Supplementary Information for

Teaching systematic, reproducible model development using synthetic biology

Kate E. Dray, Kathleen S. Dreyer, Julius B. Lucks, and Joshua N. Leonard[%]

[%]corresponding author

Affiliations, Address, and Contacts:

Dray: Northwestern University, Department of Chemical and Biological Engineering and Center for Synthetic Biology, Evanston, IL 60208

katelyndray2022@u.northwestern.edu

Dreyer: Northwestern University, Department of Chemical and Biological Engineering and Center for Synthetic Biology, Evanston, IL 60208

kathleen.dreyer@northwestern.edu

Lucks: Northwestern University, Department of Chemical and Biological Engineering and Center for Synthetic Biology, Evanston, IL 60208

jblucks@northwestern.edu

Leonard: Northwestern University, Department of Chemical and Biological Engineering and Center for Synthetic Biology Evanston, IL 60208

j-leonard@northwestern.edu

GAMES Homework Set

Submission instructions: *Submit your homework set by uploading a compressed .zip file with the following:*

- *A single Word document with your written solutions (including embedded plots when prompted)*
- *Your GAMES output folders for each relevant question (2b, 2c, 3a, 4b)*

Overview: *In this problem set, you will use a workflow for iterative ordinary differential equation (ODE) model development and parameter estimation (presented in Dray et al.) to analyze models describing gene regulatory systems. You will evaluate and apply a parameter estimation method, demonstrate iterative model formulation and analysis, and interpret results of a parameter identifiability analysis. We will focus on a case study of a gene regulatory system called COMET—the COmposable Mammalian Elements of Transcription, as described in Donahue et al[1]. COMET includes an ensemble of synthetic transcription factors (synTFs) and promoters that enable the design and tuning of gene expression, along with an ODE model that describes the relationship between transcription factor plasmid dose and reporter expression. Note that this paper preceded the development of GAMES and therefore used different, but still appropriate, model development and parameter estimation strategies. Here, we will apply GAMES to analyze and compare models of COMET synTFs. Specific learning objectives for each problem are described below following each problem statement.* Note that text referring to files or lines of code are represented in this font.

1. ***Introduction to GAMES.*** *At the end of this problem, you should be able to:*
    - *Understand key terms associated with the model development process.*
    - *Describe methods used to compare experimental and simulated data.*
    - *Compare brute-force methods for parameter estimation with traditional optimization algorithms.*
    - *Download code from GitHub, follow installation instructions, and run code on personal machine.*

    a. *Complete the following tasks:*
        i. *Read the GAMES tutorial (Dray et al.)[2] Focus on understanding the following key terms. You do not need to read the entire manuscript front to back – focus instead on understanding the introduction, the first results section (focused on the overall workflow), and the terms below.*
            - *Explanatory model*
            - *Predictive model*
            - *Model formulation*
            - *Training data*
            - *Parameter estimation method*
            - *Cost function*
            - *Hyperparameters (specifically, n_{search} and n_{init})*
            - *Parameter estimation method evaluation*

- *Parameter identifiability*
- *Parameter profile likelihood*

ii. *Download the code from GitHub: https://github.com/leonardlab/GAMES_education*

ii. *Using your command-line shell (Terminal for Max/Linux users, PowerShell for Windows users), follow the instructions in the README.md file to create and activate your poetry-based environment for this homework set. Note that you will need to have Poetry and pyenv installed, with Python 3.10 available for use.*
   - *The README.md suggests that you test your installation by immediately running module 0. Please note that this will not work for this version of the code, which was purposely released without a complete config.json file (you will complete this file in problem 2).*

iii. *Download and launch an Integrated Development Environment (IDE) of your choice (ex: PyCharm or Spyder). IDEs and Jupyter Notebooks (which you may be more familiar with) are different ways to run Python code and each is useful for different types of coding. When dealing with large, highly interconnected code bases, an IDE is generally the better option. For this problem set, you will update your code in the IDE and run the code in your command-line shell. Note that the config.json file is best updated with a simple text editor, such as Atom.*

