_2 = S_l(\rho,\alpha,\gamma)\ L + \rho\ v & \text{if} \quad \rho \text{ in Region I of bistable regime },\\ \phi_3 = S_l(\rho,\alpha,\gamma)\ (1-x)\ L + & \text{if} \quad \rho \text{ on the zero-vecity line } (0 \leq x \leq 1)\ ,\\ \qquad S_h(\rho,\alpha,\gamma)\ x\ L + \rho\ v & \\ \phi_4 = S_h(\rho,\alpha,\gamma)\ L + \rho\ v & \text{if} \quad \rho \text{ in Region II of bistable regime },\\ \phi_5 = S_m(\rho,\alpha,\gamma)\ L + \rho\ v & \text{if} \quad \rho \text{ in silenced monostable region }. \end{cases} \tag{2.30}$$

The variable x represents the fraction of the system in the S_h domain. It is present only for a particular value of ρ for which the system is located on the zero-velocity line (See Figure 2.11). Note that ϕ is a monotonically increasing function of ρ and x.

For a fixed value of γ, α and S_{tot}, to determine what the configuration of a system is and where in the bifurcation the system is located, one has to:

$$\text{find } \rho(\alpha,\gamma) \text{ and } x(\alpha,\gamma) \text{ such that} \qquad \phi(\rho,\alpha,\gamma,x) = S_{tot}\ . \tag{2.31}$$

Note that, we should have really written $\rho(\alpha, \gamma, S_{tot}, L)$ and $x(\alpha, \gamma, S_{tot}, L)$, however, for the sake of brevity, we did not write the last two parameters. How can we calculate $\rho(\alpha, \gamma)$ and $x(\alpha, \gamma)$? Let us consider some particular values of ρ which are of interest. In the active monostable region, the minimum value of ρ is 0 and the maximum value happens when we touch the bifurcation line (green line) in Figure 2.10 from below. We call this value $\rho_{bu}(\alpha, \gamma)$ (b stands for bifurcation and u for active). Let us refer to the value of ρ on the zero-velocity line by $\rho_z(\alpha, \gamma)$. As we increase ρ, we hit the green line again, this time on the boundary between bistable regime and the silenced monostable one. Let us call this value $\rho_{bs}(\alpha, \gamma)$ (b stands for bifurcation and s for silenced). With this notation, we have: $\rho_{bu} < \rho_z < \rho_{bs}$. Using this notation and the definitions given in Equation 2.30, we have:

$$\phi_1(\rho_{bu}(\alpha, \gamma), \alpha, \gamma) < \phi_3(\rho_z(\alpha, \gamma), \alpha, \gamma, x = 0) < \phi_3(\rho_z(\alpha, \gamma), \alpha, \gamma, x = 1) < \phi_5(\rho_{bs}(\alpha, \gamma), \alpha, \gamma)\,. \tag{2.32}$$

Also, note that:

$$\phi_1(\rho_{bu}(\alpha, \gamma), \alpha, \gamma) = \phi_2(\rho_{bu}(\alpha, \gamma), \alpha, \gamma) \text{ and } \phi_4(\rho_{bs}(\alpha, \gamma), \alpha, \gamma) = \phi_5(\rho_{bs}(\alpha, \gamma), \alpha, \gamma)\,.$$

The first step in determining the configuration of a system and where in the bifurcation diagram it is located is to compare S_{tot} with the 4 values given in Equation 2.32. If $S_{tot} < \phi_1(\rho_{bu}(\alpha, \gamma), \alpha, \gamma)$, the system is located in the active monostable region. Similarly, if $S_{tot} > \phi_5(\rho_{bs}(\alpha, \gamma), \alpha, \gamma)$, the system is located in the silenced monostable region. If $\phi_1(\rho_{bu}(\alpha, \gamma), \alpha, \gamma) < S_{tot} < \phi_3(\rho_z(\alpha, \gamma), \alpha, \gamma, x = 0)$ The system will be in Region I of Figure 2.10. On the other hand, if $\phi_3(\rho_z(\alpha, \gamma), \alpha, \gamma, x = 1) < S_{tot} < \phi_5(\rho_{bs}(\alpha, \gamma), \alpha, \gamma)$, the system will be in Region II. For each of these regions, one can numerically solve the corresponding ϕ_i in Equation 2.30 for different values of ρ and find the one that satisfies constraint 2.31.

The only remaining case is when

$$\phi_3(\rho_z(\alpha,\gamma),\alpha,\gamma,x=0) < S_{tot} < \phi_3(\rho_z(\alpha,\gamma),\alpha,\gamma,x=1) \ ,$$

which corresponds to a system with domains of silenced and active regions coexisting with each other. The fraction of the system in the S_h domain is determined by satisfying:

$$S_l(\rho_z(\alpha,\gamma),\alpha,\gamma)\ (1-x)\ L\ + S_h(\rho_z(\alpha,\gamma),\alpha,\gamma)\ x\ L + \rho_z(\alpha,\gamma)\ v = S_{tot} \ ,$$

which, in turn, implies:

$$x = \frac{(S_{tot} - \rho_z(\alpha,\gamma)\ v - S_l(\rho_z(\alpha,\gamma),\alpha,\gamma)\ L)}{(S_h(\rho_z(\alpha,\gamma),\alpha,\gamma)\ L - S_l(\rho_z(\alpha,\gamma),\alpha,\gamma)\ L)} \ . \tag{2.33}$$

In summary, we showed how to calculate the self-adjusting parameter ρ, as well as the configuration of the corresponding system.

Self-adjusting path in the bifurcation diagram

Imagine we have a knob which allows us to play with the value of γ. In fact, experimentally, such a knob is available. By changing the concentration of nicotinamide (NAM), an inhibitor of Sir2p, one can effectively modulate γ [41]. We would like to know how a system changes as one varies the value of γ. Let us first consider the simple case where ρ is constant, as opposed to being a self-adjusting parameter. Figure 2.12A shows the path of such a system which is simply a horizontal line (magenta line). Figure 2.12B shows the fraction of this system in the high S solution. For this particular case, since the path is close to the cusp point, the size of the Region I and II is relatively very small. However, the shape of Figure 2.12B, up to a shift along the γ axis, is independent of how close or far from the cusp the path crosses the bistable regime. As long as we are in the monostable silenced regime or Region II (above the zero-velocity line) of the bistable regime, the system is in the high S domain, namely, the fraction is 1. On the zero-velocity line itself, the fraction can be any number. As soon as we cross the zero-velocity line into the Region I and monostable active regime, the system will be in the low S domain, namely, the fraction is 0.

How about when there is a finite supply of ρ and constraint 2.29 is in action? Figure 2.12C and D show the results. Basically, the magenta line and the pink line in Figure 2.12C and D are, respectively, the functions $\rho(\alpha, \gamma)$ and $x(\alpha, \gamma)$ satisfying the Equation2.31. For very large values of γ, all the sites are in the high S solution and the system is either in the monostable silenced regime (not shown in the picture) or Region II of the bistable regime. As one decreases γ, the system hits the zero-velocity line and the fraction x start to drop to values lower than 1. Depending on the parameters, Figures 2.12B and C could have looked different. For example, on the zero-velocity line, the fraction x is not necessarily a monotonically increasing function of γ.

2.3.2 Coupling different regions via ambient SIR concentration

We want to analyze a situation which is inspired by our model system, budding yeast. Each of the 16 chromosomes in a haploid yeast has 2 telomeric regions, one at each end. In addition, there are two regions named *HML* and *HMR* located on chromosome III. The *HML*/*HMR* loci are relatively small in size (~ 10 sites). In both *HML*/*HMR* loci and telomeres, silencing is initiated by nucleation centers. One important difference is that *HML* and *HMR* loci are surrounded by boundary elements stopping the silencing domain from spreading. On the other hand, telomeric regions have free boundary between silenced and active domains.

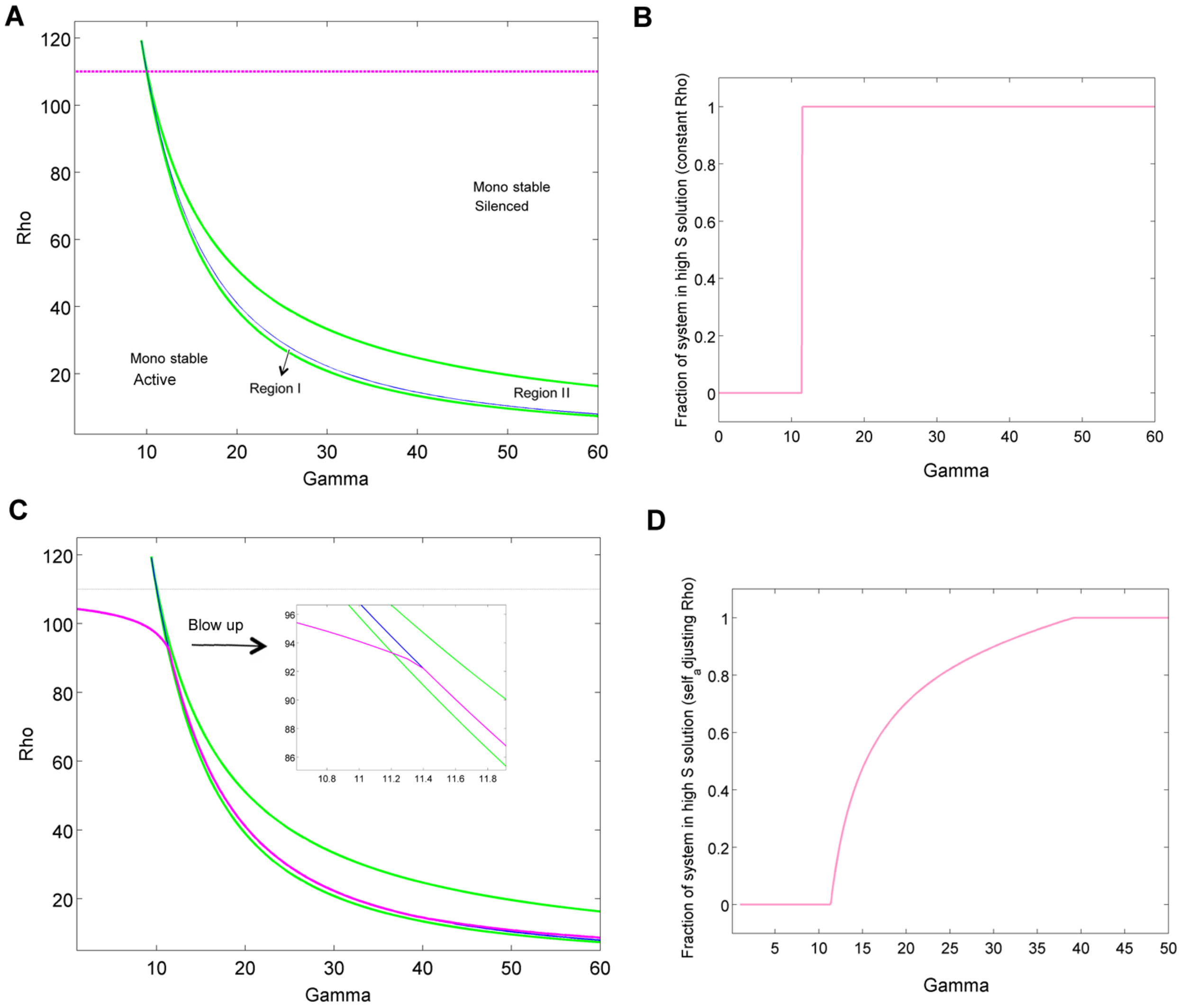

Figure 2.12: The self-adjusting path in the bifurcation diagram as γ varies. A) Assuming ρ is constant. B) Fraction of the system in the high S solution for (A). C) Assuming there is finite supply of ρ (constraint 2.29). D) Fraction of the system in the high S solution for (C). For all graphs, we chose $L = 200$ sites, $S_{tot} = 110$, $\alpha = 60$ and $v = 1$.

Our goal is to study the effect of variation in γ on this system. Since HML/HMR loci are small in size, let us ignore their contribution to the constraint 2.29. Telomeric regions have free boundary and from our discussion in the previous section, we know how to determine the self-adjusting path in the bifurcation diagram or equivalently the self-adjusting parameter $\rho(\alpha, \gamma)$. This parameter is the ambient concentration of SIR proteins which is also available to HML/HMR loci. In other words, HML/HMR loci read out the value of $\rho(\alpha, \gamma)$ as it changes due to variation of γ and the resulting effect on the state of the telomeric silenced domain. The possible states for HML/HMR loci depends on the value of $\rho(\alpha, \gamma), \alpha$ and γ. In the bistable regime, the two possible states are $S_l(\rho(\alpha, \gamma), \alpha, \gamma)$ and $S_h(\rho(\alpha, \gamma), \alpha, \gamma)$. In the monostable regime, there is only one possible solution, $S_m(\rho(\alpha, \gamma), \alpha, \gamma)$.

Let us start with the following initial condition. The system is initially on the zero-velocity line. The silenced domain at telomeric regions coexist with the active domain (free boundary separating them). Both HML and HMR loci are in the silenced state, i.e. $S_h(\rho(\alpha, \gamma), \alpha, \gamma)$. For the sake of example, imagine the system is represented by the bifurcation diagram in Figure 2.12C. In this case, the above scenario is consistent with an initial value of γ, for example, around 30. Point a in Figure 2.13 corresponds to the the value of $S_h(\rho(\alpha, \gamma), \alpha, \gamma)$ at this initial point.

As we decrease γ, to stay on the zero-velocity line, $\rho(\alpha, \gamma)$ increases. A priori, it is not obvious whether $S_h(\rho(\alpha, \gamma), \alpha, \gamma)$ is going to increase or decrease. However, it can be easily obtained numerically. Point b in Figure2.13 corresponds to a value of γ for which the system is still on the zero-velocity line. In certain range, $S_h(\rho(\alpha, \gamma), \alpha, \gamma)$ increases, whereas, in another range, it decreases. We are not sure if small changes in the value of $S_h(\rho(\alpha, \gamma), \alpha, \gamma)$ are experimentally observable. Note that, as long as the system is on the zero-velocity line, the change in S_h is smooth.

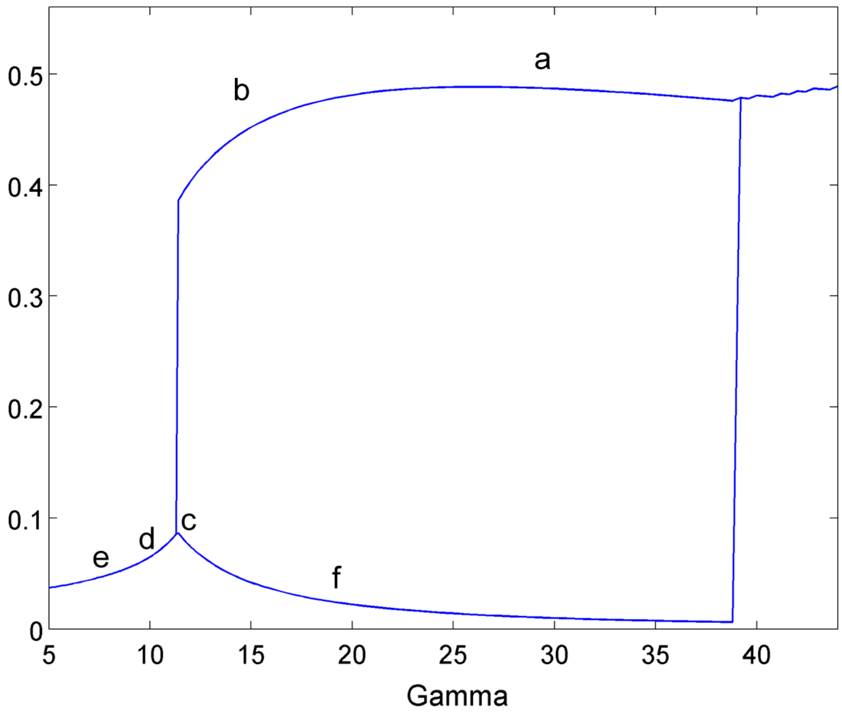

Figure 2.13: Hysteresis effect on the state of *HML* and *HMR* loci.

An interesting thing happens when the path of the system exit the zero-velocity line and enters Region I of the bistable regime ($\gamma \sim 11.4$). By this time, the silencing on the telomeric regions has shrunken to zero. Now, the $S_l(\rho(\alpha,\gamma),\alpha,\gamma)$ solution is more favorite than $S_h(\rho(\alpha,\gamma),\alpha,\gamma)$. Therefore, the state of *HML* and *HMR* loci would change to the lower, more active solution (point c in Figure 2.13). As we keep decreasing γ, the $S_l(\rho(\alpha,\gamma),\alpha,\gamma)$ solution decreases as well. Eventually, the system crosses the bifurcation line (point d) and goes into the active monostable regime (point e).

What happens if we start to increase γ? From point e to d to c in Figure 2.13, the state of *HML* and *HMR* loci goes back on the same path as before. However, at point c, where the system hits the zero-velocity line, the level of silencing at *HML* and *HMR* loci takes a new path. Previously, when we approached point c from right, *HML* and *HMR* loci were in the $S_h(\rho(\alpha,\gamma),\alpha,\gamma)$ solution, whereas this time, they are in the $S_l(\rho(\alpha,\gamma),\alpha,\gamma)$ solution. If we increase γ, the state of these loci will stay on the lower branch and move towards point *f*. The counter-intuitive behavior is that, at point f compared to point c, although γ is higher, the silencing has reduced:

$$S_l^f(\rho(\alpha,\gamma),\alpha,\gamma) < S_l^c(\rho(\alpha,\gamma),\alpha,\gamma)\ . \tag{2.34}$$

We are also performing experiments on yeast cells by monitoring the state of *HML*,*HMR* and telomeric regions, while changing γ. We have seen signs consistent with the counter-intuitive behavior of silencing reduction from point c to f in Figure2.13. One has to make sure the point where the silencing start to reduce indeed happens at point c, namely, the point where the system hits the zero-velocity line. At this point, the silencing domain at telomeric region should start to expand as well. In conclusion, we have to verify that the reduction in *HML* and *HMR* silencing happens when the silencing domain starts to form and expands at telomeric regions. This requires monitoring *HML*, *HMR* and telomeric activity at the same time. We hope to accomplish this in future.

