Min Protein Oscillations in E. Coli.

Rene Zeto.

Abstract.

The Min protein system in Escherichia coli helps the cell division process by identifying the center of the cell [1, 7]. This system has been modeled succesfully computationally under standard conditions. There has been recent experimental interest in the cell division process for significantly perturbed cell shapes. We take the existing computational model, and extend it to such a perturbed shape. We find that the existing computational model does not agree with these newer perturbed shapes, but that several interesting conclusions about the stability of the oscillations in larger cells can still be drawn. Furthermore, we explore an alternative model as a potential avenue for future work.

Acknowledgements.

I would like to thank Jeff Schulte, with whom I worked with closely on this project, Dr. Roundy, my advisor, and Dr. Tate, without whom this thesis would not be written.

I. Table of Contents.

II. Introduction

§ Section 1: Motivation.

In E. Coli, a successful cell division requires certain cell systems to be in the geometric center of the cell during the split, in order to produce two functional daughter cells. Earlier works have shown the Min protein system inside E. Coli to be vital in regulating the locations of these systems during the cell division process. If the shape of the cell is perturbed by environmental factors, as can be done experimentally, the Min protein system will undergo spatial oscillations in the cell, such that the aforementioned cell systems are, on average, located where they need to be during the division process [3, 2, 10, 12, 9].

Additionally, the Min protein system has been experimentally found to be fairly robust under different cell perturbations [11]. A recent study has shown that even with fairly significant deformation of the cell membrane, the Min system will still oscillate in an appropriate manner as to produce a successful cell division. These experimental results provide interesting insight into the conditions under which the Min protein system oscillations will occur, and invite more questions about the mechanism which causes the oscillations. Furthermore, the types of oscillatory behavior undergone by the system are not yet well understood.

Models describing the behavior of the Min protein system's oscillations, under normal conditions, do exist. The primary model, known as the continuum model, consists of a set of five coupled partial differential equations, which we discretize and solve on a three dimensional grid (see Methods) [4]. Previous works have verified the effectiveness of this model, but have not yet extended it to the diversified cell shapes created experimentally. We aim to both test the robustness of this model, study the types of oscillations that occur, and test the conditions under which these different types of oscillations occur.

§ Section 2: Experimental background.

Past experiments have shown that the E. Coli cell division process is fairly resilient under physical deformation of the cell. In a paper by Mannik *et. al*, bacteria were forced into two different etched chambers on silicon chips. One chamber was deep enough for the bacteria cells to retain their standard pill shape, whereas the other chamber compressed the bacteria, significantly deforming the cell shapes. Mannik *et. al* found that despite the stresses and deformations, the bacteria in the second chamber were able to divide just as successfully as those in the first chamber. In addition, they also found that the generational lifetimes were comparable between the two groups of bacteria [8]. Groups in which the Min protein system had been inhibited were not able to successfully divide under the conditions of deformation.

The importance of the operation of the Min protein system suggest that its behavior under these conditions of deformation may be key. A computational representation of the experimental results presented by Mannik *et. al* could be done by replicating the appropriate boundary conditions imposed by the chamber walls mathematically, and "tagging" the Min protein system as it moves around. A microscopy style view of what is going on inside the cell could then be obtained, providing insight into how the Min protein system responds to the cell deformation.

§ Section 3: Available models.

One of the first models to successfully describe the oscillations of the Min protein system was the continuum model, developed in 2003 by Huang *et. al.* The continuum model incorporates the relevant Min protein particle states in a cycle, and uses a set of reaction-diffusion equations to describe how each protein state transforms into one another and moves around in the cell. The time evolution of the system is then easy to record and analyze. With each protein state represented as a smooth number density function over the whole cell, normal calculus techniques can be applied, and the locations of extrema can be tracked in a straightforward manner.

Another model which accurately describes the Min protein system is the stochastic model. The stochastic model for the Min protein system describes each protein state as a discrete collection of particles, with an associated collection of random moves that can be made. Instead of the system as a whole evolving over time in a predictable manner, the stochastic model is constantly rolling the dice to change the state of the system. The stochastic model might have one protein bind to the wall randomly, and then another move over some amount of space, in an inherently probabilistic way. The stochastic model has been shown to be a slight improvement over the continuum model [5], except in the limit of a large number of proteins, where the models behave identically.

For this project, the continuum model was used. It is computationally less expensive, and easier to work with mathematically. The continuum model still produces good results, just not to the degree that the stochastic model does.

III. Methods.

In this chapter, the mathematical model that describes the motion of the Min protein system is introduced, and described in detail. The numerical method used to solve the equations is introduced, and the simulation parameters, such as cell shapes and diffusion constants, are shown. The data management system is explained, and some examples of the data management interface are shown. Finally, the critical simulation algorithms are listed and discussed.

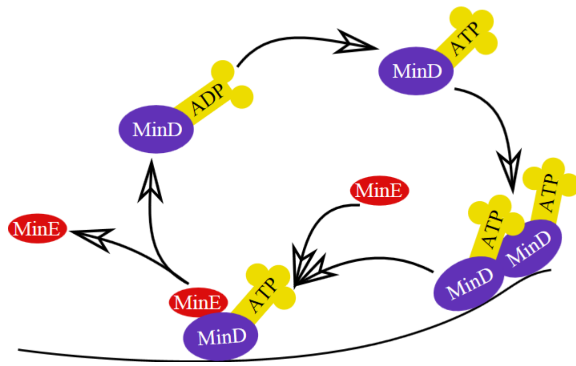

Figure 1: Reaction-diffusion cycle. We initialize our simulation in the state with loose MinE and loose MinD:ADP (see the section on initialization). [4].

§ Section 1: Computational model.

