Characterization of an experimental system designed for pulsing *dictyostelium* with chemoattractant using a homemade flow chamber

Physics Department Senior Honors Thesis
Michael C. DeSantis
Brandeis University

May 2007
Abstract

Dictyostelium is a social amoeba capable of chemotaxis, directed cellular movement towards an increasing concentration of chemoattractant. The present view of this organism’s biological system with respect to its directional sensing ability is limited; accordingly, the purpose of this research is to further our understanding of this mechanism by studying cell responses to repeated chemical signals of varying frequency and duration. To ‘pulse’ these cells, an experimental system comprising a homemade flow chamber and actuator was assembled and characterized with fluorescein for different flow rates. Using a modified strain expressing the green fluorescent protein (GFP) bound to CRAC, a molecule involved in chemotaxis, we are able to quantitatively measure the response by visualizing this CRAC-GFP fusion construct which preferentially accumulates at the cell’s leading edge and subsequently extracting the intensities with image analysis codes I have written.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>1</td>
</tr>
<tr>
<td>List of Figures</td>
<td>3</td>
</tr>
<tr>
<td><strong>Introduction</strong></td>
<td>5</td>
</tr>
<tr>
<td>The Chemotaxis Network</td>
<td>6</td>
</tr>
<tr>
<td><strong>Computational Models</strong></td>
<td>8</td>
</tr>
<tr>
<td>Perfect Adaptation</td>
<td>8</td>
</tr>
<tr>
<td>Traveling-Wave Chemotaxis</td>
<td>13</td>
</tr>
<tr>
<td><strong>Methods and Materials</strong></td>
<td>17</td>
</tr>
<tr>
<td>Experimental Setup</td>
<td>20</td>
</tr>
<tr>
<td>Characterization of the System</td>
<td>24</td>
</tr>
<tr>
<td><strong>Results</strong></td>
<td>32</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>32</td>
</tr>
<tr>
<td>Preliminary Data</td>
<td>40</td>
</tr>
<tr>
<td><strong>Discussion</strong></td>
<td>44</td>
</tr>
<tr>
<td><strong>Conclusion</strong></td>
<td>46</td>
</tr>
<tr>
<td><strong>Acknowledgements</strong></td>
<td>47</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>48</td>
</tr>
<tr>
<td><strong>A Laboratory Procedures</strong></td>
<td>50</td>
</tr>
<tr>
<td>A.1 <em>Dictyostelium</em> Growth and Media</td>
<td>50</td>
</tr>
<tr>
<td>A.2 <em>Dictyostelium</em> Starvation</td>
<td>53</td>
</tr>
<tr>
<td><strong>B The Models</strong></td>
<td>55</td>
</tr>
<tr>
<td>B.1 Perfect Adaptation Model</td>
<td>55</td>
</tr>
<tr>
<td>B.2 Traveling-Wave Chemotaxis Model</td>
<td>58</td>
</tr>
</tbody>
</table>
## C LabVIEW Code

## D MATLAB Codes

- D.1 Perfect Adaptation .................................................. 63
- D.2 Traveling-Wave Chemotaxis ........................................ 67
- D.3 Diffusion and Fitting .................................................. 69
- D.4 Image Processing ....................................................... 71
# List of Figures

1. Lifecycle of *dictyostelium* .................................................. 6
2. Chemotaxis Network ........................................................... 7
3. CRAC-GFP response to cAMP ............................................... 8
4. Perfect Adaptation: Normalized responses .............................. 10
5. Perfect Adaptation: Manual fit ............................................. 11
6. Perfect Adaptation: Predicted responses to experimental pulses 12
7. Chemotaxis: Responses at low S .......................................... 15
8. Growing Cells ................................................................. 17
9. Laboratory setup for starvation ............................................ 18
10. Peristaltic Pump (Top View) ............................................... 19
11. LabVIEW GUI to control devices ........................................ 19
12. Syringe Pump ................................................................. 20
13. Fluorescence Microscope Setup .......................................... 21
14. Cell Chamber ................................................................. 22
15. Connecting the devices ...................................................... 23
16. Controlling the actuator with attached circuit schematic .......... 23
17. Velocity profiles of the cell chamber ................................... 24
18. Average intensity as a function of pulse length ........................ 26
19. Average intensity as a function of flow rate ........................... 28
20. Manual fits to the experimental data ..................................... 30
21. Reproducibility of pulses at constant flow rate ....................... 30
22. Fluorescein Maps ............................................................ 31
23. AVI movie used for MATLAB image analysis ........................... 33
24. MATLAB: Image analysis GUI .............................................. 34
25. MATLAB: Defining boundaries by grayscale thresholding .......... 35
26. MATLAB: bboundaries command definitions ............................ 35
27. MATLAB: Illustrating the image analysis code ......................... 37
28. MATLAB: Defining boundaries by edge detection ..................... 38
29. MATLAB: Standard plot of CRAC-GFP intensities ....................... 38
30. MATLAB: Plots of CRAC-GFP intensities for a single image ........ 39


Introduction

*Dictyostelium discoideum* is considered a model organism for vast areas of research in the field of biophysics. This social amoeba displays characteristics akin to other widely studied cells including the ability to undergo chemotaxis, directed cellular movement towards an increasing concentration of chemoattractant; this response, which is shared by human leukocytes, is critical for the proper regulation of the immunological system involving wound healing, cellular differentiation and angiogenesis among other features. Moreover, scientists continue to elucidate the underlying dynamics of *dictyostelium*’s molecular network, comprising the biological pathways responsible for cell locomotion, permitting the utilization of advanced techniques such as fluorescence microscopy. Using a modified strain expressing the green fluorescent protein (GFP) bound to CRAC, a molecule directly involved in chemotaxis, we are able to quantitatively measure the chemotactic response of the cells by visualizing this CRAC-GFP fusion construct which preferentially accumulates at their leading edge. When *dictyostelium* is exposed to a gradient of the chemoattractant, cyclic adenosine monophosphate (cAMP), the cell exhibits a polarized response exemplified by the anisotropic distribution of CRAC-GFP at the membrane, indicative of chemotaxis and which may also dictate the velocity of its subsequent motion according to theoretical models. Upon removal, the cell naturally returns to its pre-stimulus state with no apparent asymmetries, a process termed “perfect adaptation.” Similar observations for the rise and decay of CRAC-GFP along the membrane have been made by simply ‘pulsing’ the cells with a known concentration of cAMP in which the proteins become uniformly distributed instead, resulting in no directed motion.

The present view of the organism’s biological system with respect to its directional sensing ability is limited; accordingly, the goal of the experiment I am conducting is to further our understanding of this mechanism by studying *dictyostelium*’s response to repeated chemical signals of varying frequency and duration and subsequently observing the localization of CRAC-GFP over a series of timed intervals. In essence, we are viewing the cell as a 'black box' whereby the nature of our input signal considering the frequency and duration of the pulse as well as the concentration of cAMP will yield a measurable output response. To study the cells in this manipulative environment, fabrication of an experimental cell chamber optimized for high-resolution imaging was required to be used in conjunction with a microfluidic pulse generator we have designed and built to accurately dispense these chemicals to starved cells. While I have obtained preliminary data that illustrates the desired response, a large amount of time was required in order to properly evaluate the conditions necessary for
conducting the experiment accurately and characterizing the system including its individual components all of which will be explored in detail. Additionally, I have written a number of codes, particularly in MATLAB (D), to analyze the recorded images, run the experiment, as well as look at various aspects of the experimental setup and of *dictyostelium*’s predicted behavioral response to chemoattractant.

**The Chemotaxis Network**

It has been observed that starved cells periodically secrete cyclic AMP as a means of cell to cell signaling. These cells respond by migrating in the direction of the source and collectively form aggregates and inevitably stalks for sporulation as depicted by the time-lapsed image below:

![Lifecycle of *dictyostelium*. Image from Copyright M.J. Grimson and R.L. Blanton, Biological Sciences Electron Microscopy Laboratory, Texas Tech University.](image)

Figure 1: Lifecycle of *dictyostelium*. Image from Copyright M.J. Grimson and R.L. Blanton, Biological Sciences Electron Microscopy Laboratory, Texas Tech University.

In *dictyostelium*, the principal receptor for the cAMP molecule is cAR1 which is linked to a seven member G-protein uniformly distributed throughout the cell membrane. Binding initiates the transfer of a phosphate group to GDP forming GTP which is further accompanied by the dissociation of the G-protein subunits thereby promoting the activation of other upstream regulators. These regulators, such as PI₃K and PTEN, are responsible for the forward and reverse conversions of PIP₂ to PIP₃ respectively as seen in figure 2.

PH-domain-containing proteins including the cytosolic regulator of Adenylyl cyclase (CRAC) are free to diffuse from the cytosol to the membrane in order to bind PIP₃. This advances the production of cAMP by membrane-bound Adenylyl cyclase which can be secreted
as an external signal to neighboring cells or is likely to be used for other cellular processes.

One of the chief uses of cAMP by the cell is for chemotaxis; however, it was our intention to use, as with previously cited experiments, cells that have been specially treated with Latrunculin-A thereby inhibiting their ability to polymerize actin necessary for locomotion. Because translocalization of CRAC from the cytosol is upstream of cellular movements, all events following the binding of this protein to PIP$_3$ is independent and therefore not likely to alter our methods for determining chemotaxis events. This statement is supported by the assertion that *dictyostelium* does not employ a temporal gradient sensing model unlike E. coli. A purely temporal model implies cells extend pseudopodia as pilot sensors in all directions to measure the relative concentrations of cAMP gradients but the data from experiments with nonmotile cells exhibiting polarized membranes in the direction of increasing cAMP sources contradict the veracity of this proposed mechanism. Other models such as a spatial-temporal one necessitates timely increases in receptor occupancy to go along with inhibition at the less stimulated side while a highly preferred purely spatial model requires that the cell be able to detect and differentiate between signals at opposing ends of the cell [2]. Since *dictyostelium* maintains an even distribution of receptors and G-proteins around the cell’s edge, an argument for the basis and existence of a spatial model can be made as can the suggestion that the cell should be able to polarize rapidly when sufficiently stimulated. Nonetheless, another mechanism, specifically amplification of the chemical signal, must be involved to ensure the cell can distinguish an external signal from the background. There is currently an ongoing debate as to the degree of effectiveness the cell can adequately discriminate between low levels of cAMP and surrounding noise while still polarizing correctly.
Computational Models

Perfect Adaptation

Addressing the cell’s behavior, there is overwhelming evidence as illustrated by the diagrams below 3, that a cell will always exhibit an equal but transient response to a uniform increase in the concentration of cAMP surrounding the cell whereas an increasing chemical gradient will induce a polarized distribution of CRAC-GFP proteins; in both situations the cell perfectly adapts by gradually acclimating to the current level of [cAMP] and consequently reverts to its resting or basal state between 20 - 30 seconds. Additionally, the cell in figure 3(b) displays the biphasic response at 120s which is marked by the re-accumulation of CRAC to the leading edge but with greater and lesser intensities at the front and back respectively:

![Diagram](image)

(a) Uniform cAMP field  
(b) cAMP gradient

Figure 3: CRAC-GFP response to cAMP. Image taken from [1].

To describe the features of chemotaxis, we have attributed these responses to be the result of a Local Excitation - Global Inhibition (LEGI) model. As I had previously mentioned, receptors are abundant and evenly spread along the membrane yielding the possibility for an immediate peak in the build up of CRAC-GFP anywhere between 5 - 10 seconds following stimulation with cAMP. To curtail the rise in the amount of PIP$_3$ by means of negative feedback machinery, a slowly diffusing cytosolic inhibitor, currently believed to be PTEN, a G$_\beta$ protein linked molecule or something yet to be defined, will suppress the response more quickly at the back. While there are numerous variations of the LEGI model (Devreotes group) which has been well developed to account for many of the cell’s behavioral aspects, one of the simplest being the “perfect adaptation” model I plan to describe shortly, there are several others which are championed by assorted dicty groups: a temporal sensing model involving the front sending an inhibitory signal to the back of the cell within a discrete time thereby preventing the subsequent response when in the presence of a cAMP gradient (Rappel); a combination of local positive (feedback) and long-range inhibitory/negative feedback...
signals generating the asymmetric response (Meinhardt, Postma and Van Haastert); and a mathematical “switch-like” model addressing LEGI’s sole amplification step which does not decisively account for the rapid CRAC localization to the front and nearly instantaneous inhibition at the back (Levine) [1] [3]. Despite the descriptive nature of these models, they do not fully incorporate the network’s molecular components. However, with recent advances in experimental techniques, it becomes possible to address quantitatively certain physical constraints and predictions of these models.