2.4 Discussion

In this chapter, we concerned ourselves with epigenetic aspect of cellular differentiation. We studied a model of chromatin silencing in budding yeast. We analyzed the bifurcation diagram of the system and found the conditions under which it becomes bistable. Our model gives rise to different dynamical behaviors possible within the same molecular model and guides the formulation of more refined hypotheses that could be addressed experimentally. The model also helped us to understand the phenotype of some mutants. One issue which still remains to be addressed is how different genomic domains (silenced vs active) get inherited during cell division.

We are also performing experiments to verify qualitative features of our model. Our goal is to sweep different points in the parameter space by changing concentration of certain chemicals affecting γ. We analyze single cell gene expression data as the system goes through different parts of the parameter space. We are looking to compare the experimental results with qualitative predictions of our model. In particular, we consider the case where there is a finite supply of SIR proteins. The resulting depletion effect gives rise to interesting counter-intuitive behaviour.

In the biology context, the discussion on the process of silencing is mostly focused on the case where the silencing propagation is initiated through a nucleation center. However, an important aspect which has not received much attention yet is the degree of the robustness of the system to spontaneous nucleation. One of our future goal is to analyze the stability of solutions of our model to the noise and the switching between different states. It is very well a possibility that there is more to the control mechanism of epigenetic states and our theoretical considerations might shed some light on this aspect, for example, by suggesting specific signatures to look for in experimental studies.

Chapter 3

De Novo Genome Assembly Using Paired Reads

In this chapter, I will present SOPRA (Statistical Optimization of Paired Read Assembly), a new tool for *de novo* assembly of paired reads produced by next-generation sequencing platforms.[1]

3.1 Background

The instruction set of living organisms is stored in their DNA using a language composed of four letters A, T, C and G. Expectedly, knowledge of the DNA sequence is of central importance. To name a few, the sequence is used to look for genes, regulatory elements and pathways, evolutionary comparison of different species, relation between mutations/rearrangements and diseases, etc. In this section, an overview of the sequencing history and the bioinformatic challenge faced in extracting information from the sequencing data is presented.

Before we continue, I should mention that DNA is composed of two parallel strands attached together by hydrogen bonds. The sequence of each strand uniquely determines the sequence of the other one, which is why they are named *complementary* strands. The reason is that hydrogen bonds only happen between A and T, or between C and G. Therefore, A, T, C or G on one strand will be complemented respectively by T, A, G or C on the other strand. Figure 3.1A shows what a piece of DNA would look like.

[1] SOPRA is available freely, under the GNU Public License, at
http://www.physics.rutgers.edu/∼anirvans/SOPRA/

3.1.1 Brief history

The field of modern DNA sequencing started in 1977 and has been followed by a rich and exciting history [42]. In 1977, Frederick Sanger used his method to sequence a virus (bacteriophage $\Phi X174$ with genome length of 5386 bp), the first organism to be fully sequenced [43]. A series of improvements in the technology, over nearly 20 years, allowed complete sequencing of a cellular genome from two bacteria in 1995 (*H. influenzae* and *M. genitalium* with genomic length in the order of 1 Mb). The first sequenced eukaryotic genome was from yeast *S. cerevisiae* (12.0 Mb), in late 1996. Today, several hundred bacterial genomes up to around 10 million base pairs and several eukaryotic genomes with up to a few billion base pairs have been sequenced and submitted to online databases [2].

The length of DNA varies from a few thousands to a few billion base pairs. There is no current technology to simply read the whole genome sequence from one end to the other. Currently, the longest read length that can be read is around 1000 bases. For this reason, the traditional method of whole genome sequencing involved a process named chromosome walking. In this process, the genome is divided into several large fragments, each of which had to carefully be constructed and mapped to the original genome. This was an inconvenient and extremely time consuming step. In 1995, Craig Venter and collaborators applied a new method referred to as *whole genome shotgun sequence* (WGS) to the organism *H. influenzae* [44]. Today, WGS is the prevalent method of sequencing. WGS simply refers to the idea of randomly breaking up the DNA into little pieces and sequencing the pieces separately. The process of breaking up the DNA is applied to many copies of the same DNA. This produces fragments which come from overlapping regions and share similar sequences. Using this overlap, one can try to stitch the small pieces back together to reconstruct the original sequence. The

[2] For example, http://www.ncbi.nlm.nih.gov/ or http://www.ensembl.org/index.html

general strategy of sequencing a DNA fragment at random places had been used before. However, in the *H. influenzae* paper, it was applied for the first time on a whole genome of relatively long length (1.8 Mb).

During the 90's, the growing need for DNA sequencing resulted in the establishment of many specialized sequencing centers based on the Sanger biochemistry, where many of the steps were performed in an automated and parallelized fashion. However, the demand of biological research required far higher throughput and lower cost.

A new generation of sequencing methodologies started to appear in 2005 [45, 46]. By producing gigabases of data per run at a moderate cost (< $50,000), the new high-throughput sequencing (HTS) platforms have dramatically increased the sequencing capacity. In contrast to factory-like Sanger sequencing centers, the new sequencers can be hosted in a room and be operated by a single person. For the above reasons, application of new sequencing technologies is engaging large communities of scientists in different areas and holds the promise of revolutionizing the field of biological research [46]. To name a few, the list of applications includes gene expression analysis, mutation mapping, non-coding RNA discovery, metagenomics, and protein binding site identification [47, 48].

We do not get into the detail of the methodology employed in various sequencing technologies (see [46] for a review). However, on the practical level, one has to deal with a set of drawbacks that came along with the advantages of HTS sequencers. The drawbacks of the new technologies are the read length and the raw accuracy. Current implementation of Sanger sequencing can achieve read length of up to 1,000 bp. However, the read length for HTS platforms is between 30 to 100 bp. The error rate of HTS technologies is also more than one order of magnitude higher than the traditional methods. These challenges have provided major motivation for our work which I present in this chapter.

Possibly, both of the above limitations drawbacks will be ameliorated by the future advances in the technology. In addition, considering the huge demand for the sequencing, there is still a need for the cost to be reduced. The catch phrase of '$1000 genome' refers to the goal of reducing the cost such that the genome of an individual person can be resequenced for $1000. This is specially going to be important in an era of personalized medicine. One can truly recognize that the discoveries done starting from early signs of existence of a hereditary element [49] to the current goal of '$1000 genome' is an amazing landmark in the history of human kind.

3.1.2 *De novo* assembly

As we mentioned, shotgun sequencing is the method of choice today. From bioinformatics point of view, there are essentially two types of problems faced in extracting data from shotgun sequencing data: alignment or mapping and *de novo* assembly. The first case is related to the situation where the reference genome (or a close-by genome) is known. Examples of this case include mutation discovery, protein binding site discovery, etc. In case the reference genome is not available, one can use the overlap between small reads to stitch them back together and build longer sequences. These longer sequences are named *contig* and the process of reconstructing the genome is named *de novo* assembly.

There are a few reasons that in *de novo* assembly, we typically do not get one contig covering the whole original sequence. The first reason is that short reads are sampled randomly from the genome. Therefore, there is always a chance that there is not enough reads from certain region of the genome and there will be a break in the contig assembly at that point. This issue is quantified in the Lander-Waterman formula given below.

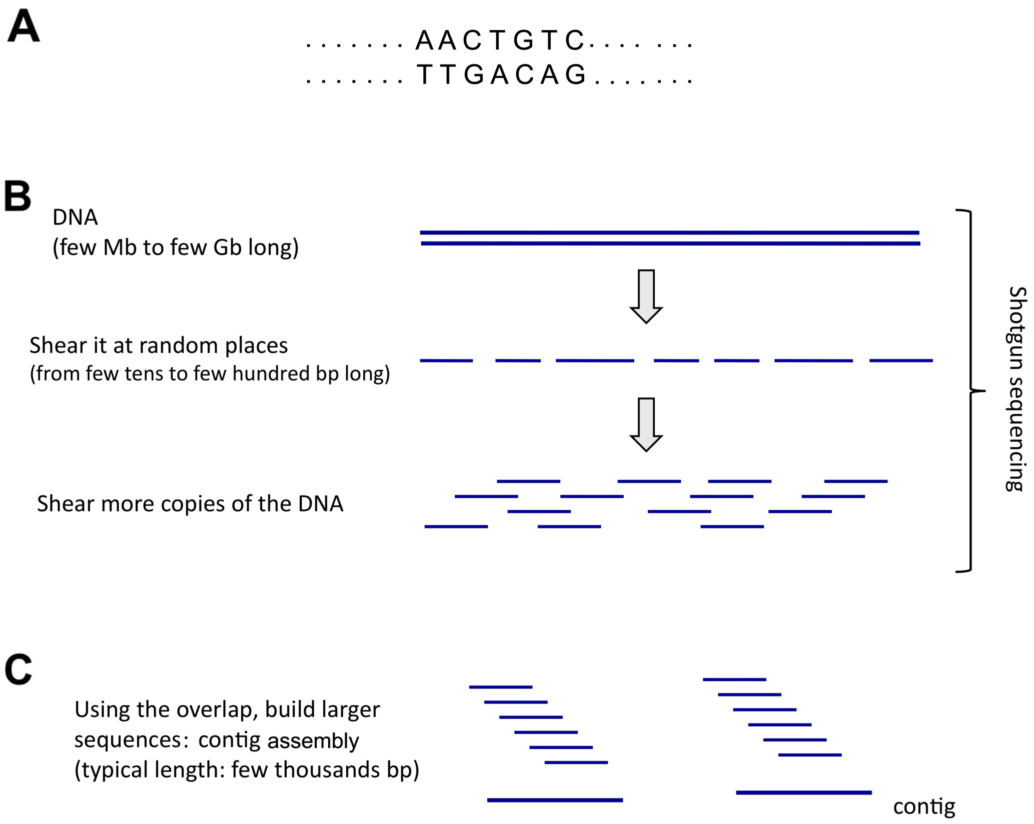
A
. . . . A A C T G T C . . . .
. . . . T T G A C A G . . . .
B
DNA
(few Mb to few Gb long)
Shear it at random places
(from few tens to few hundred bp long)
Shear more copies of the DNA
Shotgun sequencing
C
Using the overlap, build larger
sequences: contig assembly
(typical length: few thousands bp)
contig

Figure 3.1: Whole genome shotgun sequencing. A) DNA is composed of two complementary strands. Base A only pairs with base T, whereas, base C can only pair with base D. B) The maximum length of a fragment that current technology can read is less than 1,000 bp, much shorter than DNA length. One strategy, named shotgun sequencing, is to randomly break up many copies of the DNA into short pieces and sequencing the pieces separately. The short reads which come from overlapping regions will share similar sequences. Note that two overlapping pieces have to come from different copies of the DNA. Also, for each short read, it is not known which strand of the DNA it is coming from. Whole genome shotgun sequencing is the prevalent method of sequencing today (the other alternative method is called primer walking or chromosome walking). C) The process of joining back the short reads together into larger pieces goes by the name of contig assembly.

Lander-Waterman formula

Let us introduce a few notations. Let N be the total number of short reads, each with the length of L. Let G denotes the genome length. The coverage is defined as $c = \frac{NL}{G}$. This is the average number of reads covering any given point on the genome. The average number of reads starting from a particular position is $\alpha = \frac{N}{G} = \frac{c}{L}$. For two reads to be declared overlapping and be joined together, their overlap needs to be greater than a minimum length. Let us represent this minimum length by $(1-\sigma)L$. Then, the average length of contigs, $< x >$, is given by:

$$< x >= L \left[\frac{e^{c\sigma} - 1}{c} - (1 - \sigma) \right]$$

This formula is easily obtained by considering the probability for a read to be followed by another read after certain number of bases: assume you have a contig starting from one read. The contig is composed of one or more reads. The minimum length of a contig is L (this happens when the contig contains only one read). $P(x)$ is the probability that the length of a contig is equal to x.

$$P(x) = \delta_{L,x} e^{-\alpha\sigma L} + \sum_{y=1}^{\sigma L} e^{-\alpha(y-1)}(1 - e^{-\alpha})P(x - y) \ .$$

Note that $e^{-\alpha(y-1)}$ is the probability that no read starts in an interval of length $(y - 1)$. Similarly $(1 - e^{-\alpha})$ is the probability that at least one read starts from a particular position. By applying the Laplace transform, we find:

$$\hat{P}(s) = \sum_{x=1}^{\infty} s^x P(x) = \frac{s^L e^{-\alpha\sigma L}}{1 - s(1 - e^{-\alpha})\frac{1-e^{-\alpha\sigma L}s^{\sigma L}}{1-se^{-\alpha}}} \ .$$

Taking the derivative of $\hat{P}(s)$ at $s = 1$ gives the average of x:

$$< x >= L + \frac{e^{\sigma c} - 1}{1 - e^{-\frac{c}{L}}} - \sigma L \ .$$

This is the Lander-Waterman formula in the limit of small $\alpha = \frac{c}{L}$.

Apart from the random sampling, there are two more factors that affects contigs length. The first one is that the real data is error prone. As the technology advances, this factor can be greatly reduced. The second factor is the presence of repetitive sequences in the genome. To see this, assume there is a repeat sequence, R, appearing twice in the genome. Once it is flanked by two other sequences, A and B, i.e., it appears as ARB. In another place, R is flanked by C and D: CRD. After breaking up the genome and sequencing, we end up with 5 pieces: A, B, C, D and R. the right side of both A and C overlap with the left side of R. Similarly the left side of B and D overlap with the right side of R. In the process of contig assembly, we will be left with the confusion of building ARB or ARD or CRB or CRD. At this point, contig assemblers typically stop the extension of these sequences.

In the Lander-Waterman formula, the shortness of read length can be compensate by providing high coverage. However, in practice, typical contig length obtained from short read data is much smaller than the estimation. For example, using some reasonable values for the parameters in the Lander-Waterman formula, we get:

$$c = 50 \ , L = 50 \ , \sigma = .3 \ \Rightarrow < x > \sim 3 * 10^6 \ .$$

Instead, in practice, $< x >$ is typically in the order of a few hundred to few thousand base pairs. The main reason is that the shorter the read length, the higher is the chance that a read maps to several places on the genome. In fact, dealing with HTS data required a new set of strategies and algorithms, both for contig assembly and for other applications [50]. The above limitations encountered in contig assembly could be partially overcome by utilizing mate pair technology, explained below.

Mate pair technology

Consider randomly sheared DNA fragments with varying lengths and unknown sequences. Using gel electrophoresis technique, one can select the fragments which have their length approximately equal to some targeted value. Sequencing both ends of such fragments provides us with pairs of read separated by a known distance along the genome. In addition, depending on the sequencing method, one knows if the two reads are coming from the same strand or from the opposite strand of the DNA. This technique is called mate pair or paired-end technology. If two legs of a mate pair are incorporated into two separate contigs, we can infer the relative orientation (i.e. strand) and relative position of those two contigs on the genome. Such ordering of contigs using mate pair information is called *scaffold assembly.*

Mate pair/paired-end sequencing was a key innovation that allowed shotgun sequencing of large complex genomes such as humans and *Drosophila* [51]. In the following, unless we are explicitly contrasting the two methods, we will use the term mate pair to refer to both of these technologies.

Over the past few years, several algorithms have been developed for assembly of short reads. These algorithms can be divided into two broad categories. Some methods, based on 3' kmer extension, use particular data structures to efficiently search for short reads overlapping and extending a seed sequence [52, 53, 54]. In contrast, the graph-based methods pose the sequence assembly as a problem of finding paths on a graph that encodes the short read overlap information (de Bruijn graph) [55, 56, 57, 58].

The current version of some of the above-mentioned short read assemblers can handle mate pair information. However, the use of this information was not central to the concepts that led to the design of most of these algorithms. The sole exception is the ALLPATHS assembler [57], where the use of mate pairs is essential. From a practical point of view, one drawback of ALLPATHS is that it requires at least two paired libraries, with very different insert sizes. Also, the performance of this assembler degrades rapidly as the coefficient of variation of insert size in a library increases past a few percent [57]. This sensitivity is a problem for assembly of real sequence data, as we will see. In the context of previous generations of sequencing technologies with longer reads, the incorporation of mate pair information has also been addressed, either in conjunction with contig assembly [59, 60] or as a scaffolding module [61].

I worked on developing SOPRA (Statistical Optimization of Paired Read Assembly), a tool designed to exploit the mate pair/paired-end information for assembly of short reads. The main focus of the algorithm is selecting a sufficiently large subset of simultaneously satisfiable mate pair constraints to achieve a balance between the size and the quality of the output scaffolds. Scaffold assembly is presented as an optimization problem for variables associated with vertices and with edges of the *contig connectivity graph*. Vertices of this graph are individual contigs with edges drawn between contigs connected by mate pairs. Similar graph problems have been invoked in the context of shotgun sequencing and scaffold building for previous generation of sequencing projects. However, given the error-prone nature of HTS data and the fundamental limitations from the shortness of the reads, the ad hoc greedy algorithms used in the earlier studies are likely to lead to poor quality results in the current context. SOPRA circumvents this problem by treating all the constraints on equal footing for solving the optimization problem, the solution itself indicating the problematic constraints (chimeric[3]/repetitive contigs, etc.) to be removed. The process of solving and removing of constraints is iterated till one reaches a core set of consistent constraints.

Generally speaking, current scaffolding algorithms fall into two categories.

[3] Chimeric refers to contigs which are formed by mistakenly joining two or more distinct part of the genome together.