Figure 1 represents the reaction cycle of the model we are using for the Min protein system. The model involves five individual states for the relevant proteins: cytoplasmic MinE, cytoplasmic MinD bound to an ADP (adenosphine diphosphate) molecule, cytoplasmic MinD bound to an ATP (adenosphine triphosphate) molecule, MinD bound to an ATP molecule on the wall, and MinD bound to both an ATP molecule and a MinE protein on the wall. We choose to label these complexes MinE, MinD:ADP, MinD:ATP, MinD:ATP$_{mem}$, and MinD:E:ATP, respectively. These states are constantly interacting and converting into one another, according to a process described quantitatively in Equations 1 through 5 [4].

$$\frac{\partial \rho_{D:ADP}}{\partial t} = \mathcal{D}_D \nabla^2 \rho_{D:ADP} - k_D^{ADP\rightarrow ATP} \rho_{D:ADP} + \delta(d_w) k_{de} \sigma_{DE}, \tag{1}$$

$$\frac{\partial \rho_{D:ATP}}{\partial t} = \mathcal{D}_D \nabla^2 \rho_{D:ATP} + k_D^{ADP\rightarrow ATP} \rho_{D:ADP} - \delta(d_w)[k_D + k_{dD}(\sigma_D + \sigma_{DE})]\rho_{D:ATP}, \tag{2}$$

$$\frac{\partial \rho_E}{\partial t} = \mathcal{D}_E \nabla^2 \rho_E + \delta(d_w) k_{de} \sigma_{DE}, -\delta(d_w) k_E \sigma_D \rho_E \tag{3}$$

$$\frac{\partial \sigma_d}{\partial t} = -k_E \rho_d \rho_E + [k_D + k_{dD}(\sigma_D + \sigma_{DE})]\rho_{D:ATP}, \tag{4}$$

$$\frac{\partial \sigma_{DE}}{\partial t} = -k_{de} \sigma_{DE} + k_E \sigma_D \rho_E. \tag{5}$$

In this set of reaction-diffusion equations, ρ is the protein volume density in the cytoplasm, and σ is the protein surface density on the membrane. $\mathcal{D}_D$ is the diffusion rate constant for the MinD types, and $\mathcal{D}_E$ is the diffusion constant for the MinE protein types. $k_D^{\text{ADP} \to \text{ATP}}$ represents the conversion rate from the state of MinD bound to an ADP to MinD bound to an ATP. k_{dD}, k_D are the rates of attachment to the membrane for the MinD:ATP state when there is or is not a MinD:ATP already there, respectively. k_{de} is the rate of hydrolisis of the MinD:MinE:ATP complex, and k_E the rate of cytoplasmic MinE binding to membrane bound MinD:ATP complex. d_w is the distance to the nearest wall, and $\delta(d_w)$ is zero everywhere except at the cell wall. The values of these constants are listed below.

$$\begin{gathered}
\mathcal{D}_D = \mathcal{D}_E = 2.5\mu\text{m}^2/\text{sec}, \\
k_D^{\text{ADP} \to \text{ATP}} = 1/\text{sec},\ k_D = 0.025\mu\text{m}/\text{sec}, \\
k_{dD} = 0.0015\mu\text{m}^3/\text{sec},\ k_{de} = 0.7/\text{sec}, \\
k_E = 0.093\mu\text{m}^3/\text{sec}.
\end{gathered}$$

Initial conditions.

The first step in designing the simulation is choosing a set of initial conditions. Since the process is cyclic, the point in the simulation at which we begin is arbitrary. We chose to initialize our program in the state with loose MinD:ATP and MinE in the cytoplasm. To be able to compare our simulation with known results, we used the same initial number densities as did Huang *et. al.* In their one-dimensional simulation, they used an initial cytoplasmic concentration of 1,000 proteins per micron for the MinD:ATP state, and 350 proteins per micron for the MinE state, values which are not atypical for E. coli cells [6]. We converted this into a volume concentration (proteins per micron^3) for our three dimensional simulation by dividing by the cross sectional area of typical pill shaped bacteria (barrel width of .5 microns, length of 4 microns). That is, $\rho_{1D}/A = \rho_{3D}$. We obtained values of $1273\,\mu\text{m}^{-3}$ and $446\,\mu\text{m}^{-3}$ for the MinD:ATP concentrations and MinE concentrations, respectively. We conserved these two initial concentrations across different cell shapes and sizes.

To keep our simulation physically meaningful, we required that each cell shape have a total volume on the order of 1 to 3 μm^3. This is close to the average volume of normal E. coli cells. To verify that our volumes are correct computationally, we iterate through our grid, and add one discrete volume element (dx^3) to the total volume sum for each grid point that is considered to be inside the cell. This allowed us to quickly check whether a given cell shape was physically too large or too small to be worth simulating. The volume constraint, together with the density constraint described in the last paragraph, helped us make sure that our computer simulation was accurately modeling the physical situation.

Cell types.

Another important parameter to our simulation was the shape of the cell. The simulation program has a few different basic cell shapes implemented, with variable size parameters (e.g. length, width, radius, etc, where applicable) for each of them. The most basic cell shape we implemented was the pill. The pill shape features a cylindrical barrel, with hemisphere endcaps. The pill shape best approximates an actual E. coli cell membrane. In addition to the pill shape, the simulation program is capable of generating random flattened shapes, and shapes we create by hand. The arbitrary flat shape generation is done by moving and overlapping circles of different sizes around on the grid, then giving the resulting shape a thickness, and rounding the walls off. Some example shapes are presented in Figure 2.

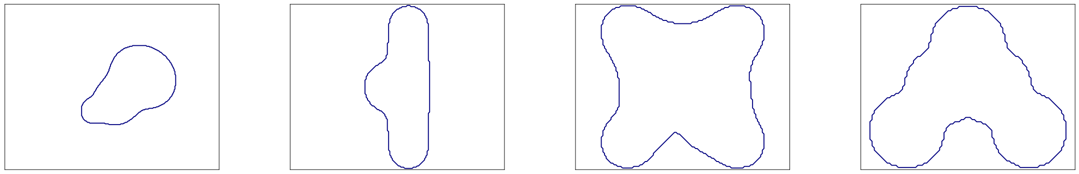

Figure 2: Sample cell shapes, "top down" view. The blue line represents the boundary of one cross section of the cell floating in three dimensional space. The cells are fairly flat, with some slight curvature in the z-direction (into and out of the page), so overall the cross section gives a good idea of what the full shape looks like.

§ Section 2: Data management.

One of the most significant challenges we faced in setting up the program infrastructure was managing the amount of data our program would be producing. Ultimately, our goal is to glean some insight into how the protein concentrations $\rho_i(\vec{r}, t)$ (or $\sigma(\vec{r}, t)$, for the membrane bound states) behave under different boundary conditions. The problem we face is that those are, mathematically, functions of four variables (x, y, z, t), we have one for each of the five different protein states, and we have five protein states for each simulation we run. We can sort of circumvent the problem of saving all this information by only recording specific details about any given simulation as it runs (e.g. extrema locations), but it became apparent that a broader solution was needed as we started running more simulations.

