Research Notes: Week 6

Subteam 1

June 22-26

With the cleaned and averaged data sets from last week, this week's work focused on applying DRAM MCMC and PSO to various parameter subsets of the model and validating the outcomes. While implementing the parameterization methods proved fairly straightforward, various alterations were made to both algorithms in order to increase accuracy and efficiency.

1 DRAM MCMC

1.1 Last Week

Last week, we were focused on understanding how to shift the Mathews data to an absolute time scale. Our solution to this was to collect time of diabetes onset and fit a lognormal distribution to these values. We could then shift our data according to this distribution. We also included the mean and standard deviation of this in the parameter set. We also honed our parameter subset to include the 7 sensitive parameters from Subteam 2's UKF algorithm, eta parameters (which impact disease onset time), and the onset time mean and standard deviation. From Subteam 2's parameter list and variances, we were able to establish a more specific range for our parameter space: $\pm$ 2 standard deviations. The results of this algorithm and parameter set can be seen in the figure below.

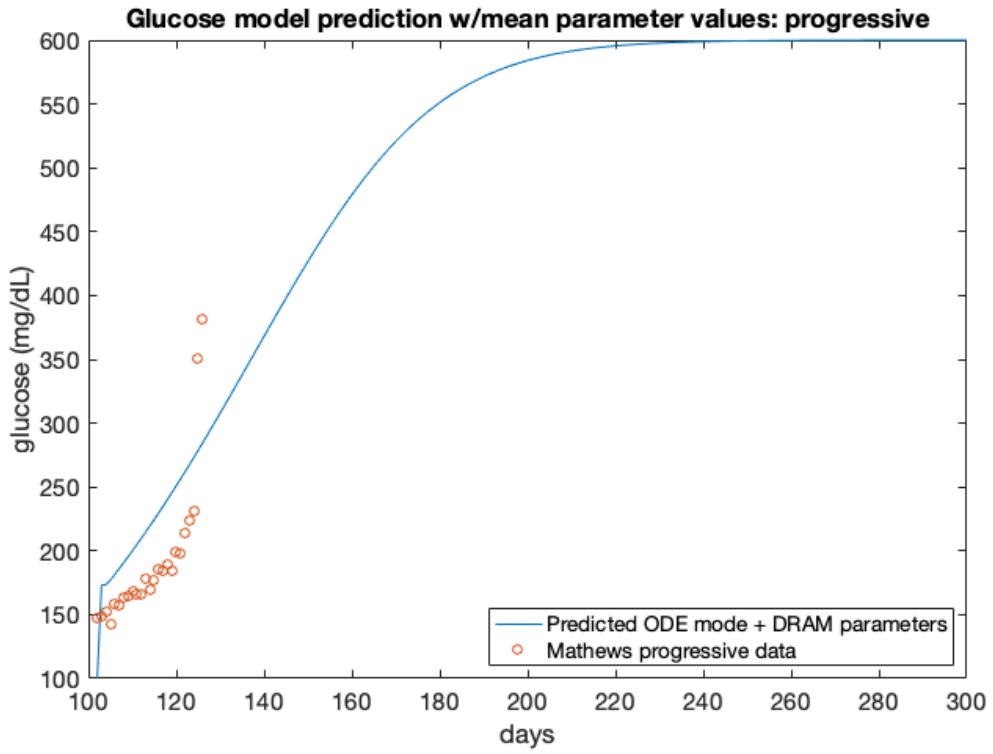


Figure 1: Predicted glucose measurement using post-DRAM mean parameter values (blue), plotted with Mathews et al. progressive data (orange).

	mean	std	MC_err	tau	geweke
e1	7.9999e-05	6.4384e-05	8.9689e-06	330.05	0.017802
e2	9.0327e-05	6.6682e-05	9.5419e-06	370.29	0.017471
delta_B	0.017754	0.0017374	0.00033513	692.21	0.77947
SI	0.10862	0.0080484	0.0016611	783.43	0.80782
GI	14.15	0.0073209	0.0014791	776.3	0.99867
mues_r	0.00064087	0.00072638	0.0001166	418.17	0.025502
mues_e	0.0014491	0.0010064	0.00019087	469.94	0.014461
alpha_eta	0.010353	0.00037342	6.8813e-05	564.24	0.92358
beta_eta	21.002	0.0034311	0.00073701	551.28	0.9997
eta_basal	0.017086	0.0063846	0.0013474	786.94	0.2176
onset_mu	4.8144	0.0022231	0.00039416	513.81	0.9989
onset_sigma	0.2137	0.0060235	0.00086382	156.13	0.99815

Figure 2: DRAM chain results for parameter set of 7 sensitive, eta, and time parameters. Most notable columns are the mean and geweke. Geweke value of > 0.7 we consider to have converged.

In Figure 1, you can see the predicted glucose model given the final mean DRAM parameter values plotted against the original progressive Mathews data. We see that the algorithm is not doing a fantastic job of fitting the Mathews data. Unfortunately at this point, we do not have a quantifiable measure of error for our model. But we can see some explanation for why the fit might not be so good in looking at Figure 2. This is a table of the results of the DRAM chain. We are most interested in the mean column and the geweke column. We used the values of the mean column to create the predicted model in Figure 1. In the geweke column, we can see that not all of our parameters have converged. We even have some values of < 0.1. From previous literature, we conclude convergence when we see a geweke value of > 0.7.

1.1.1 Some parameter justification

It is important to note some of the reasons we have chosen this parameter set:

1. We thank Subteam 2 for providing us with their list of sensitive parameters as well as their variances. In the DRAM algorithm, it is not really possible to tune your parameter ranges as the algorithm is running. Once you input an initial parameter value and range, you cannot update this range (or we have not yet discovered a way to do this). These variances allowed us to be more precise about the range we were allowing our parameters to vary.

2. We have chosen to keep the parameter subset fairly small but in previous runs we have run the DRAM algorithm on all 41 parameters. We would like to keep the number of parameters to be less than or equal to the number of data points we have available (for Mathews this is 25 data points). This is because we do not have an explicit prior function (the function that informs what kind of distribution to sample our parameters from). Instead, we have a prior variance value, which for us is the mean squared error (mse) of our model given a base set of parameters. In the worst case, if we have more parameters than data points, we can only have a defaulted uniform (uninformative) prior. (The ideal situation would to define our own explicit prior function as then we could include more parameters in our algorithm; this is something that is in the works.)

1.2 Updating DRAM

Throughout the process of tuning the DRAM algorithm, the parameter subsets have been constantly changed and updated. The major update that occurred this week was the inclusion of initial conditions as parameters and the exclusion of disease onset time parameters (mu and sigma).