Identification of key proteins involved in these signaling processes demonstrate the capacity to view the system as a series of chemical reactions dictated by rate kinetics. Quantitative measurements of the rise and decay of various species in response to a native or otherwise external concentration of signaling molecule allows us to assign chemical rate constants leading to the establishment of equilibrium steady states. Using the perfect adaptation model defined by Levchenko et al. [4], I will show that I have achieved similar results reported by the authors but will also extend the scope of the model to simulate the parameters of my own experiment, consequently making quantitative predictions by comparison with physical data of our own.

Once again, perfect adaptation describes the cell’s tendency to return to its basal uniform distribution following rapid accumulation of CRAC-GFP to the membrane upon cAMP stimulation; other organisms such as human neutrophils experience ’spontaneous polarization’ where there is a uniform increase in the concentration of membrane CRAC-GFP but which later localizes to a random location after approximately ten minutes. To simulate the dynamics involved in dictyostelium’s observed response, we can consider an external signal, $S$, or signaling concentration of molecules, [cAMP], to promote the rise in an activator species, $A$, which converts a “response element $R$ into the activated form $R^*$. “ To account for adaptation, an inactivator, $I$, must be allowed to dictate the likelihood of the reverse reaction. “$S$ activates both $A$ and $I$ in fixed proportion [4].” With these components, I can apply the formulism of Michaelis-Menten kinetics which can be used to demonstrate the rate-driven interconversions of molecular enzymes with substrates to form an $E-S$ complex and further encouraged to create a product or activated form. See appendix B.1.

The associated steady state solutions for equations B.3 B.5a B.5b I calculate to be, and which are consistent with [5] are,

$$A_{ss} = \frac{k_A}{k_{-A}} S$$

$$I_{ss} = \frac{k_I}{k_{-I}} S$$

$$R_{ss} = \frac{k_R A_{ss}}{k_R A_{ss} + k_{-R} I_{ss}}$$

Choosing to express this set of equations in dimensionless quantities such that $r =$
With concentrations and reactions gives,

\[ a = \left( \frac{k_R}{k_A} \right) A \]
\[ i = \left( k_R k_A k_{-1}/k_I k_2^2 \right) I \]
\[ s = \left( k_A k_R/k^2_A \right) S \]

\[ \frac{da}{d\tau} = -(a - s) \quad (2a) \]
\[ \frac{di}{d\tau} = -\alpha(i - s) \quad (2b) \]
\[ \frac{ds}{d\tau} = -\beta ir + a(1 - r) \quad (2c) \]

where \( \alpha = \left( k_{-1}/k_A \right) \) and \( \beta = \left[ \left( k_{-R}/k_R \right) \left( k_{-A}/k_A \right) \right]/\left( k_{-1}/k_I \right) \).

I have checked to see whether these were correct formulations and there is a requirement that \( k_{-a} = k_{-A} = k_A \) which is acceptable since the steady state solutions for the normalized concentrations of both \( A \) and \( I \) to be equal to that of the signaling molecule. “For any value of \( s > 0 \), the normalized concentration of the active response element is,”

\[ r_{ss} = \frac{a}{a/i + \beta} \quad (3) \]

Because, \( r_{ss} \) is independent of \( s \), the steady state activity will be a function of the enzyme concentrations; secondly, \( \alpha \), the ratio of inactivation rate constants for \( I \) and \( A \) respectively, will determine the magnitude of the response as illustrated below. With the model complete, considering external cAMP to be our signaling molecule, an activator being PI3K, an inhibitor molecule to be its opposing molecular analog, PTEN, and a response dictated by the conversion of PIP2 to PIP3 presumably, it is time to simulate. The following figure 4(a), analoguous to the author’s, represents the output response for various values of alpha with a 10% increase in the concentration of the signaling molecule at time 0 made by using code, perfrun.m (D.1):

![Normalized responses for different values of alpha](image1.png)

![Normalized responses for alpha = 0.1](image2.png)

Figure 4: Normalized responses for 10% increase in signaling molecule concentration.

As one can see, this model achieves perfect adaptation. I next attempted to incorporate certain aspects of the experiment I am currently conducting whose goal is to further our understanding of *dictyostelium*’s directional sensing mechanism in response to repeated chemical signals of varying duration and frequency. To simulate this feature, I was able
to add a time dependent component to the model by means of a function and/or data set defining the amplitude and occurrence of the external signal. While the use of a thresholded sinusoidal function to carry this out was convenient, ultimately a manually imported data set would prove more reliable throughout the remainder of this paper, despite the increasing lag in MATLAB. Appendix B.1 discusses several attempts at carrying this out using the same parameters as above for a 3 second pulse of cAMP I initiated repeatedly every 15 seconds.

To enhance the predictive power of these simulations I will be making use of physical data measuring the intensity response of single dictyostelium cells to a known cAMP gradient, see 5(a), and subsequently make quantitative corrections. I will be using the data from the topmost curve and the same cAMP gradient associated with it to fit my model. The output response is scaled appropriately to coincide with the reported data, specifically at the peak, which is a measure of dimensionless intensity; hence arbitrary units [au] will often be listed. Due to the nature of system of coupled differential equations, it is extraordinarily difficult to fit two sets of data that are not represented by a particular solution for the response with variable parameters. I will thus attempt to extract the best values by manually fitting the data using a larger time step so as to match the experimental results initially for the single 'pulse' or gradient:

![Graphs showing response vs distance and manual fit to experimental data.](image)

(a) A. Samadani et al. MIT Laboratory, Massachusetts Institute of Technology

(b) Figure 5: Manual Fit with experimental data.

Using the parameter values for what I have reported as the optimal fits to the data, I extended the allotted time to allow for several pulses (same cAMP gradient) to repeatedly trigger response activities (data not shown). Unfortunately, the duration of the gradient signals is too long to see how the cell would respond to more closely-spaced pulses; however, I have made intensity measurements (with a better time step) of the experimental cAMP pulse I was working with in the lab early 2007 which decays more quickly and can therefore be incorporated into this simulation to predict cell dynamics and future data. I present the
following graphs, using my cAMP data (with a higher frequency of pulses) imported into the code, perfrun6.m (D.1), for parameters same as the previous figures:

![Graphs showing predicted responses to experimental pulses of cAMP.](image)

Figure 6: Predicted responses to experimental pulses of cAMP.

Regardless of the activity which may be physically observed in lab, I have no true defense for the numbers I used as the parameters; these were values I found that worked. I was quite pleased with the fact that the model described the qualitative behavior of the cell rather well and I will eventually try to computationally fit the parameters properly. Moreover, the Local Excitation - Global Inhibition model proposes a rapid localized activator with a slow globally diffusing inhibitor. The author of the principle paper used does not specifically address diffusion effects for this particular model (as opposed to more complicated ones) but I would argue that beta accounts for this phenomena since it mediates the inhibitor-response interaction. Still noticeable, is a response peak in the experimental data occurring at approx. 8 seconds whereas these model parameters predict just over 5 seconds. This could potentially be explained by the fact that the activated response element is essentially PIP3 whereas the measured intensities are of translocated CRAC-GFP proteins to the cell perimeter which may be time retarded appropriately and therefore display comparable decays in the concentrations studied. While this model, for the parameters I have chosen, does an adequate job at fitting the data and predicting *dictyostelium*’s response, it is not definitive and several others have been proposed including a two-pathway one which involves separate control of both activator and inhibitor and also adds an amplification step for the external signal. These changes continually increase the efficiency of a model but ‘perfect adaptation’ describes the overall behavior of the cell quite well.

Moreover, one influential feature that may be worth further investigating in this model and particularly in the one to follow is incorporating receptor desensitization. According to Van Haastert [6] and Caterina et al. [7], cells that have been pre-washed with cAMP will have widely varying loss or down-regulation of ligand binding affinities ultimately dependent on exposure time and concentration and accompanied by total desensitization of adenylate cyclase for constant levels of chemoattractant occurring over the course of several minutes.
Concerning the time scales of the pulses, I cannot determine whether this will have an observable effect on the cell’s response to the external signal; nevertheless, this aspect can be employed in my code by including an additional condition, especially for sinusoidal pulses from code, perfrun4.m (D.1), which can dictate a concentration level the cell will begin responding to and secondly, altering the signal over the time of the pulse to match a function fitted to experimental data in either of the two cited works above. These actions serve to modify the shape of the pulse so as to convey what the cell would essentially feel in vivo with consideration to a concentration threshold and desensitization. As I continue to improve the experimental setup in the lab so as to achieve the desired response to cAMP pulses, we will see if experiment matches theory and whether we can more accurately evaluate these parameters from the corresponding data.

Traveling-Wave Chemotaxis

As I already stated, starved dictyostelium signal neighboring cells by emitting periodic waves of chemoattractant of a sinusoidal nature, with a peak cAMP concentration of $\sim 1\mu$M at the height of the wave. The period of these waves typically is between 5 to 8 minutes, but is often reported as 7min on average for the first few hours following onset of starvation; as cells form aggregates at least 5 hours afterwards, the period apparently decreases to 4-5min and there is a corresponding decrease in the wave velocities, from approximately $500\mu$m/min to $260\mu$m/min [8]. As cells continuously seek out higher concentrations of cAMP, responding longer over time to spatial and/or spatio-temporal gradients and given the symmetric form of the wave, cells would be expected to respond equally well to the front and back of a wave crest in view of the fact that they experience both sides for approximately the same time. While some groups have argued for net movement with the wave, ’advection,’ this has not been seen experimentally but rather the contrary: cells ’rectify the traveling waves,’ resulting in net cellular motion opposing the direction of the wave [9]. There is only conjecture regarding this supposed paradox but it is widely believed that as cells are exposed to a sudden increase in [cAMP] their chemotactic response is dramatically reduced which reasons why they do not respond to the back of the wave.

To analyze these motions as a consequence of the wave’s characteristics, Goldstein proposed the following model I will explore in limited detail, defined by the coupled differential equations (4a) and (4b) below and based on several considerations and mathematical formulisms found in appendix B.2. As in the previous model, I am working with dimensionless quantities whereby for a 1-D array of noninteracting cells of density $\rho$ responding to a wave of chemottractant defined by the wavelength, $\lambda$, and velocity, $v$ (with values discussed earlier): time $T = \omega t$ ($k = 2\pi/\lambda$ and $\omega = kv$), concentration $C = c/c_0$ ($c_0$ is the peak wave concentration), response coefficient $R = r/f_0$ ($f_0 = f(c = 0)$), adaptation function $F = f/f_0$.
and coordinate $Z = kz$ extracted from the moving frame, $z = x + vt$.

\[
\frac{dZ}{dT} = SR\frac{dC}{dZ} + 1 \quad (4a)
\]
\[
\Omega \frac{dR}{dT} = F(C) - R \quad (4b)
\]

where the new dimensionless parameters, $\Omega$, the “relaxation time in units of the wave period” and $S$, the “ratio of the characteristic speed to the wave speed” are equal to $\omega \tau$ and $f_0c_0k/v$ respectively [9]. As the chemotactic response coefficient defining the magnitude of chemotaxis normalized becomes increasingly dependent on the present level of the concentration, $c$, ‘adaptive chemotaxis’ is described whereby the associated, chemotactic velocities should be given by $r dC/dz$. Additionally, the model achieves adaptation by choosing to include an adaptation function, $F(C)$, linearly decreasing with concentration, within the response element which in turn relaxes toward $F$ over a characteristic time constant, $\tau$; the concentration waves are simulated according to the equation below which attempts to reproduce the sinusoidal traveling waves observed in experiment,

\[
C(Z) = \exp\{\beta[\sin(Z) - 1.0]\} \text{ for } \beta = 3.0 \quad (5a)
\]
\[
F(C) = 1 - C \quad (5b)
\]

The following graphs were made by employing code, gold1run.m (D.2). In the expression for the flux, from equation B.8, the variable $S$ was added so as to properly dimensionalize the inner velocity term. Figure 7(d) used a modified code which was simulated over a long series of time steps and values of $\Omega$. Specific values of $\Omega$ were chosen to illustrate the regimes in which rectification theoretically should and should not occur.

As one observes, particularly in figure 7(d), varying values of $\Omega$ which approach $\sim 1.0$ produce net rectified motion. Due to possible issues regarding the chosen timestep, I selected the range of vector data that matched well with Goldstein’s before the flux data plateaus at the extrema. An appreciation for these measured fluxes can be resolved from figure 7(a) which illustrates the symptomatic behavior of the response as dictated by the change in concentration over time. In the regime of low $\Omega$ ($\ll 1.0$), the response function is heavily anticorrelated with the concentration; the cell will have experienced the concentration longer than the characteristic relaxation time resulting in a response coefficient $\approx F(C)$ unable to differentiate the leading and trailing sides of the wave denoting nearly instantaneous adaption. As $\Omega$ and thus the time scales permitting the response to equilibrate to $F(C)$ increase, rectification is more likely to occur until the response lags independent of $c$, a nonadaptive mode. Thus for $\Omega \approx 1.0$, rectified motion is predicted illustrated by the corresponding plot from figure 7(a) where the response is substantially different between the rising and decaying sides of the concentration wave crests.