Our broader solution was to write a light weight data management program. We currently keep a list of simulation jobs that we're interested in in one file, which the program can read from and write to. It can then look for files associated with the simulation job that was run (if it had been run already), and report missing data, delete all the data for that simulation, re-run the simulation, or generate the plots for that simulation data. An example of the simulation jobs list file and some associated files it would be able to find and manage are presented in Figures 3 and 4.

```
pill 4.00 0.50 0.00 0.00 15.00 -paper
randst 0.25 8.00 6.00 97.00 15.00 0.00 10.00 -hires -paper
```

Figure 3: Two simulation jobs added to the jobs list file. The plots and data associated with these two jobs (defined by their simulation parameters) would be carefully tracked. This allows us to auto-delete any simulation data we are not interested in tracking.

```
nATP-p-4.00-0.50-0.00-0.00-15.00-101.dat
NDE-p-4.00-0.50-0.00-0.00-15.00-123.dat
nADP-pill-4.00-0.50-0.00-0.00-15.00-126.dat
... (hundreds more)

hires-arrow-plot-D_nADP-randst-0.25-8.00-6.00-97.00-15.00.dat
hires-arrow-plot-E_nE-randst-0.25-8.00-6.00-97.00-15.00.dat
hires-ave_plot--randst-0.25-8.00-6.00-97.00-15.00.dat
... (hundreds more)
```

Figure 4: Sample of files able to be managed by adding two lines to the jobs list file. Note the matching simulation parameters. If the simulation parameters do not match something that is in the jobs list, the simulation data would be "untracked" and is delete-able.

§ Section 3: Algorithms

List of functions:

```
mem_f() - sets the membrane function (shape) of cell
main() - main simulation loop
set_membrane() - wall calculation
set_curvature() - sets wall curvatures, useful for doing wall area calculation
find_intersection() - computes total wall area
set_insideArr() -  sets inside array status.
inside() - true/false check for if a grid point is inside the cell walls
get_J() - returns the next current density step
set_density() - initializes the protein density inside the cell walls

struct protein - data structure containing the protein information
```

Figure 5: Important functions used in the simulation code structure. An overview of the algorithms implemented in theses functions is detailed in the algorithms section.

The functions listed in Figure 5 make up the core of the protein simulation program.

mem_f()
mem_f() is a function that is run before the simulation begins. Its job is to define the cell shape which will be used over the course of the simulation. It creates a three-dimensional array (call it f_{ijk}) the same size as the space our cell is floating in. Each triple of indicies represents one point in the space the cell is sitting in. For example, $f_{5,3,2}$ would contain the value of f at $\vec{r} = 5\Delta x\hat{x} + 3\Delta x\hat{y} + 2\Delta x\hat{z}$. $f(\vec{r})$ is a pre-defined mathematical function, which is a positive distance for distances outside the cell walls, zero on the cell walls, and a negative distance inside the cell. This convention allows us to define and manipulate cell shapes. For example, if we wanted to define a bacteria cell as a sphere with radius 3 µm located at the origin, we would write $f(\vec{r})$ as

$$f(\vec{r}) = \sqrt{x^2 + y^2 + z^2} - 6\Delta x. \tag{6}$$

set_density()
set_density() is a function that is run before the simulation begins. It initializes the density concentrations for each protein state, across the inside of the cell. As mentioned earlier, we made sure to keep the average protein densities in line with those reported in Huang's paper. This was done to keep the simulation parameters physically reasonable. set_density() creates an array for each protein type which is the same size as the cell's space. It then iterates over the arrays, and calculates the appropriate protein concentration to place at each gridpoint. We often start the simulation with a very lopsided density, to kickstart the oscillations. This is commonly done by dividing the cell into two unequal partitions, filling one with a nonzero protein concentration, and the other with no protein concentration.

set_curvature(), find_intersection()
We will eventually want to calculate the area in each cell, since the mathematical model involves cells attaching to and leaving from the cell membrane. We implemented two new functions to carry out this calculation, set_curvature() and find_intersection(). set_curvature() will look for regions on the cell walls that contain non-zero curvature, and calculate the area of these regions. find_intersection() will calculate the cell wall area for each gridpoint that is in a flat region of the cell wall, by calculating the area of the plane which is cutting through each Δx^3 cube.

get_J()
Calculates the current density for each protein state. The numerical method we use to solve involves tracking the proteins as they move around in the cell. First, we calculate the current density of the protein concentration:

$$J_x(\vec{r}) = -\mathcal{D}_D \frac{dn}{dx} \tag{7}$$

$$J_y(\vec{r}) = -\mathcal{D}_D \frac{dn}{dy} \tag{8}$$

$$J_z(\vec{r}) = -\mathcal{D}_D \frac{dn}{dz} \tag{9}$$

We then make the substitution:

$$\frac{dn_x}{dt} = \vec{J} \cdot \hat{x} \frac{dA}{dV} \tag{10}$$

$$\Delta n_x = \vec{J} \cdot \hat{x} \frac{dA}{dV} \Delta t \tag{11}$$

Using that dn_x, which represents the amount of protein concentration that moves over one cell in the x direction, we can update the protein concentration positions, which allows us to calculate where the proteins move over time, allowing us to model the system.

main()
main() is the main simulation loop for the program. It initializes all the necessary arrays, performs setup calculations (such as running mem_f(), set_curvature(), etc), and then lets the simulation run. It records data both as the simulation is running, and records information on the final state of the cell as well. When main() has finished running, the program exits and the simulation is over.

set_membrane(), set_insideArr()
set_membrane() and set_insideArr() are functions for pre-setting information which will be called over and over again during the simulation. set_membrane() records the results of the area calculations at each gridpoint, and set_insideArr() is a boolean array which will tell you whether any given gridpoint is inside or outside the cell. It is useful to cache this information before the simulation is run, so the calculations do not have to happen over and over again.

IV. Results.

We found that the continuum model did not generally produce results comparable to the experimental findings of Mannik *et. al.* in many of the perturbed cell shapes. As an example, Figure 6 presents one case study of a perturbed cell which we replicated from Mannik's experiment. The initial densities, cell boundaries, and volumes were obtained from experimental reports.

