

rate_extrapolator.py

Functional Specifications

Updated: 11/18/21 (Sarah Wait)

Table of Contents:

Background	2
User Profile	2
Experimentalist	2
Computationalist	2
Use Cases	3
Rate Constant Determination	3
Objective	3
Expected Interactions	3
Model Determination	3
Objective	3
Expected Interactions	3

Background

In order to properly model any molecular interaction, the kinetics law needs to be determined. This can be determined through trial and error, where the user tries an array of constants and chooses the fit that best fits wet lab observations. This method is time consuming and leads to issues with model reproducibility. Another method is through determination of rates by analyzing empirical data. Though, this method requires demanding signal processing and is typically handed over to the computationalist to make decisions about how the rate fits the observed phenomena. My hope is to simplify the process such that the experimentalist can observe the extrapolated rate constants and make their own decisions regarding how well the constants are able to describe the data.

User Profile

Experimentalist

The expected experimentalist user will have some coding experience in python and familiarity with the tellurium package. Ideally, the user will be able to import python packages and understand how to call python functions. The user should be able to write tellurium antimony and identify rates as strings. However, the function is meant to be simple enough to be a “plug and chug” analysis of rates.

Computationalist

For the computationalist user who is well versed in tellurium, they will have the capabilities to control some portions of the generated antimony model. They will be expected to be familiar with the inputs to the tellurium packages modeling such that they can modulate said functions when calling the function.

Use Cases

Rate Constant Determination

Objective

The objective of this function is to provide approximate rate constants that are governing the changes affecting their observed substrates. The function will determine rate constants that describe the behavior of their systems and provide statistical confidences of model/parameter fitness.

Expected Interactions

The users will input experiment derived data as a single list describing the observed concentrations of a substrate. Users will call the function and input experimental data as data type `pd.DataFrame` **OR** `csv`. They will also supplement an antimony string with unknown constants and a list of parameters (constants) to be estimated as a string. Output will include plots of experimental data vs. the model with the best estimated rates. It also generates a histogram of extrapolated rate values with heatmapped R^2 . Counts will be the number of folds. Table generated will pull data from SBstoat model fitness information and provide rates determined, fitting method, R^2 , Chi-squared, Reduced chi-square, Akaike info crit, Bayesian info crit. The user will need to be able to understand the plots and make educated decisions of the rates from the statistics and modeling.

Model Determination

Objective

For substrates with unknown upstream reactions or multiple reactions governing subtraction concentration, the function provides the possibility of decomposing if the suspected model actually describes the experimental phenomena. In this use case, the user could supplement different antimony strings + extrapolation rates to determine which proposed model best fits their data. The function would not make these decisions alone, the user would need to analyze the resultant graphs/stats/simulated data to decide which model best fits the data.

Expected Interactions

An advanced user would be able to write multiple antimony strings and determine which best fits their data by analyzing the graphs and statistics outputted by the function. However, the expected interactions between the user and the function are identical to the previous use case. The users will input experiment derived data as a single list describing the observed concentrations of a substrate. Users will call the

function and input experimental data as data type `pd.DataFrame` **OR** `csv`. They will also supplement the suspected model as an antimony string and list of parameters to be estimated as a string. In this use case, the output will include plots of experimental data vs. the model with the best estimated rates. Table generated will pull data from SBstoat model fitness information and provide rates determined, fitting method, R^2 , Chi-squared, Reduced chi-square, Akaike info crit, Bayesian info crit. After fitting with multiple antimony strings, the user can analyze resulting graphs and make educated decisions on which model best fits their experimental results.