2011

Optimization and Analysis of a Model of the Mammalian Circadian Clock

Andrew M. Cox
Colby College

Optimization and Analysis of a Model of the Mammalian Circadian Clock

An Honors Thesis
by Andrew M. Cox, 2011J

Advisor: Stephanie R. Taylor

Department of Computer Science

Colby College, Fall 2010

This Honors Thesis is approved by:

_____

Stephanie R. Taylor
Clare Boothe Luce Assistant Professor of Computer Science
Advisor

_____

Bruce A. Maxwell
Associate Professor of Computer Science
Department Chair

_____

Andrew M. Cox
Honors Candidate
Author

Table of Contents

Abstract

Circadian rhythms are found in plants, animals, fungi, and bacteria and are responsible for the regulation of many biological, physiological, behavioral, and metabolic activities. In mammals, the "master" clock is embedded in the suprachiasmatic nuclei (SCN) of the anterior hypothalamus. It is composed of thousands of cells signaling each other to synchronize and produce a unified rhythm. It is hypothesized that this communication enables the clock to rescue rhythms in gene knockout experiments that destroy oscillations at the single cell level. Henry Mirsky developed a model of this gene regulatory network within a single cell in 2009, but the published model is unable to accurately predict some knockout phenotypes in tissue. We determined that binary 'arrhythmic' and 'rhythmic' classifications were unsuitable for describing cellular behavior, so we utilized a novel 'damped' classification to describe cellular phenotype. We then used a genetic algorithm optimization technique to fit the model's parameters to better mimic damped Cry1 single-cell knockout behavior. This optimization produced a parameter set that is able to correctly predict all single-cell knockout phenotypes as well as accurately predict the oscillatory phenotypes for two additional SCN tissue knockouts not previously demonstrated. The rescue of oscillations was confirmed by simulating SCN tissue—a process that involves reproduction of intercellular signaling and normal variation of unique cellular parameter sets. Further work seeks to explain the causes for the ability of this new parameter set to rescue oscillations in tissue simulations where the published parameter set was unsuccessful.

Acknowledgements

First and foremost, I would like to thank my wonderful advisor, Stephanie Taylor. She has provided guidance and insight on this project at every stage, and she has offered me the unique opportunity to be involved in asking questions to which no one yet knows an answer, let alone how to discover one. I would also like to thank Hannah Coulson (Class 2010) for being a member of the research group prior to her graduation. Hannah provided some much-needed motivation at times and was a wonderful collaborator. She also took the lead on a ton of data crunching that paved the way for much of this analysis.

Finally, and most importantly, I would like to thank my family for giving me the opportunity to return to Colby for a final semester to complete this honors research. I would not be the person I am today without their unwavering support.

I. Introduction

Circadian rhythms are found in plants, animals, fungi, and bacteria (Goldbeter 2002, Wager-Smith 2000). In mammals, these rhythms are controlled by the auto-regulatory gene network endogenous to the cells of the suprachiasmatic nuclei (SCN) in the anterior hypothalamus called the "master clock" (Weaver 1998). Indeed, the master clock controls peripheral clocks throughout the body that are genetically and functionally similar. The specific biology and chemistry of this gene regulatory network is not yet fully understood, however the clock is generally thought to consist of a negative primary feed-back loop and a positive ancillary loop. The negative loop consists primarily of Period/Cryptochrome (PER/CRY) dimers that are controlled through auto-inhibitory gene regulation. This negative feedback regulation induces an oscillatory protein concentration with a stable period caused by the time delay of transcription, translation, and dimerization; in this model, we do not include gene product localization. The positive loop helps to fine-tune the clock and is not required for periodicity—indeed, simpler models leave out the positive loop entirely; here, it consists of positive regulation of the core loop by Rorc, Clock (Clk), and Bmal1, and of negative regulation by Rev-erbα. (Liu 2008)

Computer modeling allows for powerful approximations of the true system through which it is possible to quickly and cheaply study the "what-ifs." Our group has been working with a model published in 2009 by Henry Mirsky, et al, (Mirsky 2009) that describes the network with 21 biological states—including mRNA and protein

concentrations—and 135 descriptive parameters—including Michaelis and Hill constants, rate constants, and other rate parameters. The model is implemented by a series of ordinary differential equations (ODEs) in MATLAB (MathWorks, Inc.) that are solved by built-in ODE-solving functions in the programming language.

When first published, the Mirsky model had been optimized to reproduce the behavior of single-cell explants of cells from the SCN, both in wild type and in gene knockouts[1]. However, it was unable to predict the tissue knockout phenotypes of the primary loop genes: Per1, Per2, and Cry1. Through unpublished work in bifurcation and rank-ordered sensitivity analysis, we determined that the published parameter set was not similar to a parameter set that would be able to predict tissue behavior. We therefore decided to perform an additional round of parameter optimization to discover a parameter set that had more predictive power. We determined that we should include some other piece of information that had not been previously included in the training or testing data sets, so we incorporated a new metric for measuring and classifying the behavior of a cell in single-cell simulations as "damped," adding a finer description to the biological data. We applied this technique to the Cry1 single-cell knockout and trained the model to this behavior.

[1] A more detailed description of the methods used to collect the biological data can be found in the paper by Liu, et al. (Liu 2007). Briefly, a luciferase knock-in system was used to allow the measurement of the bioluminescence of cells, which would be an indicator of protein concentrations (Yoo 2004, Yamazaki 2005). This luciferase gene was co-expressed with Per2, meaning that the bioluminescence patterns were a measurement only of that protein concentration, in monomer and dimer forms. Next, knockouts of *Mus musculus* variants were created with a single gene of the regulatory network knocked out. Two types of cells were analyzed: neurons from the SCN and fibroblast cells that had an endogenous clock. The data we base our analysis on are the neuronal data.

