GSMMC

Brownian Chain Evolution: A Propulsion Mechanism?

David Boy, UCSB
Michael Case, Clemson University
Holly Clark, University of Tennessee
Sohan Dharmaraja, MIT
Nicole Greaney, RPI
Timur Milgrom, Clemson University
Yogesh Joshi, NJIT
Katie Newhall, RPI

Mentor
Prof. Donald Drew, Rensselaer Polytechnic Institute

Rensselaer Polytechnic Institute
July 17, 2007

Abstract

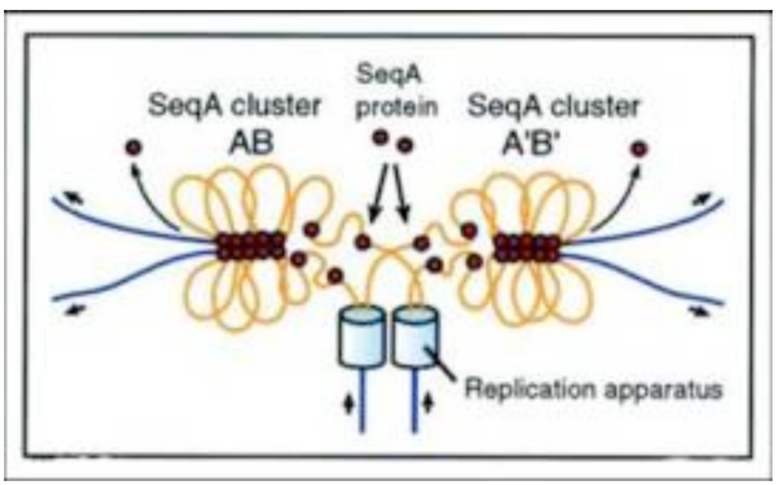
We will attempt to model DNA straightening phenomena using a discrete and a continuous approach. The discrete model uses Markov property assumptions to model the motion of the DNA strand as a Markov chain. The continuous approach models the behavior using a SDE. Both of the method tested numerically and compared to the physical experiment done by the biologists.

Contents

1 Introduction

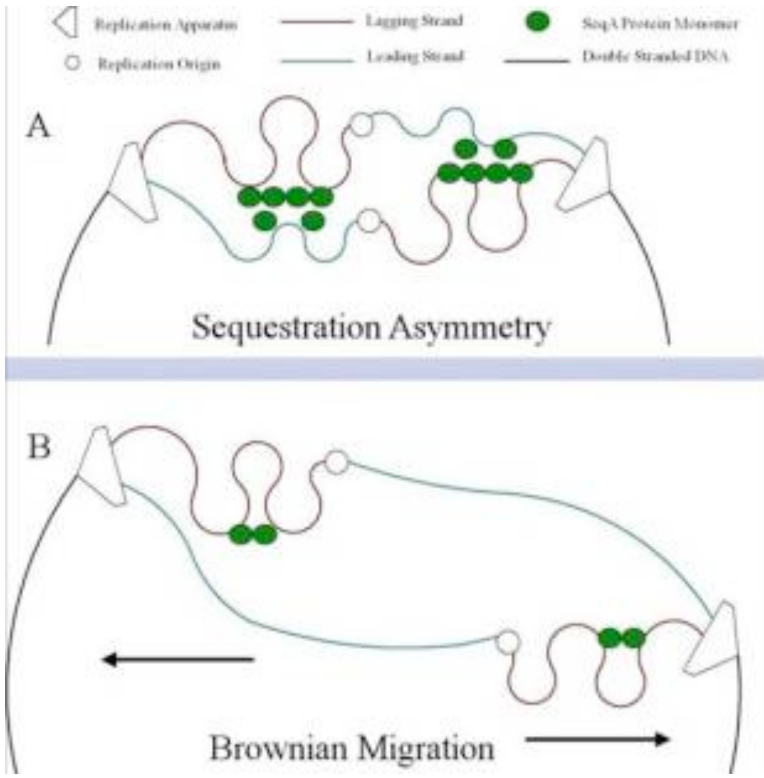
E. coli is a simple organism that is easy to study, and has evolved clever ways to accomplish its several tasks. One of these tasks involves the control of starting DNA replication. Another is the propulsion system for migration of the replicated DNA into opposite ends of the cell in preparation for cell division. The modeling effort examines a mechanism involving a single protein which may have a role in both processes.