Initial conditions: The non-zero values of the initial conditions (IC) are resting macrophages, beta cell levels, glucose levels, and insulin levels. It is important to vary the initial conditions because in the T1D model, the given values are assuming that the model will always be evaluated starting at time 0. However, since we have shifted Mathews data to absolute time, we are no longer starting at time 0. Thus, we have to account for the shift in time and how these values might vary in that period. We can do this by allowing those parameters a range and parameterizing them in our DRAM algorithm. (Note: initial conditions that were initialized as 0 were not parameterized)

Time parameters: After discussion with Prof. Shtylla, we have determined that it is not useful for us to parameterize the diabetes onset time (mu and sigma). Our initial thought process was for each sample in the algorithm we

1. Evaluate the ODE for a large time span (about 40 weeks or 280 days)

2. Use parameters mu and sigma of the diabetes onset distribution to choose an onset time

3. Build a 25 day time span out of that onset time

4. Grab that section of time from the evaluated ODE

However, remember that we are using a built-in MATLAB function, *fitdist* to create our diabetes onset distributions which we use to shift the Mathews data to absolute time. It is unlikely that parameterizing the mean and standard deviation of this distribution in the DRAM algorithm would produce a significant improvement in overall results. Therefore, we decided to discard these time parameters. We now only use the diabetes onset distribution to shift the data prior to running the algorithm. This time span is now constant throughout the DRAM process (so every time we sample, we are only solving the ODE for this 25 day time span).

Lastly, we decided to increase the number of samples taken in the algorithm from 6,000 to 10,000. In previous runs last week, we noticed that some parameters were still not reaching convergence. By increasing our samples, we give our algorithm more time to reach convergence.

1.2.1 Roadbumps

Once we determined our parameter subset of interest and began running our algorithm, we encountered several issues along the way. We have since 'fixed' these issues, although it is still a little unclear why or why the fixes implemented truly resolved problems, but it is important to be transparent about the process especially for future use.

1. In the debugging phase, we noticed that our ODE model was only evaluating glucose at a single value of 100 for the entire time span that we fed it. However, this was a simple fix, we found a typo inside our ODE system that resolved the issue.

2. The library that we utilize, mcmcstat, comes with very fancy plotting functions. For example, this library has a function that plots the model prediction based on the results of the DRAM chain parameter values and computes and plots a confidence interval - mcmcpredplot. By stepping through this function and also our main DRAM function, we finally noticed that our DRAM algorithm did not produce all of the necessary inputs for mcmcpredplot, most notably the variance of the error. More stepping through led us to implement a model.S20 of 1 and a model.N0 of 1, which correspond to the uniform prior from the previous section. By defining these two terms, the variance could be saved and used in the mcmcpredplot. It is unclear why or how this seems to fix the issue, but further investigation would be needed to be sure.

1.3 Results

After implementing the changes to parameter sets and making adjustments to allow for proper plotting. We ran the DRAM algorithm on both the acute and progressive Mathews et al data sets. You can see the results of these runs below.

1.3.1 Progressive Results

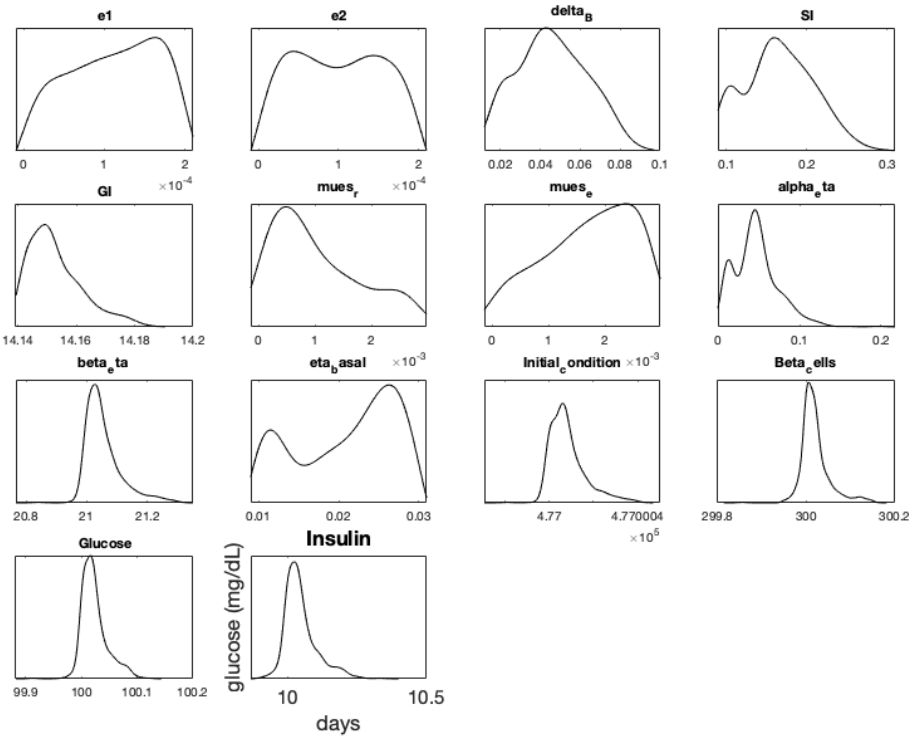


Figure 3: Posterior density plots for progressive Mathews et al mouse data.

In this figure, we get a visual sense of the parameter distributions. Many of the parameters do not seem to have an easily recognizable distribution. We can see that some, like e1 and e2, look vaguely uniform. Others like GI, $beta_eta$, $mues_r$, glucose, and insulin, look more lognormal. Still others appear to be bi-modal.

	mean	std	MC_err	tau	geweke	
e1	0.00010791	5.825e-05	5.4991e-06	64.508	0.90849	
e2	9.9354e-05	5.8922e-05	3.6826e-06	32.808	0.77314	
delta_B	0.045966	0.017079	0.0032703	1140.6	0.1249	
SI	0.16573	0.041917	0.0085364	1190.7	0.29203	
GI	14.153	0.0091062	0.0016664	1131.3	0.99838	
mues_r	0.001026	0.00078446	0.00011008	502.61	0.28468	
mues_e	0.0016997	0.00078948	7.7221e-05	103.92	0.86591	
alpha_eta	0.046788	0.027128	0.0050702	1087.8	0.050901	
beta_eta	21.055	0.059693	0.010608	633.15	0.99457	
eta_basal	0.020334	0.0065641	0.0011452	752.16	0.27501	
rest_macrophage	4.77e+05	0.081101	0.014528	637.94		1
Beta_cells	300.02	0.032506	0.0053242	525.99	0.99986	
Glucose	100.02	0.023269	0.0037851	599.48	0.99956	
Insulin	10.047	0.057837	0.0094988	790.12	0.98992	

Figure 4: DRAM chain results for parameter set of 7 sensitive, eta, and initial conditions for progressive Mathews et al mouse data. Most notable columns are the mean and geweke. Geweke value of > 0.7 we consider to have converged.