There are, however, multiple inconsistencies with Goldstein’s proposed model, namely his simplified form of the chemotactic response coefficient (directly proportional to the velocity)
(a) Chemotactic response with accompanying concentration signal at varying frequencies of $\Omega$.

(b) Phase portrait.

(c) Chemotactic velocities at varying frequencies of $\Omega$ for $S = 0.2$.

(d) Flux as a function of relaxation time ($\Omega$) for different values of $S$.

Figure 7: Responses at low $S$ for $F(C) = 1 - C$. 
as well as his “adaptation function,” $F(C)$, which is not justified by the preponderance of experimental evidence dictating how the cell’s response behaves for constant levels and gradients of chemoattractant; this data does not favor a linear decreasing function as it currently stands. In the appendix (B.2) I look at responses suggesting rectified motion obtained with a new expression for $F(C)$ by fitting experimentally determined normalized responses to various concentrations of cAMP. Alternatively, $F(C)$ could well be described by an equation fitting the desensitization of cell receptors curve as a function of $[cAMP]$ instead or integrating over the output response from the Perfect Adaptation model. In one case, we attempted to analyze the response function by examining the experimental velocities reported by Wessels et al. [10]. Despite its usefulness for determining chemotactic velocities at various $\Omega$, this model has been intentionally simplified to account for adaptive chemotaxis, and consequently leaves itself open to a host of attacks due to the significant lack of evidence governing the basis and his choice of response functions. I will therefore leave this model. While there is currently no physical data conveying net motion away from the source of chemoattractant waves as predicted at the extremum values of $\Omega$ according to Goldstein, this can possibly be experimentally verified using our system which I will now discuss.
Methods and Materials

As previously stated, a strain of *dictyostelium* cells obtained from A. Samadani expressing the CRAC-GFP construct were used throughout the duration of the experiment. These cells were initially allowed to grow in petri dishes (figure 8) to densities \( \sim 5 \times 10^6 \text{cells/ml} \) in HL-5 media containing 20\( \mu \text{g/ml} \) G418 antibiotic according to the laboratory procedures detailed in the appendix A.1. Throughout the year we have continuously augmented the procedures required to obtain viable cells for experimentation. Optimal conditions necessitate growing the cells in greater volume and in shaking suspension several days prior to study. Furthermore, to induce a state in which *dictyostelium* are more responsive to chemoattractant, cells are harvested by centrifugation the day of the experiment, washed, resuspended in 10ml of developmental buffer (DB2) and shaken at 100rpm for 6-8 hours illustrated by the setup in figure 9 while being repeatedly pulsed with cAMP every 6 minutes by means of a peristaltic pump, figure 10, so as to achieve a final concentration of >100nM within the flask, see appendix A.2. Starved cells are distinguished from vegetative ones by their elongated appearance and tendency to form clumps, or aggregates, hours following the initiation of starvation protocol. The pump is electronically controlled by an adjacent computer running LabVIEW (figure 11); the versatility of this program, full block code in section C, which acts as a timed on/off switch within a while loop when used with the connected NI-DAQ Board, ultimately allowed me to regulate the release of cAMP and buffer during the experiment as I will discuss shortly.
Figure 9: Laboratory setup for starvation including the use of an electronically-controlled peristaltic pump to directly pulse a shaking flask of cells with [cAMP] every 6 minutes.
Figure 10: Peristaltic Pump (Top View).

Figure 11: LabVIEW Graphical Use Interface (GUI) to electronically-control devices.
Experimental Setup

To study and record the cell’s behavior, we used a Nikon *Eclipse* TE2000-E inverted fluorescence microscope designed for high precision live cell microscopy with automation, see figure 13, and the companion imaging program, NIS-Elements. We designed a cell chamber (R&D #44795) shown in figure 14(a), that was produced by GraceBio Labs according to the specified dimensions with a T-channel depth of 0.12mm and width of 3.15mm that serves as an analog to a microfluidic device ideal for achieving low flow rates. Figure 14(b) displays how the chamber is assembled by placing the T-channel between a thin cover glass, which allows for higher microscope objectives, and the base of the plastic chamber with predrilled dual injection ports designed for use with PE-60 tubing and added epoxy glue. These ports allow for the administration of chemicals, specifically buffer and cAMP/fluorescein, into the chamber and is controlled by an actuator running the LabVIEW program already mentioned. The actuator itself, see figure 15(a), is a three-way solenoid pinch valve purchased from Cole-Parmer (EW-98305-00) with recommended C-Flex Tubing (1/32” ID x 3/32” OD) requiring a 12DC input voltage to operate; since the maximum voltage provided by the NI-DAQ Board is 5volts, I built an electronic circuit comprising a miniature DPDT relay used in conjunction with a DC power bench supply as illustrated in figure fig: actuator. A third port, located at the end of the T-channel, is connected to a three-way luer stopcock valve, figure 15(b), which permits the injection of starved cells into the chamber while minimizing the possibility of introducing air bubbles into the system; this valve is placed on the pump side of the chamber 1) since a single air bubble, upstream of the channel, would likely flush all the contents, including cells, away and 2) to avoid any possible disparities in the steady flow of buffer and chemoattractant between the two feeding tubes. Flow is maintained throughout the system by a syringe pump (Harvard Apparatus PHD 2000 Infuse/Withdraw), seen in the accompanying figure 12 to the right, downstream of the stopcock valve set for Turemo 3cc luer-lock syringes (diameter = 8.95mm).
Figure 13: Fluorescence Microscope Setup.
(a) Cell Chamber with PE tubing connected to individual ports via epoxy glue.

(b) Making the Cell Chamber.

Figure 14: Construction of the Cell Chamber.
(a) Connecting the devices.  
(b) Three-way luer stopcock valve.

Figure 15: Connecting the devices.

(a) Controlling the actuator.  
(b) Schematic view of actuator circuit relay.

Figure 16: Controlling the actuator.
Characterization of the System

Understandably, the above-mentioned setup needs to be calibrated so as to achieve the optimal conditions required for successful experimentation with live cells. In order to evaluate the behavior of the system with consideration of several parameters such as velocity and the degree of cAMP saturation within an enclosed rectangular channel, a series of measurements using fluorescein, which has a diffusion coefficient comparable to cAMP, $4.9 \times 10^{-10} \text{m}^2/\text{s}$, were performed. The possibility of establishing gradients of chemoattractant from microfluidic devices was not feasible at the time, instead we proposed how varying the frequency and duration for pulses of cAMP, externally controlled, may influence the response of dictyostelium. In several simulations performed by Li et al. [11] [12], square-wave pulses of chemoattractant were employed and while this type of pulse would have been ideal for our circumstances, it is neither practical nor does it reflect the shape of naturally occurring waves secreted by cells. A gaussian profile for the pulse was thus preferred and we began conducting trials with fluorescein for several pumps that provided us with a range of flow rates. Ultimately a syringe pump, described in the previous section, afforded the desired linear velocities with acceptable deviations in the flow which become more pronounced at much lower volumetric flow rates. Since our setup involves flow rates within the microfluidic regime, it becomes necessary to understand the dynamics of flow within the chamber. Using yellow-green fluorescent beads I was able to map the linear velocities of the chamber as a function of these volumetric flow rates as well as the height, see figure 17.

![Figure 17: Velocity profiles of the cell chamber](image)

The velocities were averaged over several dozens of length measurements for the beads’ fluorescent tracks divided by the exposure time. For very low flow rates, where the tracks become indiscernible over a finite time of the exposure, the initial and final positions of the
bead, as it progresses from the left to the right side of the camera’s frame, were measured instead. Both of these approaches should yield the true value of the linear velocity at a given height and also be consistent, for the maximal linear velocity (at $\frac{1}{2}h_0$), when compared to the velocity dictated by the fluorescein data that I will present shortly. While I would have liked to fit the velocity profiles to the equations describing flow within rectangular channels, I was not able to do so at this time. A back-of-the-envelope calculation can nevertheless be performed whereby the mean linear velocity within the chamber can be asserted from simply,

$$\bar{v} = \frac{\bar{Q}}{A}$$

(6)

where $\bar{Q}$ is the volumetric flow rate and $A$ is the cross-sectional area of the chamber. For the dimensions listed in figure 14(b) with a width slightly greater than the reported value based on the measured distance between the bead-defined top and bottom surfaces of the chamber, $\bar{v}$ is 302$\mu$m/s and 151$\mu$m/s for 500$\mu$L/hr and 250$\mu$L/hr respectively since the velocity scales linearly with the volumetric flow rate according to equation 6, of course. To examine the type of pulses the cell would actually be ‘seeing,’ I made serial dilutions of fluorescein with buffer for a peristaltic and a gravity-driven pump that I was initially using at the beginning of the year to achieve flow rates of several millimetres per second. The resultant time-measurements of the intensity, averaged over a very small defined rectangular region that can be considered uniform, for different pulse lengths are found below in figure 18(a) in which I have adjusted only the times of the RAM-captured movies such that the observed pulses begin at approximately the same time.

The pronounced asymmetry of the wave which we have been calling ‘capacitor-like’ was unexpected as was the observance of saturation at such small pulse times. To explain these findings, we consulted Howard Berg’s widely-referenced book, Random Walks in Biology [13], in which he describes the same characteristic pulses for the diffusion of particles from a micropipette calculated for various diffusion coefficients, $D$, with held variables including the velocity, $v$, the length of the pulse, and the distance from the source, $r$. For an arbitrary number of injected particles, $N = i dt$ where $i$ is the rate, the solution to the spherically symmetric form of Fick’s second equation concerning the change in concentration with respect to time, $dC/dt$, is,

$$C(r, t) = \frac{N}{(4\pi Dt)^{3/2}} e^{-r^2/4Dt}$$

(7)

This describes a three-dimensional Gaussian distribution; consequently, an observer will witness the swift rise and steady decay in the concentration of particles, being highest at the source, and decreasing with $t^{3/2}$ [13]. Integrating equation 7 for a pulse of length $t_0$ yields the following expression for the corresponding wave,

$$C(r, t) = \frac{i}{4\pi Dr} \text{erfc} \left( \frac{r}{(4Dt)^{1/2}} \right)$$

for $t \leq t_0$ \hspace{1cm} (8a)

$$\frac{i}{4\pi Dr} \left\{ \text{erfc} \left( \frac{r}{(4Dt)^{1/2}} \right) - \text{erfc} \left( \frac{r}{4D(t-t_0)^{1/2}} \right) \right\}$$

for $t > t_0$ \hspace{1cm} (8b)
(a) Time-measurements of average intensity.

(b) Simulation of concentration waves.

Figure 18: Average intensity as a function of pulse length for a high velocity.
Taken together, these two equations \(8a\) and \(8b\) adequately illustrate the behavior of the wave as a function of pulse time. In spite of the number of variables, an explanation as to why saturation occurs so quickly with respect to the limited range of pulse times assessed (constant velocity), I would surmise is a consequence of the diffusion coefficient which is average among other chemicals’ extremes, \(10^{-5}\) cm\(^2\)/s and \(3 \times 10^{-8}\) cm\(^2\)/s. Comparatively, waves will tend to disperse smoothly for small \(D\) whereas large diffusion coefficients dictate greater spreading which should see particles accumulating naturally over time in any observable region as Gaussian pulses diffuse on top of one another, thus the approach to saturation for the mild diffusion coefficient of fluorescein, relative to the time length of the pulse, is consistent within this formalism. There is, however, still a natural competition between the diffusion time and the flow rate. To properly evaluate the data, and that which I collected for smaller linear velocities within the chamber, I must introduce flow into equation \(7\), which until now has only considered spreading waves due to a pulse from a point source,

\[
C(r, t) = \frac{N}{(4\pi Dt)^{3/2}} e^{-(r-vt)^2/4Dt}
\]

I have relied on MATLAB to numerically integrate equation \(9\) by quadrature for all subsequent simulations and fitting analyses. Once again, the integral is evaluated for two different regimes, \(t \leq t_0\) and \(t > t_0\), such that the resulting boundary conditions account for the addition of only the diffusing Gaussians that contribute to the concentration of the wave \((t - t_0)\). Whereas figure 18(a) reported the intensities as a function of the pulse time for drastically higher velocities than what we would prefer to work with, figure 19(a) illustrates how the shape of the pulse is affected by the velocities, which were maintained by the syringe pump for a different diameter, and therefore different volumetric flow rates, than the 8.95mm standard. Regardless, one is able to see that for smaller velocities, the pulse begins to appear more symmetric or ‘Gaussian-like;’ note that figure 19(a) once again has the pulses overlayed such that each starts at approximately the same time while the adjoining figure (19(b)), made with MATLAB by integrating equation \(9\), has the pulses spaced appropriately given that the velocities, which are only estimated to be of the same order as those used experimentally, predominately dictate when the concentration wave’s peak reaches the observation point, \(r = 2\) cm from the source.