The DNA of E. coli forms a loop. When it divides, E. coli replicates its DNA starting at a location called oriC. The replication proceeds both ways along the loop, meeting at a location called ter. E. coli has evolved a regulation mechanism so that re-initiation cannot occur too soon during replication. This mechanism involves the binding of a protein called SeqA to newly replicated sites on the DNA marked by the sequence GATC. SeqA is also capable of binding to itself. Researchers have identified clumps of SeqA near the origin sites of replicating DNA, which split and migrate toward the opposite sides of the cell (see Figure 1). Biologists have sought a "propulsion molecule" in E. coli, but have been unsuccessful. We shall derive a model for, and study a mechanism by which the binding of SeqA to the newly-replicated DNA and to itself could result in loops of DNA, which when released by the SeqA bundles, could straighten to propel the remaining SeqA and DNA apart. The proposed mechanism is as follows: As the newly formed DNA emerges from the



DNA polymerase family, it is bound by SeqA at GATC sites, and the clumping property of SeqA cause the DNA to form loops. These loops have a high degree of "order", which can be thought of as an energy stored as bending. As the SeqA molecules become unbound from the GATC sites, the loops can straighten. If the straightening is unsymmetrical, the different DNA strands can be pushed in different directions (see Figure 1).

We believe that straightening is an important ability for any DNA strand to possess, as it acts as a mechanism by which materials can be transported from one end of the cell to another. The question we intend to address is: Can we formulate a mathematical model by which we can simulate the strands tendency to straighten? Further, does our model provide



a reasonable simulation consistent with biological evidence? In the following sections we will propose a discrete and a continuous mathematical model for the straightening of the DNA strand.

2 Discrete Model

We considered the DNA strand to have a chain like structure where the position of the chain at certain time can be though of as chain being at certain stage. We will make some basic assumptions about the chain. First we will assume that the chain moves according to random Brownian motion. The second assumption is that the future movement only depends on the current position of the chain. The third assumption is the chain wants to straighten or tends to favor the straightest stage.

Based of these assumptions we will construct a Markov chain that will model the movement of the chain. In order to bias the chain to move towards the straightest state we will bias the probabilities towards that state. In the following sections we will go into more detail of how we defined the DNA chain and what we mathematically mean by it being straight. We also attempt to have a systematic assignment of probabilities for a transition matrix of the model.

2.1 Defining straightness

To avoid ambiguity, we have devised two methods for determining the straightness of a given DNA chain mathematically. The DNA chains which we are working with were randomly generated in Matlab, using the "1"," 0","-1" idea explained by Dr. Drew in his presentation. Graphically, a "1" signifies a left turn in the path along the chain, at an angle of 60° degrees, and similarly, a "-1" signifies a right turn. A zero signifies that the path remains straight at that point and there is no angular change in direction. Therefore, each DNA chain represents a vector consisting of only ones, zeroes, and negative ones.

2.1.1 Penalty Method

In the first method for determining straightness, called the Penalty Method, we devised a way of assigning and deducting points for certain numbers and sequences of numbers in a given vector, which will give us its straightness score. We know that zeroes, and especially a long sequence of zeroes, are good, so we assign 2 points for each to the total straightness score. As a "1" or a "-1" represent a left and a right turn respectively, we would assign only 1 point for each time we encounter a "1" or a "-1" immediately after a zero, and the same for a "1" that follows a "-1" (or vice versa). In other words, any time there is a change in dyad of components (unless the second one is a zero, in which case we add 2 points like before). Finally, if there is a sequence of repeated ones or negative ones, we know that this signifies a looping of the chain, which we consider to be worse or less straight than just a simple change in direction of 60 degrees. We, thus, deduct 1 point for every repeated number in the sequence, and we obtain our total straightness score.

2.1.2 Euclidean Method

The other method was designed to measure the factor of straightness by measuring the Euclidean distance between the first node in the chain and the last node in the chain. The method that accomplished the task of calculating the distance worked in the following systematic way. We started with the first node being placed in the origin of our coordinate system and the "0" in the first slot of the chain would imply that the string moved in the direction of the $(1,0)$ unit vector. The "$+1$" and the "-1" would imply that the $(1,0)$ unit vector would be rotated by "$+60°$" and "$-60°$" respectively. So we would calculate the unit vector pointing in the correct direction by the use of the rotation matrix

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

where θ is the angle of rotation. Once we calculate the rotated unit vector we add the x and the y component to the starting point to get the location of the second point. In order to get the location of the third node we once again rotate the unit vector that we remember from the previous step and add the x and y components to the location of the second node. We continue this process until we get the locations of all the nodes. Finally once we get the the location of the last node we calculate the Euclidean distance between the first and the last node and divide it by the number of nodes. We define this quantity to be the straightness factor of the chain. The straightness factor is from 0 to 1 with 1 being the straightest score possible.