This thesis is designed to explain the process of model optimization and analysis. As such, it will focus on methods used and attempt to explain the thought processes of my advisor and myself. While I discuss our results, particularly in light of how they were able to improve the model, the major role of this piece is to allow future research students to learn from my experiences and to make immediate use of what we have learned thus far, without repeating work that gave us negative results. I assume that the reader is conversant in general principles of biochemistry, chemistry, biology, and computer modeling. Where feasible, I add brief descriptions of appropriate terms for those who are not adept in all aspects of systems biology.

i. Systems Biology: An Overview

The field of systems biology is diverse in both its scope and definitions. It is the intersection of computer science and biology, chemistry, mathematics, statistics, and myriad other fields, depending on the type of problem being tackled. Researchers in this field work in collaborative groups that bring scientists who are experts in these fields together, as the Taylor Group does with researchers from UCSB, Washington University-St. Louis, and elsewhere. Loosely, systems biology can be described as the study of system structures, dynamics, control mechanisms, and design methods (Kitano 2002). The goal of such an approach is to bring the predictive and analytical power of computer science and mathematics to help broaden the understanding of biological data; indeed, recent developments in high-throughput wet-lab[2] processing technologies have resulted in an

[2] Wet-lab techniques can be described as any laboratory technique that might generally be encountered in a biology or chemistry course. More broadly, it describes the lab techniques of the physical sciences.

enormous amount of data. These data particularly challenge computer scientists to utilize cluster analysis and other large-scale data analysis techniques to try to automate discovery of biological function (Kitano 2002). Practically, wet-lab technology has surpassed the ability of individual scientists to interpret the data.

However, understanding a system, such as the endogenous circadian clock or the metabolic pathways of *E. coli*, requires iterative refinement of a model that estimates the system. One simply does not know where to look to see the larger picture. The iterative process begins with the construction of a simple model that is an approximation of the current understanding. This model is then used to predict the behavior of the system under novel conditions. Such predictions help to show where the scientific community's understanding is lacking or where further research should be conducted. Investigation in these areas leads to the development of a more complicated model that describes the system more completely. In this way, computer modeling is able to assist in the discovery of novel gene pathways and foster deeper understanding of the biological and chemical bases of life.

As illustrated by Kitano (2002), the Taylor lab focuses on a subset of a larger framework that is the iterative refinement of a model of the mammalian circadian clock. Our work in concerned with analyzing the biological data to construct a better mathematical model, analyzing the model, and probing its predictive power. This project has spanned that entire spectrum, and future work seeks to continue in the cycle to discover *how* the model can explain the biology. As discussed in the Future Works section, we have several hypotheses as to why this might occur. However, more work needs to be performed in order to educe its potential.

II. Optimization

We look at our model as a set of constants (e.g. rates associated with certain biological processes) and state variables (e.g. the concentration of a certain protein, which changes over time). Because we are able to quantify parameters and states, we are able to develop metrics to compare one parameter set to another. By doing so, we can determine which of several parameter sets is "best" according to this metric. If we can do this while simultaneously creating new, (hopefully) better parameter sets, then we can optimize the model to fit a desired functionality. This process is called optimization. The metric that we developed to compare parameter sets is called the cost function. There are many techniques utilized that optimize a parameter set using the cost function as a basis, however the one chosen for this project is an implementation of a genetic algorithm.

i. Genetic Algorithms[3]

In systems biology, a widely used and well-regarded method of parameter optimization is a genetic algorithm (Goldberg 1989, Lee 1998). Each implementation and variation offers its own benefits and demerits, including time to convergence on a good solution and maintenance of a diverse population from which to choose parents. Indeed, the distinction between various methods can be vague, as they are based on ideas from natural selection and can be modified to be more similar than distinct. The goal is three-part: first, to represent individuals as a set of parameters; then, to start with random

[3] For a more detailed discussion of terminology and how it relates to a computer science application of genetic algorithms, see Appendix B.

individuals that evenly sample all possible diversity; and finally to mate the individuals according to some selection criteria to, in the end, get a fitter solution. Genetic algorithms are designed with two principles in mind: selection and mutation. From a set of individuals in generation *n*, two parents are chosen by some method, and parameters are uniformly sampled from the two parents to produce a single child in generation *n+1*. This child may then be subjected to some sort of minute mutation (e.g., +/- 1% in each parameter), ensuring greater diversity within the total population, which can be particularly sensitive to a phenomenon similar to genetic drift. The cost associated with that child's parameter set (i.e., the specific model described by that child) is then evaluated and stored with the child object, which is then placed in the set for generation *n+1*. In this way, children can be efficiently generated and evaluated in parallel.

We decided on a genetic algorithm that used proportional selection to choose parents to "reproduce" at each generation. Proportional selection was chosen over the several other approaches to genetic algorithm implementations because it showed faster convergence to a good answer while still maintaining adequate diversity of the parent population, in preliminary testing. These were both important aspects to consider because we wanted to ensure that our optimization did not exclude a portion of parameter space where it should not have; we were not able to estimate the topology of the parameter space. However, we also did not want the optimization to continue searching a space that could safely be excluded. In the end, we assumed that the exact method used was not paramount because any correct solution that was within the continuum of working solutions would be appropriate. However, the methodology used is interesting from the perspective of implementation of future algorithms despite its apparent lack of impact on our final result.

Our final algorithm was implemented with 100 parents and 110 children, the best 100 of which were passed on to the next generation; this ensured that all parents described a valid model. We added in a small element of another strategy that maintained an elite count of 5 individuals that automatically cloned themselves into the next generation, from the perspective of biological genetics. We also included a mutation coefficient of +5% in random parameters, adding to the general diversity of the population. In effect, we combined the best components of each strategy to ensure that we did not lose a good candidate solution and that we maintained sufficient diversity in the population without compromising the algorithm's ability to discover a good solution.

ii. The Cost Function

The cost function originally tested a model against the wild type and against Rorc and Rev-erbα knockouts. The cost function used for the published model was designed to favor models that matched period and relative state amplitudes per biological data. In our revised cost function, we added a term to include a measure of the damped behavior of a Cry1 knockout. To do this, we relied on the unpublished work of Alexis Webb, which had determined an appropriate measure of dampedness for the Mirsky model. In a binary (i.e., rhythmic or arrhythmic) world, the Cry1 knockout was classified as arrhythmic because its oscillations eventually fell below the detection limit of the luciferase system used. However, we realized that not all arrhythmic systems are created equal; some fall quickly to undetectable levels and some progress more slowly to the same steady state. We determined that including this new measure of Cry1 knockout damped behavior in the cost

function might allow the model to synchronize in tissue simulations. It was our hypothesis that this seemingly minor discrepancy belied a larger incongruence in the parameter set. It is important to note that we did not simulate the tissue directly in our optimization cost function because that calculation can take hours to complete whereas a single-cell simulation would take seconds; the time to compute the cost function had to be minimized because it is called several hundred times throughout the optimization process.