From this table, we can see that the algorithm did a little better in terms of convergence (than our pre-parameter-adjustment). We have more parameters that are converging. A few however are still not converging: $delta_B$, SI, $mues_r$, $alpha_eta$, and eta_basal. This is a little concerning as the eta parameters impact the model's diabetes onset time. It is possible that the small number of parameters is limiting the

overall parameter search space.

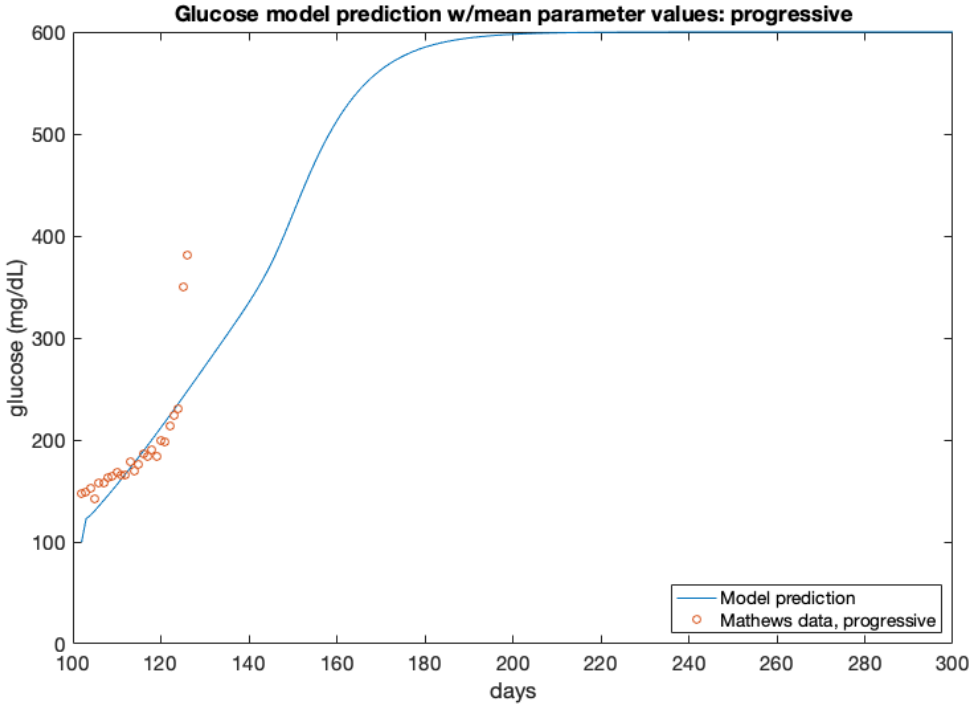


Figure 5: Predicted glucose measurement using post-DRAM mean parameter values (blue), plotted with Mathews et al. progressive data (orange). The model is predicted beyond the scope of the Mathews data to capture end behavior of the model.

In this figure, we can see how well the model using DRAM mean parameter values fit the Mathews data. The general trend seems to be in the same direction with a similar slope. However, the algorithm seems to have difficulty capturing the sharp spike that we recognize as diabetes onset. We plotted the predicted model for a longer time span so that we could see its end behavior.

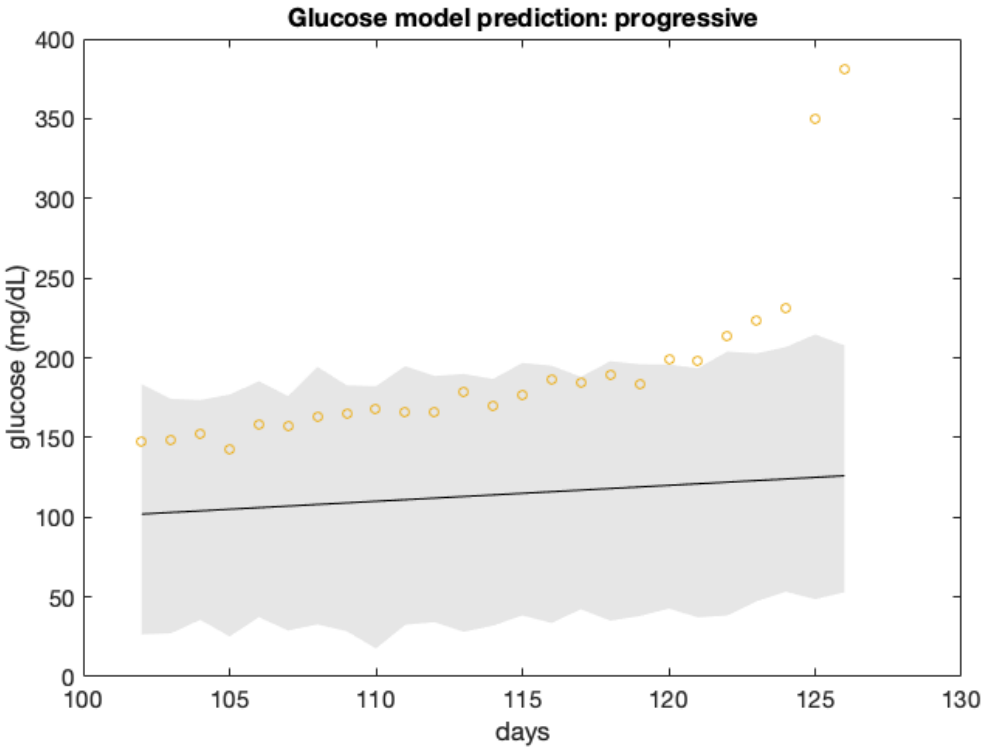


Figure 6: *mcmcstat* library model prediction plot. The black line indicates the best model prediction given the DRAM chain results. The grey range indicates the — of the predicted model. Plotted over this prediction is the original progressive Mathews et al mouse data.

We are still having difficulties with the built-in mcmcstat library model prediction plotting as you can see in this figure. This figure shows that the original data is partially captured in the predicted confidence interval. However, the prediction fails to capture the spike that indicates disease onset.