From the reliability of the syringe pump, we are attaining experimental velocities matching the last pulse generated in figure 19(a) and several hundred \(\mu\)m/s lower. Fluorescein data collected within this range of corresponding volumetric flow rates, specifically for \(500\mu\)L/hr and \(250\mu\)L/hr, is presented in figure 20 for \(r\) measured to be roughly equal to 2cm and 3cm and in figure 21 which demonstrates the reproducibility of such pulses at the given settings. Before I discuss the apparent theoretical fits to the data, one important addendum to the codes which allow MATLAB to simulate the pulse waves must be addressed. Since the flow is generally laminar with a profile appearing to be parabolic transverse to the true flow, marked by extreme velocities within the middle of the chamber and falling off to zero along the boundaries, hydrodynamic (Taylor) dispersion becomes increasingly significant; this form of dispersion dictates the spreading of suspended particles along the direction of flow and must be appropriately accounted for with an effective diffusion coefficient. For rectangular
(a) Time-measurements of average intensity.  
(b) Simulation of concentration waves.

Figure 19: Average intensity as a function of flow rate - Transition from high to low velocity regimes.
where the Péclet number, a dimensionless quantity relating the rate of a flow’s advection to its diffusion, is defined as $P \equiv \frac{v h_0}{D}$. Taylor dispersion is consequently dependent on the channel’s height and the side walls reduce the above equation to its 2D form [14]. This change results in approximately two orders of magnitude increase in $D$ of fluorescein for the volumetric flow rates being considered, see the text boxes in figure 20 for the effective diffusion coefficients calculated and simulated for using code, diff-drift.m (D.3) which calls upon another function, f.m (D.3). These coefficients were also used to make figures 18(b) and 19(b) which are akin to their partner graphs. This correction was determined to be crucial in order to achieve better theoretical predictions of the pulse to our experimental data for fluorescein since a smaller diffusion coefficient results in a substantially narrower width than those seen. Neither of the so-called fits in figure 20 I should point out were determined computationally, instead I manually varied what should be the only free parameter, $v$, considering the pulse time is fixed and $r$ is a given measurement, until I achieved the best visual plots as reflected by the R-square values I subsequently calculated.

Two additional areas meriting further attention include the disparity between the velocity data obtained by both the fluorescent beads and fluorescein as well as the average intensities which are currently measured in arbitrary units. Comparing figures 17 and 20, it is clear that for the same volumetric flows, the mean linear velocities extracted from the latter were roughly twice as great. As I already discussed, the data taken from beads should match well with fluorescein assuming no instrumental error. A possible explanation that partially accounts for this discrepancy is the consequence of using a ‘used’ flow chamber to measure fluorescein intensities. I discovered that a number of air bubbles, which may be accompanied by the loss in adhesion of the chamber’s sealed elements, had attached themselves to the sides of the channel resulting in a total reduction in the cross-sectional area by nearly 30%. From equation 6, this translates into a gain of 129.4 $\mu$m/s at 500 $\mu$L/hr with analogous increases in the linear velocity for other volumetric flow rates. We are therefore confident that the velocities detailed in figure 17 are correct following repeated measurement with a new chamber; thus it is resolved as it was always our intention to construct a chamber for separate experiments anyway. Furthermore, it was determined that the peak concentration of the pulse varies considerably when flow is included while the differences associated with the change in the observation region’s location for these velocities, at least 1.5cm away from the source, are negligible (data not shown). Therefore, to expose cells with an ideal concentration of $\sim$1$\mu$M, we made a map, see figure 22, of the average intensities for fluorescein in the chamber as a function of the concentration and exposure times used; unlike cAMP, fluorescein has low solubility in DB and the concentrations made by serial dilutions may be slightly less than reported. Based on our measurements, we believe 60$\mu$M to be an appropriate concentration with respect to the parameters chosen in figure 20.
Figure 20: Manual fits to the experimental data with calculated R-square values collected at 500\(\mu\)L/hr and 250\(\mu\)L/hr for observational distances 2cm and 3cm from the source.

Figure 21: Reproducibility of 5s fluorescein pulses at 500\(\mu\)L/hr. Separation between pulses was 30 seconds with an exposure time of 400ms.
Figure 22: Fluorescein Maps relating average intensity to exposure time and concentration of fluorescein.
Results

Data Analysis

In order to quantify the degree as to how well a cell detects and responds to a pulse of cAMP, it is necessary to determine the amount of CRAC that accumulates along the periphery of the cell which is directly proportional to velocities associated with a chemotaxis event. This is accomplished by extracting the intensities from a pixellated image of particularly high-resolution. To perform these analyses, I made extensive use of MATLAB’s image processing toolbox and wrote several codes, found in the appendix (D.4), which I will discuss in detail.

To demonstrate the function and behavior of these codes, I will use a movie made by Parent, see figure 23. The figure (24) that follows shows the first frame (image) of the movie, displaying three cells labelled accordingly; MATLAB assigns label values to regions based on their cartesian position (the positive y-axis begins at the upper left hand corner while the positive x-axis, by usual convention, starts at the bottom left corner). This latter figure is a screen capture of the GUI program I wrote in MATLAB, vid4.m (D.4), which I used exhaustingly to continuously modify my imaging codes. This GUI can be called when one is solely in a directory of images, such as TIFF files. It will subsequently read in all images in alphabetical order and can present all images by moving the bottom slider, initially set at 1, or the first image. Additionally, my main image analysis code, analyze.m (D.4), actually acts as a function whereby the user can select to read in a directory of images, a single image, or AVI movie for processing. MATLAB stores the information for each image as a matrix usually containing real scalars in the range [0,255] whereas an associated colormap or RGB values for certain file types will range from 0 (black) to 1.0 (full intensity). One feature of this GUI, highlighted by figure 25, is the option to render the image in grayscale, in which the intensities of the new matrix are in the range [0,1], and threshold at an appropriate value. Ultimately this did not prove practical for our experimental goals but nevertheless offers the user the ability to define a labelled perimeter or boundary for each object as a function of the thresholding value.
Figure 23: High-resolution AVI movie (18 frames) used for MATLAB image analysis depicting three cells exposed to a local source of cAMP injected via a micropipette at the bottom right. Movie taken from supplementary material of [15] and can be found at http://www.sciencemag.org/feature/data/990801S3.AVI.
Figure 24: MATLAB image analysis Graphical User Interface (GUI) called using the vid4.m code (D.4) and displaying frame 1 of the above movie.
The CRAC-GFP construct proteins accumulate along the periphery of a cell, thus it is of importance to isolate this region for individual cells in frame, apart from the background or anomalous fluorescence within the cytosol, and determine the corresponding intensities from the original image. These regions are defined as the enclosure by two boundaries seen in figure 28(b) following GUI button press in erosion. To illustrate how these boundaries are established I have selected the first frame from the above movie, added various objects within and along the sides of the image to demonstrate the functionality of my code as seen in the first box from figure 27, and show how the image is resolved in a stepwise fashion by each of the pertinent MATLAB commands. The first and arguably most important step involves the canny edge detection technique which determines edges by looking for local maxima of the gradient” of the image matrix (box 2); this method proved to be the optimal choice with allowed conditions for specifying high and low sensitivity thresholds.\(^1\) Several additions to the code include incorporating an image structuring element to help contrast the image further by dilating (box 3) and then eroding it while filling in any small areas within a confined region (box 4). Moreover, objects of a

specified size and/or shape located outside these well-defined regions (box 5) or along the image’s border (box 6) are removed. The command, \([B,L,N] = \text{bwboundaries}(\text{bw1,’noholes’})\), first excludes all possible holes and children within a region, see adjacent figure (26), and returns \(B\), a cell array consisting of the cartesian coordinates for the boundary pixels.\(^2\) \(L\), as the second argument, contains integer values representing each different yet uniquely labelled contiguous region while \(N\) reports the number of objects found. MATLAB plots the boundaries (green) on top of the original image which should mark the cells’ outer membranes (box 7) and figure 28(a).

Button press of erosion erodes the contrasted image again and recalculate secondary boundaries (red) which are plotted on top of the previous image (box8). It is important to consider the resolution of the image as this function will symmetrically shrink the regions by a designated number of pixels in order to define these additional boundaries inside of the first. By subtracting the labelling regions from both sets of boundaries, \(Z\), one now has the image matrix elements corresponding to the desired regions (the periphery of the cells where CRAC-GFP accumulates). These matrices are passed to the MATLAB workspace by means of the assignin command; GUIs and functions operate outside the MATLAB workspace environment and therefore do not typically share information.

This description essentially details the majority of my image analysis code, polar2.m (D.4), which is called by default, after analyze.m reads in the image matrices for the desired visual data. The cell’s geometric center, based on the labelled regions of \(Z\), will redefine a new origin \((0,0)\) for all position data of each cell. Furthermore, since a pulse will not necessarily saturate the cell inducing a uniform yet transient translocalization of CRAC-GFP, it becomes convenient to evaluate the distribution of CRAC at the membrane as a function of an angle designated as zero along the positive x-axis and which rotates from 0 to 360 degrees counterclockwise; I have converted to polar coordinates. Choosing to bin every 10°, I obtain 36 averaged intensities along a cell’s perimeter as well as its total average intensity. This is repeated for each cell and and for each image using several for loops. This information is saved in an Excel workbook, each sheet containing the data for a particular frame. Additionally, it is quite useful to plot the intensities as a function of the angle \([0,360]\), see figure 29, which can be saved as an AVI movie too. Alternatively, considering the objects being studied, another graphing technique, specifically the compass plot (figure 31) which utilizes a polar coordinate system, demonstrates how the intensity vector for each cell changes in time.

Depending on the experimental system (direction of flow) and/or prerogative of the analyst, I have included an excerpt from another optional code, polar3.m (D.4), that can be called with the analyze.m function, which behaves exactly as the previous code but for an angle that ranges from -180 to 180 degrees.

Figure 27: Illustrating the image analysis code in a stepwise fashion for frame 1 (Box 1 - top left, Box 2 - top right, and so forth).
Figure 28: MATLAB: Defining inner and outer cell boundaries by canny edge detection method utilizing the GUI’s polar2.m code (D.4) for frame 1.

Figure 29: Standard plot of CRAC-GFP intensities from figure 23 for each cell as a function of the angle, binned every 10° counter-clockwise starting along the positive x-axis.
(a) MATLAB: Standard plot of CRAC-GFP intensities.

(b) MATLAB: Compass plot of CRAC-GFP intensities.

Figure 30: MATLAB: Plots of CRAC-GFP intensities for each cell as a function of the angle, binned every 10° counter-clockwise starting along the positive x-axis for frame 1 of figure 23.

Figure 31: Compass plot of CRAC-GFP intensities from figure 23 for each cell as a function of the angle, binned every 10° counter-clockwise starting along the positive x-axis.
Preliminary Data

While we have been continuing to improve our experimental system, I have been conducting weekly experiments 6 - 8 hours following the onset of starvation. Although I never achieved a high percentage of distinctly starved cells, marked by their elongated appearance when viewed under a microscope, I nevertheless diluted small aliquots of cells and attempted to identify possible candidates for observation. Past studies have shown that starved *dictyostelium* only respond to cAMP above a threshold concentration of 1nM and that anything beyond 1µM is likely to saturate the cell inducing a uniform, yet transient response. Prior to characterizing the system and with the use of a rotary peristaltic pump, I managed to make several movies with conditions specified in the captions of the following figures:

![Figure 32: Uniform translocation events for 11-30-06 experiment. 2 second pulses of 1.25µM cAMP separated by 30 second pulses of buffer (DB2). The pump speed was set at 20% which corresponds to a mean linear flow rate within the chamber of approximately 2mm/s. Images were captured every 5s with a resolution of 0.67µm per pixel.](image)

In the above figure (32(b)), I have identified two distinct translocation events, at 40.1s
and 135.1s; these were not the only translocation events observed nor was this the only cell in frame that exhibited this behavior at the same time (data not shown). To illustrate the periodicity of these translocations, occurring every 30 to 35 seconds as expected, particularly when they are difficult to see, especially as the cell’s response efficiency dies with prolonged exposure to emitted light (I cannot address how the cell’s response may be affected as a function of continued pulsing as this is one of the ultimate goals of this research), I have plotted the mean intensity versus the area (scaled) of the cell in figure 33.