This damped measure could easily be incorporated directly from the code provided by Webb that computed the peak concentration of the first complete cycle and then computed the number of cycles before the peak concentration of a cycle fell below 20% of the nominal. If the number of cycles was between 4 and 12, then the oscillator was classified as damped. The "ideal" number of cycles was assumed to be 8, a seemingly arbitrary value, in order to ease the calculation of the cost. We determined that we could easily use this damping number in a bounded cosine function to smoothly modulate the cost:

$$C_{damp} = \lambda \frac{\cos\left[\frac{\Pi}{4} \cdot (numAmp - 4) + 1\right]}{2}$$

where λ is the scaling factor for the cost of damping. We then bounded this by a simple code:

```
if numAmp > 12,
      numAmp = 12;
elseif numAmp < 4,
      numAmp = 4;
end;
```

This allowed us easily to select for parameter sets that exhibited the "ideal" damping behavior. However, we quickly realized that there existed more than one way to follow this mathematically damped behavior. The sought oscillatory behavior (Figure 1A) could be passed over for a behavior contradictory to the biological data (Figure 1B). Thus, the second task was to figure out how to ensure that the mean concentration was decreasing.

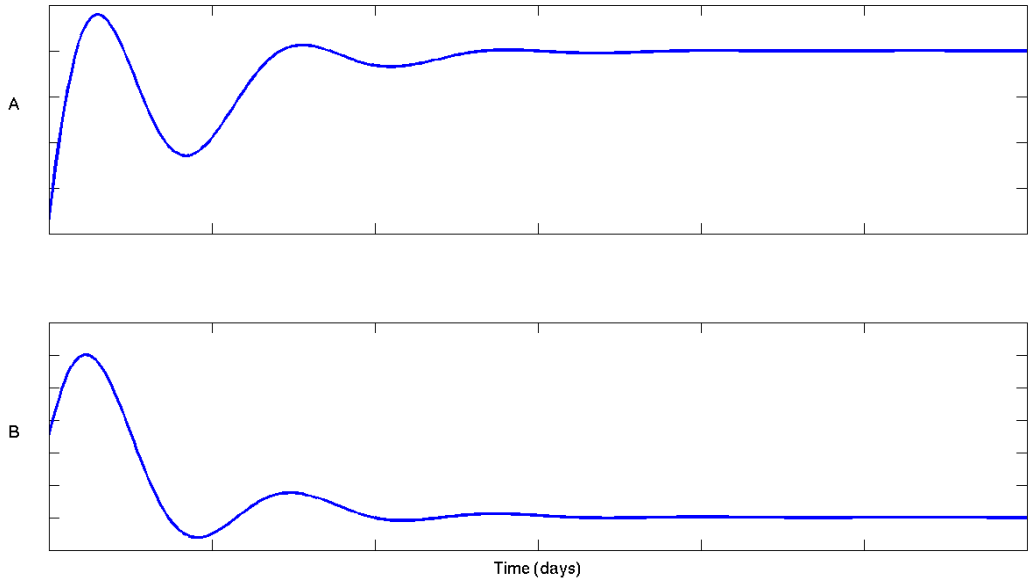

Figure 1. Two types of damped oscillations. A: Example plot of an increasing mean average of the concentration of a biological state with a decreasing (or damped) amplitude. B: Example plot of a decreasing mean average of the same.

It was important to account for the actual concentration of the states instead of accounting only for their relative behavior because of the difference illustrated in Figure 1. Although the oscillator described technically is damped, the biological data show that there is no detectable level of PER2 protein in this case. We therefore wanted to force the optimization algorithm to find a model that also satisfied this condition. Essentially, we wanted to ensure that the average amplitude for the newly optimized parameter sets was lower than the average amplitude of the wild type. We accomplished this by ensuring that

the mean average of PER2 total protein for the entire run was lower in the knockout simulation than in the wild type simulation:

```
wtavg = mean(sum(yTot(:,[10 19 20])));
koavg = mean(sum(yTot(:,[10 19 20])));
```

where yTot is the field that holds the state variable trajectories and the indices into yTot represent the states containing PER2 in monomer (10) and dimer (19 and 20) forms. We then ensured that the cost associated with the difference would be modulated as an inverse function such that the mean average in the knockout should be minimized:

$$C_{amp} = \frac{koavg}{koavg + wtavg}$$

Trial and error showed us that the optimization was easily tricked into falling prey to a simple trick: the model could easily show damped behavior by favoring systems that forced an artificially high amplitude initially, which would then stabilize to a lower stable amplitude within a few periods. We determined that this was unrealistic because, although the concentration units are arbitrary (and the model is not directly relatable to the biology in terms of numbers or scale), the difference in relative order of magnitude was not reflective of biological plausibility. As described, we implemented our cost function to take this into account. I then evaluated the cost of the initial parents and was set to begin the optimization itself.

iii. Parallel Implementation and MATLAB Quirks

There were several snags that we encountered on our journey to optimization. We made use of the parallel tools built in to MATLAB to speed the optimization process. We were able to do this because the method for generating children was implemented such

that there was no parametric interdependence within a generation. First, I randomly chose a set of initial parents from sampled parameter space that we knew produced a working model based on unpublished work by Hannah Coulson ('10). In order to determine the appropriate weight that should be given to the new Cry1 knockout criterion that we added to the cost function, I first evaluated the published parameter set using the unmodified cost function, which yielded a cost of approximately 12. Some may ask what this number signifies. Put simply, this number is a unitless and almost arbitrary measure, while nonetheless being representative of the fitness of a parameter set. The next step was to determine the appropriate cost associated with the newly defined terms of the Cry1 knockout.

Arguably, one of the most useful advancements in modern computing is parallelism and distributed computing. We were able to take advantage of the built-in parallel computing toolbox in MATLAB in order to reduce the time of computation. Our biggest hurdle in this implementation was seeding a random number generator, which seemed to be reseeding the same number upon instantiation of a parallel thread. For obvious reasons, this proved problematic as our stochastic optimization algorithm relies on pseudo-randomness to provide semi-uniform sampling of parameter space. We were able to avoid this problem by seeding the random number generator, a globally accessible object in MATLAB, within each function call instead before instantiation of all threads (see appendix A for coding example).