1.3.2 Acute Results

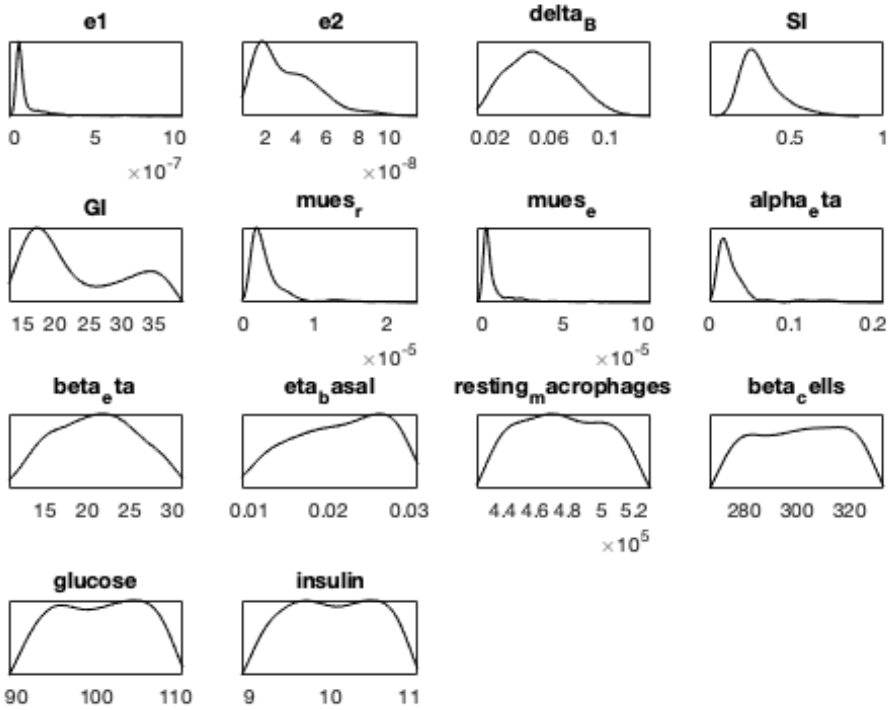


Figure 7: Posterior density plots for acute Mathews et al mouse data.

As we saw in the progressive results, we get a visual sense of the parameter distributions. Many of the parameters do not seem to have an easily recognizable distribution. We can see that some, like eta_basal, resting macrophages, beta cells, glucose, and insulin, look vaguely uniform. Others like e1, SI, $mues_e$, $mues_r$, and $alpha_eta$, look more lognormal. Still others, like e2 and GI appear to be bi-modal.

	mean	std	MC_err	tau	geweke	
e1	8.7978e-08	1.4509e-07	2.9873e-08	851.54	0.099927	
e2	3.3498e-08	1.9339e-08	3.8556e-09	970.83	0.14437	
delta_B	0.054001	0.020788	0.0028048	470.08	0.60101	
SI	0.33609	0.10862	0.014555	298.36	0.55712	
GI	24.176	7.7704	1.4866	1241.1	0.34601	
mues_r	3.5917e-06	3.217e-06	5.8075e-07	846.7	0.29911	
mues_e	9.2282e-06	1.5099e-05	3.1593e-06	880.59	0.085752	
alpha_eta	0.031893	0.030041	0.0060669	723.85	0.11125	
beta_eta	21.018	4.7455	0.39152	79.043	0.89458	
eta_basal	0.021334	0.0056186	0.00055972	128.49	0.98418	
resting_macrophages	4.761e+05	26685	2164.7	66.921	0.99569	
beta_cells	300.76	17.369	1.6113	107.62	0.96956	
glucose	100.41	5.7044	0.44281	60.32	0.95069	
insulin	10.032	0.573	0.048931	68.003	0.97666	

Figure 8: DRAM chain results for parameter set of 7 sensitive, eta, and initial conditions for acute Mathews et al mouse data. Most notable columns are the mean and geweke. Geweke value of > 0.7 we consider to have converged.

From this table, we can see how well our algorithm is performing in terms of convergence. We only have 6 parameters with geweke values of > 0.7 that we would classify as having converged. It is possible that the small number of parameters is limiting the overall parameter search space.

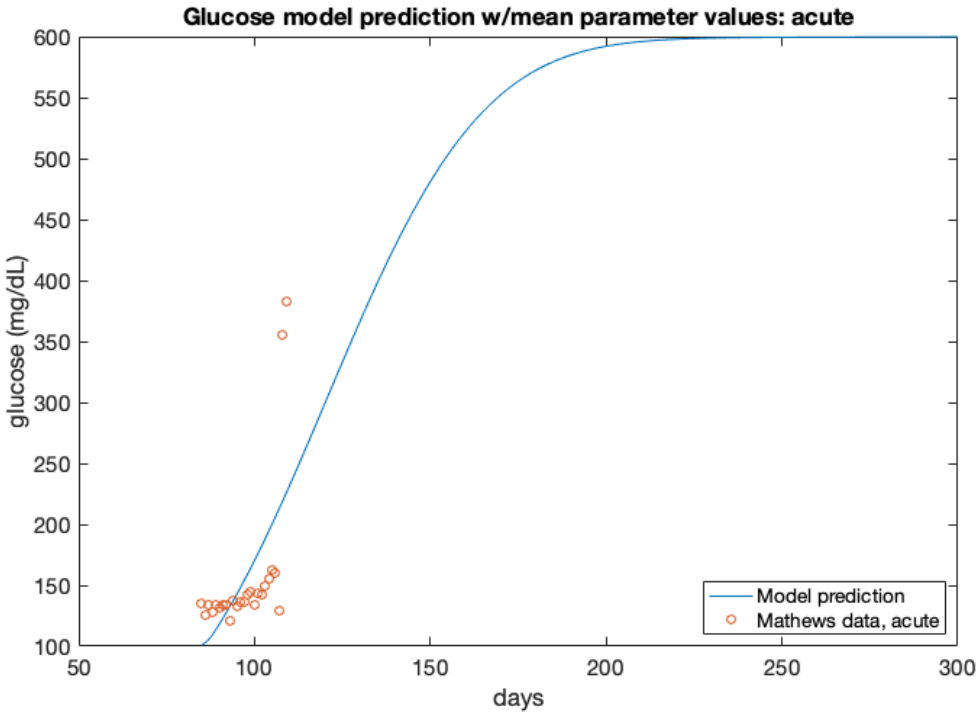


Figure 9: Predicted glucose measurement using post-DRAM mean parameter values (blue), plotted with acute Mathews et al. mouse data (orange). The model is predicted beyond the scope of the Mathews data to capture end behavior of the model.

In this figure, we can see how well the model using DRAM mean parameter values fit the Mathews data. Just as in the progressive data, the general trend seems to be in the same direction with a similar slope. However, the algorithm seems to have difficulty capturing the sharp spike that we recognize as diabetes onset. We plotted the predicted model for a longer time span so that we could see its end behavior.