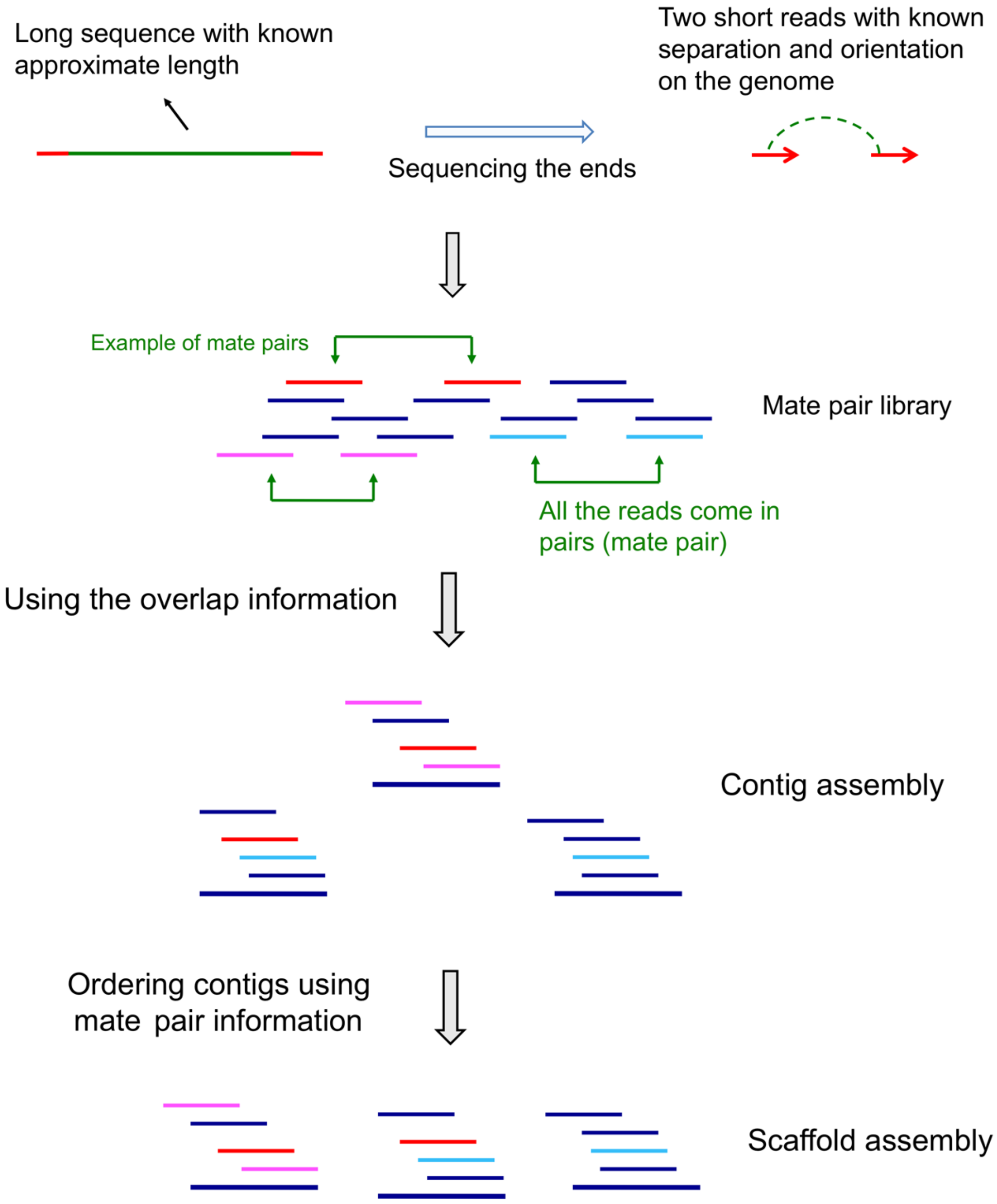
Long sequence with known approximate length
Sequencing the ends
Two short reads with known separation and orientation on the genome
Example of mate pairs
Mate pair library
All the reads come in pairs (mate pair)
Using the overlap information
Contig assembly
Ordering contigs using mate pair information
Scaffold assembly

Figure 3.2: Mate pair technology. By sequencing the ends of size-selected fragments, one obtains pairs of short reads separated by a known distance along the genome. From the sequencing method, the relative orientation (strand)

Prominent de Bruijn graph based contig building algorithms (e.g. Velvet [58] and Euler [59]) utilize mate pairs to improve the path/walk in the same de Bruijn graph. The other category of scaffolding algorithms [60, 61], formulate the problem in terms of graph theoretic constructs in which vertices of the graph are associated to contigs and edges encode mate pair information. Although our approach to the scaffolding problem has partial similarity to this last category, our solution strategy is different, as we will explain. Our algorithm could be implemented, in principle, for any kind of mate pairs, from Sanger reads to the HTS data. However, the special challenges inherent in scaffolding with short read data necessitate an approach that is more sophisticated than those developed so far. That is why we implemented and tested SOPRA in the context of short reads from next-generation technologies.

Existence of repetitive regions in DNA, errors in the sequencing process and mis-assembly of short reads into contigs are all factors which contribute to the complexity of scaffold building using mate pair information. This complexity arises in the form of apparent inconsistency among the set of constraints laid by the mate pairs. Detecting and eliminating the sources of these inconsistencies is essential for the success of any algorithm dealing with mate pair data. This issue is especially important in the context of short read data, since, we expect a higher number of problematic mate pair constraints in the process of scaffold building.

Existing scaffolding algorithms follow a greedy approach, starting with certain schemes of ordering the contigs and pairing information. The mate pairs are then iteratively incorporated as long as the new information does not conflict with the previously assembled scaffolds. In other words, at each step, only a subset of contigs and links in between are considered to improve the assembly. Given the nature of short read data, solution strategies employed in previous studies face difficulties for such kind of data [50]. In the next chapter, I will explain our approach in detail and compare it with existing algorithms.

Color-space data

SOLiD (Sequencing by Oligonucleotide Ligation and Detection) is a novel HTS platform. It uses four fluorescent color probes (coded as 0-3) for reading dinucleotides, namely, two neighboring bases at a time. The sixteen possible dinucleotide combinations are divided into groups of four, each of which is assigned a unique color (e.g. color 2 is assigned to combination AG, GA, TC and CT). However, the groups are designed in such a way that, every combination of the first base and the color call uniquely determines the second base. In other words, each color encodes a transition matrix in the base-space.

Each SOLiD read starts with a reference base, the last base in the primer (usually T or G), followed by a certain number of color calls e.g. G10223330. Using the reference base and the first color call, we can find the first letter base, which in turn can be combined with the second color call to obtain the second letter base. Continuing so forth, we can translate the whole sequence from color-space to the conventional base-space.

The issue is if one of the color calls is wrong (because of an error in the sequencing process), the whole translation from that point on will be wrong. In other words, one error in the color-space will propagate into many errors in the base-space. It is because of this error rate magnification that we do not simply translate the SOLiD output directly to the letter-space. Instead, SOPRA translates the resulting color-space assembly using a dynamic programming method that avoids such error propagation, as we will explain below.

Among the available de novo assemblers, as far as we are aware, Velvet [58] is the only one that can handle color-space data. Adapting available assemblers for color-space data is not a trivial task, since, naive translation from color-space to base-space leads to serious error amplification [62]. Particular attention was paid so that SOPRA could handle data from the SOLiD platform. The final output, given in base-space, is constructed from the color-space assembly, as well as from additional information obtained by translating only the first color call of all the reads. This method will prevent the propagation of the error that can happen in the naive translation.

3.2 Methods

The design of SOPRA is especially targeted to exploit the mate pair information in the process of scaffold assembly. In other words, SOPRA is a module that can be combined with any of the available algorithms for contig assembly. Such a modular design allows greater flexibility and control over the scaffold building process, as has been noted before [61]. SOPRA proceeds in an iterative fashion where at each step problematic mate pair constraints are detected and removed. At each step, one finds a solution consistent with most of the constraints by statistically optimizing over a cost function. Then, one relaxes the most violated constraints. This alternation between removing suspicious data and optimization continues, till we get scaffolds consistent with the remaining trusted constraints.

For color-space data, there is one additional step of translating the assembled contigs to base-space. For SOLiD sequencer data, SOPRA uses a dynamic programming approach to robustly translate the color-space assembly to base-space. For assessing the quality of an assembly, we report the no-match/mismatch error rate as well as the rates of various rearrangement errors. Conclusions: Applying SOPRA to real data from bacterial genomes, we were able to assemble contigs into scaffolds of significant length (N50 up to 200Kb) with very few errors introduced in the process.

The flow chart of the assembly process is shown in Figure 3.3. Below, we will explain each section in more details.

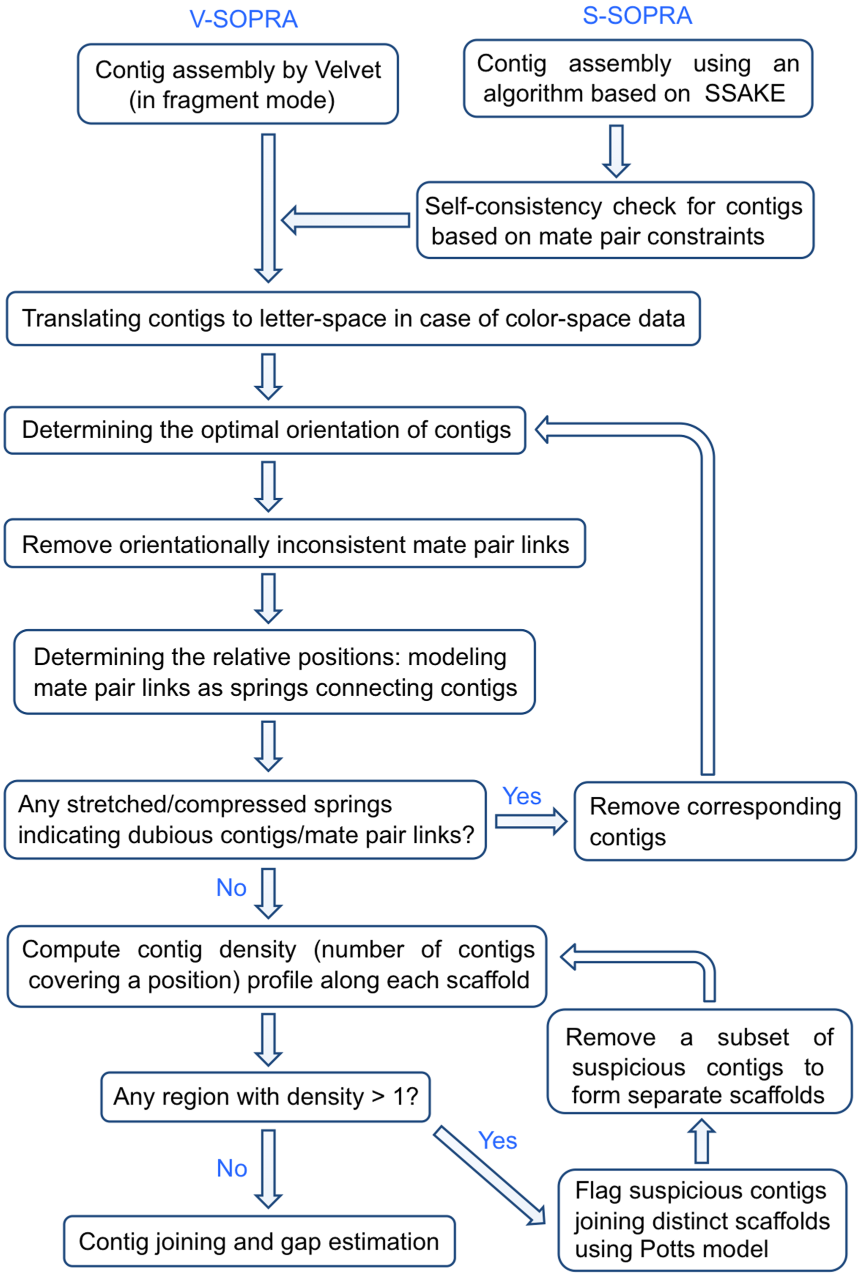
V-SOPRA
S-SOPRA
Contig assembly by Velvet (in fragment mode)
Contig assembly using an algorithm based on SSAKE
Self-consistency check for contigs based on mate pair constraints
Translating contigs to letter-space in case of color-space data
Determining the optimal orientation of contigs
Remove orientationally inconsistent mate pair links
Determining the relative positions: modeling mate pair links as springs connecting contigs
Any stretched/compressed springs indicating dubious contigs/mate pair links?
Yes
Remove corresponding contigs
No
Compute contig density (number of contigs covering a position) profile along each scaffold
Remove a subset of suspicious contigs to form separate scaffolds
Any region with density > 1?
Yes
No
Flag suspicious contigs joining distinct scaffolds using Potts model
Contig joining and gap estimation

Figure 3.3: Flow chart of the algorithm. In principle, the contig assembly can be performed using any of the available contig assembly algorithms. SOPRA uses the mate pair information to assemble contigs into scaffolds. S-SOPRA and V-SOPRA correspond to the integration of SOPRA with SSAKE and Velvet respectively.

3.2.1 Contig assembly preliminaries

As we mentioned, SOPRA is focused on scaffold assembly. The information SOPRA needs from a contig assembler is the computed positions of reads in each contig. SOPRA reconstructs the contigs based on this information. Note that, in the case where these reads do not show perfect overlap, reconstruction of the contigs by SOPRA may not agree with the output of the original contig assembler.

In this work, I present the performance of SOPRA integrated with two particular contig assembly algorithms, namely, SSAKE [52] and VELVET [58]. We will refer to these two versions as S-SOPRA and V-SOPRA, respectively. This integration is relatively straightforward and described in Appendix F. However, for color-space data, there is one additional step of translating the assembled contigs to base-space.

Robust translation of contigs assembled in color-space

As we pointed out, the issue with the naive translation of color-space to base-space is that if one of the color calls is wrong (because of an error in the sequencing process), the whole translation from that point on will be wrong. In other words, one error in the color-space will propagate into many errors in base-space. It is because of this error rate magnification that we do not simply translate the SOLiD output directly to the base-space. Instead, SOPRA translates the resulting color-space assembly using a dynamic programming method that avoids such error propagation, as we will explain below.

We only translate the first color call (using the reference base) to the base-space but keep the rest of the sequence in color-space. This means a library of sequences, each of which consists of a reference base and L color calls, will become a library of sequences that start with a DNA base followed by $L-1$ color calls. If we ignore for a moment the first DNA base, we can use the $L-1$ base long sequences for contig assembly in the same way as in regular base-space data. Of course, the result of this assembly will be contigs in the color-space. Although we do not use the first letter base of the sequences in the assembly process, once a sequence is used in building a contig, we record where on the contig the first letter base of the corresponding sequence lies (Figure 3.4). Notice that the first letter base lies between two color calls and serves as a suggestion for what the DNA base at that position should be. On the other hand, each color call is located between two neighboring DNA bases and provides information about the corresponding dinucleotide.

At this point, the assembly result is a sequence in color-space, C, plus some letter base suggestions at certain locations of each contig, F. In Figure 3.5, the color-space contig is represented using blue numbers 0-3, whereas, base-space suggestions are shown in magenta. Now, we pose the following question: Given a color-space sequence plus its letter base suggestions, what is the most likely DNA sequence which gave rise to this data? We will set up a model that allows for mistakes in the base suggestions as well as in the assembled color-space contigs. To each arbitrary base-space sequence, the model assigns a probability for that sequence to be the real DNA sequence. The final translation output would be the base-space sequence that maximizes this probability.

The reason why this method prevents propagation of error can be counter-intuitively understood as follows. If the presence of a color call error is ignored, the nave translation will disagree with most of the base-space suggestions. If this disagreement goes on for a long stretch, from the perspective of the probability function, it is better to declare that particular position to be a color call error and replace it with another color such that the translation becomes consistent with the stretch of base-space suggestions. The ability to alter a color call to enhance the consistency with base suggestions in long stretches helps not only with substitution errors, but also helps to compensate for inconsistency arising

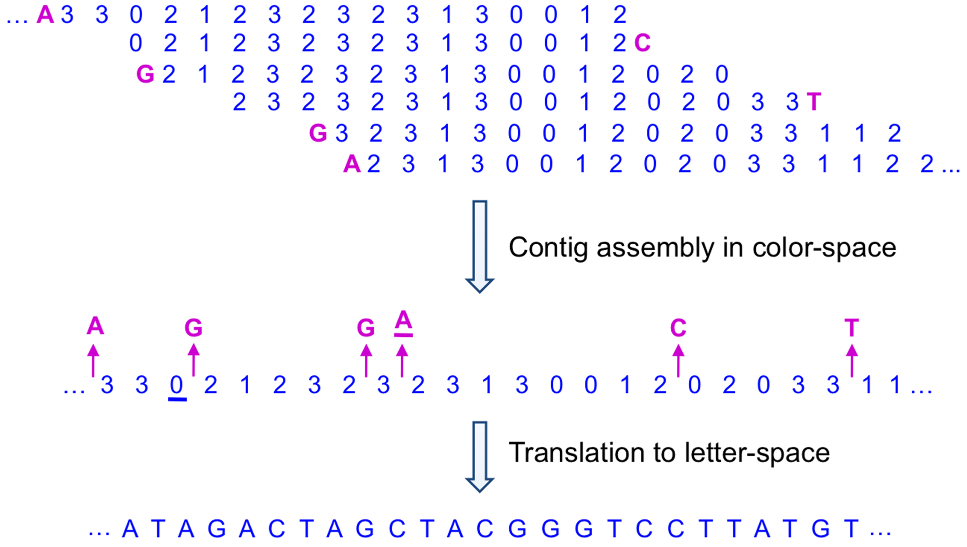
... A 3 3 0 2 1 2 3 2 3 2 3 1 3 0 0 1 2
0 2 1 2 3 2 3 2 3 1 3 0 0 1 2 C
G 2 1 2 3 2 3 2 3 1 3 0 0 1 2 0 2 0
2 3 2 3 2 3 1 3 0 0 1 2 0 2 0 3 3 T
G 3 2 3 1 3 0 0 1 2 0 2 0 3 3 1 1 2
A 2 3 1 3 0 0 1 2 0 2 0 3 3 1 1 2 2 ...
Contig assembly in color-space
A G G A C T
... 3 3 0 2 1 2 3 2 3 2 3 1 3 0 0 1 2 0 2 0 3 3 1 1 ...
Translation to letter-space
... A T A G A C T A G C T A C G G G T C C T T A T G T ...