The final decision to make in the optimization endeavor was to decide when an answer that was "good enough" was obtained. Because the parameter space of this optimization problem is so large, it is difficult to prove that any solution is optimal; indeed,

it is unknown how to define "optimal" or if there exists a single solution for such an evaluation. I determined that a good solution was reached when the optimization algorithm was unable to reach a better solution within about 70 generations. This determination was based on two rationales: 1. the diversity of the population decreases with each subsequent generation, so it becomes increasingly unlikely that population is capable of reaching a better solution; and 2. by the time the optimization algorithm reaches a point where the same solution is optimal over 70 generations, it has been running for several days with a trend of exponential time to optimize the solution further (Figure 2).

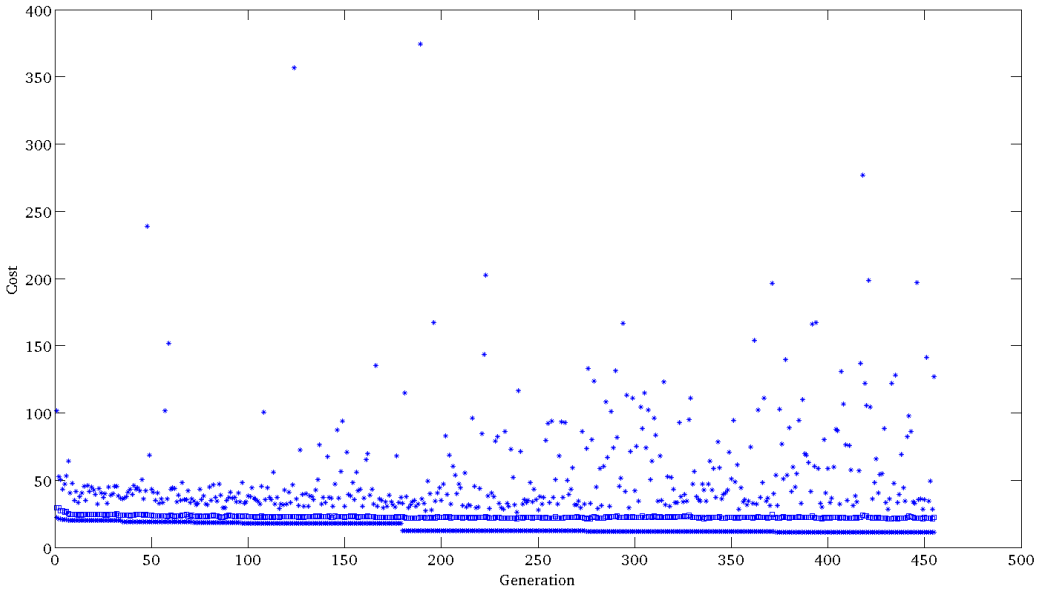

Figure 2. Plot of the lowest, highest, and mean average cost of individuals in each generation of the optimization. Note that the lowest cost decreases in steps because of the elite individuals passed on in each generation.

III. Optimization Results

Recall that, in systems biology, the model is always being revised, revisited, and further refined. It is not enough to have written a cost function that we believe optimizes the parameter set to behave in any particular fashion. Indeed, we must verify that the model behaves in the way that we expect, and, more importantly, we must confirm that the new parameter set does not destroy some other good aspect of the model. We therefore test the new parameter set in single-cell and tissue simulations to confirm its ability to accurately predict the behaviors previously predicted by the published parameter set and to test its ability to further predict known behavior. If it succeeds in these tests, then we can use the new parameter set to predict behavior under conditions for which data do not exist and have faith that the model demonstrates a close approximation of the biology.

i. Single-cell Comparisons

Table 1. Summary of Single-cell Data. The newSet parameters performs the same approximations of the experimental data as the pubSet parameters.

Knockout Type	Experimental Class	newSet class	pubSet class
Cry1	damped	damped	arrhythmic
Cry1Cry2	arrhythmic	arrhythmic	arrhythmic
Cry2	rhythmic	rhythmic	rhythmic
Per1	arrhythmic	arrhythmic	arrhythmic
Per1Per2	arrhythmic	arrhythmic	arrhythmic
Per2	arrhythmic	arrhythmic	arrhythmic
Rev-erbα	rhythmic	rhythmic	rhythmic
Rorc	rhythmic	rhythmic	rhythmic
Bmal1	arrhythmic	arrhythmic	arrhythmic

Our analysis of the resulting parameter sets helped us to iteratively refine our model. First, we wanted to compare the new parameter set against all of the single-cell

knockout data. Table 1 describes the behavior of the model under the given gene knockout conditions with the classifications of the biological data and the simulation data with the original published parameter set. Importantly, the newly discovered parameter set does not change the ability of the model to predict the behavior of single-cell knockouts.

There are two major distinctions that stand out between the newly discovered set and the published set. First, the period of the simulation with new parameter set is slightly longer in the wild type; second, the state amplitudes peak at higher concentrations. However, neither of these differences is paramount. We can convert our time scale into what is known as "circadian time," which by definition translates the model into 24 even "circadian hours." In addition, we recall that the state concentrations are of arbitrary units and are indeed of arbitrary real value (their relative values are important). This disconnect arises because we have no chemical means of measuring exact concentrations over time *in vivo* or *in vitro*, so we can not connect the simulated concentrations back to the direct biology.