a. *What is the purpose of the function solve_single_parameter_set()in the file solve_single.py (in the folder src/games/modules/)? Your response should be ~1 sentence long.*

b. *The function run_single_parameter_set() provides three different ways to evaluate the agreement between the training data and the simulated data: a plot of the training data and simulated data on the same axes, a $\chi 2$ value, and an R^2 value . List 1 advantage and 1 disadvantage of each method. Your response should be ~1 sentence long for each advantage/disadvantage (6 sentences total).*

c. *If you wanted to estimate parameters using a brute-force approach, how might you use run_single_parameter_set() to find a parameter set that yields good agreement between the training data and the simulated data? How would such a brute-force method compare to a traditional optimization algorithm in terms of the overall process (not the outcome)? Your response to each question should be ~1 sentence long (2 sentences total).*

2. ***Model formulation, normalization, and parameter estimation.*** *At the end of this problem, you should be able to:*
   - *Compare ODEs formulated with different assumptions.*
   - *List biological assumptions associated with a mathematical implementation.*
   - *Understand the impact of normalization on the comparison between experimental and simulated data.*
   - *Perform parameter estimation simulations and interpret results.*
   - *Compare parameter estimation results for models formulated with different assumptions*

a. *First, take a look at the ODEs defined in synTF.py (src/games/models/synTF.py in the function called gradient()). The model that differs from the published COMET model in 2 key ways.*
   i. *One difference is related to promoter activation. The COMET version uses a nonlinear promoter activation term that depends on the parameters b, m, and w. How is promoter activation described differently in the 'linear' mechansimID (lines 123-125 in models/synTF.py) and what assumptions were made to formulate this the portion of the model describing promoter activation? Your response should include ~1 sentence each for 3 different assumptions. (Hint: think about the the following questions: 1) How does promoter activation depend on the amount of synTF; 2) What is the relative promoter activation when no synTF is present; 3) What is the relative promoter activation as the amount of synTF is increased to a relatively large value?). Be sure to include the biological meaning of the parameter g in your response.*

   ii. *This model also differs from the model published in the COMET model in terms of population heterogeneity. The COMET model uses a statistical model to represent the heterogeneity in plasmid uptake that is observed for the plasmid delivery method used to perform experiments (transient transfection) and requires simulation of 200 single cells with different plasmid uptake parameters. The simulation results are then averaged across the cells to calculate each mean simulated data point. How is the version in the GAMES code different and what assumptions related to population heterogeneity are involved in this formulation? Your response should be ~1 sentence long.*

b. *Let's see if the GAMES version is capable of describing a set of experimental data. In this case, the training data is a synTF plasmid dose response collected via flow cytometry. Sample experimental data for use in this problem set are included in training_data_synTF dose response.csv and the data are imported into the GAMES code in the file experimental_data.py.*

   For this problem, the following conditions should be defined in config.json (content in parentheses serves as explanation and should not be entered in config.json). Fields that are not mentioned here do not need to be changed.

   - *"folder_name": "problem 2b",*
   - *"modelID": "synTF",*
   - *"dataID": "synTF dose response",*
   - *"mechansimID": "linear" ,*
   - *"normalizationID": "mean",*
   - *"parameters": [1, 1], (These numbers are the initial guesses for each parameter in parameter_labels. If the parameter is not free, it will be fixed to the number set "here.)*
   - *"parameter_labels": ["g", "s"]*
   - *"free_parameter_labels": ["g"] (Only parameters defined here are free in a given simulation.)*
   - *"num_parameter_sets_global_search": 100, (n_{search} in the paper)*

- *"num_parameter_sets_optimization": 5, (n_{init} in the paper)*

i. *Run the parameter estimation module to estimate the parameter g by typing the following in your command-line shell:*

   python run.py --modules='2'

   What is the g value that yields the best agreement between the training data and the simulated data (the best g value is printed to your console following optimization)? Include the GAMES-generated plot of the training data and the simulated data on the same axes.