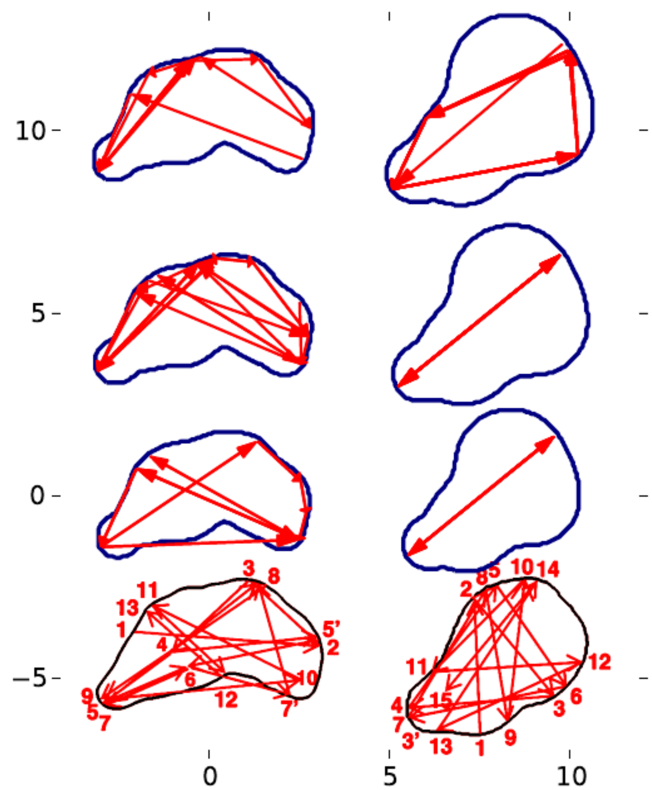

Figure 6: Locations of protein maxima over the course of one cell simulation. Top three rows represent our simulations, using the continuum model. The bottom row contains the experimental findings of Mannik *et. al.*

At the bottom of Figure 6 are two cross sections of the bacteria that were deformed by the channels experimentally. Above those are our replica membranes (the computer models), at different size scales. The smallest ones are just above the experimental figures, getting larger toward the top of the figure. It is in a direct comparison such as this that the comparison between the continuum model and the experimental results is most distinct.

In the left column of Figure 6, we see experimental results that exhibit what seem to be three primary poles of the cell, with some transient behavior mixed in. In our computer simulation, we never quite see this behavior. The smallest scale replica, directly above the experimental membrane, the extrema seem to travel mostly along the cell wall, rather than directly through the middle. This behavior continues as the cell increases in size, shown in the middle row, although three distinct poles do appear to develop. At the largest scale shown however, the three poles behavior is again nonexistent. The "membrane travelling" behavior present in the simulation, but not the model, is also problematic.

In the right column of Figure 6, the strongest feature of the experimental cell shown appears to be the "billiards" style motion of extrema, clearly seen by following the numbering indicated on the outside of the cell. The smallest simulation result however, shows clear pole to pole oscillator behavior, which is very regular and not at all consistent with the behavior seen experimentally. This continues in the medium sized cell simulation, with absolutely no extrema motion around other parts of the cell. In the largest scale cell, this is slightly different, as we do have some extrema located on the sides of the cell, but looking at the data directly, it is clear that those were more transient extrema than the two dominant pole extrema:

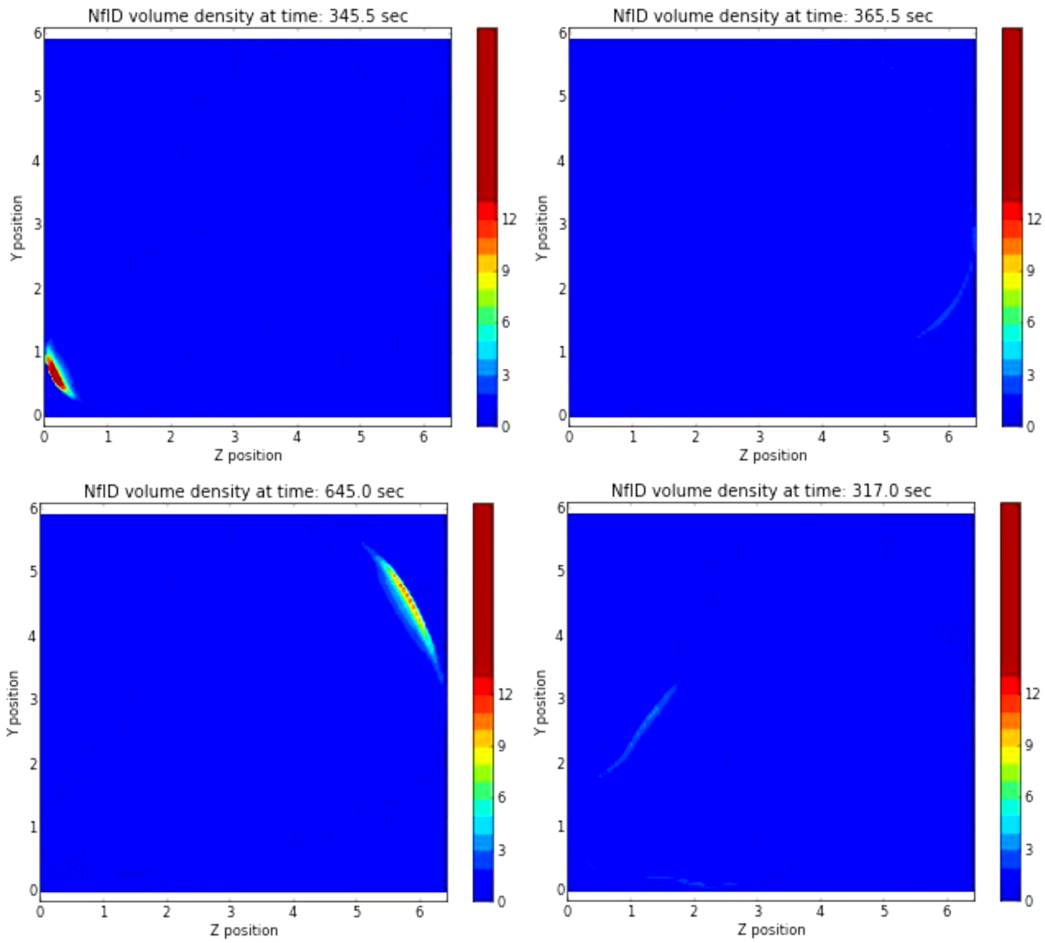

Figure 7: Left column: real maxima in protein number density over the course of the simulation. Right column: misleading "maxima" that appeared in the extrema plot of Figure 6, top right.