Figure 3.4: Robust translation from color-space to base-space. The base-space suggestions, obtained by translating only the first color call of each read, are shown in magenta. Contig assembly is performed using only the color part (indicated by numbers 0-3) of each sequence. Inconsistencies between the color-space calls and the base-space suggestions, signals the presence of errors. We use an error probability model to find the most likely DNA sequence consistent with this data. The underlined color calls and suggestions in the figure are declared as mistakes in the final translation.

from indels. The details of the model are explained in Appendix B.

Contig self-consistency check

We implemented the self-consistency checks described below only in S-SOPRA. The reason for these checks is that the programs, like SSAKE, which use a greedy algorithm for contig assembly, are particularly vulnerable to generating chimeric contigs. If two legs of a mate pair are located on the same contig, then their relative orientation and position in the contig should match the ones suggested from the mate pair link. If we observe more than certain number of times (threshold is a parameter of the software) cases where the orientation disagrees or the separation between reads is more than one standard deviation different from the insert size, we discard that contig. This method, however, does not necessarily detect chimeric contigs where two or more regions from different parts of the genome have been mis-assembled into one contig. Mate pair information can be used to detect such mis-assemblies, as explained below.

If a contig is genuine, there should be several mate pairs connecting different locations on the same contig (assuming the contig is at least a few times longer than the insert size of mate pairs). However, if it is the case that the contig is composed of two or more sequences coming from different parts of the genome, there should not be as many mate pair links connecting those sequences together. For each point on a contig, we count how many mate pair links connect the right side of that point to the left side. If this number is particularly low for some region, we cut the contig into two at that position.

Estimation of insert size

In the case where there are enough long contigs, the typical value of the insert size can be estimated from the mate pairs located on the same contig. To do so, we first remove the outliers for which the separation between the pair is different from the suggested insert size by more than the value of the suggested insert size (or equivalently, more than five times the standard deviation, if we assume it is 20% of the suggested insert size). The empirical insert size is equal to the mean value of the separation for the remained pairs. The user needs to know only an approximate value for the insert size based on the library preparation protocol. Prior knowledge of the typical insert size needs to be accurate only when almost all contigs are smaller than the typical inserts.

In case the insert size targeted by the library preparation methods is not available to the user, he/she could take advantage of the empirical distribution of insert sizes output by SOPRA and determine the typical insert size by inspection. In any case, it is a good idea to inspect this distribution, to ascertain the quality of the mate pair library.

Removal of reads in high coverage regions from scaffolding process A contig containing repetitive regions can provide conflicting mate pair constraints and cause mis-assembly in the scaffolding process. Although, one could take up the problem of resolving the repeat structures, our approach currently is to identify and remove such contigs from the scaffolding process. One way of detecting repeats is by looking for high coverage regions in each contigs. If a contig has high mean coverage (determined by a parameter of the software) we remove such a contig from scaffold assembly before starting the process. Some contigs have high coverage locally without having high mean coverage. We exclude mate pairs with reads falling in such local high coverage regions for the scaffolding considerations as well (the threshold is a parameter of the software).

3.2.2 Scaffold assembly

If two legs of a mate pair are incorporated into two separate contigs, we can infer the relative orientation and relative position of those two contigs on the genome. However, such ordering of contigs is not an easy task, since, the constraints imposed by mate pairs are often not self-consistent. The best one can do is to assign the orientations and positions so that as many constraints as possible are satisfied. In addition, there can be misleading or incorrect information. These dubious constraints arise not only from issues like erroneous contig assembly, but also from innate problems in mate pair data itself.

To elucidate this point, let us examine the two real libraries discussed below in the performance comparison section. In Figure 3.5, we plot the histogram of separation between the two reads belonging to a mate pair, obtained by matching the reads to the reference genome. As we can see, the distribution of separation could be thought of as a combination of a sharp peak and a broad background that spans over the entire length of the genome. Even if we limit ourselves to the sharp peak (Figures 3.5B and 3.5D), the standard deviation is around 20% of the mean value. The variability in separation is much larger than values used for generating simulated data in some studies [57, 58]. The algorithm for position assignment has to be robust to such large degree of uncertainties. As will be discussed in the coming sections, in our approach, this goal is achieved by identifying and removing those mate pairs that belong to the broad background as well as from averaging effect of imposing all the remaining constraints together.

For contig building, it is often convenient to represent the sequence overlap information using graph theoretic constructs, e.g. in terms of an overlap graph or a de Bruijn graph. Similarly, it is useful to encode the constraints given by mate pair information into a graphical model. In this model, the underlying undirected graph has vertices corresponding to each contig. Any two contigs connected through mate pairs have an edge in between. We call this graph the contig connectivity graph. This graph is similar to the contig-mate-pair graph introduced in [60], except that here each contig is represented by a single vertex as opposed to two. This kind of graph structure has been used in other studies as well [61]. The structure of the contig connectivity graph, at different stages of the assembly, can be visualized with the help of programs such as GraphViz package [30].

In our formulation, orientations and positions for each contig are variables living on the vertices of this graph. If we introduce the mate pair information as probabilistic constraints on relative orientations and positions of neighboring vertices on the graph, this graphical model has the structure of a Markov random field model [63]. Markov random field models were originally inspired by problems in statistical physics. There are relatively obvious connections between finding the ground state (the most probable configuration of Markov random field) of certain statistical physics models and well-known graph optimization problems as was pointed out by several researchers in the eighties [64]. Such analogies

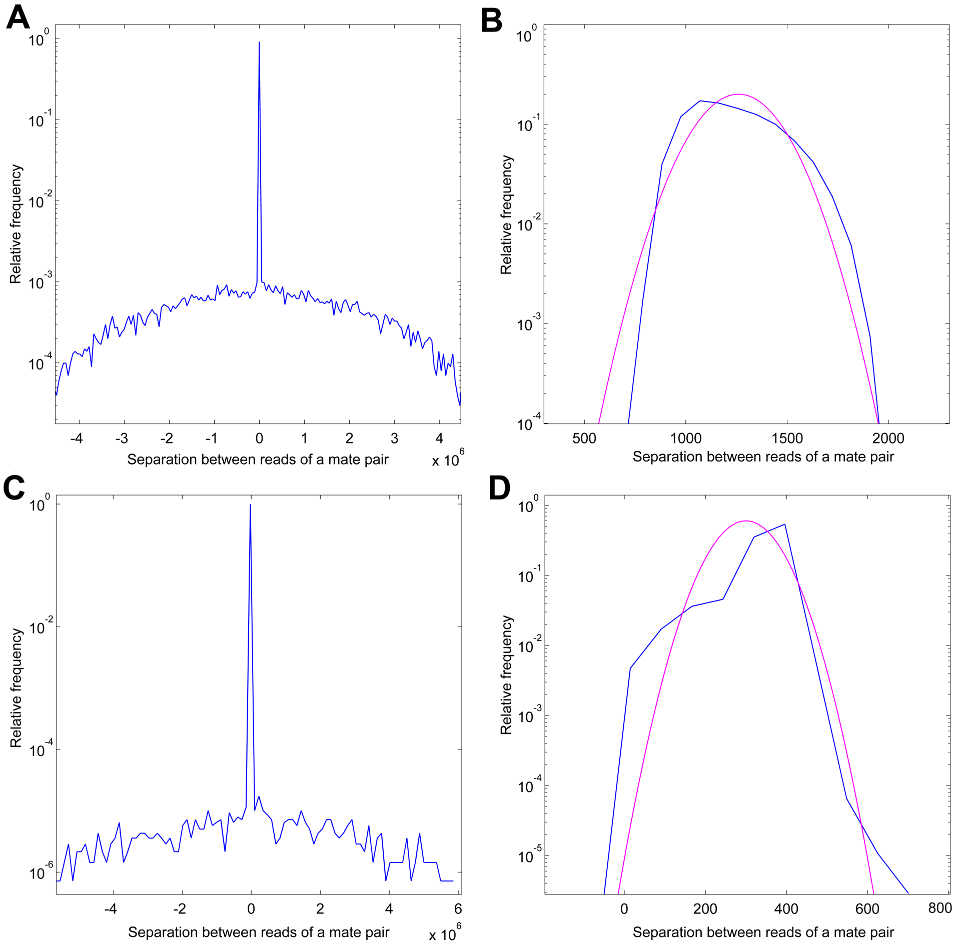

Figure 3.5: Histogram of separation between locations of two reads of a mate pair on the reference genome. This histogram appears to be a combination of two parts. One part is a distribution peaked around the insert size of the mate pair library, as expected. However, in addition, there is a broad background. (A) *E. coli* data from SOLiD platform. (B) *E. coli* dataset, but limited to pairs for which the separation is around the peak region in (A). (C) *P. syringae* data from Illumina platform. (D) *P. syringae* dataset, but limited to pairs for which the separation is around the peak region in (C). Both distributions in (B) and (D) have large standard deviations, each around 20% of the corresponding mean values.

also led to the simulated annealing [65] as a heuristic method for solving hard combinatorial optimization problems (see [66] for a review). We will explain our procedure by invoking the physical analogies, but one could often describe the same procedure using a language familiar to computer scientists.

We perform the scaffolding in two steps. We first assign the orientation of contigs, without considering their positions. Once the orientation is determined, in the second step, we calculate the position of contigs. In this second step, we only use those mate pair links which are consistent with the orientation assigned in the first step. In principle, one could have optimized for orientation and position together, however, our two steps process simplifies the algorithm.

One additional constraint is that distinct contigs cannot be assigned to the same or overlapping positions. This should be true for every possible pair of vertices. This means that if we want to impose this condition in the contig connectivity graph, every possible pair of vertices will be connected by an edge representing this non-overlapping condition. In other words, every vertex will be directly connected to all other vertices. In this sense, the Markov random field structure on the contig connectivity graph is violated. We first solve for orientations and positions ignoring the non-overlapping constraints. The resulting solution typically includes some scaffolds for which the non-overlap condition is not satisfied. We segment these scaffolds into smaller scaffolds satisfying the non-overlap condition using another Markov random field model living on a new graph obtained by augmenting the contig connectivity graph with additional edges between apparently overlapping contigs.

Determining the relative orientation

We indicate the two possibilities for the orientation of contig by $S_i = 1$ and $S_i = 2$. If two contigs i and j are connected through mate pair links, we associate a number to it, denoted by $J_{i,j}$. The sign of $J_{i,j}$ is positive if the links suggest that two contigs have the same orientation, otherwise it is negative. The absolute value of $J_{i,j}$ is equal to the number of links that connect the two contigs. If all the mate pairs connecting two contigs do not agree with each other, we require that at least a significant majority do. To be a significant majority, we require the percentage of the mate pairs in the dominant group to be higher than a certain threshold, which is a parameter in the software. Otherwise, all the links between those contigs are ignored.

The reason for rejecting all these links is as follow. For two close-by genuine contigs, not belonging to repeats, the source of orientational conflicts is the presence of spurious mate pairs. Usually, these inconsistent spurious links form a small minority. However, when a part of a contig belongs to repetitive regions or one of the contig is chimeric, the nature of the orientational conflicts is different. For example, it is likely that part of the mate pair information suggests the contig belongs to one strand while some other part of the information suggest it belongs to the other strand. In such cases, the majority group and the minority group can have comparable number of links. If a significant majority of links do agree, the minority links are ignored suspecting that they are spurious. If the numbers are comparable, then all links are ignored for the reason mentioned above.

For each configuration of orientations, $S = (S_1, S_2, ..., S_N)$, N being the number of contigs, we define the following cost function:

$$E[S] = \sum_{<ij>} J_{i,j} S_i S_j \ . \tag{3.1}$$

This quantity, a measure of how many of the mate pair links are satisfied, could be thought of as the energy of an Ising spin system with interactions $J_{i,j}$. If it were possible to find a configuration to satisfy all the constraints, we would have:

$sign(J_{i,j}) = sign(S_iS_j)$, $\forall\ i, j$. The energy of this configuration would be:

$$E_{min} = \sum_{<ij>} |J_{i,j}|\ .$$

As we mentioned before, it is often the case that such a configuration does not exist. Therefore, our goal is to find the best configuration in which as many mate pair links as possible are satisfied. Effectively, we want to find the orientation assignment that minimizes the energy function in Equation 3.1 (Figure 3.6A). This minimization is equivalent to the maximum weight cut problem, which appeared in the context of shotgun sequencing [67] and of scaffold assembly [61]. Given that this problem is NP-complete [68, 69], it is natural to search for heuristic methods. The approach of these earlier studies is to resolve the constraints in the scaffold assembly problem through particular greedy algorithms that depend upon ad hoc schemes of ordering the contigs. The contrast between such approaches and ours will become clear, as we will explain our algorithm in the Appendix C.

Determining the relative position

For determining the relative positions of contigs, we only use the mate pair links that are orientation-wise consistent with the optimal configuration found in the previous section. Consider a set of contigs connected through mate pair links. Let $X = (0, x_2, ..., x_N)$, denotes the positions of the start points of these contigs. By putting $x_1 = 0$, we have chosen a particular system of coordinates. Each mate, r, connecting contigs i and j, provides us with some information about $x_i - x_j$, encoded in the probability distribution $p^r(x_i - x_j)$. This distribution is picked around certain value, $l^r_{i,j}$, which can be determined from the location of the two reads in the corresponding contigs and the insert size of the mate pairs (the formula is presented in the Appendix D).

Had we not assigned the orientations, one could still define $l^r_{i,j}(S_i, S_j)$, with the

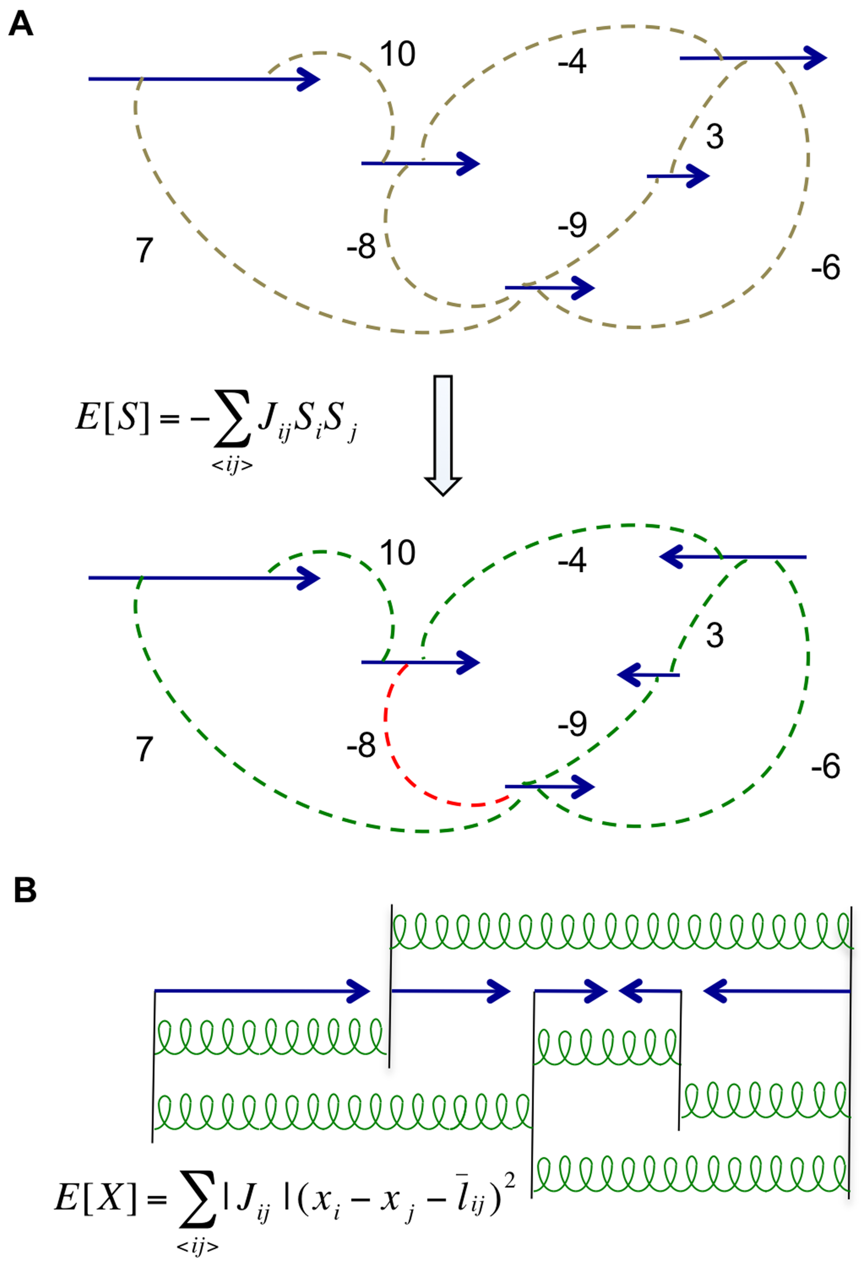
A
10
-4
3
7
-8
-9
-6
$E[S] = -\sum_{<ij>} J_{ij} S_i S_j$
10
-4
3
7
-8
-9
-6
B
$E[X] = \sum_{<ij>} |J_{ij}| (x_i - x_j - \bar{l}_{ij})^2$

Figure 3.6: Modeling constraints on the contig connectivity graph. (A) For two contigs i and j connected through mate pairs, the quantity $J_{i,j}$ encodes the information about relative orientation (sign of $J_{i,j}$) and number of mate pairs connecting those contigs (absolute value of $J_{i,j}$). Minimizing the energy produces an orientation assignment that satisfies as many constraints as possible. The constraints that are not satisfied in the optimal configuration (shown in red) are ignored in the next part. (B) To determine the relative position of contigs, we model the collection of mate pairs connecting contigs i and j as a spring attached to the start points of those contigs. The relaxed length of this spring, $\bar{l}_{ij}$, is equal to the average suggested distance between the start points of those contigs given by mate pair constraints.

orientations only affecting the sign of the quantity. Note that $|l^r_{i,j}|$ is the suggested distance between the corresponding contigs, whereas, the sign determines the ordering (i.e. which one is to the left and which one is to the right). In Figure 3.6A, next to each edge, we just show $J_{i,j}$'s. However, each edge also carries the additional information on the relative position of the corresponding contigs ($l^r_{i,j}$'s). Before assigning the orientation, the contig connectivity graph does not fully capture the ordering of contigs, since, as we explained, $l^r_{i,j}$ is determined up to a sign. After the orientation assignment, the full information about relative position of contigs is captured by this graph.