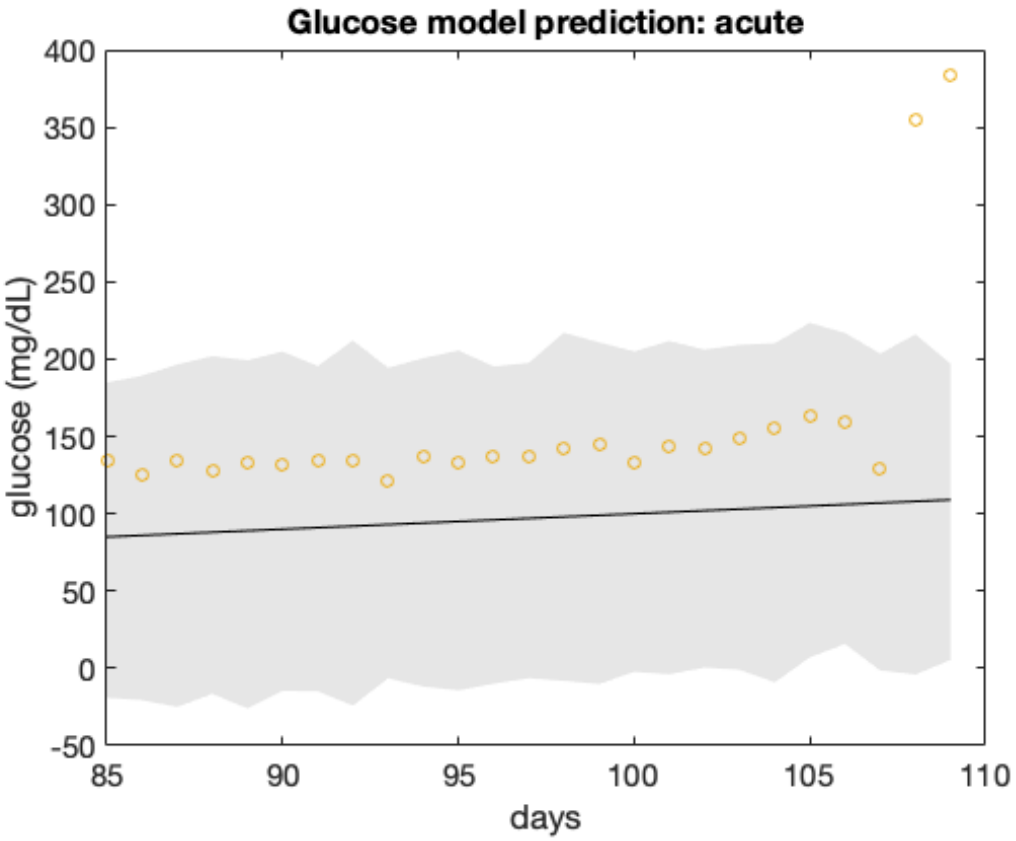


Figure 10: *mcmcstat* library model prediction plot. The black line indicates the best model prediction given the DRAM chain results. The grey range indicates the — of the predicted model. Plotted over this prediction is the original acute Mathews et al mouse data.

The mcmcstat library model prediction shows that part of the original data is partly captured in the predicted confidence interval. However, the prediction fails to capture last two data points and the spike that indicates disease onset.

1.4 Validation

There is not yet a quantification of how well DRAM is parameterizing and fitting the T1D model, but we can do a preliminary visual check using the Li et al data (doing a comparison similar to that in Figures 5 and 9). The hope is that the predicted model should be similar to the Li et al data.

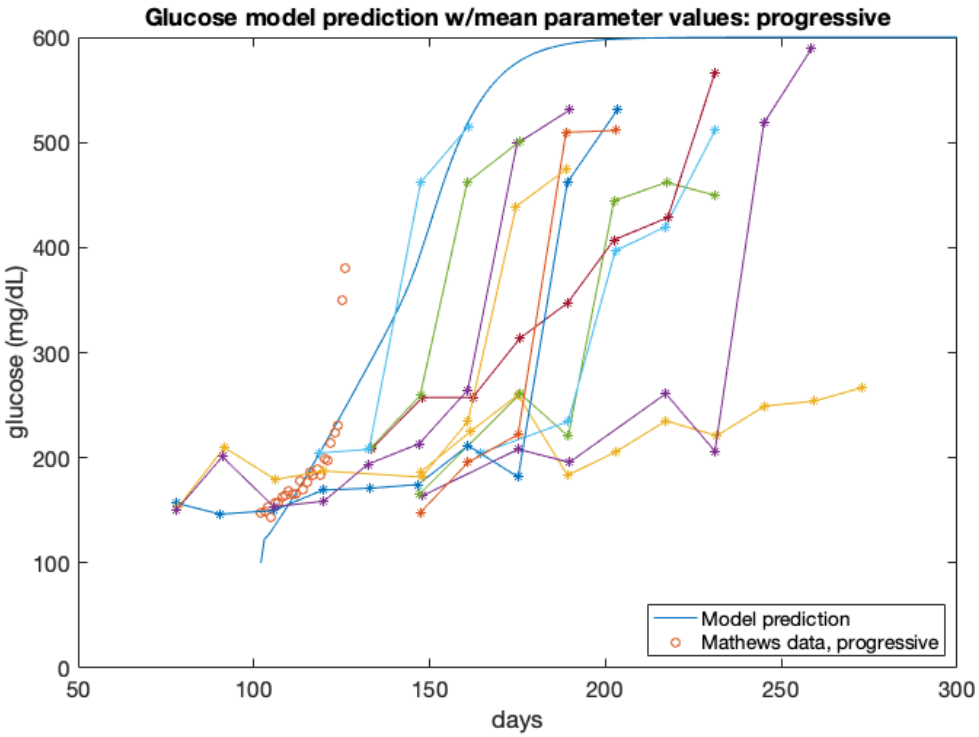


Figure 11: Plotting DRAM predicted model for progressive mice using mean parameter values, plotted against 11 mice from the Li et al data set

Here, we can see that the predicted model does a pretty good job of matching the overall shape of what we call the "z-shaped" mice of Li et al data (excluding the mice who do not have such a sharp increase in glucose). However, we note that the predicted model is on the earlier side of time of diabetes onset in comparison to the Li data. We attribute this discrepancy to the relatively large difference between the mean of the distribution of the onset times for Mathews vs. Li (about 25 weeks vs. about 28 weeks).