Figure 33: Anticorrelation between mean intensity and area (cringe effect observed). Pearson coefficient = -0.215.

Initially when I began the experiment at night I observed a peculiar change in the shape of all the cells as I pulsed them: they all became dark and shrunk in size. This is the so-called cringe effect which causes cells to round up (‘cringe’) and slow down briefly when exposed to a sudden elevation in [cAMP]. Upon seeing this behavior I switched to the fluorescence filter and later discovered the above translocation events. Hence the direct anticorrelation between the intensity of the cell (high during a translocation event) with the area is indicative of the desired response.

In another attempt, we managed to observe a polarized, asymmetric localization of CRAC-GFP along the periphery of the cell at 38.56s in figure 34(b) accompanied by a change in the cell’s morphology. This change occurs very rapidly between the previous frame and the current one, presumably due to an increase in actin polymerization at the cell’s front (left), which is facing the wave of cAMP passing over and persisted at 41.68s before the CRAC-GFP accumulation is no longer visible to the eye in the last frame. Unlike the former movie, the higher resolution of these frames may allow for image analysis; this has not been performed as of now.

In spite of these results, the likelihood that we managed to observe this latter translocation event was diminished since figure 22 reveals that we should have been using a much
Figure 34: Asymmetric translocation events for 1-25-07 experiment. 2 second pulses of 1.2µM cAMP separated by 20 second pulses of buffer (DB2). The pump speed was set at 10% which corresponds to a mean linear flow rate within the chamber of approximately 480µm/s. Due to shutter slowness, images were captured every 3.12s with a resolution of 0.16µm per pixel.
higher concentration of cAMP. It is reasonable, however, to conclude that the linear flow rates associated with this peristaltic pump were still within acceptable ranges for *dictyostelium* to perceive the spatial concentration differences over the cell (up to ~2%) and polarize as the pump speed was reduced from 20 to 10%; it should be noted that such flow rates typically ‘shear’ the cells (physically altering their morphology in alignment with the direction of the flow), eventually causing the cells to detach from the chamber’s floor and to be subsequently swept into the flow as the movie later shows.
Discussion

Notwithstanding the initial success of these experiments, we have encountered a number of problems that hinder the reliability of conducting future research on a regular basis. Throughout the year, the strain of cells I have been growing have been plagued by several bouts of contamination, presumably yeast, as seen in adjoining figure 35. Understandably, contamination decreases the uniformity of the population and can potentially eradicate our batches of *dictyostelium* but the presence of yeast is most obstructive during starvation since they provide the cells with a constant source of food. Even without contamination, however, *dictyostelium* cells that were at confluent densities and ready to be experimented with never became sufficiently 'starved,' illustrated by their elongated appearance; despite my continued variations to the starvation protocol such as increasing the internal concentration of cAMP or density of cells, they did not respond as efficiently as the strain supposedly did prior to its transfer to the lab.

Notes to future researchers

With regards to the MATLAB image analysis codes, there are several issues which may need to be addressed later on. As I already stated, depending on the resolution of the image, the length of the strel element used to erode the outer boundaries for the cells will have to be changed accordingly to obtain the desired peripheral region. While the analyze.m code (D.4) is capable of reading in a single image or a directory of images, there cannot be any anomalous files located within the folder in order for the code to work. Furthermore, the AVI movie that one can also attempt to extract the intensities for could have additional levels of information and may not function properly with the polar2.m code (D.4) which considers only a three-dimensional image matrix. If required to, one can correct this by writing another if loop within analyze.m in which the size of the image matrix (proabably 4-D) determines the correct version of polar2.m; this other version can be created by adding
an additional dimension (:) to each workspace variable in the code. Other enhancements include modifying the compass plot command so that the magnitudes of the intensity vectors are contained within the bounds of the graph’s axes which should be set permanently as well as changing the binning size for the angle.

While we have collected a considerable amount of data that is capable of characterizing the overall properties of the system, due to time constraints I was unable to give full attention to the region at the bottom of the chamber in which flow around cells slightly deviates from the velocity profiles that I have mapped. However, it is possible to adequately describe the fluid dynamics in this locale from numerous papers published on the subject given only the trends from the data which can be fitted more precisely.
Conclusion

We have developed a device capable of generating spatio-temporal pulses of chemoattractant thereby reproducing the periodic, symmetric waves *dictyostelium* naturally emits when starved. The experimental setup allows us to control the frequency and duration of these pulses which can yield potential insight regarding *dictyostelium*’s gradient sensing mechanism. To characterize the system with respect to the flow, fluorescein, which has a comparable diffusion coefficient to cAMP, as well as yellow-green fluorescent beads were used. As one transitions from regimes of high to low linear velocities within the chamber, the shape of wave becomes more ‘Gaussian-like.’ Additionally, it was discovered that Taylor dispersion, arising from parabolic velocity profiles transverse to laminar flow, significantly affects the width of the chemoattractant wave by increasing the effective diffusion coefficient ultimately allowing for the successful fitting of our experimental data to theoretical models describing diffusing three-dimensional Gaussian distributions with flow. Overall our testing accounts for most experimental factors influencing flow but this should be considered a coarse-grained view of the system for the reasons discussed in the previous section. Using a syringe pump, we have achieved flow rates within the chamber that the cells are able to tolerate and while only preliminary data for cells exhibiting the desired response, both a uniform and an anisotropic accumulation of CRAC-GFP to the cell’s periphery, has been collected, the system is sufficiently characterized such that one should be able to determine how the adaptive response changes as a function of the input signal or cAMP wave as dictated by several LEGI models. Moreover, according to the traveling-wave chemotaxis model by Goldstein, future researchers should be able to create pulses for the frequencies that have been predicted to elicit motions away from the source which have yet to be observed experimentally. The use of high magnification objectives optimized for fluorescence microscopy will improve resolution permitting the measurement of intensities via image analysis.
Acknowledgements

Who would have thought slime mold could be this exciting? Over the course of my research I have not only developed an appreciation for this social amoeba but to the study of cell signaling and motility in general, due largely in part to my advisor, Azadeh Samadani, who I must thank. In addition to her valuable suggestions, she has afforded me the opportunity to learn such laboratory techniques as fluorescence microscopy and MATLAB, especially. The ability to program in MATLAB has been considerably helpful to me this past year and had I not been forced to use it during the previous summer to perform data analysis, this thesis would be a mere 40 pages. Furthermore, my colleague, Samuel Rauhala, has served as a constant source of motivation (this statement is intended to be humorous). Lastly, I would like to extend special thanks to the department staff, who, in addition to lending an ear to our usual airing of grievances related to work or otherwise, have been providing poor undergraduate students with sustenance for many years at either the club meetings or the 3:30pm snacks for the colloquia.

Figure 36: Image taken from “Piled Higher and Deeper” by Jorge Cham http://www.phdcomics.com/.
Bibliography


Appendix A

Laboratory Procedures

A.1  *Dictyostelium* Growth and Media
Recipe for Developmental Buffer (DB2, 1 liter)
5mM Na₂HPO₄
5mM NaH₂PO₄ [pH 6.2]
2mM MgSO₄
200µM CaCl₂
MilliQ H₂O

Make sure to clean and/or autoclave all glassware prior to preparation in order to reduce possible sources of contamination.
Wear gloves throughout procedure.
It is wise to prepare 3, 100x stock solutions of the above chemicals. The following are the appropriate amounts and desired concentrations to make 100ml of each:

500mM Phosphate Buffer
5mM Na₂HPO₄ 7.098g
5mM NaH₂PO₄ 5.998g
MilliQ H₂O ~100ml

200mM MgSO₄
MgSO₄ 2.407g
MilliQ H₂O ~100ml

20mM CaCl₂
CaCl₂ 0.2219g
MilliQ H₂O ~100ml

To a 1L bottle, add 10ml of the Phosphate Buffer and fill to proper level with autoclaved MilliQ H₂O (~970ml). It is necessary to adjust the pH of this solution to 6.2 by addition of HCl. Add 10ml of the remaining two stock solutions. Autoclave to sterilize.

Recipe for HL-5 (1 liter)
Yeast Extract (Oxoid) 5.0g
Thiotone E Peptone (Difco/BBL) 7.5g
Proteose Peptone (Difco/BBL) 7.5g
Na₂HPO₄·7H₂O 0.35g
KH₂PO₄ 1.20g
Dextrose (Mallinckrodt/Fisher) 10g
MilliQ H₂O* ~900mL

*The amount of MilliQ water listed above is less than the final volume to compensate for other ingredients.

Make sure to clean and/or autoclave all glassware prior to preparation in order to reduce possible sources of contamination.
Wear gloves throughout procedure.
Depending upon how often HL-5 will be used, one may choose to make stock solutions of the chemicals, particularly a 10x diluted dextrose solution. To make this, add 50g of dextrose slowly to a 500ml flask containing ~300ml distilled water initially so as to avoid unwarranted volume changes. Use a magnetic stirrer and/or heater to facilitate dissolution and add water to reach 500ml mark. Autoclave to sterilize. It is also useful to have several liters of autoclaved MilliQ H₂O to begin with.
To a 1L beaker, add the desired amount of water and all of the required ingredients except dextrose. A magnetic stirrer is helpful to mix the ingredients so that they enter solution faster. After all solutes have dissolved, adjust pH of solution to 6.5 by addition of HCl. Autoclave for 20 minutes.

Keep the beaker of HL-5 (without dextrose and containing ~900ml of solution – it may be necessary to add additional water if excessive evaporation occurred while autoclaving) and the dextrose solution (10x dilution) separated. When you are ready to grow dictyostelium, have 50ml of HL-5 at hand by adding 5ml of dextrose (10x dilution) to 45ml of HL-5 (without dextrose). Store all media at 4°C.

To begin growing dictyostelium, keep a 50ml of HL-5 from stock in an available sorvette tube (see above).

-Extract 10ml and pour into a sterile Petri dish. Add 500µl of HL-5 to frozen stock of dictyostelium in a microfuge centrifuge tube and continuously pipette up and down to melt; gradually place suspended cells in Petri dish. Given a stock solution of 20mg/ml of G418 antibiotic, add 10µl to cells to achieve a concentration 20µl/ml. Place Petri dish cover on and monitor every 3 days to watch for contamination allowing growth to densities of ~5x10⁶cells/ml.

Note: it is wise to replace the media after the first day to remove unwanted contaminants from frozen stock.

If the concentration is greater than desired density, re-dilute cells 100x fold accordingly in either 5 or 10ml of HL-5 to keep them in their exponential growth phase.

-Alternatively, one may wish to begin growing dictyostelium by shaking suspension. Typically one would add approximately 20ml of HL-5 to an autoclaved 250ml Erlenmeyer flask (with aluminum foil covering mouth to prevent contamination). Add 500µl of HL-5 to frozen stock of dictyostelium in a microfuge centrifuge tube and continuously pipette up and down to melt; gradually place suspended cells in flask. Given a stock solution of 20mg/ml of G418 antibiotic, add 20µl to cells to achieve a concentration 20µl/ml. With foil, place flask in a shaker set to ~200rpm (speed 5) and monitor every 3 days to watch for contamination allowing growth to densities of ~5x10⁶cells/ml. If the concentration is greater than desired density, re-dilute cells 100x fold in HL-5 accordingly to keep them in their exponential growth phase.

To count and estimate the density of cells, use a standard hemacytometer (it may be necessary to dilute a sample in order to improve accuracy).

- Mount coverslip over counting chamber and add aliquot (~10µl) at the edge of the “V” shape of the chamber. Place on microscope stage.

- Locate the 25 individual squares (within a total area of 1mm x 1mm) and begin counting the number of cells in each square. If the population of cells is relatively constant between squares, you are permitted to count approx. 6 squares and average their populations; otherwise, average the cell counts over all 25 squares.

- To determine the density of cells (#cells/ml), multiply the average by: 25, 10⁴, and any dilution factor that may have been used:

\[
\text{density(#cells/ml)} = (\text{avg})*(25)*(10^4)*(\text{dilution factor})
\]
A.2 *Dictyostelium* Starvation
Starvation

To grow cells for starvation use an autoclaved (with foil) 1L Erlenmeyer Flask. Suspend cells in approximately 200ml of HL-5 with added G418 antibiotic. Place flask in a shaker set to ~250rpm (speed 7) and watch for contamination allowing growth to densities of ~5x10^6 cells/ml; it is best to harvest cells while they are in their exponential growth phase. If the concentration is greater than desired density, re-dilute cells 100x fold in HL-5 accordingly.