The overall information provided by all the mate pairs linking contigs i and j is given by

$$\prod_{r=1}^{|J_{i,j}|} p^r(x_i - x_j)$$

Note that $|J_{i,j}|$ is the number of mate pairs bridging between contigs i and j. We do not know the exact form of $p^r(x_i - x_j)$; however, if we take it to be a Gaussian centered around $l^r_{i,j}$, we will have:

$$p^r(x_i - x_j) \propto e^{-(x_i - x_j - l^r_{i,j})^2/2\sigma^2} \quad , \tag{3.2}$$

where σ corresponds to the variance in the insert size of mate pairs. Our approach is to determine the relative position of contigs by maximizing the joint probability distribution:

$$P(X) = \prod_{<ij>} \prod_{r=1}^{|J_{i,j}|} p^r(x_i - x_j) \propto \prod_{<ij>} \prod_{r=1}^{|J_{i,j}|} e^{-(x_i - x_j - l^r_{i,j})^2/2\sigma^2} \propto \prod_{<ij>} e^{-|J_{i,j}|(x_i - x_j - \bar{l}_{i,j})^2/2\sigma^2}, \tag{3.3}$$

where $\bar{l}_{i,j} = (\sum_{r=1}^{|J_{i,j}|} l^r_{i,j})/|J_{i,j}|$ is the average suggested distance between the start points of contigs i and j. Equivalently, one could minimize the function:

$$E(X) = \sum_{<ij>} \frac{|J_{i,j}|}{2} \left(x_i - x_j - \bar{l}_{i,j}\right)^2 \ . \tag{3.4}$$

. This function has an alternative interpretation as the energy of a coupled system. In this analogy, the collection of mate pairs between two contigs i and j is replaced by a spring connecting the start points of those contigs. The spring constant is equal to $|J_{i,j}|$, and the relaxed length of the spring is given by $\bar{l}_{i,j}$. In this way, the original system of contigs connected through a network of mate pairs is modeled as a system of objects connected through a network of springs (Figure 3.6B). The solution maximizing the probability given in Equation 3.3 corresponds to the equilibrium position (X^*) of the objects in the spring system. These positions could be calculated by solving a set of linear equations corresponding to the force on each object being zero.

In the equilibrium position, if the distance between two contigs is equal to the distance suggested by the mate pairs connecting them, then the corresponding spring is relaxed; otherwise, the spring is either stretched or compressed. In other words, we could define $\Delta_{ij} = |x_i^* - x_j^* - \bar{l}_{ij}|$ as a measure of the degree to which the mate pair constraints are violated. If all the suggested distances were self-consistent, all Δ_{ij}'s would be nearly zero (no stretch/compression in the springs). In real data, it is possible that some sequences match in several locations on the genome, and therefore, mate pair information would not uniquely determine the position of contigs. In our model, the sign of this non-uniqueness is that in the equilibrium solution, X^*, some of the springs will be stretched or compressed. The same situation can arise because of contig mis-assembly where two separate regions of the genome are joined into one contig.

When there is a stretched or compressed spring, we remove the contigs attached to the end of that spring from the system and restart the scaffold assembly on the remaining contigs. In other words, we go back to the orientation assignment step (Figure 3.3). The cycle stops when in the equilibrium position, all the springs are close to their relaxed state, namely, all Δ_{ij}'s are below a certain threshold. Note that X^* is the positions of the start points of contig. If the orientation of contig i is positive, it means that it covers the interval $(x_i^* \ , \ x_i^* + length_i - 1)$ on the scaffold. If i has negative orientation, we assign the reverse complement of i to the interval $(x_i^* - length_i + 1 \ , \ x_i^*)$.

The greedy algorithms, previously applied to the combinatorially difficult problem of assigning relative positions, consider contigs in a certain order; an order that depends on the number of links associated with each contig [61, 60]. Potentially, such methods could be prone to incorporating repeats/chimeric contigs which could have significant number of links associated with it. In contrast, our method has the advantage of providing an unambiguous means for flagging misleading distance constraints with having to commit to any such order.

Detecting tangled scaffolds by the contig density profile

We calculated the position of the contigs in a scaffold from a set of linear equations based on the assumption in Equation 3.2. Of course, position intervals corresponding to distinct contigs should be non-overlapping. However, the solution of these linear equations is not guaranteed to satisfy this non-overlap condition. In fact, such overlapping configurations do arise in practice. Below, we explain some of the causes leading to this problem.

Consider the scenario described in Figure 3.7A. There are two sets of contigs, shown in green and magenta, belonging to distinct regions of the genome. Contigs within each set are self-consistently connected through mate pairs. Assume during contig assembly, contig 3 from the first set and contig 7 from the second set get mis-assembled into one contig. In this case, we obtain a scaffold that contains all the contigs and yet, does not have any stretched or compressed spring.

In addition to contig mis-assembly, existence of repetitive regions in the genome is another factor that can cause improper joining of multiple scaffolds. In that case, contigs 3 and 7 in Figure 3.7A are seen as one contig in the assembly, whereas they are really copies of the same sequence that matches on multiple places on the genome. Each copy can cause the mis-incorporation of a new set of contigs from its neighbors.

In order to detect this type of complication, we define the 'density profile', a quantity that represents how many contigs cover each region of a scaffold. In the final assembly output, this density should be near one for all regions of each scaffold (except for gaps where the density is zero). For a configuration like in Figure 3.7A most of the points in the problematic region are covered by two contigs, leading to a higher density. Therefore, by inspecting the density profile, we expect to detect these cases where two or more scaffolds are mis-assembled into one scaffold. Figure 3.7B shows the density profile obtained in the assembly process of a real dataset from *E. coli* genome (discussed below in the performance comparison section). Notice that there are two regions with density above the background density of one, and that those high densities are in fact very close to integers (3 and 2). The nearly integral values indicate how many potentially distinct scaffolds have been joined together.

Scaffold segmentation

After detecting high-density regions, we need a procedure to identify and remove the problematic contigs that lead to the merger of disjoint scaffolds. We will call these contigs 'junctures' for future references. We wish to assign the rest of the contigs into distinct scaffolds in such a way that each scaffold has an acceptable density profile. With that goal in mind, we provide each contig i with a variable σ_i. One could think of σ_i's as a putative scaffold label. From the density profile, we can determine q, the total number of distinct labels (scaffolds) that we need.

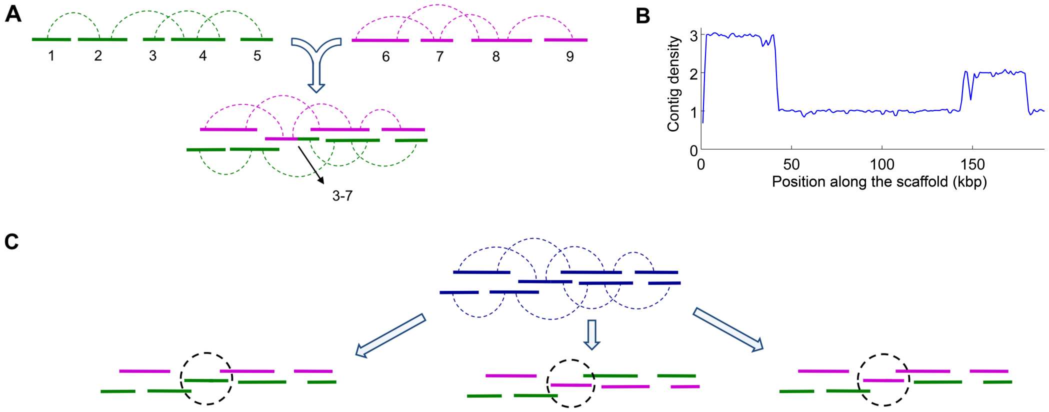

Figure 3.7: Detecting and resolving scaffold mis-assembly using density profile and Potts model. (A) Two scaffolds, shown in green and magenta, belong to the different regions of the genome. Mis-assembly of a chimeric contig composed of contig 3 from the green scaffold and contig 7 from the magenta scaffold causes the two distinct scaffolds to join together. In the new scaffold, many positions are covered by two contigs. (B) For a genuine scaffold, the density profile (see text for definition) should be close to one (or zero for gaps). The plot shows the density profile for a mis-assembled scaffold obtained in the assembly process of a real dataset from the E. coli genome. Each point along the x-axis represents a window of length 1000 bases along the scaffold. The y-axis shows the average density for positions located within each window. From this profile, we can infer that at least four scaffolds have been mis-assembled together. (C) Our labeling method for dividing contigs into distinct groups for the case shown in (A) can lead to any of the three possibilities shown here. We use color to present different labels. Note that the problematic contig (3-7) always lies at the boundary between different groups.

For example, the profile in Figure 3.7B implies $q = 4$.

We want to assign the labels according to two criteria. On one hand, we want the contigs that are directly connected by mate pairs to have the same label. On the other hand, we want the contigs that lie over each other to have different labels. To present these criteria mathematically, we define two matrices D and O. If contigs i and j are directly connected by mate pairs, the matrix element D_{ij} is one; otherwise, it is zero. The matrix element O_{ij} is a positive number monotonically increasing with the length of the region that contigs i and j cover simultaneously. We want to find the label assignment that minimizes the following cost function:

$$E[\sigma] = -\sum_{i,j} D_{ij}\ \delta_{\sigma_i,\sigma_j} + \sum_{i,j} O_{ij}\ \delta_{\sigma_i,\sigma_j}\ . \tag{3.5}$$

Here, $\delta_{\sigma_i,\sigma_j}$ is the Kronecker delta; it is one if σ_i and σ_j are equal and zero otherwise. This cost function is exactly the energy of a q-state Potts model with both ferromagnetic and antiferromagnetic interactions. We use a simulated annealing method [65] to find a configuration of label assignment that minimizes the above energy (details explained in the Appendix E).

In the minimum energy configuration, neighboring contigs belonging to the same scaffold prefer to have the same label while contigs belonging to different scaffolds, juxtaposed in position space, prefer to have different labels. This is a direct consequence of the two criteria with which we began. However, these two criteria cannot be satisfied everywhere at the same time. Around the junctures, namely, contigs joining such juxtaposed scaffolds, the two criteria are at conflict with each other. The result of this conflict is the formation of domain boundaries (change of label) in the neighborhood of the junctures. To get a better sense of this phenomenon, let us revisit the example in Figure 3.7A. The result of label assignment by our algorithm could give rise to any of the three configurations in Figure 3.7C (different labels are shown by different colors). Note that the juncture is always located at the boundary where different labels meet.

Motivated by this discussion, we form an initial list of suspected junctures from the contigs located at label boundaries, namely, contigs having at least one neighbor with a different label. This list often has much fewer members than the original set that we started with. Ideally, one would like to consider the result of removing all the different combinations of suspected contigs from the original set to check if it resolves the problems in density profile. An exhaustive search over all combinations becomes possible when the list is small. Otherwise, one has to limit the list to members located at the densest part of the scaffold. If the list is still too large, we have to proceed with a randomly chosen subset.

After removing any subset of these suspected junctures from the original set of contigs, the remaining set of contigs will form one or more connected components. We score each subset by combining two numbers, one penalizing the formation of too many small components and the other penalizing the presence of high-density regions. We choose the best scoring subset to be removed and focus on the resulting connected components.

For each connected component, we check whether the corresponding density profile is free of high-density regions. All connected components with satisfactory density profiles are declared to be new scaffolds. For the rest, we restart the labeling process individually for each component, and continue this process until all the components have satisfactory density profiles. The removed contigs, either in the Potts model or in the spring model, are reported as single contigs at the end of the assembly.

The Potts model based approach is different from the formulation in terms of non-self-overlapping path introduced in Pop et al. [61]. The method of arbitrarily picking the longest non-self-overlapping path [61] through the tangle might end up joining two scaffolds wrongly. In our method, we remove the problematic contigs, even if, in some cases, it could lead to some good scaffold breaking up. If there are mate pairs overarching the removed contigs, it is possible for scaffolds to have the correct continuation. This is the case for the example in Figure 3.7A, since contigs 6 and 8 are connected by a mate pair overarching contig 7.

Contig joining and gap estimation

In the last stage of scaffold assembly, we decide whether neighboring contigs in a scaffold are to be joined or be separated by a gap. Notice that according to the computed positions, the end of two neighboring contigs might still have a small positional overlap (the density profile is sensitive only to overlaps larger than a few bases); otherwise, they will be separated by a gap. In either the case of positional overlap or the case where the estimated gap is smaller than certain value (e.g. 10 bases), if the ends of neighboring contigs are similar, we join those two contigs. For the rest of the cases, we insert a sequence composed of letter 'N' between the contigs. The length of each sequence is decided by rounding the length of the corresponding gap to the closest multiple of 50. In the special case where there is no sequence similarity, despite the positions indicating a small overlap, we separate the contigs by a 50 base long sequence of 'N'.

3.2.3 Assembly performance on real data

Metrics of assembly quality

Before we discuss our results, we need to define how we assess the quality of a *de novo* assembly. The first obvious measure of performance is the typical size of assembled contigs and scaffolds. This quantity is often reported in terms of an N50 value. Roughly speaking, half of the bases are covered by contigs/scaffolds that are longer than the N50 value. However, N50 provides no indication of the accuracy of the assembled contigs/scaffolds. In order to evaluate the quality of the assembly, it is common to study the performance of the algorithm on data from known genomes. While comparing the assembled components to the reference genome, we need to pay attention to different kinds of errors that could arise and define the metrics of performance accordingly.

To define such metrics, let us bear the following question in mind: In order to map a contig to the reference genome, what type of different operations do we need to do? For example, it might be possible for an entire contig to be matched to a continuous part of the genome with a few mismatches and indels. However, it could also be the case that the contig cannot be matched to a continuous region of the genome; instead, different parts of the contig might match to different regions of the reference genome. Of course, for some contigs, one might not find any significant match at all. In addition to errors in the contigs, there would also be errors in the assignment of relative positions and orientations of contigs in a scaffold.

It is common in the sequence assembly literature to single out mismatch rates and combine some of the other kinds of errors in the no-match category. The emphasis of our algorithm is on using the mate pair information for orienting, positioning and joining contigs. Improper execution of these tasks leads to the formation of chimeric contigs, dislocation and inversion of contigs in a scaffold, as well as merger of distinct scaffolds. Metrics for quality assembly corresponding to these categories of errors are essential for fair comparison among different algorithms. In general, for each algorithm, there is a trade-off between building large scaffolds and making small number of mistakes. For example, a cautious algorithm might produce smaller scaffolds rather than keep on joining suspicious fragments together.

Following the spirit of the above discussion, we will define four categories of errors in order to assess the quality of the assembly. We used MegaBLAST [70] with a minimum identity threshold of 92% to align the sequences against the reference genome (Refseq: NC_007005 for *P. syringae* and NC_010473 for *E. coli*). The sum of the length of all the contigs for which no BLAST hit is found, expressed as a percentage of total assembled bases, is reported as the no-match error rate, ϵ_{no_m}. Each BLAST hit for a contig comes with a number of mismatches and short indels. Mismatch error rate, ϵ_{mis_m}, reports the total number of mismatches and indels as a percentage of total assembled bases. In addition, if only some parts of a contig do not match to the reference genome, the total length of those parts contributes to mismatch counts as well.

As we discussed above, there are other types of error that lead to large-scale 'rearrangements' of genomic sequence. The use of the term 'rearrangement error' is inspired by the analogy with the process of genome evolution. Just as local errors in assembly have similarity to mutations and indels, the large scale errors in assembly, have their evolutionary analogues: inversion, translocations etc.

These rearrangement errors, measured in the unit of number of events per Mbp of assembly, are divided into the following categories. The error rate ϵ_{ch} is associated with chimeric mis-assemblies, namely, the cases where two distinct parts of the genome have been joined into one contig. For chimeric contigs, we would like to differentiate between the cases where the real gap between mis-assembled parts is in the order of few hundred bases and the cases where this gap is in the order of, for example, a few megabases. Therefore, overall error rate ϵ_{ch} is broken down to two parts, ϵ^{s}_{ch} and ϵ^{l}_{ch}, accounting for chimeric contigs involving gaps smaller or larger than 500 bases, respectively.

Apart from the issue of chimeric contigs, we also have erroneous assignment of orientations and positions of contigs in a scaffold. Each time the relative orientation of two neighboring contigs on a scaffold disagrees with that in the reference genome, we have an event contributing to the error rate ϵ^{o}_{sc}. In addition, for any two consecutive contigs in a scaffold, we have an estimated separation, which decides the number of 'N' bases we insert in between those contigs in the final output. For consecutive contigs with verified relative orientations, we compare the estimated separation with the real separation on the reference genome. The last category of rearrangement error rate, ϵ_{sc}^{p}, is associated with the cases where the difference between those values is greater than 500 bases. The two categories of error, presented in this paragraph, keep track of events where two contigs from different strands or from far apart regions have been brought together.

Description of the libraries

We present the assembly result for two real datasets, one being a mate pair library from SOLiD, while the other is of the paired-end kind from Illumina. In paired-end technology, mainly used by Illumina, two reads in a pair come from the opposite strands. In mate pair technology, both reads in a pair are from the same strand. The insert size is also typically larger for the mate pair libraries, which is beneficial for many applications. At the same time, owing to the particular enzymatic steps required to make the mate pairs, there is a higher rate of production of molecules which do not represent true ends of the large DNA molecule. The sequence information from these molecules has to be properly identified and handled so as not to lead to inconsistent scaffolds.