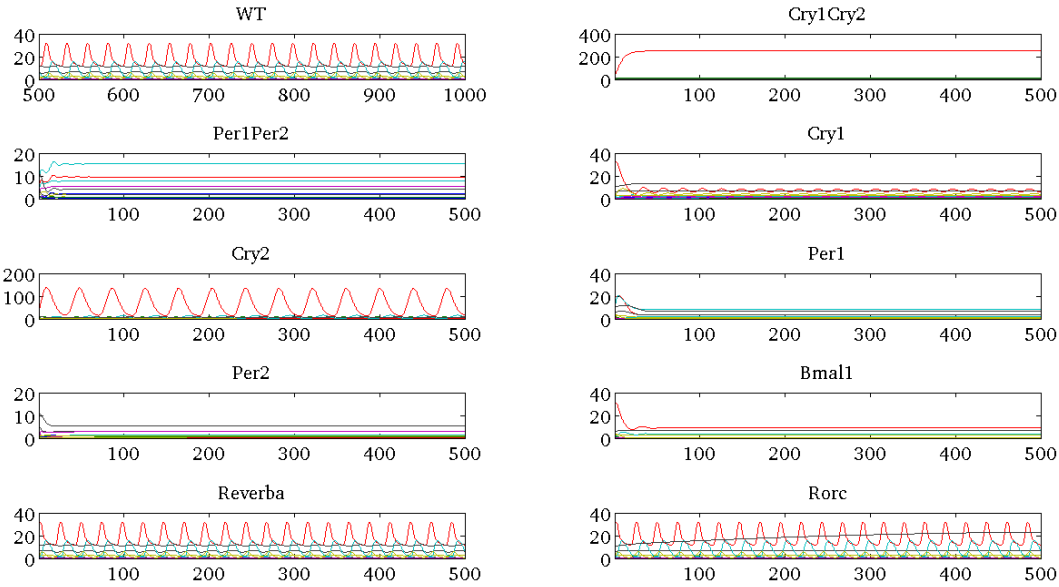

Figure 3. Single-cell simulations with the newSet parameters. Note the biological state concentration amplitudes as being higher in all simlations as compared to the pubSet simuations.

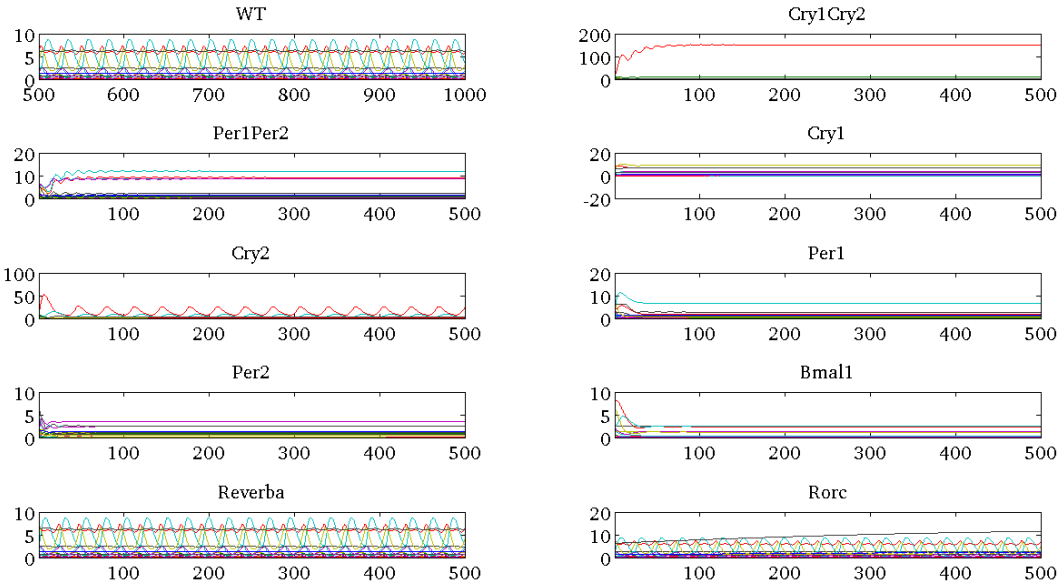

Figure 4. Single-cell simulations with the pubSet parameters.

Although we are unable to connect our simulations quantitatively to the biology, there are still aspects of the state trajectories that remain important to recognize. For example, in the single-cell simulation of the Cry1 knockout, the new parameter set shows that the system is a damped oscillator that falls into a low, stable oscillation. We can assume that this oscillation is acceptable because we presume that the peak amplitude is well below the detection limit of the method used in the wet lab. We also note that the simulations of Rev-erbα and Rorc show very little distinction between themselves and the wild type, for both of the parameter sets, as is illustrated in Figures 3 and 4.

ii. Multi-cell Comparisons

Much of the interesting data come from the multi-cell, or tissue, analysis. As we recall, the training data were from single cell explants of the SCN, but our goal was to create a model of the tissue.

a. A little bit of background

This tissue model was accomplished by modifying the ODEs from the single-cell model to include intercellular signaling in a connected graph of n nodes, where n is the number of cells and is input at runtime; this is called coupling. Coupling has been shown to rescue oscillations in systems that are arrhythmic with individual cells (Liu 2007, Vasalou 2009, To 2007). We simulated graphs for intervals ranging from 50-250 cells. Intercellular signaling acts through a VIP pathway (Aton 2005) that is pegged linearly to the total concentration of *per1* and *per2* mRNA. This signal activates transcription of CREB, which activates transcription of *per1* and *per2* in the target cell; CREB is also auto-inhibitory, so there is no infinitely increasing feed-forward loop but rather an overall effect of the phase

of the clock (Pulivarthy 2007, Lim 2007). The rest of the model remains unchanged. The code is added through an additional 19 parameters (adding the CREB parameters and the signal parameters to the *per* ODEs) and a single extra biological state equation (describing CREB).

The same algorithms for solving ordinary differential equations that were used previously are still able to solve these networked ODEs, however the solution takes much longer to compute because all of the ODEs for every cell in the network must be computed simultaneously. We specified the non-zero pattern of the system's Jacobian matrix. This allowed the ODE solver to work much more efficiently and led to a significant speed-up in computation. The Mirsky group previously optimized the 19 additional parameters that had been added to enable intercellular communication, and we did not think it necessary to re-optimize the parameters because our goal was to select only for the single cell behavior and to be able to predict the tissue behavior.