1) Centrifuge cells (use setting 4 on our centrifuge) at 2500rpm for 5 minutes.
2) Pour off supernatant (loss of cells is acceptable).
3) Wash in 5ml of Developmental Buffer (DB2). Ensure that cells go back into solution.
4) Repeat steps 1-3.
5) Centrifuge cells again and pour off supernatant.
6) Cells are now ready to be transferred to a flask for cAMP pulsing. Suspend cells in DB2, transferring in approximately 3mL at a time such that the total volume in the flask should be 10ml.
7) Place flask in shaker set to ~200rpm (speed 5). Pulse cells with cAMP, dropwise, every 6 minutes for up to 8 hours. The initial concentration of cAMP and the duration of the pulse should be calculated in order to achieve a final concentration of 100nM in the flask and a total change in volume of ~4ml.

Notes on procedure:

- If cells are reddish/brown, they are likely overgrown.
- If you are combining cells from multiple stocks, it is probably best to do so after the first centrifugation.
- Cells begin to die roughly 6-8 hours after starvation is initiated. Do not begin starvation unless you will be able to experiment with cells within that period of time.
- For Samadani Lab purposes whereby we are using a voltage-controlled Cole-Parmer L/S Digital Standard Drive with L/S 13 tubing threaded through L/S 15 tubing to be used with a High-Performance Precision Pump Head set to L/S15 sized tubing in conjunction with a USB-6008 DAQ board (0-5volts analog DC output):
  - Pulse duration ~ 3.8 seconds
  - [cAMP]_{initial} = 20μM (concentration is greater than what is specified in step 7)
  - 0.4 volt DC input from DAQ board to generate a flow rate of 1.3ml/min
  - Droplet volume is approx. 0.0470ml
  - See LabVIEW Program.
Appendix B

The Models

B.1 Perfect Adaptation Model

Rate kinetics of the model adapted from [4]

In general, the external signal can be represented in either an active, $W^*$, or an inactive form, $W$, whose activation is regulated by an enzyme, $E_1$, an enzyme-substrate complex, $U_1$, and various rate constants for the association, disassociation and catalysis reactions designated as $k_{c1}$, $k_{u1}$ and $k_{a1}$ respectively; inactivation is controlled by another enzyme, $E_2$ with similar factors to yield the following set of coupled differential equations relating the birth and decay of the signaling molecule and the abovementioned complexes,

\[
\begin{align*}
\frac{dW}{dt} &= -k_{c1}WE_1 + k_{u1}U_1 + k_{u2}U_2 \\ \\
\frac{dU_1}{dt} &= k_{c1}WE_1 - (k_{u1} + k_{a1})U_1 \\ \\
\frac{dW^*}{dt} &= -k_{c2}W^*E_2 + k_{u2}U_2 + k_{a1}U_1 \\ \\
\frac{dU_2}{dt} &= k_{c2}W^*E_2 - (k_{u2} + k_{a2})U_2
\end{align*}
\]

In this scheme, as when one usually derives enzyme kinetics using Michaelis-Menten models, it is customary to consider the first step to occur quickly resulting in a rapid steady state approximation for the intermediate complexes ($dU_1/dt = dU_2/dt = 0$). Thus I can rewrite the time rate in change of $W^*$ as that given above. From the perspective of *dictyostelium*, I have assumed that the signaling molecule is directly proportional to the concentrations of the
activator \( (A^*) \) and inactivator \( (I^*) \) which function as \( E_1 \) and \( E_2 \) respectively and can thus be equated as an indicator of the response element, \( R^* \). Furthermore, since these enzymes do not appear to reach saturation levels within the cell, a valid statement can be made to relate these enzymes to the “total rather than free concentrations,” resulting in a new expression pertaining to the response rate,

\[
\frac{dR^*}{dt} = -k_{-R} IR^* + k_R AR
\]  

(B.3)

I should at this point clarify that while \( W \), or similarly the response, occurs in both an activated and inactivated state, so do the related enzymes whose total concentrations are given by \( A_{tot} \) and \( I_{tot} \).

\[
\frac{dA}{dt} = -k_{-A} A + k'_A S (A_{tot} - A) \quad \text{(B.4a)}
\]

\[
\frac{dI}{dt} = -k_{-I} I + k'_I S (I_{tot} - I) \quad \text{(B.4b)}
\]

While the measured ratio of the reaction rate and Michaelis-Menten constant demand that the occurrence of inactivating \( A \) and \( I \) be more likely, a consequence is that there exists significantly more substrate for \( S \), supported by research showing a dense distribution of G-protein receptors around the cell, than the concentrations of these two enzymes. Since the external signal dictates the formation of activated \( A \) and \( I \), \( (A_{tot} \gg A \) and \( I_{tot} \gg I) \), the above equations can be easily transformed to,

\[
\frac{dA}{dt} = -k_{-A} A + k_A S \quad \text{(B.5a)}
\]

\[
\frac{dI}{dt} = -k_{-I} I + k'_I S \quad \text{(B.5b)}
\]

where \( k_A = k'_A A_{tot} \) and \( k_I = k'_I I_{tot} \) and I will now refer to the active enzymes as simply \( A \) and \( I \) (without *).
Initial pulses simulation

Results of the simulation for a 3 second cAMP pulse every 15 seconds by means of code, perfunc2.m (D.1):

![Graphs showing normalized responses to square pulses.](image)

Figure B.3: Predicted normalized responses to square pulses.

While one rightfully observes the response to adapt before the end of the pulse, I did not expect this negative output relative to the initial normalized value. I attempted to rationalize this result by explaining that the presence of increased amounts of inhibitor may suppress the basal levels of PIP$_3$ but this is not experimentally justified. To account for this (with some fortunate parameter value changes) I was able to obtain better (positive response) results, plotting the absolute activity, which should still reflect the desired dynamic of the response, with initial concentrations of \{a, I, r\} equal to 0:
Although this is obviously incorrect, I believe that I am making significant progress since the signal adapts by returning to its pre-stimulus state, the response is positive and the magnitude of its first peak is greater than the subsequent ones before relaxing into steady state equilibrium as reported in lab.

### B.2 Traveling-Wave Chemotaxis Model

**Equations of the model adapted from [9]**

A simplified model designed for producing rectified motion only when adaptation is incorporated must meet certain criteria as described by Goldstein. Primarily, there is a single characteristic adaptation time which should also yield the greatest net positive flux when this value is of same order as the wave’s period. Additionally, a response function should decrease with concentration and that the calculation of this chemotactic flux can be performed by analyzing the area confined by plots of the response and concentration vectors as
a function of time [9]. Equations (4a) and (4b) are derived from the following series of mathematical formulisms which considers a one-dimensional array of cells, along x(t), capable of responding to simulated chemoattractant waves with the above-mentioned parameters and whose motions due to gradients of cAMP can be inferred from 'overdamped dynamics,'

\[ \frac{dx}{dt} = r \frac{d}{dx} c(x + vt) \]  \hspace{1cm} (B.6)

\[ \tau \frac{dr}{dt} = f(c) - r \]  \hspace{1cm} (B.7)

Determination of the chemotactic flux, averaging over one wave period is,

\[ J = \langle J \rangle / \rho v = \langle [1 + r (dc/dz)/v]^{-1} \rangle^{-1} - 1 \]  \hspace{1cm} (B.8)

I also calculated S, and if we are to assume that the peak concentration for normal chemoattractant waves is 1µM and that if f0 is 1, it should therefore be an order of magnitude greater than his chosen values, specifically 3.59 for v = 500µm/min and T = 7min. As such, I attempted to re-run these simulations but the system apparently breaks down for S > 2.65 under any possible value of Ω (I have manually searched over a wide range of Ω and could not find any distinct region that maintains oscillatory behavior at this time). This is nicely illustrated by figure B.5 where phase portraits for S = 2.65 and S = 2.7 reveal oscillatory behavior for certain regimes and relaxations to a steady state equilibrium point, seen as a constant concentration and response, respectively. The periodicity of the chemoattractant waves and response function is lost since both are intrinsically dependent on the dimensionless coordinate, Z.

Figure B.5: Phase Portraits at high S.
Simulation of fitted model for $F(C)$

In lieu of altering the entire model which may ultimately be required, I chose to redefine $F(C)$ as first a gaussian exponential fitted to the data collected by Van Haastert, see figure 3 of [18] whereby the normalized response of cells exposed to varying concentrations of cAMP appears as a sigmoid, saturating at roughly $1\mu$M. While this runs contrary to how we have established an adaptive model for chemotaxis, it attempts to correct the response function. My simulations, figure B.6 for $S = 3.59$, selecting $f_0 = 1$, elicits the desired effect: rectified motion around $\Omega = 1.0$ quite apparent from the velocity data. The shape, albeit smaller in scale, of the response trajectories versus concentration are analagous for $F(C) = 1 - C$ with oscillatory behavior for all $\Omega$. The following graphs were made by employing code, golds1run.m (D.2), which contains the data used for creating the $F(C)$ fit. Figure B.6(d) used a modified code which was simulated over a long series of time steps and values of $\Omega$. 
(a) Chemotactic response with accompanying concentration signal at varying frequencies of $\Omega$.

(b) Phase portrait.

(c) Chemotactic velocities at varying frequencies of $\Omega$ for $S = 3.59$.

(d) Flux as a function of relaxation time ($\Omega$) for $S = 3.59$.

Figure B.6: Responses for $S = 3.59$ using fitted model for $F(C)$. 
Appendix C

LabVIEW GUI Code

Figure C.1: Block code written in LabVIEW for electronically controlling the peristaltic pump and the actuator by choosing on and off wait times.
Appendix D

MATLAB Codes

D.1 Perfect Adaptation Codes

perfrun.m

```matlab
%Michael C. DeSantis
clear all

% Time span
tspan = linspace(0,50,5000);
a0 = 1;
i0 = 1;
r0 = 1/101;

[T,Y] = ode23(@perf,tspan,[a0 i0 r0]);
plot(T,Y(:,3)/r0,'Color','b','MarkerSize',0.001,'LineWidth',0.001)
xlabel('Normalized Time')
ylabel('Normalized Activity')
```

perf.m

```matlab
function dy = perf(t,y)
%Michael C. DeSantis

alpha = 0.1;
beta = 100.0;
s = 1.1;

assignin('base','alpha',alpha);
assignin('base','beta',beta);
assignin('base','s',s);

dy = zeros(3,1);
```
\begin{verbatim}
13  dy(1) = -(y(1) - s);
14  dy(2) = -alpha*(y(2) - s);
15  dy(3) = -beta*y(2)*y(3) + y(1)*(1 - y(3));
\end{verbatim}

**perfrun2.m**

```matlab
1 %Michael C. DeSantis
2 clear all
3
4 t0 = 0;
5 tf = 50;
6 \Delta t = 0.01;
7 steps = tf/\Delta t;
8 tspan = linspace(t0, tf, steps);
9 a0 = 0;
10 i0 = 0;
11 r0 = 0;
12 alpha = 0.1;
13 beta = 100.0;
14
15 freq = 15;
16 pulse_t = 3;
17
18 for t = 1:length(tspan)
19    for i = 1:(tf/freq)+1
20       if (tspan(1,t) \geq freq*(i-1)) & (tspan(1,t) < (freq*(i-1) + pulse_t))
21          B(i,t) = 0.1;
22       else
23          B(i,t) = 0.0;
24       end
25    end
26  \Delta s(1,t) = sum(B(:,t));
27  clear B;
28 end
29
30 span = tspan;
31 n = \Delta s;
32
33 [T,Y] = ode23(@perf2, tspan, [a0 i0 r0], [], alpha, beta, span, n);
34 plot(T,Y(:,3)/r0)
35 %plot(T,Y(:,1)/a0,T,Y(:,2)/i0,T,Y(:,3)/r0)
36 plot(T,Y(:,1), T,Y(:,2), T,Y(:,3))
37 xlabel('Normalized Time')
38 ylabel('Normalized Activity')
39 legend('Activator', 'Inhibitor', 'Response')
```

**perf2.m**

```matlab
1 [T,Y] = ode23(@perf2, tspan, [a0 i0 r0], [], alpha, beta, span, n);
2 %plot(T,Y(:,3)/r0)
3 %plot(T,Y(:,1)/a0,T,Y(:,2)/i0,T,Y(:,3)/r0)
4 plot(T,Y(:,1), T,Y(:,2), T,Y(:,3))
5 xlabel('Normalized Time')
6 ylabel('Normalized Activity')
7 legend('Activator', 'Inhibitor', 'Response')
```
function dy = perf2(t,y,alpha,beta,span,n)