This direct comparison showed us that the continuum model would not be sufficient to analyze the behavior of the Min system directly under the significant cell perturbations we were considering. Our implementation of the model was reasonable at reproducing results for regular shaped bacteria.

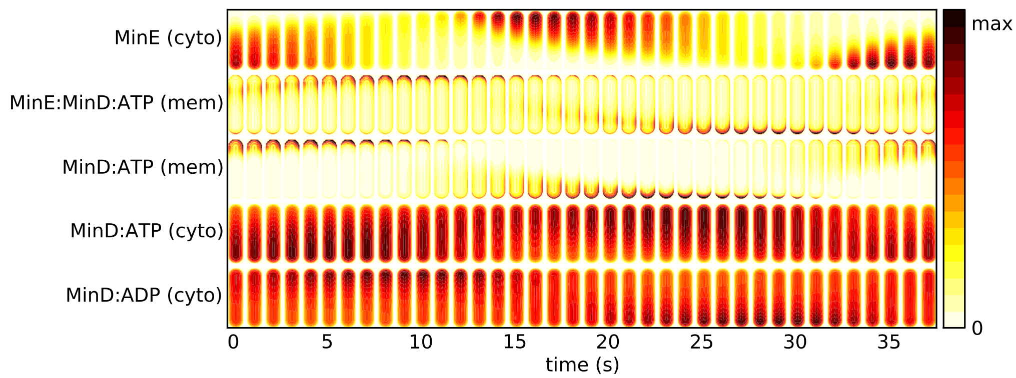

Figure 8: Cross section of a pill shaped cell, with real time densities plotted along the horizontal axis. The period of each oscillation is on the order of 40 seconds.

In Figure 8, a frame-by-frame cross section of each of the five protein states is presented. From this information, the typical distribution of each kind of protein can be seen, and compared to the results from Huang *et. al.*, who first came up with this model. Huang's work was done in a purely cylindrical system, rather than a more realistic pill-shaped bacterium, but the behavior is similar enough.

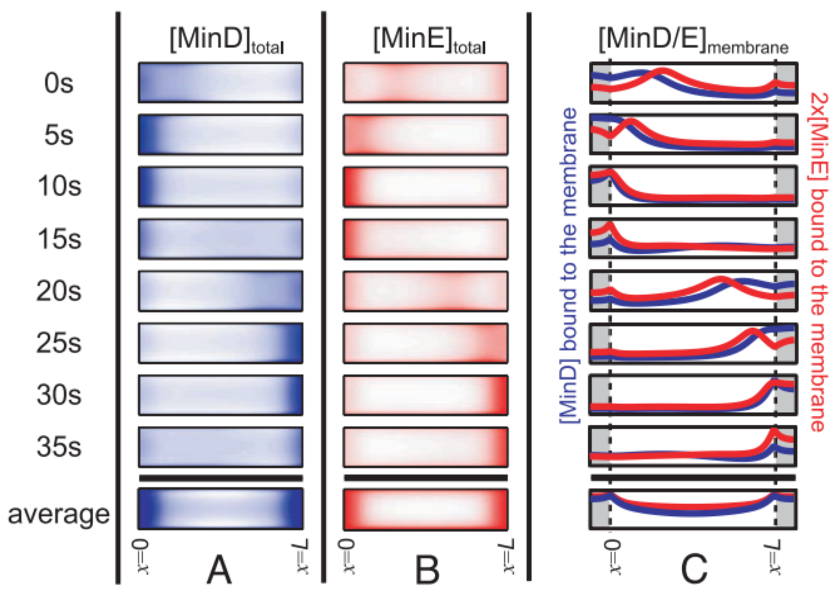

Figure 9: Periodicity of protein oscillations from Huang *et. al*, 2003. [4]

From Huang's figure [4], we can also ascertain the period of the oscillation, which is on the order of 40 seconds. We obtain a similar period of oscillation from the same cell, shown in Figure 10. This period is a direct measurement from our simulation data, but has to do with the constants defined in the methods section, such as the reaction rates, diffusion rates, and so on. Those constants determine the overall speed of the cycle.

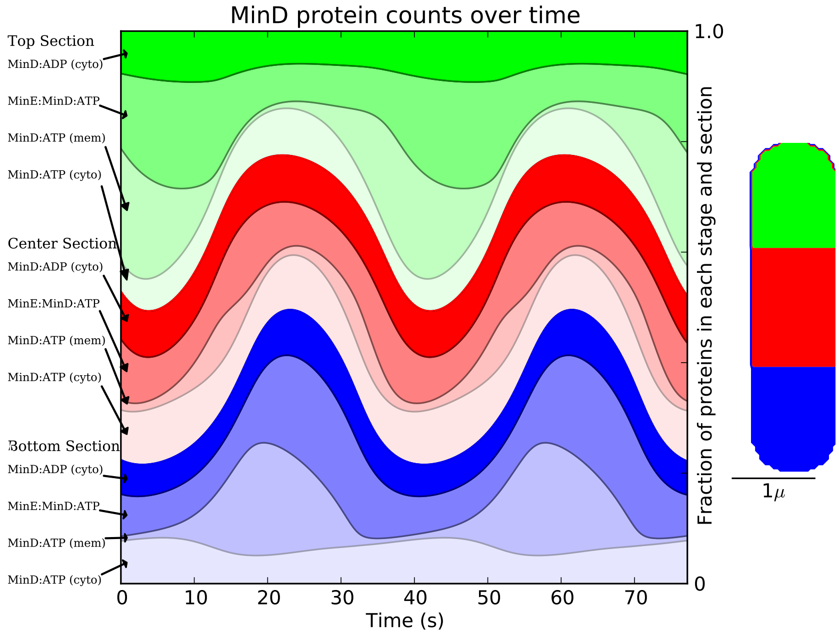

Figure 10: Oscillatory behavior in the time domain. The thickness of each shaded region represents the number of that type of protein in a particular section of the cell, at one instant of time.

V. Discussion.

§ Section 1: Length scales.

One thing we learned from simulating the continuum model under these conditions is that the physical scale of the cell membrane is meaningful in determining the types of oscillations that will occur. When originally performing the cell simulations, we found a wide variety in the types and frequencies of oscillations that occured in different shaped cells with similar initial conditions. We hypothesized that it was not just the shape of the cell that would determine these features of the oscillations, but the general length scale of the cell. A stability analysis performed by Jeff Schulte determined a critical length scale criterion, supporting this idea.