In addition, we introduced some randomness to the network of cells by perturbing the values of a set of random parameters by 1-10%. This set was randomized for each cell such that all parameters in the network were equally varied between all cells. We hypothesized that the cells in the SCN were slightly dissimilar from each other because not all cells respond in the same way to the same stimuli (Aton 2005, Herzog 2004, To 2007). Because of this, some cells could be arrhythmic and some could be rhythmic or damped rhythmic, but that intercellular communication would "rescue" oscillations in the entire network. Thus, this variation in parameters would help to better mimic this rescue by heterogeneity.

b. The results

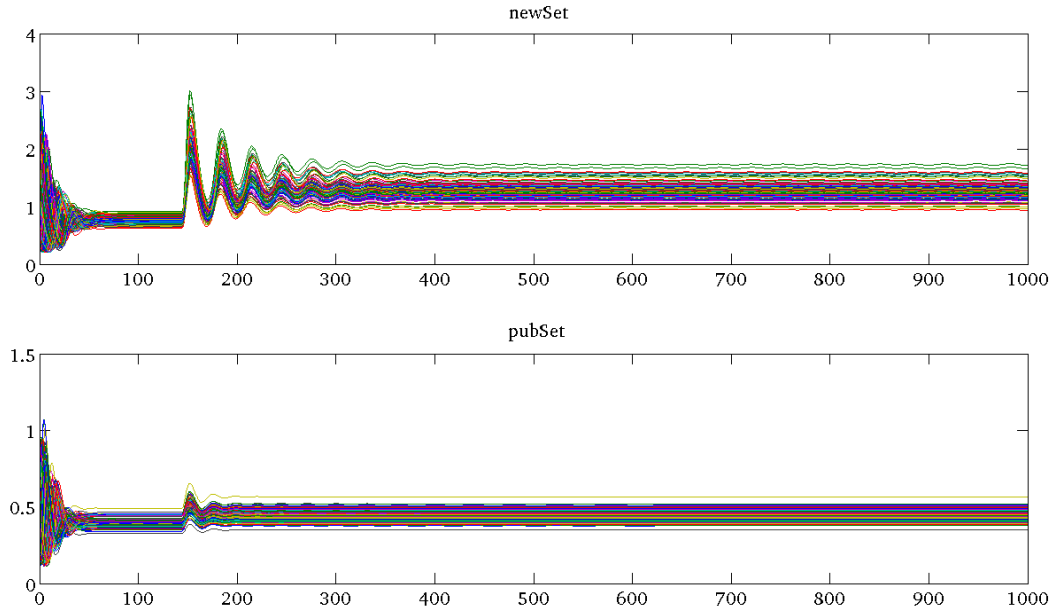

Figure 5. Per1 gene knockout simulation using the two parameter sets. Note the major distinction in the newSet that the average *per2* mRNA concentrations with a 3% standard deviation in the cells' parameter sets. Signalling is disabled until time t=150. (ncells=100)

The additional ability of the model to predict two other known tissue phenotypes shows that the newly discovered parameter set captures the relative biology better than the published set (Table 2). One of the major differences between the two models is that the new parameter set creates a model with much higher concentrations of the biological states. This can be seen Figures 5 and 6, which show the behavior using the two parameter sets in the two gene knockouts that the new model is able to correctly predict. Here, we see the Per1 gene knockout unable to synchronize in the published set (Figure 5, bottom). The new set (Figure 5, top) synchronizes for about 6 days before settling into its own steady state. In the Cry1 knockout, the published set is unable to produce stable oscillators with

intercellular communication (Figure 6, bottom). The new parameter set, however, rescues the oscillations with a high amplitude in perfect synchrony (Figure 6, top).

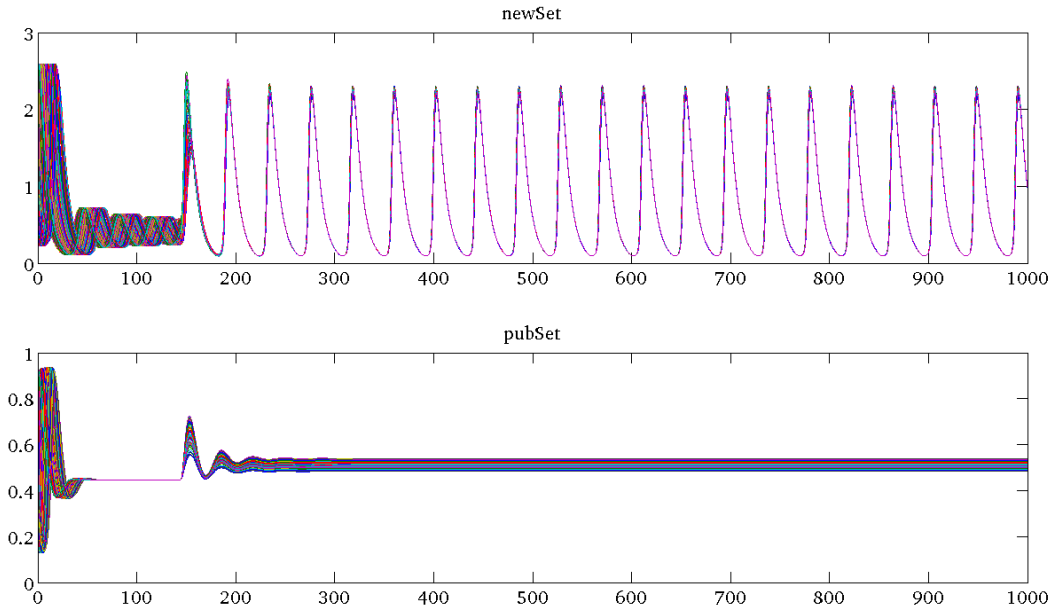

Figure 6. Cry1 knockout simulations using the two parameter sets. Again, note the difference in the ability of the two models to syncronize. This simulation is without any variation in the parameters (i.e. all cells are identical). Signaling is disabled until time t=150. (ncells=100)

The iterative refinement that was described earlier has now reached the point where we can begin to ask new questions. Namely, we can begin to investigate *how* the new parameter set is able to predict the additional tissue phenotypes. This comparison is the focus of future work in the Taylor Group. More specifically, we first notice that the period of both models is longer when coupled compared to its single-cell equivalent. Our second observation is that the state peaks in the new parameter set simulation are much higher than that of the published parameter set.

Table 2. Summary of Tissue Data. The newSet parameters are able to predict the biological data in the Cry1 and Per1 gene knockout simulations. Biological Bmal1 gene knockout data is not available.
1. The biological Per2 gene knockout data are for a circadian clock for fibroblasts, which are not necessarily similar the SCN cells. 2. The pubSet Per1Per2 gene knockout data was unable to complete in simulation after several months of continuous computation, leading us to postulate that it is arrhythmic.