%Michael C. DeSantis

s = interp1(span,n,t);

dy(1) = -(y(1) - s);

dy(2) = -alpha*(y(2) - s);

dy(3) = -beta*y(2)*y(3) + y(1)*(1 - y(3));

dy = dy(:);

perfrun4.m

%Michael DeSantis

clear all

t0 = 0;
tf = 300;
\Delta t = 0.1;
steps = tf/\Delta t;
tspan = linspace(t0,tf,steps);
a0 = 0;
i0 = 0;
r0 = 0;
al = 0.02;
b = 100.0;

[T,Y] = ode23(@perf4,tspan,[a0 i0 r0],[],alpha,beta);

%plot(T,Y(:,3)/r0)
%plot(T,Y(:,1)/a0,T,Y(:,2)/i0,T,Y(:,3)/r0)
%xlabel('Normalized Time')
%ylabel('Normalized Activity')
%legend('Activator','Inhibitor','Response')

for i = 1:steps
    if sin(w0*i*\Delta t) ≥ threshold
        s(i,1) = 0.1;
    else
        s(i,1) = 0;
    end
end

plot(T,Y(:,1),T,Y(:,2),T,Y(:,3))
xlabel('time [s]')
ylabel('Activity')
legend('Activator','Inhibitor','Response')
figure, plot(T,Y(:,3),T,s)
perf4.m

```matlab
function dy = perf4(t,y,alpha,beta)
%Michael DeSantis

w0 = (2*pi)/15;
assignin('base','w0',w0);
threshold = 0.95;
assignin('base','threshold',threshold);

if sin(w0*t) ≥ threshold
    dy(1) = -(y(1) - 0.1);
    dy(2) = -alpha*(y(2) - 0.1);
    dy(3) = -beta*y(2)*y(3) + y(1)*(1 - y(3));
else
    dy(1) = -(y(1) - 0);
    dy(2) = -alpha*(y(2) - 0);
    dy(3) = -beta*y(2)*y(3) + y(1)*(1 - y(3));
end

%dy(1) = -(y(1) - (1 + abs(sin(w0*t))));
%dy(2) = -alpha*(y(2) - (1 + abs(sin(w0*t))));
%dy(3) = -beta*y(2)*y(3) + y(1)*(1 - y(3));
dy = dy(:);
```

perfrun6.m

```matlab
%Michael C. DeSantis
clear all

t0 = 0.1;
 tf = 46.2;
 Δt = 0.05;
 steps = tf/Δt;
 tspan = linspace(t0,tf,steps);
 a0 = 0.25;
i0 = 1;
r0 = 0.548;
 alpha = 0.055;
beta = 0.18;
 span = tspan;

q = [pulse data]';
q = q/1000;
n = repmat(q,1,length(tspan)/length(q))';
```
perf6.m

1 function dy = perf6(t,y,alpha,beta,span,n)
2 %Michael C. DeSantis
3 s = interp1(span,n,t);
4 dy(1) = -(y(1) - s);
5 dy(2) = -alpha*(y(2) - s);
6 dy(3) = -beta*y(2)*y(3) + y(1)*(1 - y(3));
7 dy = dy(:);

D.2 Traveling-Wave Chemotaxis
gold1run.m

1 %Michael C. DeSantis
2 clear all
3 tspan = linspace(0,20,2000);
4 omegavar = [0.1,1,10];
5 initialvar = [1,0.97,0.78];
6 for i = 1:length(omegavar)
7     omega = omegavar(i);
8     initial = initialvar(i);
9     [T(:,i),Y(:,:,i)] = ode23(@gold1,tspan,[-0.5,initial]);
10     C(:,i) = exp(beta*(sin(Y(:,1,i)) - 1.0));
11     figure(1)
12     subplot(3,1,i),plot(T(:,i),Y(:,2,i),'r',T(:,i),C(:,i),'-.b')
13     axis([0,max(tspan),0,1])
14     figure(2)
15     plot(C(:,i),Y(:,2,i))
```
hold all
for j = 1:length(Y(:,1,i))
    v(j,i) = Y(j,2,i)'.*beta*cos(Y(j,1,i))'*C(j,i);
end
flux(:,i) = (mean((1 + S*v(:,i)).^(-1)))^(-1) - 1;
end
```

gold1.m

```
function dy = gold1(t,y,beta,S,omega)
    beta = 3.0;
    S = 0.2;
    omega = evalin('base','omega');
    assignin('base','beta',beta);
    assignin('base','S',S);
    dy = zeros(2,1);
    dy(1) = S*y(2)*beta*cos(y(1))*exp(beta*(sin(y(1)) - 1.0)) + 1;
    dy(2) = ((1 - exp(beta*(sin(y(1)) - 1.0))) - y(2))/omega;
```

golds1run.m

```
clear all

tspan = linspace(0,20,2000);
omegavar = [0.1,1,10];
initialvar = [0.06,0.055,0.028];
for i = 1:length(omegavar)
    omega = omegavar(i);
    initial = initialvar(i);
    [T(:,i),Y(:,:,i)] = ode23(@golds1,tspan,[-0.5,initial]);
    C(:,i) = exp(beta*(sin(Y(:,1,i)) - 1.0));
    figure(1)
    subplot(length(omegavar),1,i),
    plot(T(:,i),Y(:,2,i),'r',T(:,i),C(:,i),'-.b')
    axis([0,max(tspan),0,1])
    figure(2)
    plot(C(:,i),Y(:,2,i))
    hold all
    for j = 1:length(Y(:,1,i))
        v(j,i) = Y(j,2,i)'.*beta*cos(Y(j,1,i))'*C(j,i);
    end
```
flux(:,i) = (mean((1 + S * v(:,i)).^(-1)))^(-1) - 1;
end

for i = 1:length(C(:,1))
    sinu(i,1) = (C(i,1)^4)/((10^(-2))+(C(i,1)^4));
end

figure(3)
plot(T,C,'k',T,g,'r',T,sinu,'b')
legend('conc.','current F(C)','a sinusoidal F(C)')

golds1.m

function dy = golds1(t,y,beta,S,omega)
%Michael C. DeSantis
beta = 3.0;
S = 3.59;
omega = evalin('base', 'omega');
assignin('base','beta',beta);
assignin('base','S',S);

C(y(1)) = exp(beta*(sin(y(1)) - 1.0));
%F(C) = 1 - C;
%conc = [0,10^-7,5*10^-7,10^-6,5*10^-6,10^-5];
%resp1 = [0,0.28,0.6,0.73,0.83,0.9];
%resp2 = [0,0.4,0.7,0.92,1,1];
%logc = log10(conc);
%F(x) = fittedmodel1(x) = a1*exp(-((x-b1)/c1)^2)
  a1 = 0.95;
b1 = -5.2;
c1 = 1.7;
assignin('base','a1',a1);
assignin('base','b1',b1);
assignin('base','c1',c1);

dy = zeros(2,1);
dy(1) = S*y(2)*beta*cos(y(1))*exp(beta*(sin(y(1)) - 1.0)) + 1;
dy(2) = ((a1*exp(-((log10(exp(beta*(sin(y(1)) - 1.0)))-b1)/c1)^2)) - y(2))/omega;
dy = dy(:);

d.3 Diffusion and Fitting Codes
diff-drift.m
%Azadeh Samadani and Michael C. DeSantis

clear all

timevec = 0:0.38:60;
tpulse = 5;
ii= 100;
int = [];

for vid = [5.32]
    v = vid*0.01;
    disp(['Working on v = ' num2str(v)]);
    for tt = 1:length(timevec)
        currenttime = timevec(tt);
        if currenttime < 0
            int(tt,vid+0.68) = 0;
        elseif currenttime ≤ tpulse
            int(tt,vid+0.68) = quad(@f,0.01,currenttime,[],[],v);
        else
            int(tt,vid+0.68) = quad(@f,currenttime−tpulse,currenttime,[],[],v);
        end
    end
end

norm_int = int*ii./((4*pi*D).ˆ(3/2));
figure(1);
plot(timevec,norm_int(:,end),'r');
end

function y = f(tau, v)
%Azadeh Samadani and Michael C. DeSantis

v = evalin('base','v');
Df = 4.9*10^-6;
h = 0.0146;
Pe = v*h/Df;
Deff = (1/210)*Pe^2;
assignin('base','Df',Df);
assignin('base','Deff',Deff);
D = Df*Deff;
r = 2;

y = exp( -(r−v*tau).^2/(4*D*tau) )./(tau.^((3/2));
assignin('base','D',D)
assignin('base','r',r)
diff-drift-fit.m

1 %Michael C. DeSantis
2 clear all
3 warning off
4 data = xlsread('2.15.07 - Syringe Pump - 60uM - 2.3cm.xls',2,'R2:S159');
5 int1 = data(:,2);
6 diff_drift
7 hold all
8 int1_norm = (int1 - mean(int1(1:20)))*
9 (max(max(norm_int)))/(max(int1-mean(int1(1:20))));
10 plot(timevec,int1_norm, 'bo')
11 xlabel('time [s]')
12 ylabel('Average Intensity [au]')
13 legend('theoretical fit','experimental data','Location',2)

D.4 Image Processing Codes

vid4.m

1 function varargout = vid4(varargin)
2 % VID4 M-file for vid4.fig
3 % VID4, by itself, creates a new VID4 or raises the existing
4 % singleton*.
5 %
6 % H = VID4 returns the handle to a new VID4 or the handle to
7 % the existing singleton*.
8 %
9 % VID4('CALLBACK',hObject,eventData,handles,...) calls the local
10 % function named CALLBACK in VID4.M with the given input arguments.
11 %
12 % VID4('Property','Value',...) creates a new VID4 or raises the
13 % existing singleton*. Starting from the left, property value pairs
14 % are applied to the GUI before vid4_OpeningFunction gets called. An
15 % unrecognized property name or invalid value makes property appli-
16 % cation stop. All inputs are passed to vid4_OpeningFcn via varargin.
17 %
18 % *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
19 % instance to run (singleton)".
20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help vid4

% Last Modified by GUIDE v2.5 28-Oct-2006 21:47:50
% Michael C. DeSantis
% 3-3-2007

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @vid4_OpeningFcn, ...
'gui_OutputFcn', @vid4_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% —— Executes just before vid4 is made visible.
function vid4_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to vid4 (see VARARGIN)

% Choose default command line output for vid4
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes vid4 wait for user response (see UIRESUME)
% hObject    handle to figure

% —— Outputs from this function are returned to the command line.
function varargout = vid4_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
73 varargout{1} = handles.output;

74 % —— Executes during object creation, after setting all properties.
75 function thresholdtext_CreateFcn(hObject, eventdata, handles)
76 set(hObject,'String',0);
77
78 if ispc && isequal(get(hObject,'BackgroundColor'),
79     get(0,'defaultUicontrolBackgroundColor'))
80     set(hObject,'BackgroundColor','white');
81 end

82 function thresholdslider_CreateFcn(hObject, eventdata, handles)
83 set(hObject,'min',0.0);
84 set(hObject,'max',0.2);
85 set(hObject,'SliderStep',[0.005 0.05]);
86 set(hObject,'Value',0);
87
88 if isequal(get(hObject,'BackgroundColor'),
89     get(0,'defaultUicontrolBackgroundColor'))
90     set(hObject,'BackgroundColor',[.9 .9 .9]);
91 end

92 function framenum_CreateFcn(hObject, eventdata, handles)
93 set(hObject,'String',1);
94
95 if ispc && isequal(get(hObject,'BackgroundColor'),
96     get(0,'defaultUicontrolBackgroundColor'))
97     set(hObject,'BackgroundColor','white');
98 end

99 function framenumslider_CreateFcn(hObject, eventdata, handles)
100 warning off;
101 dircontents = dir(cd);
102 set(hObject,'min',1);
103 set(hObject,'max',(length(dircontents)-2));
104 set(hObject,'SliderStep',
105     [1/(length(dircontents)-2) 1/(length(dircontents)-2)]);
106 set(hObject,'Value',1);
107 imshow(imread(dircontents(3).name),[]);
108
109 if isequal(get(hObject,'BackgroundColor'),
110     get(0,'defaultUicontrolBackgroundColor'))
111     set(hObject,'BackgroundColor',[.9 .9 .9]);
112 end

113 % executes on change of text in threshold textbox
function thresholdtext_Callback(hObject, eventdata, handles)
    threshold = str2double(get(hObject, 'String'));

% Check that the entered value falls within the allowable range
if isempty(threshold) | (threshold < 0.0) | (threshold > 0.2),
    % Revert to last value, as indicated by the slider
    OldVal = get(handles.thresholdslider, 'Value');
    set(hObject, 'String', OldVal);
else
    % Set the value of the slider to the new value
    set(handles.thresholdslider, 'Value', threshold);
end