ii. *How well does the best g value describe the data? What features of the data are well-described and not well-described? Be sure to comment on all three evaluation methods described in problem 1b. Your response should be ~3-4 sentences long.*

iii. *Consider the global search and optimization results (global search results.csv and optimization results.csv). These results show that every value of g yields the same cost function. Rationalize this result (hint: consider how the data are normalized for comparison between the experimental and simulated data in lines 201-202 in models/synTF.py and lines 47-49 in config/experimental_data.py). Your response should be ~2-3 sentences long.*

c. *Now, let's change the normalization strategy and run parameter estimation again. This time, we'll use a scaling factor approach. Here, we will add a new free parameter, s, that will be multiplied by each simulated data point instead of the previous normalization strategy. This data set will then be compared with the experimental data (which will no longer be normalized for comparison with the simulated data).*

   For this problem, make the following changes to config.json.

   - *"folder_name": "problem 2c",*
   - *"normalizationID": "scaling factor",*
   - *"free_parameter_labels": ["g", "s"],*

i. *Run the parameter estimation module to estimate the parameters g and s by typing the following in your command-line shell:*

   python run.py --modules='2'

   What are the g and s values that yields the best agreement between the training data and the simulated data (the best g value is printed to your console following optimization)? Include the GAMES-generated plot of the training data and the simulated data on the same axes.

ii. *How well does the best set of g and s values describe the data? What features of the data are well-described and not well-described? Be sure to comment on all three evaluation methods described in problem 1b and compare these metrics to those that you calculated in 2bii. What is the major improvement of the parameter estimation process in this problem vs. that in 2bii (hint: take a look at the range of $\chi 2$ values in global search results.csv)? Your response should be ~4-5 sentences long.*

3. ***Evaluation of parameter estimation method.*** *At this point, it is unclear whether (I) the GAMES version of the model cannot accurately describe the experimental data, or (II) the parameter estimation method (PEM) is simply incapable of identifying a satisfactory parameter value with the current PEM and hyperparameters. The purpose of module 1 (evaluate parameter estimation method) is to build confidence in the PEM by generating simulated data with known parameters and then evaluating the ability of the PEM to fit the simulated data. This represents a "ground truth" situation for which we know that a satisfactory fit exists. Therefore, if we cannot identify a good fit, we know that there is something wrong with our PEM and we can tune it accordingly (such as by increasing the hyperparameter values or trying a different algorithm). For this problem, use the same conditions as defined in 2c. At the end of this problem, you should be able to:*
    - *Differentiate between scenarios in which a model cannot describe the training data and in which the parameter estimation method is not well-suited to the problem.*
    - *Perform parameter estimation method evaluation simulations and interpret results.*
    - *Understand features of parameter estimation problems that make the problems more difficult to solve.*

    a. *For this problem, make the following changes to config.json:*
        - *"folder_name": "problem 3a"*

        Next, run the parameter estimation evaluation module by typing the following in your command-line shell:

        python run.py --modules='1'

        Include the GAMES-generated plot of the fit to the first set of PEM evaluation data. Is the PEM evaluation data well-described by the simulated data? How can you tell?

    b. *Include the GAMES-generated plot of the PEM evaluation criterion using the R^2. What are the highest R^2 values for each PEM evaluation data set following optimization?*

    c. *The PEM evaluation criterion is used to benchmark success of PEM evaluation simulations. What is the value of the PEM evaluation criterion?*

    d. *Based on the fit metrics ($\chi 2$ and R^2) and the plot of the PEM evaluation data with the experimental data, is the parameter estimation method appropriate for this problem? Provide a 1-2 sentence justification.*

e. *This is a relatively simple problem with only 2 free parameters. If you were to run a similar analysis on a larger model with more free parameters, what might the results look like if the PEM was not appropriate for the parameter estimation problem? If the PEM evaluation criterion is not met, what are 2 different steps that you might you take to alter the PEM to make it appropriate for the given problem?*