As the cell size decreases in some direction, the oscillations will relax into a motionless steady state. That is, when a cell membrane was physically smaller than this size parameter along one direction, we did not find oscillatory behavior along that axis. Instead, the proteins would oscillate along an orthogonal axis, if the cell was long enough in that direction. This was clearly seen in most of the pill-shaped cells, as they characteristically have a long axis and a short axis. In most of the elongated pill shapes, we saw a distinct pole-to-pole oscillation behavior, which is consistent with the length scale idea. Shorter, more spherical pill shapes did not have the same robust oscillatory behavior as the elongated ones.

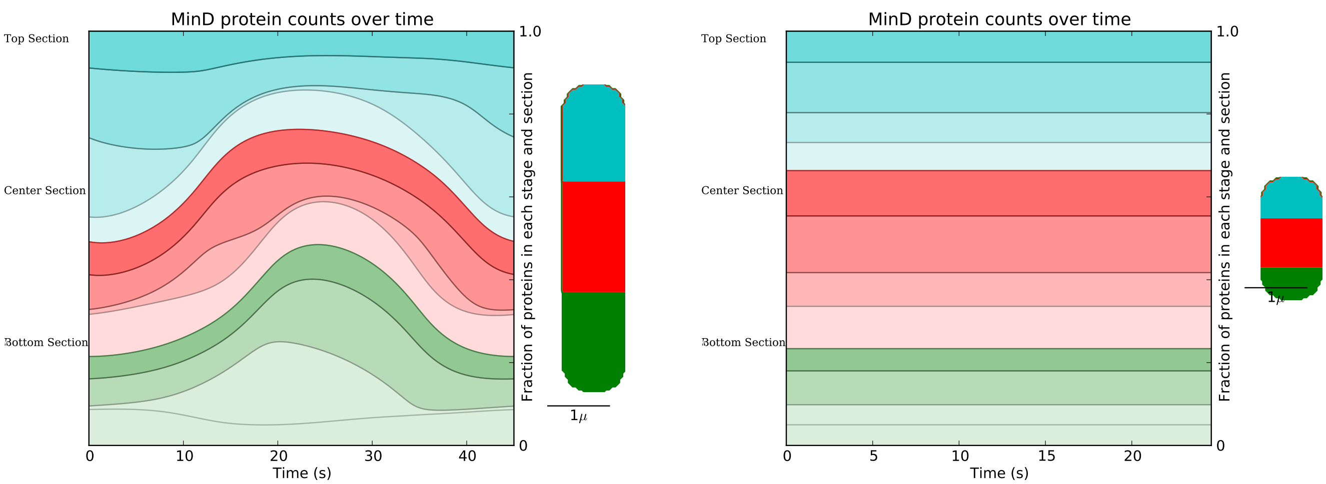

Figure 11: Left: elongated pill shape, showing distinct oscillatory behavior. Right: shorter pill shape, showing no oscillatory behavior, due to a lack of a "long axis."

In Figure 11, on the left side, we have a box plot for the long and skinny pill. Distinct spatial oscillations are visible, as the cell is much larger than the characteristic size parameter along its long axis. On the right side of Figure 11, we have the shorter, fat pill shape, which has smaller oscillatory behavior, as both axes are too short for oscillatory behavior to occur, consistent with the size parameter hypothesis. We saw that this behavior extended generally to other types of cell shapes we tested, but it was most clearly present in the pill shapes.

When a particular cell approached the characteristic size parameter along one axis, we saw oscillatory behavior begin to occur, despite being significantly damped. The damping was severe enough for the oscillatory behavior to stop after only a few cycles. After those cycles, the protein systems remained in the steady state solution, with no spatial oscillations occuring. This information is presented graphically in Figure 12.

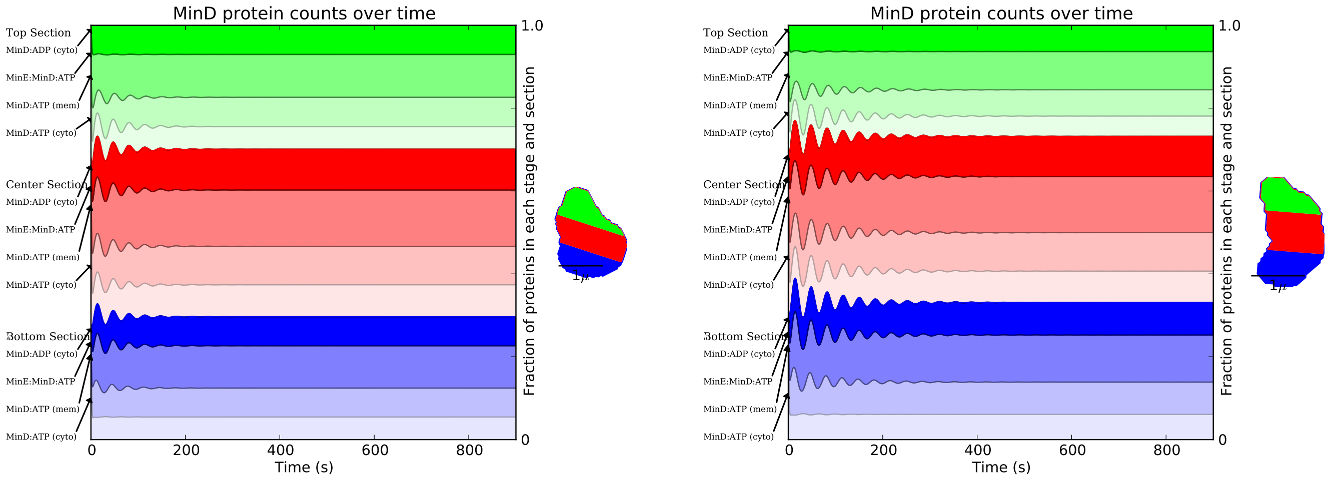

Figure 12: Slightly damped oscillations shown in cells which are on a comparable length scale to the size parameter.

Two different cell shapes are shown in Figure 12. On the left, a cell which is short, but still above the characteristic size parameter, is presented. On the right, a cell which is slightly longer along the relevant oscillatory axis is presented. As can be seen, increasing the length above this characteristic size parameter decreases the decay time of the oscillations. This can be most clearly seen in cells that are around the length scale of the size parameter. In cells that are significantly longer than their characteristic size parameters, no decay types of behavior are seen in the oscillations.