2.2 The Markov Model

We decided to implement a discrete Markov model, to model the Brownian effects on our DNA chain. The naïve approach of enumerating every possible state of our chain would be impractical: a chain of length n has 3^n possible configurations. To overcome this, we made several simplifying assumptions (with the intention of greatly reducing the size of our state space)

1. The chain *wants* to become straight (maximizing entropy)
2. Chains prefer to make *smaller* changes in straightness, rather than larger ones.
3. Each node can only make transitions to *adjacent* angles.

We captured the fact that the chain wants to become straight by implementing an 'acceleration factor' - meaning at each transition, the transition probabilities were computed dynamically, based on the straightness factor of the current state.

2.3 Finding Average Length is "Complex"

We thought that we could develop a systematic way of assigning probabilities in the transition matrix based on the length of the chain and how it relates to the average. So we attempted to calculate the average length of a chain with certain number on nodes. Let $\alpha = e^{i\triangle\theta}$, $\triangle\theta = \frac{\pi}{3}$, $r_i = \{\pm 1, 0\}$, consider $Z_n = 1 + \alpha^{r_1} + \alpha^{r_1+r_2} + \cdots + \alpha^{r_1+r_2+\cdots+r_{n-1}}$ to be a complex representation of the chain with n nodes. Based of this representation we could calculate the magnitude of the distance between the first and the last node by

$$\begin{aligned}
||Z_n||^2 &= (1 + \alpha^{r_1} + \alpha^{r_1+r_2} + \cdots + \alpha^{r_1+r_2+\cdots+r_{n-1}}) \\
&\quad \cdot(1 + \alpha^{-r_1} + \alpha^{-r_1-r_2} + \cdots + \alpha^{-r_1-r_2-\cdots-r_{n-1}}) \\
&= n + (\alpha^{r_1} + \alpha^{-r_1}) + (\alpha^{r_2} + \alpha^{-r_2}) + \cdots + (\alpha^{r_{n-1}} + \alpha^{-r_{n-1}}) \\
&+ (\alpha^{r_1+r_2} + \alpha^{-r_1-r_2}) + (\alpha^{r_1+r_2+r_3} + \alpha^{-r_1-r_2-r_3}) + \cdots \\
&+ (\alpha^{r_1+r_2+\cdots+r_{n-1}} + \alpha^{-r_1-r_2-\cdots-r_{n-1}}) \\
&+ \cdots + (\alpha^{r_{n-2}+r_{n-1}} + \alpha^{-r_{n-2}-r_{n-1}}).
\end{aligned}$$

Observe that the formula consists of powers with one r_j, sums of two r_j 's, sums of three r_j 's, etc. Since each r_j is either 1,-1, or 0, we can list the possible sums as shown below.

Sum=	-4	-3	-2	-1	0	1	2	3	4
Sum of Singles=				1	1	1			
Sum of Doubles=			1	2	3	2	1		
Sum of Triples=		1	3	6	7	6	3	1	
Sum of Quadraples=	1	4	10	16	19	16	10	4	1

If we wish to get an expected value for the distance between the first and last node in a chain with n nodes, we want to sum all possible lengths and divide by the total possible number of chains. After writing a program to do this (since we could find no closed form), we discovered that the length behaved like $\sum_{j=0}^{n-1}(\frac{2}{3})^j$ where n was the number of nodes.

3 Stochastic Model

The deterministic system is modeled by

$$\dot{\mathbf{x}} = \mathbf{v} \tag{1}$$

$$\dot{\mathbf{v}} = f(\mathbf{x}) = k\left(\|\mathbf{x}_{i-1} - \mathbf{x}_i\| - L\right)\frac{\mathbf{x}_{i-1} - \mathbf{x}_i}{\|\mathbf{x}_{i-1} - \mathbf{x}_i\|} + k\left(\|\mathbf{x}_{i+1} - \mathbf{x}_i\| - L\right)\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|} - c\mathbf{v} \tag{2}$$

This system, if started static would remain static. What causes the system to move is , or thermal excitation. Brownian motion is an approximation to the physical force of atoms bumping into each other. As it is a random force in any direction, with Gaussian distributed amplitude, it can be approximated by the Weinner Increment. Thus the Ito stochastic differential equation to model this system is

$$d\mathbf{x} = \mathbf{v}dt \tag{3}$$

$$d\mathbf{v} = f(\mathbf{x})dt + bdW(t) \tag{4}$$

where b is the magnitude of the Wiener process, which would be related to the temperature of the surroundings.