The first dataset is a 50 bp mate pair dataset, generated by SOLiD platform, for the 4.7 Mb genome of *Escherichia coli DH10B*[4]. After we used an in-house filter [71] to remove polyclonal and error-laden reads, we were left with 7.4 million pairs of 50 bp long sequences. For this mate pair library, we used the insert size of 1350 bp (Figure 3.5). Assembly of these reads resulted in very poor quality output. Therefore, we decided to trim down the reads to 35 bp, expecting most of the sequencing errors are concentrated towards the end of the reads [71]. Even after filtering and trimming, the remaining reads provided 100x coverage, and produced better assembly than the raw data set (data not shown).

[4] http://solidsoftwaretools.com/gf/project/ecoli2x50/

The other dataset contains 3.5 million pairs of 36bp long reads from the Illumina platform, providing 40x coverage of the 6.09Mb genome of *Pseudomonas syringae pv. syringae B728a* [72]. For this paired-end library, we used the insert size of 350bp (Figure 3.5).

Performance comparison

We compare the performance of our algorithm to that of Velvet [58]. One reason for selecting Velvet is that several studies found that the performance of Velvet was either better or at least competitive with other available programs [72, 73, 55]. The other reason is that we wanted to study the platform dependence of the performance of SOPRA. Velvet is the only program among the popular assemblers that handles color-space data. For *P. syringae* dataset from the Illumina platform, the original study [72] from which we obtained the library has compared performance of several assemblers. The authors attempted assembly using EULER-SR [56] and SHARCGS [54], but they ran out of random access memory (32 Gb available). It also turned out that Velvet outperforms SSAKE [52], VCAKE [53] and EDENA [55]. These last two assemblers do not incorporate mate pair information and were run only in unpaired mode. ALLPATHS [57] requires multiple paired libraries with different insert sizes. Given the above issues, we decided to proceed with comparison Velvet.

In many areas, including biological data mining, a common exercise for assessing the performance of a binary classifier is to consider the DET or ROC curve [74, 75]. As one reduces the stringency of the classifier, false negative rate decreases at the cost of increasing the false positive rate. DET/ROC curves provide a quantitative representation of this trade-off and are essential for finding optimal operating point that balances the conflicting goals of keeping both of these error rates down. As we mentioned before, in the context of *de novo* assembly, there is a similar trade-off between N50 and the assembly quality [72]. In this analogy, smaller N50 corresponds to having a high false negative rate, while low quality of the assembly plays the role of high false positive rate.

The comparative assembly performance, in the form of N50 versus error rate, is shown in Figures 3.8 and 3.9. Ideally, one would like to be on the top left corners of these graphs, which corresponds to large sizes and low error rates. We present the performance of the algorithms both for contig assembly (triangles) and scaffold assembly (circles).

In the case of *E. coli* data produced by SOLiD platform, for contig assembly, the mismatch rate for V-SOPRA is lower than that for Velvet (Figure 3.8A). This is partly because of error correcting feature of our algorithm for translating color-space data. In contrast, S-SOPRA produces much shorter contigs compared to the other two. Running Velvet with the paired option did not particularly improve the N50, but it increased the mismatch rate significantly. In comparison to Velvet, both V-SOPRA and S-SOPRA perform better in term of scaffold size and error rate, with V-SOPRA outperforming S-SOPRA.

In contrast to the case of the *E. coli* mate pair dataset from SOLiD, pairing information helps Velvet generate much larger scaffolds from the *P. syringae* paired-end Illumina dataset. Figure 3.8B shows the results of running Velvet, with paired option, on the *P. syringae* reads, for two different parameter sets. Note that the two-fold increase in N50 comes at the cost of increasing the error rate by more than one order of magnitude. This trade-off pattern is consistent with a study comparing, among other things, the performance of Velvet for many combinations of parameters [72]. V-SOPRA produces comparable N50 at a much lower mismatch rate. For this particular dataset, the contig building performance of V-SOPRA and Velvet is nearly identical. Like in the *E. coli* dataset, the performance of S-SOPRA is worse than V-SOPRA.

More or less the same pattern continues with the large-scale rearrangement error rates. In Figure 3.9 we report N50 versus the combined rearrangement error

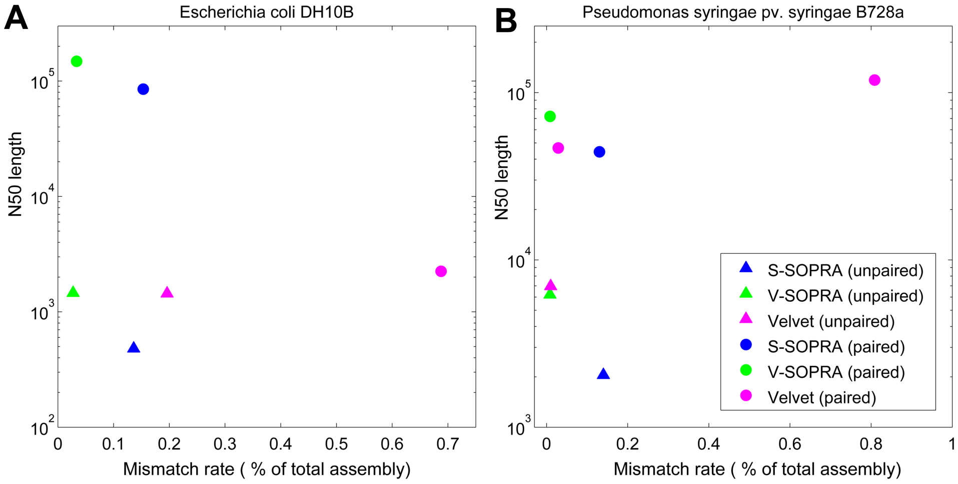

Figure 3.8: N50 vs. combined mismatch and no-match error rate for de novo assembly of real data. See main text and the caption for Table 3.1 for explanation of the error rates.

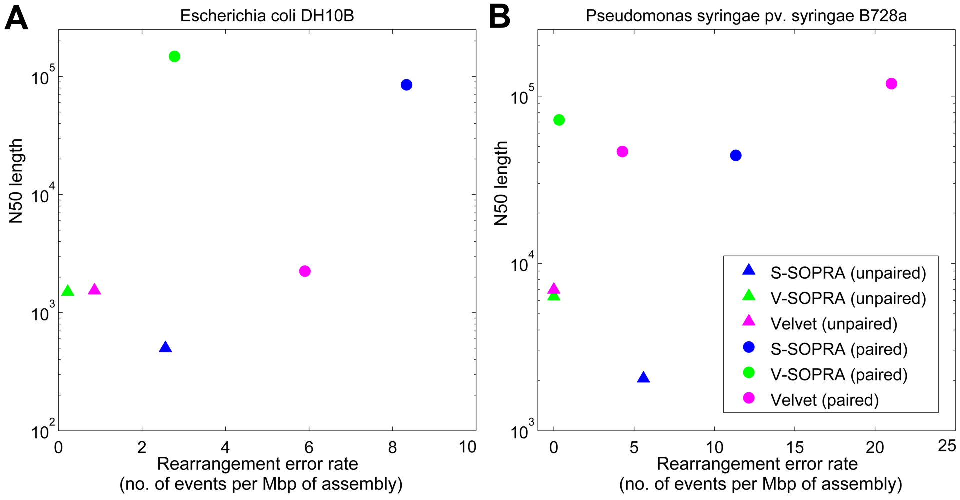

Figure 3.9: N50 vs. combined rearrangement error rate for de novo assembly of real data. See main text and the caption for Table 3.1 for explanation of the error rates.

rates. In the case of Illumina dataset, V-SOPRA did not produce any errors in certain categories (Table 3.1).

In general, for both datasets and all categories of error, our algorithm utilized the mate pair information to enhance N50 by one or two orders of magnitude without significantly increasing the error rates (see details in Tables 3.1 and 3.2). The N50 gain from contigs to scaffolds, for the SOLiD dataset is remarkable for SOPRA when compared to the corresponding gain for Velvet. We believe, based on our simulations (data not shown), that our gain for the Illumina dataset would have been much larger if, instead of being around 350 bases, the insert size of this library were close to a kilobase. Another reassuring aspect of SOPRA as compared to Velvet is that for SOLiD dataset, the algorithm managed to keep the mismatch error rate low, partly thanks to the robust handling of the color-space translation.

We also used MegaBLAST to analyze the contigs which SOPRA removed from the scaffolding process during the assembly. The result is presented in Table 3.3. For the *P. syringae* dataset from Illumina platform, most of the removed sequences were either chimeric or belonged to repeats (referred to as problematic contigs). For the *E. coli* dataset from SOLiD sequencer, slightly more than half of removed sequences were determined to be problematic. In both cases, the total length of removed sequences remains a small fraction of the total assembly. It should be noted that for a removed contig which was not determined to be problematic, there is a possibility that it contains a short stretch of sequence belonging to repeats which was not identified by MegaBLAST.

Assembler	ϵ_{no_m}	ϵ_{mis_m}	ϵ_{ch}^{s}	ϵ_{ch}^{l}	ϵ_{sc}^{o}	ϵ_{sc}^{p}	N50	Genome coverage
	% of tot.	assembly	No. of events/Mbp of assembly				Kbp	%
S-SOPRA (unpaired)	.2	.14	.33	5.25	-	-	2.1	98.4
V-SOPRA (unpaired)	.17	.01	0	0	-	-	6.6	97.7
Velvet (unpaired)	.16	.01	0	0	-	-	7	97.2
S-SOPRA (paired)	.3	.13	0.49	5.58	0.66	3.12	44.2	98.4
V-SOPRA (paired)	.18	.01	0.33	0	0	0	74	97.7
Velvet (paired1)	.16	.02	3.28	0.82	0	0.16	46.7	97.7
Velvet (paired2)	.14	.81	4.93	4.1	1.64	7.56	118.8	96.6

Table 3.1: *De novo* assembly statistics for *P. syringae*. The error rate ϵ_{no_m} represents the sum of length of the contigs/scaffolds with no BLAST hit as a percentage of total assembled bases. Mismatch error rate ϵ_{mis_m} reports the total number of mismatches and indels as a percentage of total assembled bases. The error rates ϵ_{ch}^{s} and ϵ_{ch}^{l} are associated with chimeric mis-assemblies, involving gaps smaller or larger than 500 bases, respectively. The error rate ϵ_{sc}^{o} accounts for the number of cases where the relative orientation of two neighboring contigs disagrees with that in the reference genome. The cases where the estimated separation between two consecutive contigs on a scaffold differs from the real separation in the reference genome by more than 500 bases are associated with ϵ_{sc}^{p}. These last four categories of errors are measure as the number of erroneous events per megabases of assembly.

Assembler	ϵ_{no_m}	ϵ_{mis_m}	ϵ_{ch}^{s}	ϵ_{ch}^{l}	ϵ_{sc}^{o}	ϵ_{sc}^{p}	N50	Genome coverage
	% of tot. assembly		No. of events/Mbp of assembly				Kbp	%
S-SOPRA (unpaired)	.2	.14	.43	2.13	-	-	.5	92.7
V-SOPRA (unpaired)	.02	.03	.22	0	-	-	1.5	94
Velvet (unpaired)	.02	.2	0.22	0.64	-	-	1.5	94.3
S-SOPRA (paired)	.2	.15	0.43	2.13	0.43	2.55	125.5	92.7
V-SOPRA (paired)	.02	.03	0.21	0	0.43	1.7	200.6	94
Velvet (paired)	0.06	0.67	2.55	1.7	0.65	0.87	2.3	94.2

Table 3.2: De novo assembly statistics for *E. coli.* For the definition of different error rates, see the caption for Table 3.1.

	E. coli dataset		*P. syringae* dataset	
	V-SOPRA	S-SOPRA	V-SOPRA	S-SOPRA
Total number of removed contigs	106	338	61	189
Total genomic length of removed contigs (% of total assembly)	192 kb (4.1%)	313 kb (6.7%)	77 kb (1.3%)	272 kb (4.5%)
Number of problematic contigs	130 kb (2.8%)	184 kb (3.9%)	76 kb (1.2%)	233 kb (3.8%)
Total genomic length of problematic contigs (% of total assembly)	58	128	60	164

Table 3.3: Analysis of contigs removed from the scaffolding process. Problematic contigs refer to contigs which are either chimeric, belong to repeats, or do not match to the reference genome. Genomic length means that for repeats, the length is multiplied by the corresponding copy number.

3.3 Discussion

The goal of scaffold assembly is to arrange contigs such that most of the mate pair constraints are satisfied. Given the inconsistencies in the constraints, any solution strategy inevitably has to decide upon a subset of constraints to be ignored. In our algorithm, this choice is made iteratively, going back and forth between the optimization step and removal of offending constraints. For example, in the process of assigning the optimal orientations, we also detect the links that are not satisfied and are to be removed. The same was true for the next step, where, by modeling the links as springs, we both assign the positions and remove the constraints that cause stretch/compression in this solution.

Taking the entire set of remaining mate pair constraints into account simultaneously at each round of optimization is critical to the success of our approach. Some algorithms, at each step, consider only a small subset of contigs and links in between to improve the assembly in a particular region [61, 59, 60]. This manner of local processing of mate pair information stands in stark contrast to our global approach.

In a sequencing project, the issue of large variability in separation of mate pairs (Figures 3.5B and 3.5D) has an important implication for the choice of the insert size in the library preparation. The insert size should preferably be large enough to bridge over most of the small repeats or the shallowly sequenced regions. However, as the typical insert size increases, so does the standard deviation of the separation for individual mate pairs. The averaging effect from having multiple mate pairs between two contigs depends upon the number of such pairs, which, in turn, is limited by the size of the corresponding contigs. Therefore, beyond a certain point, larger insert size might result in higher uncertainty in contig positioning. We expect the optimal insert size to be dependent upon the typical size of the contigs, the depth of coverage, and most importantly, the ability to restrict size variation in the library preparation. In our simulations for assembly of some bacterial genomes, the optimal insert size is typically around 1 Kb, if we were to choose only one insert size (data not shown). However, if the contig assembly mostly produces small fragments, namely, the contig N50 is much less than 1 Kb, the quality of scaffold assembly suffers significantly.

In our study, we emphasized the possible conflict between getting larger scaffolds and avoiding mis-assembly. We showed that the N50/error rate trade-off characteristics for V-SOPRA is excellent. In a practical *de novo* assembly project, mis-assembly rates are hard to estimate. As a result, one may be tempted to increase the N50 without consideration of accumulating inaccuracies [76]. Therefore, it is important for such projects to develop a set of independent benchmarks to assess the accuracy of assembly. The N50/error rate trade-off curve, based on such benchmarks, can be used to set the optimal parameters for the assembler.

Currently, SOPRA is quite conservative and it errs on the side of breaking up things whenever there is any confusion. As we have seen, this tendency has not resulted in smaller N50s compared to other algorithms. However, it is possible that a more sophisticated algorithm could partially reconstruct the structure of repeat regions while solving the orientation and positions of different contigs. One may also be able to breakup some chimeric contigs at the right place rather than remove the whole contig. We hope to include these features in the future versions of the algorithm.

The current HTS platforms not only read sequence fragments but also generate additional information regarding relative position and orientation of pairs of reads. Our methodology is particularly adept at exploiting this extra information. The approach developed here could be easily adapted to any new technology that provides additional positional and orientational constraints on multiple reads. Combination of efficient algorithms for utilization of such constraints and improvements in accuracy of reads leading to better quality contig building will bring us closer to the goal of assembling genomes from the next generation of HTS data.

Chapter 4

Appendix A: Steady State Solutions of the Silencing Model

Equation 2.5 is a third degree equation in S and has 3 solutions. Either 1 or all 3 of the solutions are real. The solutions are easily found using the formula for third degree equations. Let us use a few notations:

$$a = (1+\rho)\gamma^2 \ ,$$
$$b = -\rho\gamma^2 + 2\gamma(1+\rho) + 2\alpha\gamma \ ,$$
$$c = -2\gamma\rho + (1+\rho) + 2\alpha + \alpha^2 \ ,$$
$$d = -\rho \ ,$$
$$k = (-2b^3 + 9abc - 27a^2d + (4(-b^2+3ac)^3 + (-2b^3 + 9abc - 27a^2d)^2)^{.5})^{1/3} \ .$$

Using these notations, the three solutions are:

$$S_1 = -b/(3a) - (2^{(1/3)}(-b^2+3ac))/(3ak) + k1/(3a2^{(1/3)}) \ ,$$
$$S_2 = -b/(3a) + ((1+i3^{.5})(-b^2+3ac))/(2^{(2/3)}3ak) - (1-i3^{.5})k/(6a2^{(1/3)}) \ ,$$
$$S_3 = -b/(3a) + ((1-i3^{.5})(-b^2+3ac))/(32^{(2/3)}ak) - (1+i3^{.5})k/(6a2^{(1/3)}) \ .$$

Chapter 5
Appendix B: Robust Translation of Color-space Data

We saw how the output of our color-space contig assembly consists of a sequence in color-space, C, plus some base-space suggestions, F, at certain locations (Figure 3.4). However, it may not be possible to find a base-space sequence that agrees with all the color-space calls and base-space suggestions. Therefore, we turn the issue of translating this color-space sequence into a search for the most likely DNA sequence that gave rise to this data (C and F). Basically, we set up a hidden variable model. The hidden states of the model are the real letter bases. The color calls and letter base suggestions are the observations. There are two unknown parameters: the probability that a given color call is wrong, and the probability that a letter base suggestion is wrong. For the sake of convenience in calculations, we parameterize these two probabilities as $1/(1+e^{r_c})$ and $1/(1+e^{r_s})$, respectively.