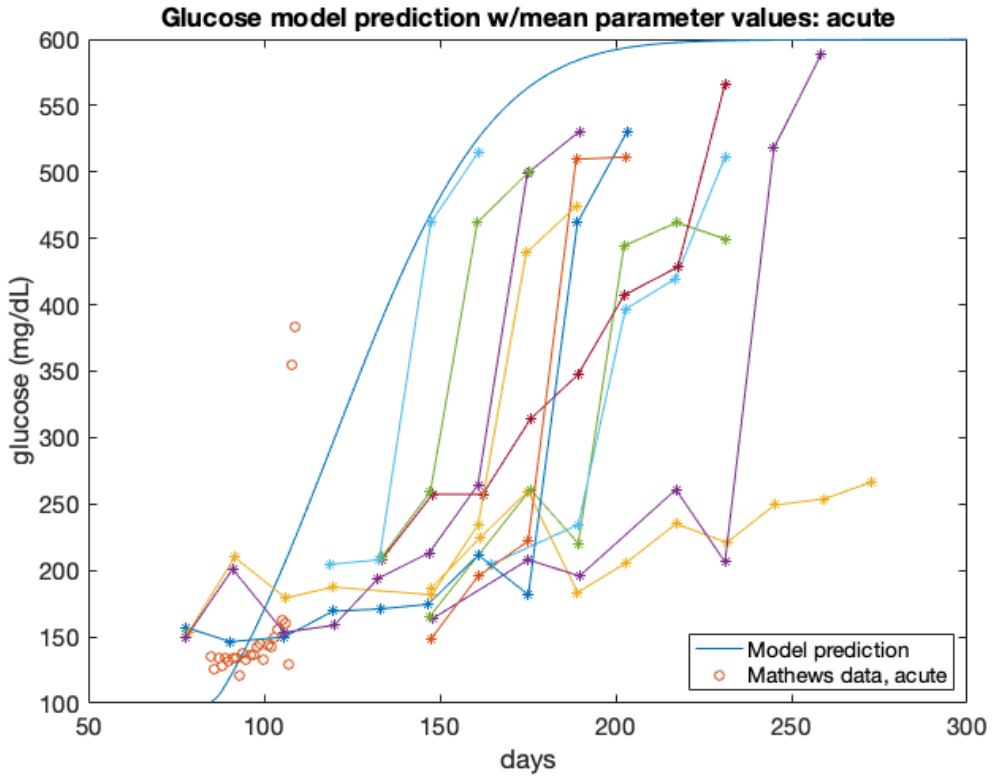


Figure 12: Plotting DRAM predicted model for acute mice using mean parameter values, plotted against 11 mice from the Li et al data set

Here, we see a bigger difference between the predicted model and the Li data. The predicted model begins with a lower overall glucose level than the Li data. The overall trend of the predicted model seems to be similar to some of the mice of the Li et al data (although it should be noted that we have not definitively classified the Li et al data as progressive and acute). Again, the predicted model is offset in time from the Li data. We hypothesize that the difference between onset time distribution means plays a role in the offset onset times.

While these visual validations are useful, eventually we would like to create a computational measure of error between our predicted model and the data.

2 PSO

2.1 Last Week

The 'sliding window' mechanism was implemented on the averaged and shifted Li et al. post-onset data in order to fit parameters to both the shape of the data and to its placement in time. Results of this, unfortunately, were highly variable. This variability was attributed to the lack of observed data before diabetes onset; essentially, the PSO algorithm could not correct for errors in this timespan, which lead to very naive fits. Additionally, the most likely point of onset identified by PSO was drastically different than that observed in the data. This was because the PSO algorithm assumed the onset time distribution from the Mathews et al. data, however the average onset time of the Li et al. data was over 7 weeks greater. Finally, the initial classification of the Li dataset into progressive and acute cases was incorrect. This was not immediately visible in the average data, however made a difference in the biological validity of the optimal parameters. Because of these errors and inconsistencies in the data, changes had to be made.

2.2 Data Changes

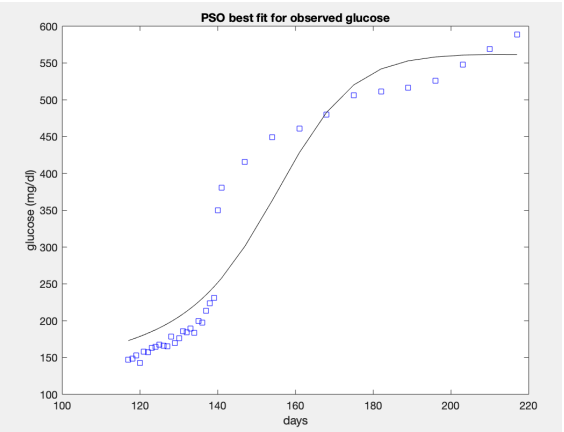
First, the Li data was re-categorized and averaged according to the algorithm described in Week 5's summary. To allow for greater knowledge regarding the pre-onset glucose behavior, data was also incorporated from Mathews et al. The Mathews data was shifted from its original, relative timespan to an absolute timespan using code by B. Shtylla (also referenced in Week 5's summary), and the Li data was concatenated to its end. Because the Li data used represented purely glucose levels *after* onset, and the Mathews purely *before*, the two were able to meet at the point of onset to form a reasonable picture of a progressive NOD mouse's glucose over time.

Running PSO on this dataset yielded a more confident fit and parameter set, as well as greater consistency. However, many assumptions go into creating the combination of the datasets; assumptions that cannot be biologically justified. This is why the idea of combining the datasets was eventually abandoned for a training-validation strategy (section 2.6).

2.3 PSO Algorithm Changes

To increase the efficiency and accuracy of the PSO algorithm itself, steps were taken to reduce the parameter search space fed to MATLAB's *particleswarm* function. Initially, disregarding parameter subsets, all parameters were allowed to vary a percentage of their value above and below the baseline provided in the original ODE system. This was problematic because of the lack of biological relevance. Next, the ranges found through eFAST analysis and *a priori* knowledge of biology were used. These were helpful to guarantee that only reasonable parameters were being selected, but the search space was still quite large. This meant that many iterations were required to reach the global minimum: a computationally intensive task. Finally, to reduce the search space even further, variance values from Team 2's UKF were manipulated to create bounds. Assuming the parameters followed a Gaussian distribution about the baseline, the 95 percent confidence intervals were used as upper and lower bounds. This allowed for significant variation in parameters, but prevented them from existing in the very extreme tails of the distribution. With this, it is important to note that PSO does not assume a normal distribution and its sampling occurs in a uniform fashion, just within what would be the confidence intervals of the normal.

Using the variance to calculate bounds not only decreased runtime by reducing the number of required iterations to find the optimal parameter set, but it seemingly improved the fit itself. The onset time was more consistent with the data, and the fit of the pre-onset data was much better. Unfortunately, beyond the 'eyeball test', we did not have a definite way of quantifying this, so a metric was developed.



(a) Fit using eFAST bounds

(b) Fit using variance-determined bounds

Figure 13: Comparison of best fit model before, after structural and data changes

2.3.1 Scoring Fit

Sum-of-squares in an effective way of evaluating the numerical differences between a model and observed data. However, it lacks intuition regarding how good a fit may 'look', and whether it represents the trend that we, as humans, may want it to. This means that, though it makes a good objective function, it it less informative for quantifying how good a fit may be.

To get around this, a method was developed to translate the two main benchmarks of the visual test into math. When examining the model fit, the two most important aspects are 1) the onset time, and 2) the shape of the curve. In sum-of-squares, these aspects cannot be separated as the onset time shifts the data points being compared to determine the shape. In this new fit scoring metric, two separate scores (one for onset time, one for shape) negate this.

