**Summary Documentation of 2021\_ParameterOptimization\_withUQ**

5/26/2021 – Hayley Boigenzahn

Purpose of program: Fit parameter values to a set of ODEs representing peptide ligation/hydrolysis reactions.

**Important Note:** Highly recommend running in MATLAB 2019a.

* (MATLAB was screwing around with how optimization problems are initialized in the past couple of years. As I recall I tried it in MATLAB 2020a and it threw errors, so it’s safest to assume that it’s extremely version dependent.)

**Important Note:** This code sort of grew bigger than I anticipated when I first started working on it. If you see something that looks like a weird decision or an error, it may or may not be intentional. Let me know.

SolvedModel.m

This file does the plotting/post-processing stuff. **This you can run.** It assumes that you have the variables saved at the end of a run of ODE\_optimizer already loaded into the Workspace. **(Ie, you need to double click on FinalFit-1.mat first.)**

This file sets up the optimization problem in reverse, solving the ODEs using the parameters found in the ODE\_optimizer file. Then it plots the results of solving that ODE for various initial conditions with the experiments corresponding to those initial conditions.

It also should theoretically calculate the final value of the error function for the consensus parameters, but for some reason I’m having some difficulties getting this to line up with the results from the ODE\_optimizer file, so there may be an error somewhere.

reactions.m

Implements and documents all reactions being fit. Use these numbers to assign reactions to the r values produced by the code.

ODE\_optimizer\_allvers.m

This is the main file of the program. **(Warning: Without adjusting mc\_iterations and samples, this file will take many hours to run, so I recommend against it.)**

Because I’m a bad programmer, it does basically everything internally instead of splitting it into functions. The basic sections, separated by %% lines, are as follows:

1. Load experimental data and experimental standard deviations
2. Declare initial conditions and time spans for the ODE model
3. Create the reaction parameters, which will be varied to try to fit the experimental data
4. Create the optimization function using SolveWithFoundParams.m and reactions.m
   1. I don’t declare the error function until the next section, but I’m using ((x-μ)/σ)2 instead of the usual sum squared error because the concentrations of longer peptides are 3 or 4 orders of magnitude smaller than the monomers and I don’t want them ignored completely
   2. The error for each dataset is calculated individually and summed together because I couldn’t think of a better way to handle fitting multiple datasets simultaneously
5. Psuedo-Monte Carlo procedure to select the initial guesses for the reaction parameters
   1. This is necessary because the initial conditions sometimes have a significant effect on how parameters converge.
   2. I don’t use a strict Monte Carlo method where I’m looking to get below a certain error threshold because the error varies wildly depending on which experimental data I’m using. Instead, I just randomly vary the guesses for a certain number of steps, keeping the guess with the lowest error unless an unfavorable step is randomly accepted.
6. Uncertainty Quantification procedure to estimate how much the parameters vary due to relatively small changes in the experimental data.
   1. I just make a bunch of noisy replicates of my experimental data and fit the system repeatedly from the same initial guess for reaction parameters
7. Take the average of the uncertainty quantification solutions as the final parameter solution and save all the variables in the MATLAB workspace.