README and Data Documentation

Python code for: Modeling and Optimization of Complex Enzymatic Reactions: A Practical Guide for Biotechnologists and Bioprocess Engineers

June 2026

1 Project Information

Project Title: Python code for: Modeling and Optimization of Complex Enzymatic Reactions: A Practical Guide for Biotechnologists and Bioprocess Engineers
Related Publication:
Tillmann, M. T., Ibañez, M., Dittmer, K. R., Lu, J., Rodriguez Manotas, G., Ohde, D., Liese, A., Burger, J., Skiborowski, M., and Paschalidis, L. (2026). Modeling and Optimization of Complex Enzymatic Reactions: A Practical Guide for Biotechnologists and Bioprocess Engineers. *Reaction Chemistry and Engineering*, Submitted.
Corresponding Author: Leandros Paschalidis (leandros.paschalidis@tuhh.de)
Authors:

- Märthe Theresa Tillmann, Hamburg University of Technology
- Manuel Ibañez, Technical University of Munich
- Kayla Reata Dittmer, Hamburg University of Technology
- Jixia Lu, Hamburg University of Technology
- Gustavo Rodriguez Manotas, Hamburg University of Technology
- Daniel Ohde, Hamburg University of Technology
- Andreas Liese, Hamburg University of Technology
- Jakob Burger, Technical University of Munich
- Mirko Skiborowski, Hamburg University of Technology
- Leandros Paschalidis, Hamburg University of Technology

Funding Acknowledgement:
Financial support from the Collaborative Research Center (CRC) "SMART Reactors for Future Process Engineering," funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number: 503850735 at the Hamburg University of Technology (TUHH), is gratefully acknowledged.

2 General Information

Data Set Title: Python code for: Modeling and Optimization of Complex Enzymatic Reactions: A Practical Guide for Biotechnologists and Bioprocess Engineers
Short Description:
This repository provides a comprehensive computational toolkit written in Python designed to guide biotechnologists and bioprocess engineers through kinetic modeling, parameter estimation, and process optimization of complex enzymatic cascades. The code follows a step-by-step tutorial methodology spanning initial rate analysis, kinetic parameter determination under uncertainty (including global and Bayesian methods), dynamic simulation via progress curves, sensitivity analysis, validation against unseen data, and advanced reactor network synthesis via superstructure optimization.
Date of Code Creation: 2025-2026
Geographical Coverage: Hamburg, Germany
Keywords: Enzymatic Cascades, Parameter Estimation, Superstructure Optimization, Bioprocess Engineering, Kinetic Modeling, Python, Optimal Control, Bayesian Optimization, Sensitivity Analysis

3 Methodological Information

Data Collection and Processing:
The computational scripts process only synthetic kinetic data (created and used for education purposes). Model validation, parameter sweeps, and process optimizations are managed computationally. Dynamic simulations are carried out using standard numerical integration techniques to model enzyme systems suffering from complex inhibition networks (e.g., competitive, non-competitive, and substrate/product inhibitions) described in the accompanying text.
Software Packages and Solvers Used:

- Python
- Core Scientific Libraries: `numpy`, `scipy` (`optimize`, `integrate`), `matplotlib`, and `PYOMO and IPOPT`.

Experimental Design / Study Context:
The software workspace replicates the sequential learning curve detailed in the practical guide publication, categorizing tasks into primary lifecycle scripts and specialized multi-reactor structural optimization modules:

1. **Kinetic Characterization & Parameter Estimation:** Scripts dedicated to fitting early-stage initial velocities, modeling continuous long-term progress curves, calculating confidence intervals, and utilizing global/Bayesian frameworks to escape local optimization minima.
2. **Process Control & Network Synthesis:** Scripts focused on finding the absolute performance upper limits of biocatalytic cascades via time-dependent dosing strategies (optimal control) or macro-level equipment flow design configurations (`superstructure.zip`).

4 Data and File Overview

List of Files and Structure:
The repository consists of self-contained root scripts focusing on specific instructional modeling steps alongside a compressed archive hosting the multi-reactor network configurations:

File Name	Description	Format
Kinetic Characterization & Parameter Fitting Foundations		
`initial_rate.py`	Fits initial velocity data to Michaelis-Menten kinetic variants to extract initial parameter estimates ($v_{\max}$, K_m).	.py
`progress_curve.py`	Performs dynamic regression over full concentration time-series tracks.	.py
`confidence_ellipses.py`	Calculates parameter covariance matrices and plots parameter confidence ellipses to visually represent parameter correlations and statistical uncertainty.	.py
Advanced Global Estimation & Validation Suite		
`Bayesian_optimization.py`	Implements surrogate-driven Bayesian sampling algorithms to discover robust, global parameter maps across highly nonlinear or poorly initialized spaces.	.py
`validation_unseen_data.py`	Validates estimated kinetic structures against an independent, separate validation dataset to test model predictive power and safeguard against overfitting.	.py
Sensitivity Testing & Process Optimization		
`sensitivity_analysis.py`	Executes local parameter perturbations to identify the effect of certain parameters.	.py
`simulation.py`	Standard dynamic solver script configured to run forward time-series integration for user-defined multi-enzyme cascade models.	.py
`optimal_control.py`	Solves dynamic optimization trajectories (e.g., time-varying substrate dosing profiles or temperature schedules) to maximize product metric yield.	.py
Structural Synthesis Modules		
`superstructure.zip`	The zip folder contains two subfolders. One for the single reactor and one for the two reactor case. To run them, open the meta start file and press run.	.zip

5 Access and Licensing Information

Repository and Persistent Identifier: Published via GitLab / institutional repository.
License for Use: MIT License
Access Restrictions: Open Access
Title for Citation:
Python code for: Modeling and Optimization of Complex Enzymatic Reactions: A Practical Guide for Biotechnologists and Bioprocess Engineers.

6 Reproducibility and Software Dependencies

Software Required:

- Python 3 interpreter (Anaconda environment or standalone virtual environment recommended).
- Required tracking packages: `numpy`, `scipy` (`optimize`, `integrate`), `matplotlib`, and `PYOMO and IPOPT`.

Scripts and Workflow:
To run the primary modeling and analysis scripts:

1. Ensure all individual `.py` workspace scripts are arranged in your active project path.

2. To rub standalone files (essentially all files except the superstructure files), run the respective root python file directly inside your python compiler (e.g. spyder).

3. To run the superstructure case study extract `superstructure.zip` and execute:

   ```
   python meta_start.py
   ```

   The scripts reproduce the results in the paper. There is an indication within the paper, which codes can be used to produce which results.

7 Ethical and Legal Aspects

Data Protection: No personal or sensitive data are contained in this repository.
Consent Statement: Not applicable. No human participants or animals were involved.

8 Versioning and Updates

Version Number: v1.0
Date of Release: [17-06-26]
Change Log:
Version 1.0: Initial unified public release combining the complete tutorial framework scripts—ranging from initial rates up through the compressed superstructure optimization workspace directories.