ODE Modeling Exercises

Exercise Bonus: ODE Model of Glycolysis and Parameter Estimation

Assume you have conducted an experiment, where *E. coli* cells were grown in a medium where **Glucose** was the sole carbon and you have measured the concentrations of all metabolites that appear in Figure 1. The data for every metabolite and the corresponding sampling times are provided in the following files:

- **ExperimentalData.mat**: a 30x7 matrix containing the data for every sampling point (1 to 30) of every metabolite (1 to 7).
- **StandardDeviation.mat:** a 1x7 matrix containing the standard deviations of every metabolite (we assume that every metabolite has the same standard deviation over time, for simplicity)
- **SamplingTime.mat**: a 30x1 matrix containing the information regarding which time point corresponds to which data point (which row of ExperimentalData corresponds to which time point).
- **MetaboliteNames.mat**: a 1x7 matrix containing the information regarding which metabolite corresponds to which column of the ExperimentalData.

The goal of this exercise is to use the available experimental data of the 7 metabolites we measured in order to make our computational model better, in terms of how accurately it describes the real biological system's behavior. With this as a goal, you will perform the following tasks/exercises.

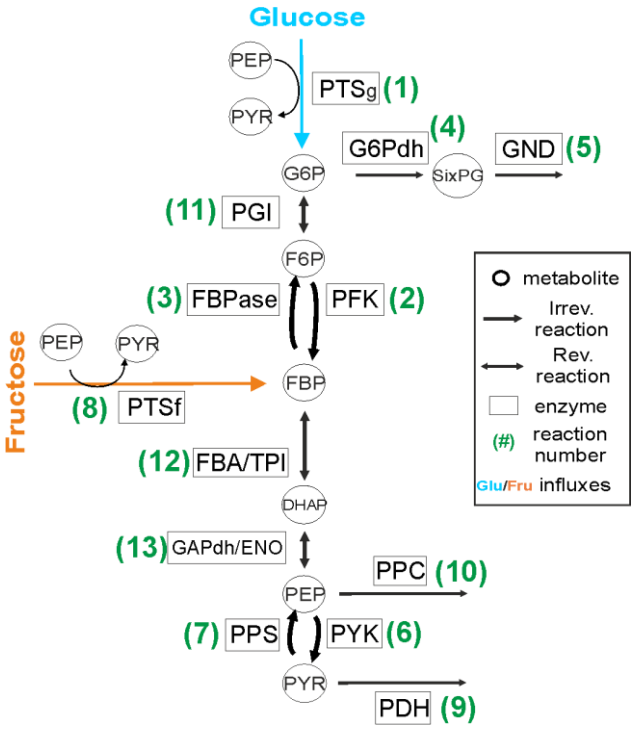

Figure 1: Reaction scheme of Glycolysis

Evaluation of a model's simulation results with respect to available experimental data.

In order to evaluate how accurate our current model is compared to the available experimental data (goodness of fit – GOF), we need a metric. We will use the distance between simulation and experimental data, taking measurement error into account, as a metric . The formula is the following:

$$\sum_{i=1}^{m}\sum_{t=1}^{n}\left[\frac{(y'_{i,t}-y_{i,t})}{\sigma_i}\right]^2 \text{, Equation 1}$$

where m is the total number of metabolites (here 7), n is the total number of time points where we have experimental data (here 30), $y'_{i,t}$ is the simulated value of metabolite i in time point t, $y_{i,t}$ is the (real) value of metabolite i in time point t, as determined by the hypothesized experiment and σ_i is the standard deviation of metabolite i (here the standard deviation of every metabolite is assumed to be constant over time, for simplicity). Ideally, if the simulated results are perfect, we get a score of zero (perfect accuracy). If not, we get positive values and the higher the value is the worse our model is.

You are given a function named "**scoreModels.m**", which implements this metric. What it needs as arguments are the following three matrices in this order:

1. ExperimentalData matrix,
2. SimulationData matrix
3. StandardDeviation matrix

and gives as output the "score" of this particular model: how well the simulation results that come from this parameterization of the model agree with the experimental data.

1) Read the code of the function "scoreModels.m" and try to understand how it works and if it implements correctly Equation 1 .
2) Load the following files in your workspace: **ExperimentalData.mat, StandardDeviation.mat, SamplingTime.mat** and **MetaboliteNames.mat.**
3) Simulate the model of Glycolysis for the default parameters that were given to you in Exercise 4. In case you have not managed to complete the model of Glycolysis in Exercise 4, a function that implements it has been provided to you (GlycolysisModel.m). You will simulate the model with the default parameters for **30 seconds** and you will simulate it for the exact time points for which we have experimental information for (SamplingTime). This is crucial, because we want to compare the simulated results with the corresponding experimental data in the **same time points**.
   Hint: To do so, you have to set the time vector for which we will simulate our model, in the code in the following way, before you run the "sbiosimulate()" function:
   `set(configsetObj.SolverOptions, 'OutputTimes', TIME_POINTS_OF _SAMPLES)`
4) Plot the results of your simulation with the experimental data you loaded, for every metabolite, in different plots.
5) Use the **scoreModels** function and provide the score of the current model with respect to the experimental data.

Parameter uncertainty and parameter estimation

From the results of 3) and 4) you can see that our model does not perfectly recapitulate the data we have from the experiment. This could happen for many reasons, but we have very good reasons to believe that one parameter that we use in the model is not the correct one. Let us assume now that we know that this parameter in our case is the value of **Vmax2**. We will try to use the available experimental data and the scoring function in order to try and identify a **better** value for parameter Vmax2. We will attempt to **estimate** its value.

6) Implement a parameter scan for parameter Vmax2 of your computational model of Glycolysis, in the range from 0.5 mM/sec up to 3.5 mM/sec with a step of 0.05. For every different parameter, simulate the model for **30 seconds**, for the exact time points for which we have experimental information for (SamplingTime). Use the provided scoring function and the provided information regarding the experimental data in order to evaluate the goodness of fit of every different parameterization of the model. Store this value for every different parameterization of the model in a matrix.
   Hint: in order to set a new value for a parameter you need to do the following:
   First get a handle for the parameter.
   ```
   parameter=sbioselect(modelObj,'Type','parameter','Where','Name','==','Vmax2');
   ```
   Then set the parameter with the value you want
   ```
   set(parameter,'Value',VALUE_YOU_WANT);
   ```

7) Identify the parameter value of Vmax2 for which the model performs best (has the lowest value in score).

8) Plot the different scores you acquired for every different parameterization of the model against the different parameters.