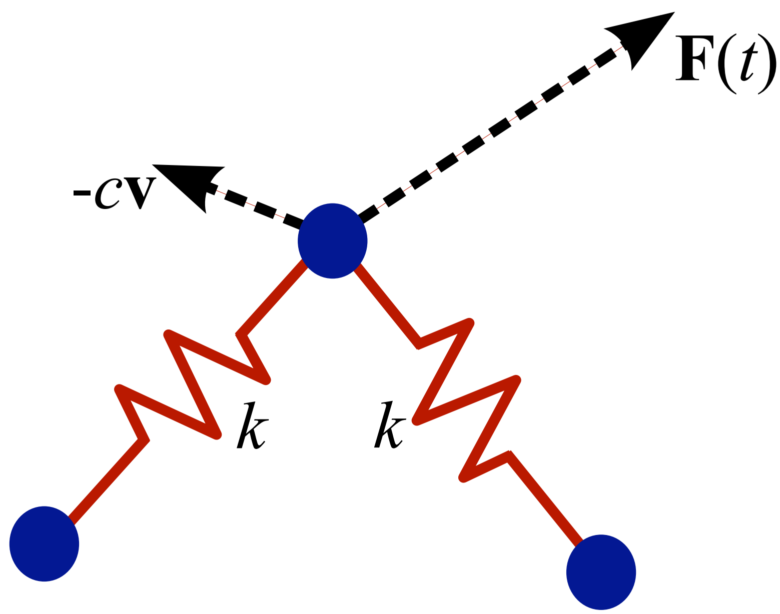
This nonlinear system is solved numerically in two parts. The deterministic part is handled by the ??????? scheme, while the simple stochastic part ($bdW(t)$) is handled by the Euler-Marayama scheme. For this 2D system, $dW(t)$ is taken to be symmetrically distributed around the circle, with Gaussian distributed magnitude, then broken down into its x and y components.

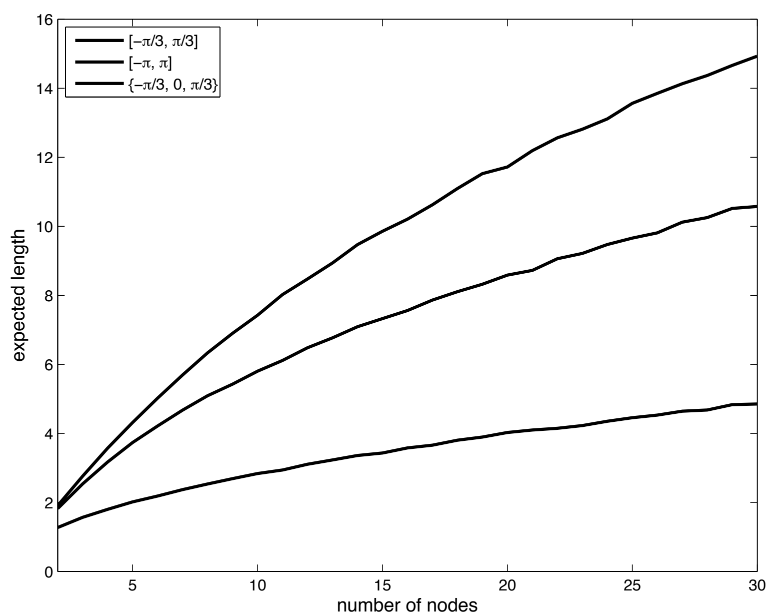
$$\dot{\mathbf{x}} = \mathbf{v} \tag{5}$$

$$\dot{\mathbf{v}} = k\left(\|\mathbf{x}_{i-1} - \mathbf{x}_i\| - L\right)\frac{\mathbf{x}_{i-1} - \mathbf{x}_i}{\|\mathbf{x}_{i-1} - \mathbf{x}_i\|} + k\left(\|\mathbf{x}_{i+1} - \mathbf{x}_i\| - L\right)\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|} - c\mathbf{v} + \mathbf{F}(t) \tag{6}$$

$\mathbf{F}(t)$ is a stochastic process simulating Brownian motion.

Support for this model from Cohen, Moemer 2007.





$E(n) \propto \sqrt{n}$

$E(n)$: expected length.

4 Conclusion and Results

References

[1] Sota Hiraga, Chiyome Ichinose, Toshinari Onogi, Hironori Niki and Mitsuyoshi Yamazoe, Bidirectional migration of SeqA-bound hemimethylated DNA clusters and pairing of oriC copies in Escherichia coli, Genes Cells, 5 (2000), 327-41.

[2] Toshinari Onogi, Hironori Niki, Mitsuyoshi Yamazoe and Sota Hiraga, The assembly and migration of SeqAGfp fusion in living cells of Escherichia coli, Molecular Microbiology, 31 (1999), 17751782.