We can then ask for a given C, F, r_c and r_s, what is the probability for a particular base-space sequence, B, to be the real DNA sequence? Let c_i represent the color call between position i and $i+1$ of a contig. At each position, we can have different first base suggestions (one for each short read starting at that position). Let $f_{i,b}$ denote the number of times a particular base $b \in \{A, T, C, G\}$ is suggested at position i. If at certain position there is no suggestion for a particular base, the corresponding $f_{i,b}$ is equal to zero. Let us represent a base-space sequence of length N as $B_{1,N} = b_1 b_2 ... b_N$, where $b_i \in \{A, T, C, G\}$ for all $1 \leq i \leq N$. For each sequence $B_{1,N}$, there is an associated sequence $\tilde{C}_{1,N} = \tilde{c}_1 \tilde{c}_2 ... \tilde{c}_{N-1}$ in color-space such that $\tilde{c}_i$ is the color associated to the dinucleotide $b_i b_{i+1}$. Let us also represent the probability of $B_{1,N}$ being the real DNA sequence, given C, F, r_c and r_s, as: $p_{1,N}(B_{1,N}) = prob(B_{1,N}|C, F, r_c, r_s)$. Using the above notation, we have:

$$p_{1,N}(B_{1,N}) = \left[\prod_{i=1}^{N} \prod_{b \in \{A,T,C,G\}} \left(\frac{e^{r_s \delta_{b_i,b}}}{1+e^{r_s}}\right)^{f_{i,b}}\right] \times \left[\prod_{i=1}^{N-1} \frac{e^{r_c \delta_{\tilde{c}_i,c_i}}}{1+e^{r_c}}\right] . \tag{5.1}$$

$\delta_{\tilde{c}_i,c_i}$ is the Kronecker delta; it is equal to one if the color call between position i and $i+1$ (i.e. c_i) agrees with the color associated with the dinucleotide $b_i b_{i+1}$ (i.e. $\tilde{c}_i$); otherwise, it is zero. $\delta_{b_i,b}$ is the Kronecker delta as well. The next step is to find the base-space sequence that maximizes the above probability. The particular structure of this model allows us to efficiently solve for the optimal sequence using dynamic programming as follows. Consider an arbitrary position k. Equation 5.1 can be written as:

$$p_{1,N}(B_{1,N}) = p_{1,k}(B_{1,k}) \left[\frac{e^{r_c \delta_{\tilde{c}_k,c_k}}}{1+e^{r_c}}\right] p_{k+1,N}(B_{k+1,N}) \ .$$

The middle term on the right hand side contains $\tilde{c}_k$, which depends on both b_k and b_{k+1}. The term $p_{k+1,N}(B_{k+1,N}$ does not contain any variable which corresponds to positions smaller than $k+1$, however, it depends on b_{k+1}. Similarly, the term $p_{1,k}(B_{1,k})$ does not contain any variable which corresponds to positions greater than k, however, it depends on b_k. There are four possibilities for b_k, namely, A, T, C and G. For each of these possibilities, we can ask what $B_{1,k-1} = b_1 b_2 ... b_{k-1}$ will optimize $p_{1,k}(B_{1,k})$. Imagine we know the answer to this question for some arbitrary k. Then, we can easily find the answer to the following question: For each of the four possibilities for b_{k+1}, what $B_{1,k} = b_1 b_2 ... b_k$ will optimize $p_{1,k+1}(B_{1,k+1})$? The reason is that we can write:

$$p_{1,k+1}(B_{1,k+1}) = p_{1,k}(B_{1,k}) \times \left(\frac{e^{r_c \delta_{\tilde{c}_k,c_k}}}{1+e^{r_c}}\right) \times \prod_{b \in \{A,T,C,G\}} \left(\frac{e^{r_s \delta_{b_{k+1},b}}}{1+e^{r_s}}\right)^{f_{k+1,b}} .$$

For each particular choice of b_{k+1}, there are four possibilities for b_k. For each of these possibilities, we know the first term in the right hand side and we can calculate the second and the third term. The information that we have to save at step $k+1$ is that for each b_{k+1}, what is the maximum value of $p_{1,k+1}(B_{1,k+1})$ and what base b_k corresponds to this value.

We start with $k = 1$ where for each of four possibilities for b_1 we can calculate:

$$p_{1,1}(B_{1,1}) = \prod_{b \in \{A,T,C,G\}} \left(\frac{e^{r_s \delta_{b_1,b}}}{1 + e^{r_s}} \right)^{f_{1,b}} .$$

We continue as explained above to find, for each of four possibilities for b_N, what sequence $B_{1,N-1} = b_1 b_2 ... b_{N-1}$ will maximize $p_{1,N}(B_{1,N})$. We have four options for b_N and four corresponding values for $p_{1,N}(B_{1,N})$. We pick the b_N for which the probability $p_{1,N}(B_{1,N})$ is highest. We then go backward and check, for this choice of b_N, what base b_{N-1} was used. We continue this backward process until we get the whole optimum sequence.

The only remaining issue is the choice of values for r_c and r_s. Ideally, we would like to choose these values such that the quantity

$$\sum_B prob(C, F|B, r_c, r_s)$$

is maximized. This quantity represents the probability of observing the data, namely, the color-space contig and first base suggestions. One could use iterative methods like expectation maximization in order to find the optimal values of error rates. However, the translation result is robust for a wide range of parameters and training the rate is not particularly essential in all cases that we encountered, for simulated and for real data. counter-intuitively, the reason for this robustness is as follows. If an error were propagated, it would disagree with most of the subsequent base pair suggestions. The relative strength of r_c versus r_s decides how many such mismatches would be tolerated before a color call error is declared. If the density of first base suggestion is high, color call errors get found out within a few bases, as long as the ratio r_c over r_s is within a reasonable range. The density of first base suggestions is usually high for short read data, given the high coverage and the fact that there is one base suggestion for each incorporated short read. As a first estimate, we can put the probability for a letter base suggestion to be wrong equal to, e_s, the sequencing error rate generated by SOLiD platform. The rough estimate for the probability of a color call being wrong would be e_s^d, where d is the average depth of coverage of the corresponding contig.

Chapter 6

Appendix C: Optimization Strategy for Orientation Assignment

We solve the orientation assignment problem by finding the ground state of an Ising model. In general, this is an NP-complete problem [69, 68]. However, for moderate quality mate pair data, the typical optimization problems that we face have a redeeming feature. In many cases, most of the vertices in the contig connectivity graph are connected to only a few neighboring contigs, thanks to the nearly linear structure of the scaffold. This feature allows us to partition the graph into smaller components on which the optimization can be performed independently. We can then put the partitioned components back together to find the optimal configuration. Below, we explain this procedure in more detail.

An articulation vertex is defined as a vertex such that by removing it from the graph, the graph splits into two or more disconnected components. For each connected component of the graph, we search for articulation vertices that have more than two neighbors (an articulation vertex with only two neighbors is just part of a linear chain in the graph for which the energy optimization can be solved efficiently). After finding an articulation point, we split the graph into the corresponding disconnected components. We give a copy of the articulation vertex to each of these newly formed components. We iteratively continue this procedure on each of these components until we end up with non-reducible ones i.e. components without articulation points that have more than two neighbors. Finding the articulation points and dividing up the graph takes $O(N^2)$ time, where N is the total number of the vertices. We can separately optimize the orientation configuration for these non-reducible components. Notice that, in each component, the optimal configuration has a degeneracy of two, namely, if we reverse all the orientations, we get the same energy ($E[s] = E[-S]$).

Once we have the optimized configuration for each of these components, we reverse the process of iterative partitioning. At each step we join back components formed by removal an articulation vertex. Each of these components was provided with a copy of the articulation vertex. Using the freedom of an overall flip within each component, we arrange to have the same orientation for the copies of the articulation vertex in different components. We can stitch the components together by merging the different copies into a single vertex. The order of merging the articulation vertices is the reverse of the order in which they were split. The reason we can find the global optimum solution by separately optimizing non-reducible components and joining them back together is as follows. Given the definition of the articulation points, there is no edge connecting the non-reducible components in the original graph. In other words, in the energy function, there is no term that includes two vertices which belong to different non-reducible components. As a result, the total energy can be broken up into sums of energies of the non-reducible components. Thus, we can optimize the orientational configuration for each of these components separately, up to an overall reversal within each component. The only set of constraints that has to be satisfied is that the copies of each articulation vertex should have the same orientation. This goal can be easily achieved using the freedom of overall reversal within each component.

In order to optimize the non-reducible components, we proceed as follow. For a given component, we pick a random vertex i and name the singleton set $\{i\}$ to be Z_1. Next, take all the vertices connected to the vertex in Z_1 and call this new set Z_2. We will then consider all the vertices adjacent to the vertices in Z_2, and for each of them, if it does not already belong to Z_1 or Z_2, we put it in a new set called Z_3. We continue until all the vertices in the corresponding connected component have been visited.

For a general graph, the size of Z_k, denoted by $|Z_k|$, grows exponentially as k increases. However, for the contig connectivity graph, because of the linear structure of the scaffolds, in many cases $|Z_k|$ remains a small number and does not grow as increases. For a given non-reducible component, depending on the sizes of s, we choose different strategies. In the case where all the sizes are smaller than a threshold value (e.g. six), we use a dynamic programming approach, similar to the Viterbi algorithm, to optimize the energy, $E[S]$ (Equation 3.1). In the other case, we use the simulated annealing method as explained in Appendix E.

The dynamic programming approach is very similar to the procedure explained above for translation of color-space data into base-space. Note that by construction, a vertex belonging to a set Z_k can only be connected to the vertices belonging to Z_{k-1}, Z_k or Z_{k+1}. In other words, we can write:

$$E_{1,N} = E_{1,k} + E_{k,k+1}^{connection} + E_{k+1,N} \ ,$$

where the expressions for $E_{1,k}$, $E_{k,k+1}^{connection}$ and $E_{k+1,N}$, only contain orientations from vertices belonging to the sets $Z_1 \bigcup Z_2 ... \bigcup Z_k$, $Z_k \bigcup Z_{k+1}$ and $Z_{k+1} ... \bigcup Z_N$, respectively. This means that if we fix orientations of all the vertices belonging to Z_k (there are $2^{|Z_k|}$ possibilities for the choice of these orientations), we can optimize $E_{1,k}$ without any knowledge of the orientations associated with vertices belonging to Z_l, $\forall \ l > k$. At this point, it is clear how we can implement the dynamic programming procedure.

Let $o_k = (S_1^k, S_2^k, ..., S_{|Z_k|}^k)$ be an arbitrary set of orientations for all the vertices belonging to Z_k. There are $2^{|Z_k|}$ possibilities for o_k. For each of these possibilities, we can ask what choice of $O_{1,k-1} = (o_1, o_2, ..., o_{k-1})$ will minimize $E_{1,k}$. If we know the answer to this question for some arbitrary k, then, we can easily find the answer to the following question: For each of the $2^{|Z_{k+1}|}$ possibilities for o_{k+1}, what $O_{1,k} = (o_1, o_2, ..., o_k)$ will minimize $E_{1,k+1}$? The reason is that we can write: $E_{1,k+1} = E_{1,k} + E_{k,k+1}^{connection}$. For each particular choice of o_{k+1}, there are $2^{|Z_k|}$ possibilities for o_k. For each of these possibilities, we know the first term in the right hand side and we can calculate the second term. The information that we have to save at step $k + 1$ is that for each choice of o_{k+1}, what is the minimum value of $E_{1,k+1}$ and what choice of o_k corresponds to this value.

We start with $k = 1$ where for each of 2 possibilities for o_1 (note that Z_1 only has one member), we can calculate $E_{1,1}$ which is equal to zero in both cases. We continue as explained above to find, for each of $2^{|Z_k|}$ possibilities of o_N (N being the total number of Z_k's), what choice of $O_{1,N-1} = (o_1, o_2, ..., o_{N-1})$ will minimize $E_{1,N}$. We have $2^{|Z_N|}$ options for o_N and $2^{|Z_N|}$ corresponding values for $E_{1,N}$. We pick the o_N for which the energy is lowest. Note that because of the degeneracy in the energy function ($E[S] = E[-S]$), there are two choices of o_N with exactly the same energy. We can arbitrary pick either one of them. We then go backward and check, for this choice of o_N, what set of orientation o_{N-1} was used. We continue this backtracking until we get the optimum orientation for all the vertices.

As mentioned before, for a generic graph, size of Z_k's grow with k and the step of going from k to $k+1$ requires a large number of calculations. This is expected as the problem of minimizing Ising energy on an arbitrary graph is NP-complete [69, 68]. However, if the structure of a particular graph allows efficient use of the dynamic programming approach, then the above procedure results in an exact solution. We might have to abandon this method and adopt a heuristic one when there are highly-connected components of moderate or large size.

Figure 6.1A shows a typical region of the contig connectivity graph for the *E. coli* dataset. As one can see, the contig connectivity graph is mostly quite sparse. Assume if we only consider a small part of the graph, similar to the one shown in Figure 6.1B, and defines the Z_k sets starting from an arbitrary point. Given the typical structure in the graph, it is clear why the size of Z_k's do not often grow as k increases. If by removing the articulation points we manage to break up parts of the contig connectivity graph into small components, the above exact method can be applied to most of such components. Some of the branches in Figure 6.1A are part of bigger loops which cannot be seen here. When several such relatively big loops get interconnected, the above optimization strategy often becomes impractical.

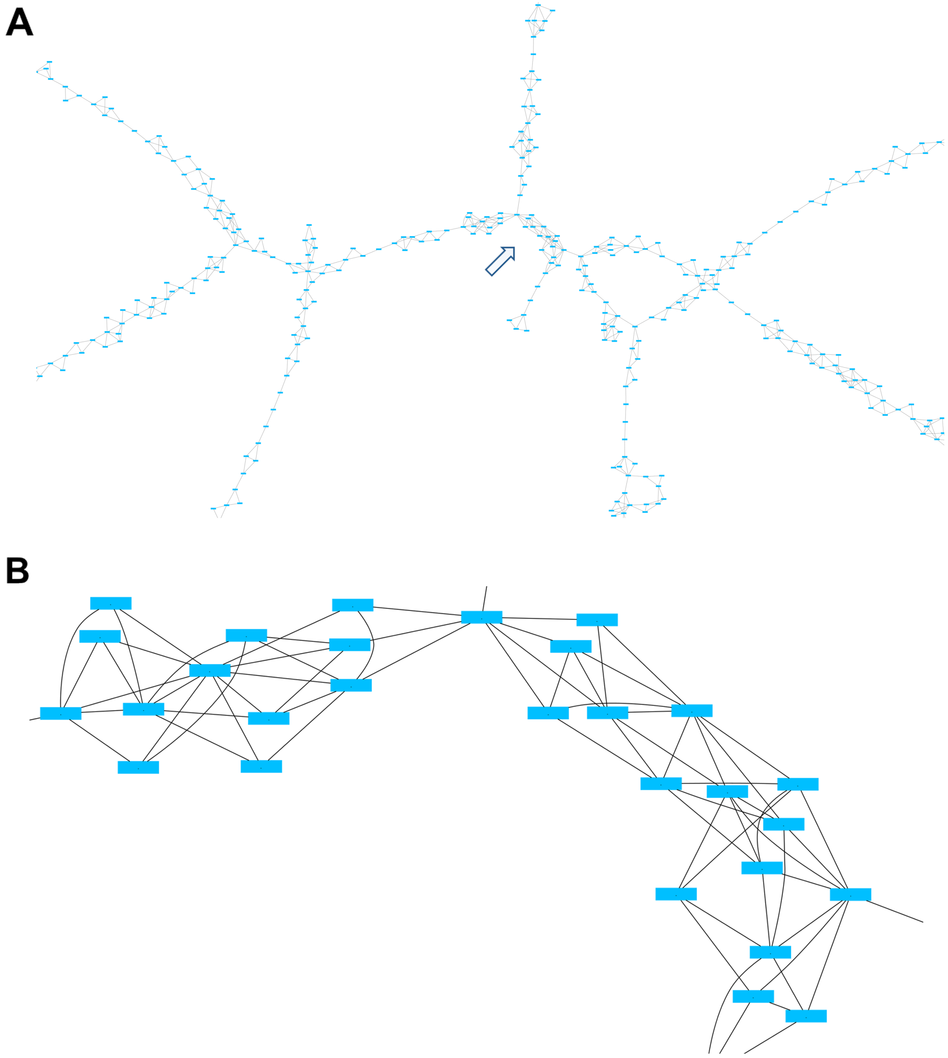

Figure 6.1: A typical region of the contig connectivity graph for the E. coli dataset. (A) The graph typically has a sparse structure. Some of the branches shown are part of bigger loops which cannot be seen here. (B) The blow up of the region indicated by arrow in (A).

Chapter 7

Appendix D: Calculation of $l_{i,j}$

In a SOLiD mate pair library, each pair is composed of two reads, denotes by $R3$ and $F3$. They come from the same strand and $F3$ read is located to the right of $R3$ as one goes from 5' to 3'. Imagine the $R3$ read was used in contig i_R and the $F3$ read was used in contig i_F. Now, let us define the variables τ_R and τ_F. If the $R3$ read itself (and not its reverse compliment) was used in contig i_R, then $\tau_R = 1$; otherwise. $\tau_R = -1$. Similarly, if the $F3$ read itself (and not its reverse compliment) was used in contig i_F, then $\tau_F = 1$; otherwise. $\tau_F = -1$. The position of the $R3$ and $F3$ reads in contigs i_R and i_F is denoted by p_F and p_F, respectively. Also, let Ins denote the insert size of the library. Then, for the suggested distance between contigs i_R and i_F (i.e. $x_F - x_R$), we have:

$$l_{i_F,i_R} = \tau_R \; S_{i_R} \; (Ins + \tau_R \; p_R - \tau_F \; p_F) \; .$$

Here, S_{i_R} is the orientation assigned to contig i_R. For an Illumina paired-end library, the two short reads are located on the opposite strand and face each other. Let us still use the same notation as above, namely, call the first read R and the second one F, etc. Then, the above formula becomes:

$$l_{i_F,i_R} = \tau_R \; S_{i_R} \; (Ins + \tau_R \; p_R + \tau_F \; p_F) \; .$$

Each mate pair, connecting contigs i_R and i_F, provides us with its own suggested distance which we calculate using the above formula. The average of all these suggested distances for contigs i_R and i_F is denoted by $\bar{l}_{i_F,i_R}$.