The next interested regime of cell sizes are those which are significantly longer on one axis than the cell's characteristic size parameter. Shown in Figure 13 is a progression of cell shapes of the same type, that are getting larger in scale (in order of top left, bottom left, top right, bottom right). A scale is presented next to the cell membrane outline.

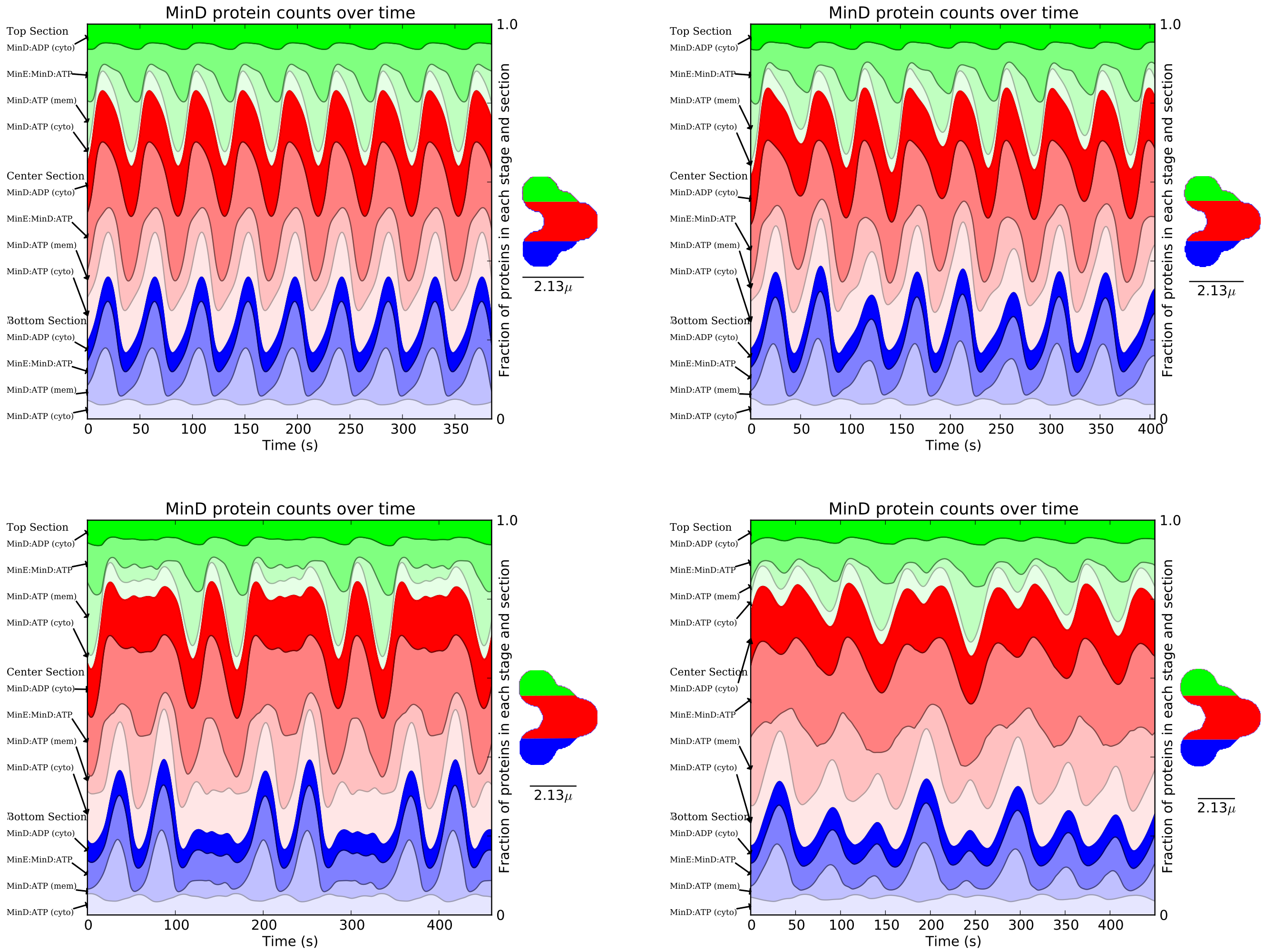

Figure 13: The number of frequency components of the oscillatory behavior increases with cell size.

The top left box plot shows a typical cell, which is larger than its size parameter, so that there is no decay, but not significantly larger. Normal, almost sinusoidal oscillations occur along the relevant axis of the cell. In the next larger cell (bottom left), we see a slight deviation from the regular oscillatory behavior of the smallest cell. It appears as though another frequency oscillation is present, leading to the double peaks. This trend continues into the second largest cell, with yet another frequency oscillation distorting the sinusoid present in the smallest cell. It appears that larger cell sizes, those much larger than their characteristic size parameter, begin to accumulate multiple oscillations of the Min system.

While this may not exactly model experimentally perturbed E. Coli cells, it is nonetheless an interesting extension of the computational model developed by Huang *et. al.*, showing oscillatory behavior which was not seen at the typical length scales of unperturbed E. Coli bacteria.

§ Section 2: Alternative models.

We believe that perhaps the best alternative model to try in the future would be the stochastic model [5], described in the introduction. The stochastic model takes into account the fact that these reaction/diffusion actions do not actually happen on a large scale continuum like the continuum model's differential equations describe, but rather in a more random, protein-by-protein basis. The larger, cell-scale phenomena that was observed experimentally may be accounted for in this statistical approach, instead of being overlooked in the continuum model.

The stochastic model essentially considers the probabilities of each protein inside the cell to take a certain action. These event probabilities for each protein are interrelated. For example, a Min D protein located near the cell wall might have a higher probability of binding to that part of the cell wall than a Min D protein further from the wall. Or as another example, if there are high concentrations of Min E near the cell membrane, the probability of a cytoplasmoic Min D to bind to a Min E might be reduced. After each event occurs, the stochastic model should update the probabilities of each other event accordingly.

There are significant computational challenges with implementing such a model on our scale. The Min protein cycle, described at the beginning of the methods section, has many possible actions that one individual protein could take. There are also many proteins, and the goal of recalculating every probability in the cell after a single event has occured would simply be computationally unfeasible. There are some computational approaches that can be used to reduce the number of calculations needed, however. One such approach is to take into account that the probabilities only need to change locally; a protein event occuring on one end of the cell does not affect the proteins on the other side of the cell too much. Given that our model is a full three dimensional simulation, such approximations will be necessary in the future implementation of this model.