% Apply threshold value
    dircontents = dir(cd);
    framenum = round(get(handles.framenumslider, 'Value'));
    fnamel = dircontents((framenum)+2).name;
    A = imread(fnamel);
    bw = im2bw(A, threshold);
    axes(handles.axes1);
    imshow(bw)

% --- Executes on slider movement.
function thresholdslider_Callback(hObject, eventdata, handles)

% Get the new value from the slider
threshold = get(hObject, 'Value');

% Reduce precision of threshold to display easily
threshold = round(threshold * 1000)/1000;

% Set the value of the threshold textbox to the new value set by slider
set(handles.thresholdtext, 'String', threshold);

dircontents = dir(cd);
framenum = round(get(handles.framenumslider, 'Value'));
fnamel = dircontents((framenum)+2).name;
A = imread(fnamel);
bw = im2bw(A, threshold);
axes(handles.axes1);
imshow(bw)

% executes on change of text in framenum textbox
function framenum_Callback(hObject, eventdata, handles)
    framenum = str2double(get(hObject, 'String'));
    framenum = round(framenum);

% Check that the entered value falls within the allowable range
dircontents = dir(cd);
if isempty(framenum) | (framenum < 1) | (framenum > length(dircontents)-2),
    % Revert to last value, as indicated by the slider
    OldVal = get(handles.framenumslider,'Value');
    OldVal = round(OldVal);
    set(hObject,'String',OldVal);
else
    % Set the value of the slider to the new value
    set(hObject,'Value',framenum);
end
dircontents = dir(cd);
framenum = round(get(handles.framenumslider,'Value'));
fname1 = dircontents((framenum)+2).name;
A = imread(fname1);
axes(handles.axes1);
imshow(A,[])

% —— Executes on slider movement.
function framenumslider_Callback(hObject, eventdata, handles)
    % Get the new value from the slider
    framenum = get(hObject,'Value');
    framenum = round(framenum);
    % Set the value of the framenum textbox to the new value set by the slider
    set(handles.framenum,'String',framenum);
dircontents = dir(cd);
fname1 = dircontents((framenum)+2).name;
A = imread(fname1);
axes(handles.axes1);
imshow(A,[])

% —— Executes on button press in perim.
function perim_Callback(hObject, eventdata, handles)
dircontents = dir(cd);
fname1 = dircontents((framenum)+2).name;
A = imread(fname1);
axes(handles.axes1);
imshow(A,[])

threshold = get(handles.thresholdslider,'Value');
bw = im2bw(A,threshold);
perim = bwperim(bw);
bw_filled = imfill(perim,'holes');
bw_filled = bwareaopen(bw_filled,30);
bw_filled = imclearborder(bw_filled, 4);

[B,L] = bwboundaries(bw_filled,'noholes');
axes(handles.axes1);
imshow(A,[
hold on
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
end

% --- Executes on button press in boundary.
function boundary_Callback(hObject, eventdata, handles)
    dircontents = dir(cd);
    framenum = round(get(handles.framenumslider,'Value'));
    fnamel = dircontents((framenum)+2).name;
    A = imread(fnamel);
    [junk threshold] = edge(A,'canny');
    fudgeFactor = .5;
    bw = edge(A,'canny',threshold*fudgeFactor);
    se90 = strel('line',3,90);
    se0 = strel('line',3,0);
    bw_dil = imdilate(bw,[se90 se0]);
    bw_filled = imfill(bw_dil,'holes');
    bw_filled = bwareaopen(bw_filled,30);
    bw_filled = imclearborder(bw_filled, 4);
    seD = strel('diamond',1);
    bw_erode = imerode(bw_filled,seD);
    bw1 = imerode(bw_erode,seD);
    [B,L,N] = bwboundaries(bw1,'noholes');
axes(handles.axes1);
imshow(A,[
hold on
for k = 1:N
    boundary = B{k};
    plot(boundary(:,2),boundary(:,1),'g','LineWidth',2)
end

% --- Executes on button press in erosion.
function erosion_Callback(hObject, eventdata, handles)
dircontents = dir(cd);
framenum = round(get(handles.framenumslider,'Value'));
fname1 = dircontents((framenum)+2).name;
A = imread(fname1);

[junk threshold] = edge(A,'canny');
fudgeFactor = .5;
bw = edge(A,'canny',threshold*fudgeFactor);

se90 = strel('line',3,90);
se0 = strel('line',3,0);

bw_dil = imdilate(bw,[se90 se0]);
bw_filled = imfill(bw_dil,'holes');
bw_filled = bwareaopen(bw_filled,30);
bw_filled = imclearborder(bw_filled, 4);

seD = strel('diamond',1);
bw_erode = imerode(bw_filled,seD);
bw1 = imerode(bw_erode,seD);
bw2 = imerode(bw1,strel('diamond',10));

[B,L,N] = bwboundaries(bw1,'noholes');
[C,M,N] = bwboundaries(bw2,'noholes');
Z = imsubtract(L,M);

assignin('base','A',A);
assignin('base','B',B);
assignin('base','L',L);
assignin('base','C',C);
assignin('base','M',M);
assignin('base','N',N);
assignin('base','Z',Z);

axes(handles.axes1);
imshow(A,[])
hold on
for k = 1:N
    boundary1 = B{k};
    boundary2 = C{k};
    plot(boundary1(:,2),boundary1(:,1),'g','LineWidth',2)
    plot(boundary2(:,2),boundary2(:,1),'r','LineWidth',2)
    avg(k,:) = mean2(A(Z==k));
end

assignin('base','avg',avg);

% —— Executes on button press in send.
function send_Callback(hObject, eventdata, handles)
threshold = get(handles.thresholdslider,'Value');
assignin('base','threshold',threshold);
close(vid4);

analyze.m

function analyze(input,processmode)
    %Michael C. DeSantis
    %3-7-2007
    if(nargin == 0 || isempty(input) || input == 0)
        dirname = uigetdir(['C:\Program Files\MATLAB704\work\TIF Images\'],
            'select an image folder');
        if(dirname == 0)
            return;
        end;
        addpath(dirname)
        dircontents = dir(dirname);
        fps = 2;
    
    for n=3:length(dircontents)
        fname(:,:,n-2) = dircontents(n).name;
        A(:,:,n-2) = imread(fname(:,:,n-2));
        assignin('base','A',A);
        assignin('base','dircontents',dircontents);
        assignin('base','fname',fname);
        assignin('base','fps',fps);
    end

    if (nargin<2 || isempty(processmode) || processmode == 0)
            polar2
    else
            polar3
    end

    elseif(input == 1)
        [fname,dirname] = uigetfile('*.AVI','select an AVI movie');
        if(fname == 0)
            return;
        end;
        fileinfo = aviinfo(fname);
        fps = getfield(fileinfo,'FramesPerSecond');
    addpath(dirname);
    mov = aviread(fname);
    for j = 1:getfield(fileinfo,'NumFrames')
        A(:,:,j) = getfield(mov,{1,j},'cdata');
43  end
44  assignin('base','A',A);
45  assignin('base','fname',fname);
46  assignin('base','fileinfo',fileinfo);
47  assignin('base','fps',fps);
48  if (nargin<2 || isempty(processmode) || processmode == 0)
49    polar2
50  else
51    polar3
52  end
53
54  else
55    [fname,dirname] = uigetfile('*.tif','select an image file',
56      ('C:\Program Files\MATLAB704\work\TIF Images\'));
57    if(fname == 0)
58      return;
59    end;
60    addpath(dirname)
61    fps = 2;
62
63    A = imread(fname);
64    assignin('base','A',A);
65    assignin('base','fname',fname);
66    assignin('base','fps',fps);
67
68  if (nargin<2 || isempty(processmode) || processmode == 0)
69    polar2
70  else
71    polar3
72  end
73
74  end

polar2.m

1  %Michael C. DeSantis
2  %3-7-2007
3
4  for j = 1:size(A,3)
5    [junk threshold] = edge(A(:,:,j), 'canny');
6    fudgeFactor = .5;
7    bw(:,:,j) = edge(A(:,:,j), 'canny', threshold*fudgeFactor);
8    se90 = strel('line',3,90);
9    se0 = strel('line',3,0);
bw_dil(:,:,j) = imdilate(bw(:,:,j),[se90 se0]);
bw_filled(:,:,j) = imfill(bw_dil(:,:,j),'holes');
bw_filled(:,:,j) = bwareaopen(bw_filled(:,:,j),30);
bw_filled(:,:,j) = imclearborder(bw_filled(:,:,j), 4);

seD = strel('diamond',1);
bw_erode(:,:,j) = imerode(bw_filled(:,:,j),seD);
bw1(:,:,j) = imerode(bw_erode(:,:,j),seD);
bw2(:,:,j) = imerode(bw1(:,:,j),strel('diamond',10));

[B(:,:,j),L(:,:,j),N] = bwboundaries(bw1(:,:,j),'noholes');
[C(:,:,j),M(:,:,j),N] = bwboundaries(bw2(:,:,j),'noholes');
Z(:,:,j) = imsubtract(L(:,:,j),M(:,:,j));

[cm(:,j)] = regionprops(Z(:,:,j),'Centroid');
centroids(:,:,j) = cat(1,cm(:,j).Centroid);
centroid(:,:,j) = round(centroids(:,:,j));

for k = 1:N
    [y(:,k),x(:,k)] = find(Z(:,:,j) == k);
    xnew(:,k) = x(:,k) - centroid(k,1,j);
    ynew(:,k) = y(:,k) + centroid(k,2,j);
    [theta(:,k),rho(:,k)] = cart2pol(xnew(:,k),ynew(:,k));
    for i = 1:length(theta(:,k))
        if theta(i,k) < 0
            theta(i,k) = 2*pi + theta(i,k);
        else
            theta(i,k) = theta(i,k);
        end
    end

angle = 10;
regions = 360/angle;

for n = 0:regions - 1
    J(:,k) = find((theta(:,k) <= (n+1)*angle*pi/180) & (theta(:,k) >= n*angle*pi/180));
    xnew1(:,k) = x(J(:,k),k);
    ynew1(:,k) = y(J(:,k),k);
    A1 = A(ynew1(:,k),xnew1(:,k),j);
    avg_polar(n+1,k,j) = mean2(A1);
    D = A(:,:,j);
    avg(1,k,j) = mean2(D(Z(:,:,j) == k));
    clear xnew1 ynew1 J A1 D
end

angle_regions = [angle/2:angle:360]';
assignin('base','centroid',centroid);
assignin('base','avg_polar',avg_polar);
assignin('base','angle',angle);
assignin('base','angle_regions',angle_regions);
assignin('base','avg',avg);
assignin('base','B',B);
assignin('base','L',L);
assignin('base','C',C);
assignin('base','M',M);
assignin('base','N',N);
assignin('base','Z',Z);

warning off MATLAB:xlswrite:AddSheet
color = get(gca,'ColorOrder');
for j = 1:size(A,3)
    for k = 1:N
        %xlswrite('data1', angle_regions, j, 'A3');
        %xlswrite('data1', avg(:,:,j), j, 'B2');
        %xlswrite('data1', avg_polar(:,:,j), j, 'B3');
        figure(1)
        [xcomp(:,j),ycomp(:,k,j)] = 
        pol2cart(angle_regions*pi/180,avg_polar(:,:,j));
        subplot(5,4,j),
        w = compass(xcomp(:,j),ycomp(:,k,j));
        set(w,'Color',[color(k,:)]);
        hold all
        figure(2)
        w = compass(xcomp(:,j),ycomp(:,k,j));
        set(w,'Color',[color(k,:)]);
        if k == N
            hold
        else
            hold all
        end
        F(j) = getframe(gcf);
    end
assignin('base','F',F);
assignin('base','xcomp',xcomp);
assignin('base','ycomp',ycomp);
figure(1)
%print -djpeg compass1
%movie(F,1,fps)
%movie2avi(F,'new1','compression','None','fps',fps)
for j = 1:size(A,3)
    figure(3)
    subplot(5,4,j),plot(angle_regions,avg_polar(:,:,j))
    axis([0,400,min(min(min(avg_polar)))-5,max(max(max(avg_polar)))+5])
    xlabel('angle')
    ylabel('intensity [au]')
```matlab
figure(4)
plot(angle_regions,avg_polar(:,:,j),'-o','LineWidth',2,'MarkerSize',5)
axis([0,360,min(min(min(avg_polar)))-5,max(max(max(avg_polar)))+5])
xlabel('angle')
ylabel('intensity [au]')
G(j) = getframe(gcf);
end
assignin('base','G',G);
figure(3)
% print -djpeg plot1
% movie(G,1,fps)
% movie2avi(G,'plot1','compression','None','fps',fps)

excerpt from polar3.m

```