rate_extrapolator.py

Component Specification

Updated: 11/12/21 (Sarah Wait)

Table of Contents:

Software Components

rate_extrapolator

The rate_extrapolator function will act as a wrapper function and the main manager of the aforementioned software components and is the function that is directly interfaced with the user.

Inputs

- Experimental data as dataframe where first column corresponds to time and subsequent columns correspond to substrate
- Tellurium antimony
- List of rates as strings

Outputs (as PDF)

- Figure 1: 0xn subplot figure where n = the number of substrates given for analysis by user.
    - (0,n) Experimental Data plot on axis of concentration vs time as scatter with best fitted parameter tellurium model plotted as line
    - With residuals plotted as subfigure below x-axis
- Figure 2: 0xm subplot figure, where m = number of rates extrapolated.
    - (0,m) subplot of the histogram of rate determined by SBstoat with heat-mapped R^2 values
- Table 1: Stats table
    - Explicitly state rates determined
    - Fitting method
    - R^2
    - Chi-squared
    - Reduced chi-square
    - Akaike info crit
    - Bayesian info crit

k_folds_data_splitter

This function takes the experimental dataframe provided by the user in the inputs to the wrapper function and performs k-fold splitting of the data into train/test splits.

Inputs

- User provided data, datatype: dataframe.
- K-folds if designated by user, otherwise # of folds will be the floor of 10% of provided data.

Outputs

- Dataframe:
  - Dimensions: 2xk-folds
    - Rows (0:test,1:train)
    - Columns: (0,...,K-folds-1)
  - Contents of cells are dataframes of split data.

	0	1	2	3	4
Test	time S1 S2 S3 0...	time S1 S2 S3 1 ...	time S1 S2 S3 2 ...	time S1 S2 S3 3...	time S1 S2 S3 4 ...
Train	time S1 S2 S3 1...	time S1 S2 S3 0...	time S1 S2 S3 0...	time S1 S2 S3 0...	time S1 S2 S3 0...

SBstoat_model_fitting_to_folds:

This function determines rate constants of provided antimony model on the k-folds determined k_folds_data_splitter

Inputs

- K_folds_data_splitter dataframe
- Antimony, from user, provided as input to wrapper function
- List of rates

Outputs

- Dataframe:
  - Dimensions: K-folds-1 x n rates fit.

	k1	k2	k3
0	0.067906	8.628230	0.627521
1	220.985812	296.653293	0.315899
2	516.064715	297.694048	0.317722
3	0.001062	450.792070	0.567133
4	0.189155	2.400910	0.607697
5	131.823983	0.003567	0.564810

parameter_estimation_fittness_evaluator:

This function takes the estimated rates provided by the SBstoat_model_fitting_to_folds df and performs tellurium modeling of provided rates. The output simulation of the tellurium model is evaluated against fold's test set and R^2 value is determined and appended as a new column within the same dataframe.

Inputs

- K_folds_data_splitter dataframe
- Tellurium road-runner model
- List of rates
- Estimated rates in each fold dataframe from SBstoat_model_fitting_to_folds

	k1	k2	R Squared
21	0.496701	0.905934	0.997805
11	0.495505	0.911081	0.997346
14	0.497246	0.911616	0.996873
20	0.496156	0.907144	0.996588

Outputs

- Dataframe:
    - Dimensions: K-folds-1 x n rates fit + R^2 column.
    - Sorted by 'R Squared' column from largest to smallest.

te_model_of_best_estimated_params

The tellurium modeling section will take the top rates determined by the parameter_estimation_fittness_evaluator and simulate over the same time-course as experimental data.

Inputs

- Road-runner TE model
- Outputted parameter_estimation_fitness_evaluator df, first row.

Outputs

- Te model simulation

figure_generator

Visualization manager generates a final report that will be passed to the user. It will take the curated information and transfer it to a visualized report that the user can make decisions from.

Inputs

- Experimental data (from user)
- Te_model_of_best_estimated_params output simulation
- Parameter_estimation_fittness_evaluator output df

Outputs

- Figure 1: 0xn subplot figure where n = the number of substrates given for analysis by user.
    - (0,n) Experimental Data plot on axis of concentration vs time as scatter with best fitted parameter tellurium model plotted as line
    - With residuals plotted as subfigure below x-axis
- Figure 2: 0xm subplot figure, where m = number of rates extrapolated.
    - (0,m) subplot of the histogram of rate determined by SBstoat with heat-mapped R^2 values
- Table 1: Stats table
    - Explicitly state rates determined

- Fitting method
- R^2
- Chi-squared
- Reduced chi-square
- Akaike info crit
- Bayesian info crit

Summary of Components

- Rate_extrapolator (wrapper function)
  - k_folds_data_splitter
  - SBstoat_model_fitting_to_folds
  - parameter_estimation_fittness_evaluator
  - Te_model_of_best_estimated_params
  - figure_generator

Software Interactions

1. **Rate_extrapolator** reads the time course dataframe, generates floating variables initial time start, time end, and number of steps. Generates a list of SBstoat parameters to be evaluated from the number of rates designated by the user.
2. **Rate_extrapolator calls function k_folds_data_splitter.** K_folds_data_splitter takes the time-course data frame and parses it into a train-test for K number of folds. Output is a dataframe of dataframes that correspond to train and test data for each fold.
3. **Rate_extrapolator calls function SBstoat_model_fitting_to_folds.** SBstoat_model_fitting_to_folds function intakes dataframe produced by k_folds_data_splitter, list of SBstoat parameters to be evaluated (from rate_extrapolator, step 1), and the antimony provided by user to the rate_extrapolator function. Output is a dataframe of all estimated parameters for each fold.
4. **Rate_extrapolator calls function model_fittness_evaluator.** Model_fitness _evaluator takes inthe TE roadrunner, k-folds split dataframe, rate estimates df. Computes residuals from test set to simulated model with estimated rates. Returns model estimated fit as an additional column of R^2 values. The dataframe is also sorted from highest to lowest R^2 value.
5. **Rate_extrapolator calls te_model_of_best_estimated_params.** Te_model_of_best_estimated_params takes the best performing rates determined by model_fitness_evaluator and performes te model simulation using best rates.
6. **Rate_extrapolator calls figure_generator.** Figure_generator generates 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.

Actionable Items

- ☑ ~~11/4: Functional Specification and Component Specification Write Ups~~
- ☑ ~~11/9: Functional Specification and Component Specification Write Ups (again with the revised plan)~~
- ☑ ~~11/10: first pass run of all separate segments as different cells~~
- ☑ ~~11/11: functionalize all cells from the previous day~~
- ☑ ~~11/12: inter-link all functions and validate that everything works well.~~
- ☑ ~~11/13: packaging single wrapping function and generate testing with "fake data" (IE easy sinusoidal data with known frequencies and added noise)~~
    - ☑ ~~Function descriptors and pep8 compliance check~~
- ☑ ~~11/14: testing with more advanced data such as those contained in the wolf model and function tweaking to fix any issues that arise upon testing.~~
- ☑ ~~11/15: work with repositories and moving to a .py file, doing README, virtual env etc.~~