With these precautions in mind, the next step for the stochastic model would be to roll the dice, and calculate what events occured during a single time step of the simulation. The stochastic model would then update the state of the cell, with its new probabilities, and another time step would occur. This model has been implemented by other groups in some simulations, and they have seen similar or better results than with the continuum model [5]. We think that the stochastic model may be able to better replicate some of the results of Mannik *et. al.*, and future work with this project will be exploring that avenue.

VI. Conclusion.

§ Section 1: A review of the Min system.

The Min protein system serves as an indicator to the rest of the cell where the cell center is. During cell division, a long FtsZ polymer chain developes, which dictates the plane of cell division. MinC proteins, which move in conjunction with MinD, inhibit the development of FtsZ polymers, ensuring that the polymer chain builds in the right location. Where there is MinD, there will not be FtsZ polymers. If the FtsZ polymer chain builds in the wrong location, the cell division process will be disturbed, and viable daughter cells will not be produced. Thus, we are interested in knowing the spatial behavior of the Min protein system during the cell division process.

In particular, we are interested in knowing the behavior of the Min protein system when the cell is deformed significantly. Experimental studies have shown that the Min protein system begins to oscillate when the cell shape is changed, but these oscillations are important for the cell to still divide properly. Studying the oscillatory behavior of the Min protein system under new cell shapes is therefore important if we want to better understand the process of cell division in E. coli. We looked at types of oscillations, and the criteria for oscillations to occur, for cell shapes that are perturbed significantly from the standard pill shape.

To model the Min protein system computationally, we implemented a mathematical model developed by Huang *et. al.* This model consists of five reaction-diffusion equations, the solutions to which are the volume densities for each protein state described in the cycle presented in the methods section. These equations describe the ways each protein state diffuses across the cell, and reacts to turn into another protein state. Applying some experimentally acquired intial conditions, we numerically solve these equations, and gain five protein densities. We then plot and analyze the time evolution of the densities over the cell. This method is referred to as the continuum model.

§ Section 2: What we learned.

The first result we found was that the continuum model was not able to replicate the experimental results of Mannik et al. for perturbed cell shapes. This was despite having successful replication of the results of Huang *et. al.*, who developed the continuum model. Our simulations showed that the oscillations in the continuum model were too regular and robust when compared to the experimental data. We saw many pole-to-pole style oscillations in numerous cell shapes, when we should have seen a more random, or chaotic behavior. We believe that this may be due to the shortcomings of the continuum model, which do not directly account for the random, microscopic behavior of proteins inside the cell.

Despite this, we did find some interesting extensions of the continuum model, which had not been previously expected. A stability analysis showed that there exists a characteristic length parameter for each cell shape. This length parameter acts as a sort of minimum length at which the cell will undergo oscillations. As shown in the results section figures, cells which are about as long as this length parameter will exhibit damped oscillations. Below the length parameter, the cell will not have oscillations at all. Above the length parameter, we see the decay time increase, until the oscillations are undampened. This length parameter places a lower limit on the size of E. coli perturbations.

Future work for this project will involve developing an alternative model for use in simulation of the Min protein system. The stochastic model, described in the discussion section, is a more realistic, albeit computationally challenging model for the Min protein system. It takes into account the microscopic behavior of each individual protein, rather than treating the whole cell as a continuous density of proteins. This might prove to be a more correct description of the Min protein system, when attempting to model the protein system's behavior in a perturbed cell shape. More comparisons will be made between experimental data and the simulation results.

Bibliography

[1] David W Adams and Jeff Errington. Bacterial cell division: assembly, maintenance and disassembly of the z ring. *Nature Reviews Microbiology*, 7(9):642–653, 2009.

[2] Xiaoli Fu, Yu-Ling Shih, Yan Zhang, and Lawrence I Rothfield. The mine ring required for proper placement of the division site is a mobile structure that changes its cellular location during the escherichia coli division cycle. *Proceedings of the National Academy of Sciences*, 98(3):980–985, 2001.

[3] Zonglin Hu and Joe Lutkenhaus. Topological regulation of cell division in escherichia coli involves rapid pole to pole oscillation of the division inhibitor minc under the control of mind and mine. *Molecular microbiology*, 34(1):82–90, 1999.

[4] Kerwyn Casey Huang, Yigal Meir, and Ned S Wingreen. Dynamic structures in escherichia coli: spontaneous formation of mine rings and mind polar zones. *Proceedings of the National Academy of Sciences*, 100(22):12724–12728, 2003.

[5] Rex A Kerr, Herbert Levine, Terrence J Sejnowski, and Wouter-Jan Rappel. Division accuracy in a stochastic model of min oscillations in escherichia coli. *Proceedings of the National Academy of Sciences of the United States of America*, 103(2):347–352, 2006.

[6] HE Kubitschek. Cell volume increase in escherichia coli after shifts to richer media. *Journal of bacteriology*, 172(1):94–101, 1990.

[7] Joe Lutkenhaus. Assembly dynamics of the bacterial mincde system and spatial regulation of the z ring. *Annu. Rev. Biochem.*, 76:539–562, 2007.

[8] Jaan Männik, Fabai Wu, Felix JH Hol, Paola Bisicchia, David J Sherratt, Juan E Keymer, and Cees Dekker. Robustness and accuracy of cell division in escherichia coli in diverse cell shapes. *Proceedings of the National Academy of Sciences*, 109(18):6957–6962, 2012.

[9] David M Raskin and Piet AJ de Boer. Rapid pole-to-pole oscillation of a protein required for directing division to the middle of escherichia coli. *Proceedings of the National Academy of Sciences*, 96(9):4971–4976, 1999.

[10] Lucy Shapiro, Harley H McAdams, and Richard Losick. Why and how bacteria localize proteins. *Science*, 326(5957):1225–1228, 2009.

[11] Ahmed Touhami, Manfred Jericho, and Andrew D Rutenberg. Temperature dependence of mind oscillation in escherichia coli: running hot and fast. *Journal of bacteriology*, 188(21):7661–7667, 2006.

[12] Xuan-Chuan Yu and William Margolin. Ftsz ring clusters in min and partition mutants: role of both the min system and the nucleoid in regulating ftsz ring localization. *Molecular microbiology*, 32(2):315–326, 1999.