The time score, s_{time}, is simply defined as the absolute difference between diabetes onset time in the data, t_{dat}, and the predicted diabetes onset time in the model, t_{mod}. Onset time was defined as the time when glucose levels above 250 mg/dl were observed consecutively, in agreement with Mathews et al. Mathematically,

$$s_{time} = |t_{mod} - t_{dat}|$$

The second score, the shape score, compares the shape of the simulated and observed curves. This comparison does not consider that these two curves may occur at different times, it simply aligns them at their onset times for a fair comparison. The shape score is calculated by splitting both the data and model output at their respective onset times. The pre-onset points from the model and the pre-onset points from the observed data are compared using sum of squares. The same is done for the post-onset points:

$$s_{shape} = (t_{mod}^{pre} - t_{dat}^{pre})^2 + (t_{mod}^{post} - t_{dat}^{post})^2$$

In the way that the code is written surrounding these two scores, a user-specified function could be used form a single value. However, because of the need for many assumptions regarding the weighting in this process, these scores were left separate.

2.4 Parameter Subsets

Building on (and correcting) work from last week, several parameter subsets were tested. Parameterizing just a few values reduces complexity and allows more of the behavior of the original, ideal system to be conserved. There are many ways to select these parameters, all detailed below. Some of this work was redundant upon our work from last week, though the new data and ability to quantify the goodness of fit made repetitive work worthwhile. The identified subsets are as follows:

- Glucose-related: parameters involved in the glucose equation, as glucose is the observable of the system
- eFAST Sensitive: parameters with large impact on the final glucose readings according to eFAST
- Onset time-related: eta parameters with heavy influence on the onset time, scaling
- UKF 'Active': parameters identified through UKF that have experience significant changes ($\geq$1 percent away from baseline)

For each of these subsets, PSO was run. All parameters were allowed a very small 'wiggle' of $\pm$1 percent, but the parameters within the subset were allowed to travese anywhere within their bounds. Allowing small movements in all parameters accounts for dependencies between parameters in and out of the subset in question. When run on the Mathews-Li combined data, the scores in Table 2 were produced. Not surprisingly, allowing all of the parameters to vary produces a reasonably good shape score and the best time score of the bunch. Very generally, it appears that PSO struggles to produce both a good fit for shape and a good fit for onset time. The best shape fits sacrifice their accuracy in onset time. The same goes for the reverse. Of all of the subsets, the UKF 'Active' subset seems to best balance shape and time accuracy. Though more subset combinations could be tested, in scenarios where it isn't viable or necessary to optimize all parameters, using this subset would be best.

Parameter Subset	Time Score	Shape Score
All	2	7.618 x 10^4
Glucose-related	27	6.067 x 10^4
eFAST Sensitive	27	6.834 x 10^4
Glucose + Sensitive	27	7.392 x 10^4
Onset time-related	2	9.122 x 10^4
UKF 'Active'	2	8.840 x 10^4

Table 1: Fit scores of various parameter subsets for the combined Mathews, Li dataset

2.5 Parameter Distributions

In an effort to make DRAM MCMC and PSO more comparable, our PSO routine was altered in order to produce posterior distributions of parameters rather than simply a single, best-fit set of parameters.

PSO is a global algorithm, so ideally a distribution would not capture any information about the parameters (as all outputs would be the same). However, when working with a relatively low number of iterations (600), a small swarm (20), and a high-dimensional parameter space, PSO often returns different values; either local minima, or points close to but not equal to the global minimum. These values can be used to shape a distribution if PSO is run multiple times.

This is a computationally intensive task, so a pared-down run was completed with reduced swarm size (10) and iterations (200). This was run 100 times (about 5 hours runtime) on just the Mathews et al. data (for reasons explained in Section 2.6) varying all parameters. Histograms were produced for all parameters. These histograms are pictured for the best-fitting subset, the UKF 'Active' subset. If there were more trials run, perhaps the histograms could have been resemblant of common distributions, however this was not the case for the histograms produced with a low number of trials. We would have fit appropriate distributions if they were more obvious, but we didn't feel comfortable making such big assumptions about shape.

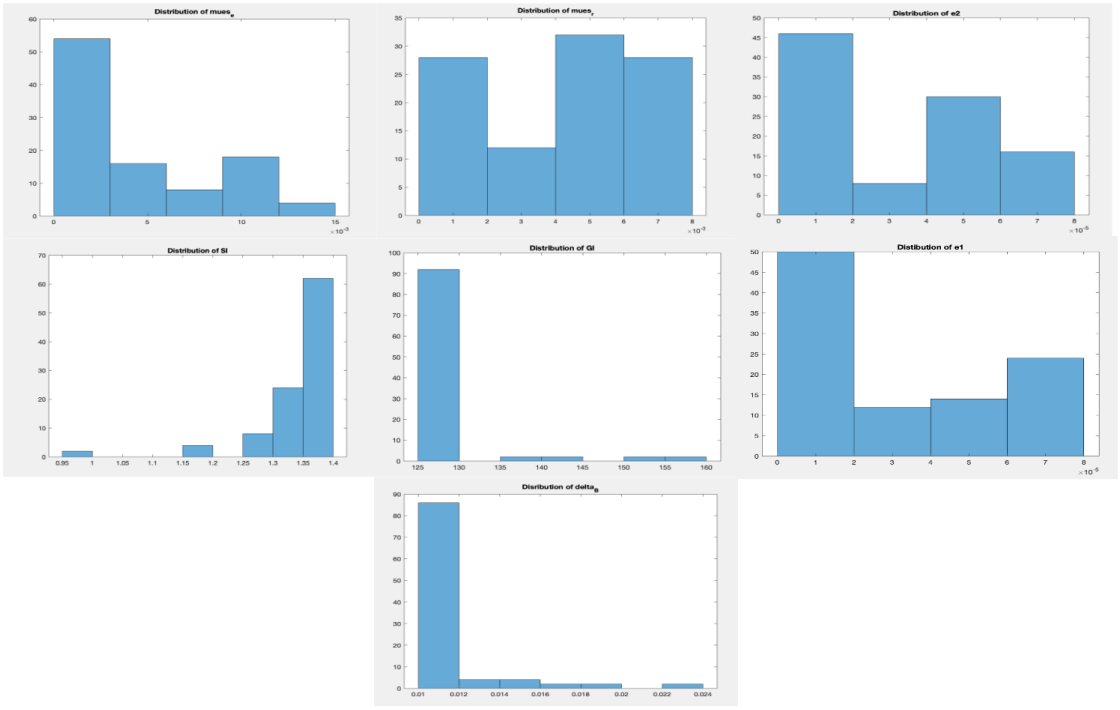


Figure 14: Histograms for UKF 'Active' parameters after 100 iterations of PSO

Parameters	Time Score	Shape Score
Mean	103	7.359 x 10_4
Median	104	7.270 x 10_4
Mode	N/A	1.147 x 10_5

Table 2: Validation fit scores of mean, median, and mode optimal parameter sets from Mathews on Li

2.6 Validation

About halfway through the week, questions arose about the composure of the Mathews/Li combined dataset. How were we sure that manipulating the data in the way that we did was a 'legal' move? Quickly we realized that we might have manipulated the data to a point where is looked nice, but we were not certain as to whether it was valid. This caused us to change our approach. Instead of initially combining the datasets, we set forth to create a training-validation model wherein the Mathews data would be parameterized, and then its accuracy assessed by scoring the fit of the optimally parameterized system against the Li data. If the model that fits well to Mathews (pre-onset only) fits well to Li (pre and post-onset), it is likely that the optimal parameters found have some validity to them. This is still a work in progress, so there is not a distinct threshold for a 'good fit', but irregularly large values for either the time or shape score would suggest a poor fit.