Knockout Type	Experimental Class	newSet class	pubSet class
Cry1	rhythmic	rhythmic	arrhythmic
Cry1Cry2	arrhythmic	arrhythmic	arrhythmic
Cry2	rhythmic	arrhythmic	arrhythmic
Per1	rhythmic	rhythmic	arrhythmic
Per1Per2	arrhythmic	arrhythmic	--[2]
Per2	rhythmic[1]	arrhythmic	arrhythmic
Rev-erbα	rhythmic	rhythmic	rhythmic
Rorc	rhythmic	rhythmic	rhythmic
Bmal1	--	rhythmic	rhythmic

IV. Future Work

In the future, we hope to explain why the new parameter set is able to predict model behavior that was not previously captured. In particular, we want to look at how changes in the gene network structure (through gene knockouts) affect the signaling effects of CREB on *per1* and *per2*. These effects are described by two parameters that are part of the 19 added parameters that were specific to the intercellular signaling. Initially, we hypothesized that these two parameters had different sensitivities with the different parameter sets that were exacerbated in the knockout settings. However, initial work shows that the response of the model to the parameters is not significantly different. Indeed, our work strongly suggests that the predictive ability of the new parameter set may be due to the fact that the concentrations of PER1 and PER2 protein are so high that they create a strong CREB signal by which cells in a tissue network can couple effectively.

However, this phenomenon needs to be investigated further. For example, one reason to question this conclusion is that the tissue simulations are able to reproduce a coupled-uncoupled-coupled paradigm, whereby intercellular signaling is interrupted and then resumed. This behavior is not reproducible by any previously published model of the biology, but I believe that it may indicate that a more complex system is at play. In addition, there are still several aspects of the intercellular signaling model that the group is currently exploring. Specifically, To et al (2007) used a different method cell network structure to simulate inter- and intra-cellular signaling. This method, and others, might affect the synchronization potential of the model as a whole.

V. Conclusion and Summary

We were able to redesign the cost function associated with a genetic algorithm to discover a new parameter set selected to specific single cell simulation conditions. In order to do this, we utilized a novel classification of oscillator behavior ("damped") to add a finer resolution to the biological data. This parameter set is able to accurately predict some aspects of the true biological tissue. Thus, the parameter set provides a new avenue through which to explore the model structure and dynamics.

One important characterization of the parameter set with respect to the rest of parameter space is the local "pointy" shape of the space. This structure describes how stably fit the discovered parameter set is. The biological data suggest, through their robustness and ability to remain nearly constant under varying conditions (indeed, many of us have experienced the phenomenon of jetlag), that the parameter space surround any appropriate parameter set would be broad, such that cells that sample the local parameter space would not be prone to exceedingly unique behavior. The tissue data suggest that this is the case for this parameter set. In addition, a brief analysis of the parameter sets that fell into the elite count at the last generation shows that their behaviors are similar with regard to single-cell wild type simulations, but are distinct in the specifics of the model simulation. Indeed, these two pieces of data suggest that local parameter space surrounding the new parameter set is (at least close to) ideal.

Although the newly discovered parameter set is unable to predict all of the desired traits, it is certainly a closer approximation of the true biology. Our future work seeks to explain how the new parameter set accomplishes this more attuned simulation. Preliminary work suggests that it may be as simple as the absolute concentration of specific biological

states within the simulation that surpass thresholds for effective intercellular signaling, however we are unable to confirm or reject this hypothesis. Despite this, we are able to continue exploring the model dynamics through adaptations of Stephanie's doctoral work, and we continue to re-evaluate and refine the model and our understanding of it.

References

Aton SJ, Colwell CS, Harmer AJ, Waschek J, Herzog ED (2005) Vasoactive intestinal polypeptide mediates circadian rhythmicity and synchrony in mammalian clock neurons. *Nat Neurosci.* 8:476-483.

Goldbeter A (2002) Computational approaches to cellular rhythms. *Nature.* 420:238-245.

Goldberg DE. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

Herzog ED, Aton SJ, Numano R, Sakaki Y, Tei H (2004) Temporal Precision in the Mammalian Circadian System: A Reliable Clock from Less Reliable Neurons. *J Biol Rhythms* 19(1):35-46.

Kitano H (2002) Systems Biology: A brief overview. *Science.* 295:1662-1664.

Lim C, Lee J, Choi C, Kim J, Doh E, Choe J (2007) Functional Role of CREB-Binding Protein in the Circadian Clock System of *Drosophila melanogaster*. *Mol Cell Biol* 27(13):4876-4890.

Liu AC, Welsh DK, Ko CH, Tran HG, Zhang EE, Priest AA, Buhr ED, Singer O, Meeker K, Verma IM, Doyle FJ, Takahashi JS, Kay SA (2007) Intercellular Coupling Confers Robustness against Mutations in the SCN Circadian Clock Network. *Cell* 129:605-616.

Liu AC, Tran HG, Zhang EE, Priest AA, Welsh DK, Kay SA (2008) Redundant Function of REV-ERBα and β and Non-Essential Role for Bmal1 Cycling in Transcriptional Regulation of Intracellular Circadian Rhythms. *PLoS Genet.* 4(2):e1000023.

Lee KY, Yang FF (1998) Optimal Reactive Power Planning Using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Strategy, Genetic Algorithm, and Linear Programming. *IEEE T Power Syst* 13:101-108.

Mirsky HP, Liu AC, Welsh DK, Kay SA, Doyle FJ (2009) A model of the cell-autonomous mammalian circadian clock. *Proc Natl Acad Sci USA* 106:11107-11112.

Pulivarthy SR, Tanaka N, Welsh DK, De Haro L, Verma IM (2007) Reciprocity between phase shifts and amplitude changes in the mammalian circadian clock. *Proc Natl Acad Sci USA* 104(51):20356-20361.

To TL, Henson MA, Herzog ED, Doyle FJ. (2007) A Molecular Model for Intercellular Synchronization in the Mammalian Circadian Clock. *Biophys J.* 92(11):3792-3803.

Vasalou C, Herzog ED, Henson MA (2009) Small-World Network Models of Intercellular Coupling Predict Enhanced Synchronization in the Suprachiasmatic Nucleus. *J Biol Rhythms* 24(3):243-254.