Chapter 8

Appendix E: Simulated Annealing Method

We explain the procedure in the context of finding the optimal orientation configuration. Simulated annealing [70] is a Monte Carlo method in which one samples the configuration, s, with probability $P[S] \propto exp(-E[S]/T)$, while slowly decreasing the temperature parameter, T, towards zero. If the energy of the system reaches a value close to E_{min} as the temperature goes to zero, it indicates that most of the orientational constraints are satisfied. The advantage of this method over certain greedy approaches is that in simulated annealing, all the contigs and the constraints are treated democratically. Also, in the presence of multiple local optima, one expects simulated annealing to perform better than various domain specific greedy algorithms. In practice, much depends on the particular greedy algorithm and the structure of the graph, as was found in the context of several optimization problems on graphs ([77]). In that study ([77]), it was found that for relatively sparse and regular graphs, simulated annealing did better than some well-established greedy algorithms. This fact, along with many other examples of successful use of simulated annealing[65, 66], motivated our choice.

In simulated annealing, we start from an arbitrary configuration, e.g. $S_i = 1, \forall\ i$. At each step, we randomly choose a contig and check whether by flipping its orientation the energy would decrease or increase. If the energy decreases, we flip the orientation. Otherwise, if the energy increases by ΔE, we flip the orientation with probability $exp(-\Delta E/T)$ where T is a parameter. We start with a large value of T which will allow orientation flip in most cases. After each step, we slightly decrease T according to an exponential cooling schedule [65]. As we go forward, the energy of the system will on average decrease and get closer and closer to E_{min}. This continues until the energy curve reaches a plateau, at which point the search is stopped.

For the Potts model, the only difference is that, instead of the variable S_i, we assign the variable σ_i to contig i. We start with a random label assignment and at each step we make a decision to whether or not change the label of a randomly chosen contigs to a new randomly chosen label. We find that, although the final label configuration may depend upon the choice of initial configuration, the domain boundaries are robustly reconstructed.

In the optimization problems that we face, if the inconsistencies were too severe, the degree of frustration in the system would be very high, and any heuristic method would typically produce a suboptimal solution. In our experience, this is not the case as evidenced by the fact that the energy of the final orientation configuration is close to the minimum energy (data not shown). This fact, on one hand, allows simulated annealing to find the solution. On the other hand, being able to satisfy most of the constraints indicates that the mate pair data is on the whole trustworthy.

Chapter 9

Appendix F: Parameters of the Softwares

SOPRA was implemented in Perl and tested both on a 64-bit Linux and on a Mac OS X server machine. The available memory for both machines was 16 GB.

9.1 V-SOPRA Parameters

For contig assembly part of V-SOPRA, we directly used Velvet v0.7 without invoking the paired option. We get the output in the format of sequence positions in contigs. For base-space data, this information is stored in the *afg* file generated by Velvet. For color-space data, Velvet is part of a pipeline called SOLiD system *de novo* accessory tools (http://solidsoftwaretools.com/gf/project/denovo/). In this pipeline, color-space data has to be preprocessed before inputting to Velvet. Velvet output also has to go through a post-processing step. We use the output of this post-processor that contains the information related to the position of sequences in contigs (the sequences are still in color-space). There is one last step in the pipeline that outputs the final contigs in base-space. However, we do not use this last step. The parameters used for running Velvet in the fragment mode as the first step in V-SOPRA are the same as those described below in the Velvet parameter subsection.

For scaffold assembly, parameter determines the minimum number of mate pairs that have to join two contigs in order for those contigs to be considered connected. For *E. coli* data, we set , whereas for *P. syringae* data we put . Parameter , determining the minimum length that a contig must have in order to be used in the scaffold assembly, was set to for both datasets.

On the Linux machine, the first step of the program, reconstructing the contigs from Velvet output and recording the mate pair information, took 50 minutes for both *E. coli* and *P. syringae* dataset. The color-space translation for *E. coli* data took 14 minutes. The scaffold assembly part took 1.2 hours for *E. coli* and 5 minutes for *P. syringae* dataset. The runtimes were similar for the Mac OS X server.

9.2 S-SOPRA Parameters

S-SOPRA performs contig assembly based upon our modification of SSAKE v3.2 which can also handle color-space data. The crucial parameter for contig assembly is the parameter that determines the minimum required overlap length between two reads. For *E. coli* data we used , whereas for *P. syringae* data we set . For scaffold assembly, we set for *E. coli* data, whereas for *P. syringae* data we put . For *E. coli* data, we set , whereas for *P. syringae* data we put .

The first step of the program that builds the contig based on SSAKE algorithm and records the mate pair information took 8.5 hours for *E. coli* and 6 hours for *P. syringae* dataset. The color-space translation for *E. coli* data took 16 minutes. The scaffold assembly part took 7 hours for *E. coli* and 1.8 hours for *P. syringae* dataset. These numbers are for the Linux machine with similar runtime for the Mac OS X server.

9.3 Velvet Parameters

For Velvet, we tried different combinations of parameters and report results for the ones giving the best performance. For *E. coli* data, Velvet in the fragment mode was run with a hash length of 19 and coverage cutoff of 6x. We ran Velvet in the paired mode using a hash length of 19, coverage cutoff of 6x and coverage expectation of 50.

For *P. syringae* data, Velvet in the fragment mode was run with a hash length of 21 and coverage cutoff of 6x. We ran Velvet in the paired mode using two different parameter sets noted by paired1 and paired2 in Table 3.1 and 3.2. Both parameter sets used hash length of 21 and coverage cutoff of 6x. The coverage expectation for the first parameter set was 12, whereas for the second parameter set we used 50.

References

[1] J.W. Little, D.P. Shepley, and D.W. Wert. Robustness of a gene regulatory circuit. *The EMBO Journal*, 18(15):4299–4307, 1999.

[2] U. Alon, MG Surette, N. Barkai, and S. Leibler. Robustness in bacterial chemotaxis. *Nature*, 397(6715):168–171, 1999.

[3] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. Molecular biology of the cell. 2002. *New York: Garland Science.*

[4] H.F. Lodish and A. Berk. *Molecular cell biology.* WH Freeman, 2008.

[5] B. Sanson. Generating patterns from fields of cells: Examples from drosophila segmentation. *EMBO reports*, 2(12):1083, 2001.

[6] Scott F. Gilbert. *Developmental biology.* Sinauer Associates, Sunderland, MA, 9th ed edition, 2010.

[7] G. von Dassow, E. Meir, E.M. Munro, and G.M. Odell. The segment polarity network is a robust developmental module. *Nature*, 406(6792):188–192, 2000.

[8] R. Heinrich and S. Schuster. *The regulation of cellular systems.* Kluwer Academic Pub, 1996.

[9] M.A. Savageau. Parameter sensitivity as a criterion for evaluating and comparing the performance of biochemical systems. 1971.

[10] K.S. Brown and J.P. Sethna. Statistical mechanical approaches to models with many poorly known parameters. *Physical Review E*, 68(2):21904, 2003.

[11] R.N. Gutenkunst, J.J. Waterfall, F.P. Casey, K.S. Brown, C.R. Myers, and J.P. Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput Biol*, 3(10):1871–1878, 2007.

[12] KS Brown, CC Hill, GA Calero, CR Myers, KH Lee, JP Sethna, and RA Cerione. The statistical mechanics of complex signaling networks: nerve growth factor signaling. *Physical Biology*, 1:184–195, 2004.

[13] D. Waxman and S. Gavrilets. 20 questions on adaptive dynamics. *Journal of Evolutionary Biology*, 18(5):1139–1154, 2005.

[14] M. Djordjevic, A.M. Sengupta, and B.I. Shraiman. A biophysical approach to transcription factor binding site discovery. *Genome research*, 13(11):2381, 2003.

[15] N.T. Ingolia. Topology and robustness in the drosophila segment polarity network. *PLoS Biology*, 2(6), 2004.

[16] R. Albert and HG Othmer. The topology of the regulatory interactions predicts the expression pattern of the drosophila segment polarity genes. *J. Theor. Biol*, 223(1):1–18, 2003.

[17] M. Chaves, E.D. Sontag, and R. Albert. Methods of robustness analysis for boolean models of gene control networks. *Arxiv preprint q-bio/0605004*, 2006.

[18] M. Chaves, R. Albert, and E.D. Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *Journal of theoretical biology*, 235(3):431–449, 2005.

[19] W. Ma, L. Lai, Q. Ouyang, and C. Tang. Robustness and modular design of the drosophila segment polarity network. *Molecular Systems Biology*, 2(1), 2006.

[20] G. Von Dassow and G.M. Odell. Design and constraints of the drosophila segment polarity module: robust spatial patterning emerges from intertwined cell state switches. *J Exp Zool Mol Dev Evol*, 294:179–215, 2002.

[21] M. Chaves, A. Sengupta, and E.D. Sontag. Geometry and topology of parameter space: investigating measures of robustness in regulatory networks. *Journal of Mathematical Biology*, 59(3):315–358, 2009.

[22] KM Cadigan, U. Grossniklaus, and WJ Gehring. Localized expression of sloppy paired protein maintains the polarity of drosophila parasegments. *Genes & development*, 8(8):899, 1994.

[23] A. Hidalgo and P. Ingham. Cell patterning in the drosophila segment: spatial regulation of the segment polarity gene patched. *Development*, 110(1):291, 1990.

[24] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323, 2000.

[25] J.B. Tenenbaum, V. Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.

[26] M. Grunstein. Yeast heterochromatin: Minireview regulation of its assembly and inheritance by histones. *Cell*, 93:325–328, 1998.

[27] A. Hecht, S. Strahl-Bolsinger, and M. Grunstein. Spreading of transcriptional represser sir3 from telomeric heterochromatin. 1996.

[28] S.I.S. Grewal and D. Moazed. Heterochromatin and epigenetic control of gene expression. *Science*, 301(5634):798, 2003.

[29] D. Moazed. Common themes in mechanisms of gene silencing. *Molecular cell*, 8(3):489–498, 2001.

[30] ER Gansner and SC North. An open graph visualization system and its applications to software engineering. *Software-Practice & Experience*, 30(11):1203–1233, September 2000.

[31] N. Suka, K. Luo, and M. Grunstein. Sir2p and sas2p opposingly regulate acetylation of yeast histone h4 lysine16 and spreading of heterochromatin. *Nature genetics*, 32(3):378–383, 2002.

[32] O.M. Aparicio, B.L. Billington, and D.E. Gottschling. Modifiers of position effect are shared between telomeric and silent mating-type loci in S. cerevisiae. *Cell*, 66(6):1279–1287, 1991.

[33] M. Sedighi and A.M. Sengupta. Epigenetic chromatin silencing. *Physical biology*, 4:246–255, 2007.

[34] X. Bi and J.R. Broach. Uasrpg can function as a heterochromatin boundary element in yeast. *Genes & development*, 13(9):1089, 1999.

[35] D. Donze, C.R. Adams, J. Rine, and R.T. Kamakaka. The boundaries of the silenced hmr domain in saccharomyces cerevisiae. *Genes & development*, 13(6):698, 1999.

[36] L. Pillus and J. Rine. Epigenetic inheritance of transcriptional states in S. cerevisiae. *Cell*, 59(4):637–647, 1989.

[37] AE Ehrenhofer-Murray, DH Rivier, and J. Rine. The role of sas2, an acetyltransferase homologue of saccharomyces cerevisiae, in silencing and orc function. *Genetics*, 145(4):923, 1997.

[38] E.Y. Xu, K.A. Zawadzki, and J.R. Broach. Single-cell observations reveal intermediate transcriptional silencing states. *Molecular cell*, 23(2):219–229, 2006.

[39] J.P. Keener. Propagation and its failure in coupled systems of discrete excitable cells. *SIAM Journal on Applied Mathematics*, 47(3):556–572, 1987.

[40] Y. Katan-Khaykovich and K. Struhl. Heterochromatin formation involves changes in histone modifications over multiple cell generations. *The EMBO Journal*, 24(12):2138–2149, 2005.

[41] K.J. Bitterman, R.M. Anderson, H.Y. Cohen, M. Latorre-Esteves, and D.A. Sinclair. Inhibition of silencing and accelerated aging by nicotinamide, a putative negative regulator of yeast sir2 and human sirt1. *Journal of Biological Chemistry*, 277(47):45099, 2002.

[42] C.A. Hutchison III. DNA sequencing: bench to bedside and beyond. *Nucleic Acids Research*, 2007.

[43] F. Sanger, AR Coulson, T. Friedmann, GM Air, BG Barrell, NL Brown, JC Fiddes, CA Hutchison, et al. The nucleotide sequence of bacteriophage [phi] x174. *Journal of Molecular Biology*, 125(2):225–246, 1978.

[44] RD Fleischmann, MD Adams, O. White, RA Clayton, EF Kirkness, AR Kerlavage, CJ Bult, JF Tomb, BA Dougherty, JM Merrick, et al. Whole-genome random sequencing and assembly of haemophilus influenzae rd. *Science*, 269(5223):496, 1995.

[45] S.C. Schuster. Next-generation sequencing transforms today's biology. *Nature*, 200:8.

[46] Jay Shendure and Hanlee Ji. Next-generation dna sequencing. *Nat Biotechnol*, 26(10):1135–45, Oct 2008.

[47] D. MacLean, J.D.G. Jones, and D.J. Studholme. Application of'next-generation'sequencing technologies to microbial genetics. *Nature Reviews Microbiology*, 2009.

[48] E.R. Mardis. The impact of next-generation sequencing technology on genetics. *Trends in Genetics*, 24(3):133–141, 2008.

[49] E. Schrodinger and Lewin. *What is life?* Cambridge University Press Cambridge, 1968.

[50] M. Pop. Genome assembly reborn: recent computational challenges. *Briefings in bioinformatics*, 10(4):354, 2009.

[51] E.W. Myers, G.G. Sutton, A.L. Delcher, I.M. Dew, D.P. Fasulo, M.J. Flanigan, S.A. Kravitz, C.M. Mobarry, K.H.J. Reinert, K.A. Remington, et al. A whole-genome assembly of drosophila. *Science*, 287(5461):2196, 2000.

[52] R.L. Warren, G.G. Sutton, S.J.M. Jones, and R.A. Holt. Assembling millions of short dna sequences using ssake. *Bioinformatics*, 23(4):500, 2007.

[53] W.R. Jeck, J.A. Reinhardt, D.A. Baltrus, M.T. Hickenbotham, V. Magrini, E.R. Mardis, J.L. Dangl, and C.D. Jones. Extending assembly of short dna sequences to handle error. *Bioinformatics*, 23(21):2942, 2007.

[54] J.C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer. Sharcgs, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Research*, 17(11):1697, 2007.

[55] D. Hernandez, P. François, L. Farinelli, M. Østerås, and J. Schrenzel. De novo bacterial genome sequencing: Millions of very short reads.

[56] M.J. Chaisson and P.A. Pevzner. Short read fragment assembly of bacterial genomes. *Genome Research*, 18(2):324, 2008.

[57] J. Butler, I. MacCallum, M. Kleber, I.A. Shlyakhter, M.K. Belmonte, E.S. Lander, C. Nusbaum, and D.B. Jaffe. Allpaths: De novo assembly of whole-genome shotgun microreads. *Genome Research*, 18(5):810, 2008.

[58] D.R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821, 2008.

[59] P.A. Pevzner and H. Tang. Fragment assembly with double-barreled data. *Bioinformatics*, 17(Suppl 1):S225, 2001.

[60] D.H. Huson, K. Reinert, and E.W. Myers. The greedy path-merging algorithm for contig scaffolding. *Journal of the ACM (JACM)*, 49(5):603–615, 2002.

[61] M. Pop, D.S. Kosack, and S.L. Salzberg. Hierarchical scaffolding with bambus. *Genome Research*, 14(1):149, 2004.

[62] K.J. McKernan, H.E. Peckham, G.L. Costa, S.F. McLaughlin, Y. Fu, E.F. Tsung, C.R. Clouser, C. Duncan, J.K. Ichikawa, C.C. Lee, et al. Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome research*, 19(9):1527, 2009.

[63] Ross Kindermann and J. Laurie Snell. *Markov random fields and their applications*. Contemporary mathematics ; v. 1. American Mathematical Society, Providence, R.I., 1980.

[64] KH Fischer and JA Hertz. *Spin Glasses (Cambridge*. Cambridge University Press, 1991.

[65] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[66] P.J.M. van Laarhoven and EHL Aarts. Simulated annealing: theory and applications. *Mathematics and Its Applications, D. Reidel, Dordrecht*, 1987.

[67] J.D. Kececioglu and E.W. Myers. Combinatorial algorithms for dna sequence assembly. *Algorithmica*, 13(1):7–51, 1995.

[68] M.R. Garey, D.S. Johnson, et al. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH freeman San Francisco, 1979.

[69] F. Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15:3241–3253, 1982.

[70] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A greedy algorithm for aligning DNA sequences. *Journal of Computational Biology*, 7(1-2):203–214, 2000.

[71] A. Sasson and T.P. Michael. Filtering error from solid output. *Bioinformatics*, 26(6):849, 2010.

[72] R.A. Farrer, E. Kemen, J.D.G. Jones, and D.J. Studholme. De novo assembly of the Pseudomonas syringae pv. syringae B728a genome using Illumina/Solexa short sequence reads. *FEMS Microbiology Letters*, 291(1):103–111, 2008.

[73] S.L. Salzberg, D.D. Sommer, D. Puiu, and V.T. Lee. Gene-boosted assembly of a novel bacterial genome from very short reads. *PLoS Computational Biology*, 4(9), 2008.

[74] J.P. Egan. *Signal detection theory and ROC-analysis*. Academic Pr, 1975.

[75] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. In *Fifth European Conference on Speech Communication and Technology*. Citeseer, 1997.

[76] S.L. Salzberg and J.A. Yorke. Beware of mis-assembled genomes. *BIOINFORMATICS-OXFORD-*, 21(24):4320, 2005.

[77] DS Johnson and CR Aragon. LA McGeoch and C. Schevon, 1989. Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning. *Operations Research*, 37(6):865–892.

Vita

Adel Dayarian

2005-10 – PhD candidate, Department of Physics and Astronomy, Rutgers University.

2004-05 – DEA de Physique Theorique, Ecole Normale Supérieure, France.

2002-04 – Magistère Interuniversitaire de Physique, Ecole Normale Supérieure, France.

2000-02 – Sharif University of Technology, Iran.