This idea of validation was performed on parameters drawn from the histograms. If we were able to produce true distributions, parameters would have been sampled from the distributions instead of the histograms. The mean, median, and mode parameter sets were extracted, overplotted, and scored against the Li data.

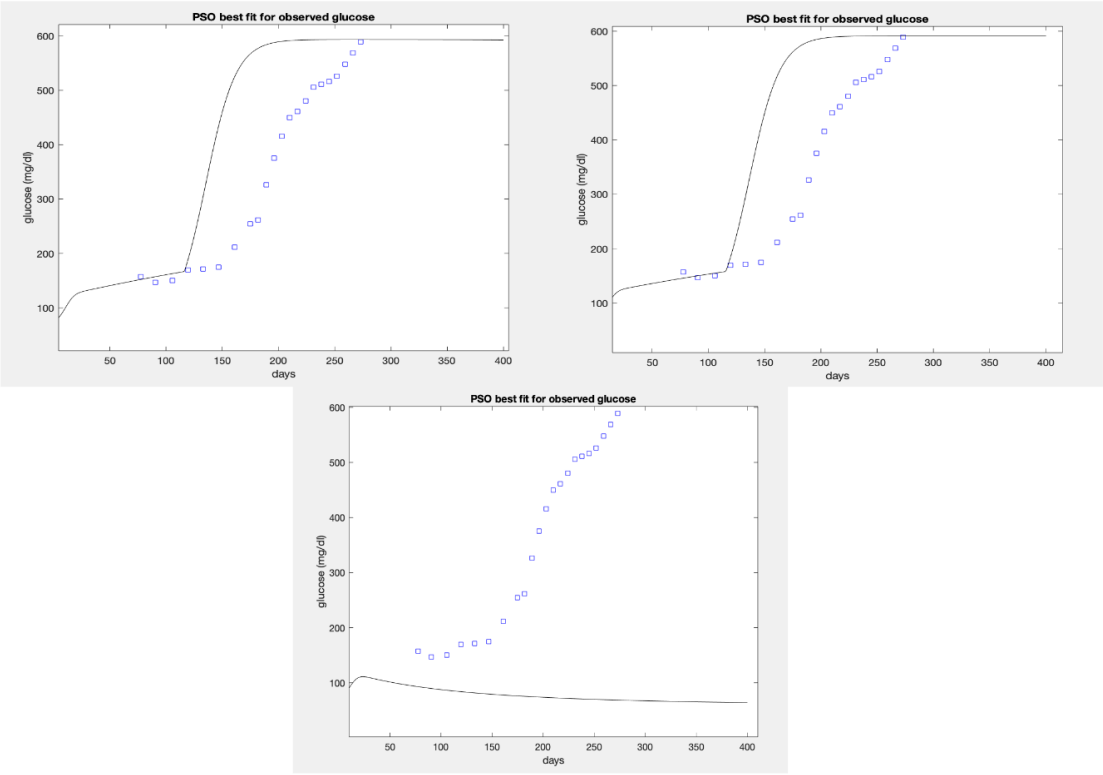


Figure 15: Mean (L), median (R), and mode (bottom) parameter sets fit to Mathews data; plotted against Li

To our surprise and relief, the shape of the data produced was fairly good for the mean and median

values. Their shape scores were consistent with the shape scores of the Mathews/Li combined dataset for all parameters. Their time scores, however, were very poor. This can partially be attributed to the difference in average onset time between the Mathews and Li mice (18 wks, vs. 25 wks). This would account for about a 49 day shift, but this still leaves over 50 days of error for both the median and mean parameter sets. As for the mode parameters, performance was very poor in both time and shape. The simulated mouse never reached the glucose threshold for diabetes. As such, the shape was incredibly different, producing a large shape score. Intuitively, since PSO is a global algorithm, it makes sense to use the mode parameters (a global algorithm should frequently reach the global minimum), but clearly the number of trials was not large enough to give this result. Additional factors such as the numerical methods employed by MATLAB to run PSO could also be in play; the accuracy and numerical methods of this specific implementation of PSO is unknown.

3 Moving forward

3.1 MCMC

There are still several tasks we would like to complete that we hope will help with analysis and fit.

1. It would be ideal to have a function to quantify the error in between the predicted model and the original data. The simplest option would be (as Christina did for PSO) to simply take the absolute value of the difference between the model and data then assign the model a score.

2. The Mathews et al data is a large data set, but as we have discovered, it is not very clean or easy to work with. Although the Li data has much less overall data, it would still be interesting to parameterize on this data set to see if our fits improve.

3. The main focus of determining how well DRAM was parameterizing the T1D model was looking at the glucose output. By plotting our data against the predicted glucose, we could get a visual sense of how well it was doing. It would also be useful to look at the other variables in the ODE system (insulin, beta cells, macrophages, etc.). Of course, we cannot directly compare these predictions to any raw data, but we could compare them to figures in the original paper, to at least see if they are biologically feasible. This should give us a better overall sense of how well DRAM is parameterizing the system.

3.2 PSO

There is much to explore in the realm of validation. Determining a reasonable function that weighs both the time and shape scores to combine them into a single value would make model comparison much more straightforward. Additionally, more work could be done with selecting parameter subsets, either selecting new groups of parameters or combining ones that have already been tested. This could even be done programatically: determining the group of n parameters that produces the closest fit. Finally, PSO would benefit from having more trials run to create more definite distributions. It would be really interesting to run a large number of trials given access to a supercomputer.

3.3 Preliminary ideas for merging UKFs and MCMC algorithms

Merging the UKF and DRAM MCMC algorithms has been tossed around in several discussions now. Since MCMC is such a general algorithm for sampling of any kind (not necessarily parameterization), there are many ways to integrate MCMC and UKFs (one of which was discussed by subteam 2 in their Friday slides). One method that has been discussed is using the final UKF parameter distributions to inform the prior distribution of the MCMC algorithm. Looking briefly into the mcmcstat library, we believe that we have found a way to do this. It would go as follows:

1. Run UKFs

   (a) Obtain parameter values and fit distributions (if possible)

2. Write a custom prior function that can take in those UKF distributions
3. Run MCMC using this new informative prior