Wager-Smith K, Kay SA (2000) Circadian rhythm genetics: from flies to mice to humans. *Nat Genet.* 26:23-27.

Weaver DR (1998) The suprachiasmatic nucleus: a 25-year retrospective. *J Biol Rhythms* 13:100–112.

Yamazaki S and Takahashi JS (2005) Real-Time Luminescence Reporting of Circadian Gene Expression in Mammals. *Methods in Enzymology* 393:288-300.

Yoo SH, Yamazaki S, Lowrey PL, Shimomura K, Ko CH, Buhr ED, Siepka SM, Hong HK, Oh WJ, Yoo OJ, Menaker M, Takahashi JS (2004) PERIOD2::LUCIFERASE real-time reporting of circadian dynamics reveals persistent circadian oscillations in mouse peripheral tissues. *Proc Natl Acad Sci USA* 101(15):5339-5346.

Appendix A – Optimization Implementation Code

```
% Loop over the generations
for g = 1 : settings.numGenerations,

    % disp(['calculating generation ' int2str(g)]);
    fn = ['Optimization' int2str(settings.filenum) 'generation' int2str(g) '.mat']; % l6
    if ~exist(fn,'file') % l7
        G     = zeros(settings.numChildren, length(lb));
        Gcost = zeros(settings.numChildren,1);
        G(1:settings.eliteCount,:) = P(1:settings.eliteCount,:);
        Gcost(1:settings.eliteCount) = Pcost(1:settings.eliteCount);
        parfor c = settings.eliteCount+1 : settings.numChildren
            % Create a child from the previous generation
            [G(c,:) Gcost(c)] = generateChild(cost_fcn, P, Pcost, lb, ub, settings, c);
        end;
        % Sort the children by cost from least to greatest
        [Gcost idx] = sort(Gcost(find(Gcost>-1)));
        G           = G(idx,:);

        % save the current state of this generation
        save(fn,'G','Gcost','P','Pcost');
    end; % l9
    load(fn); % l10
    disp(['Generation ' int2str(g) ' has ' int2str(length(Gcost)) ' children and ' ...
        int2str(length(Pcost)) ' parents.']);
    % Determine the parents of the next generation
    numParents = settings.numParents;
    % first ensure that there are enough children
    if numParents > length(G),
        numParents = length(G);
    end;
    P     = G(1:numParents,:);
    Pcost = Gcost(1:numParents);
end;
```

Note that the nested **parfor** loop calls `generateChild()`. It is within this function that the random number generator is seeded:

```
function [params, cost] = generateChild(cost_fcn, P, Pcost, lb, ub, settings, Cnum)

% Seed the random number generator so that it will generate
% a new pseudo-random number sequence each time the algorithm
% is run.
rand('state',sum(Cnum*100*clock));

% additional code for actually generating the child is not included in this appendix
end;
```

This random generator is guaranteed to be uniquely seeded for each child and each parallel thread.

Appendix B – Terminology Overview

Population: A population is simply a set of individuals. A population can be a subset of another population, as long as it is clearly genetically separable (e.g. two populations of the same species of fish in two different lakes), or it can describe the entire set of known individuals (the set of all individuals of *Mormotomyia hirsuta*, the African "terrible hairy fly").

Individual within a population: An individual of a given population fits the basic description of the encompassing characteristics of the population. For example, all individuals in a given population of wolves would bear similar physical appearance, and more importantly, any two individuals would be genetically compatible to produce a viable offspring. This last characteristic is what is important for creating the genetic algorithm: if we can describe a pseudo-genome through definite parameters, then we can create a set of "genetically" similar individuals that can produce offspring.

Parent/child: intuitively, the definition of a parent and a child makes sense. However, in reference to a genetic algorithm optimization, it's important to take a closer look at what they are. A parent is any individual chosen from a given population. A child is a new individual created from the "genetic" information contained within one or more parents; a generic genetic algorithm does not require a two-parent system, though our specific implementation does assume that.

Diversity within a population: Diversity describes the "genetic" variation within a population and is related to the similarity (or rather dissimilarity) of the genetic information of the individuals within the population. In life, genetic diversity is important for a species to be able to adapt to an evolutionary pressure (e.g. global warming or an invading predator). In the genetic algorithm optimization, diversity helps to ensure that the algorithm does not get caught in a local optimum of the parameter space. If diversity decreases too

much in the optimization process, the algorithm will stop prematurely without finding a solution near the global optimum.

Convergence: convergence of an optimization algorithm describes the phenomenon of having several individuals within the population nearing the same solution, weighing the population closer to that solution. It is not necessary that this solution be globally optimal, but it is usually true that the solution will be locally optimal. This tendency is particularly pronounced when the population is composed of few individuals because the population is sensitive to genetic drift, a common phenomenon in restricted populations.

Proportional selection: Parents produce children. Darwin's theory of natural selection tells us that the parents that are successful at reproducing are those that have a better fitness for a given environment. This is the basis for the genetic algorithm that we implemented, and the key question to ask is how likely is any given parent to reproduce. Proportional selection essentially assumes that the probability p_i of an individual being selected as a parent for the next generation is related to the individual's fitness f_i as $p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$. This is in contrast to methods such as uniform selection (every individual is equally likely), tournament selection (a subset of individuals is randomly selected to compete to be a parent), or linear ranking (the probability is linearly related to the fitness).

Elite count: The elite count is an element of the evolutionary strategy that clones individuals from one generation to the next. Essentially, genetically identical individuals are allowed to be parents in two or more subsequent generations. This helps to ensure that a good solution is not lost during the optimization. In a large enough population size with enough time to run the optimization for thousands of generations, this would probably not be needed; however, in the conditions with a restricted population size and a limited computational time, we chose to add this element to the genetic algorithm. Generally, this number remains small compared to the total population size (in our case, 5 out of 100 individuals).

Mutation: In life, mutation is the key biological error that allows for evolution to occur. It occurs when the enzymatic machinery that clones the genome makes an error and changes one or more key genes. However, our genome is much less restricted than that of DNA (recall that DNA uses four amino acids as its basis whereas we use floating point numbers). Therefore, we have to allow for mutation in a different way that will allow for the same benefits to come about. We implemented this by selecting parameters randomly and with a normal distribution and then adding a 5% mutation onto each of them. We were able to