4. ***Iteration between model development and parameter estimation***. *If a given model cannot describe the training data, the model formulation must be updated based on a mechanistic hypothesis about why the model is unable to describe the data.*

   At the end of this problem, you should be able to:
   - *Compare parameter estimation results for models formulated with different assumptions.*
   - *Understand features of parameter estimation problems that make the problems more difficult to solve.*

   a. *Update the model to include the nonlinear promoter activation term (f) from the COMET paper (see Methods: equations 5) on line 128 in synTF.py. Which features of the synTF plasmid dose response do the parameters b, m, and w describe? What assumptions are made about promoter activation with this nonlinear term? Your response should be ~2-3 sentences long.*

   b. *For this problem, the following conditions should be defined in config.json in:*
      - *"folder_name" : "problem 4b",*
      - *"mechanismID": "nonlinear",*
      - *"normalizationID": "mean",*
      - *"parameter_labels" : ["b", "w"],*
      - *"free_parameter_labels" : ["b", "w"], (here, we will set m = 1, an arbitrary value that will not affect the results because of our normalization strategy, and we will estimate b and w with the experimental data).*

      Run the parameter estimation evaluation module and the parameter estimation module in series by typing the following in your command-line shell:

      python run.py --modules='12'

      How would you interpret the PEM evaluation results? Why is it necessary to run PEM evaluation simulations every time the model is changed? Include the GAMES-generated plot of the PEM evaluation criterion. Your response should be ~2-3 sentences long.

   c. *How would you interpret the parameter estimation results? Be sure to comment on all three evaluation methods described in problem 1b. Include the GAMES-generated plot of the simulated data. Your response should be ~2-3 sentences long.*

d. *This case study is a relatively simple problem that requires estimating only 2 free parameters, and therefore one could reasonably imagine completing the previous problems with a brute-force method. List three features of a parameter estimation problem that may make the problem unsuitable for such a brute-force method and justify why each feature would make the parameter estimation problem more difficult. Your response should be ~3 sentences long.*

5. ***Parameter identifiability analysis and refinement.*** *For the final problem, we will consider a case study of a transcription factor cascade, such that synTF1 regulates expression of synTF2, which regulates expression of a reporter protein. In this case study, we will assume that the synTFs are stably integrated into the genome and therefore promoter activation can be described with a linear promoter activation term, unlike the transiently transfected TFs considered in the previous problems. Therefore, there are 2 free parameters in the system, g_1 and g_2, which describe linear promoter activation for synTF1 and synTF2, respectively. Suppose that the training data for this model is a synTF1 plasmid dose response with reporter expression as the dependent variable. This problem is fully conceptual and will not require any simulations. At the end of this problem, you should be able to:*
    - *Understand the concept of parameter identifiability and its importance in model development and analysis.*
    - *Classify parameters as identifiable, structurally unidentifiable, or practically unidentifiable.*
    - *Analyze the causes of unidentifiable parameters by investigating parameter relationships*
    - *Propose model reduction strategies to refine unidentifiable parameters.*
    - *Explain when experimental design should be used to refine unidentifiable parameters instead of model reduction.*

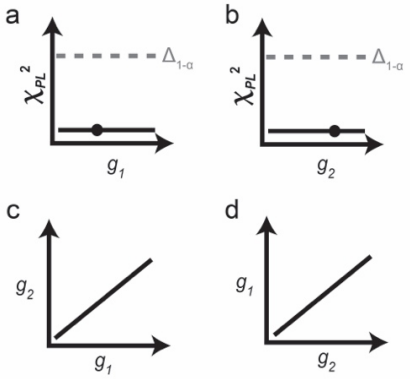

Figure 2: Example parameter profile likelihood (PPL) results for a synTF cascade*. (a) PPL results for g_1, (b) PPL results for g_2, (c) parameter relationships for g_1, (d) parameter relationships for g_2.*

a. *Ideally, g_1 and g_2 will both be identifiable. What does it mean for a parameter to be identifiable and why is it important for both explanatory and predictive models? Your response should be ~3 sentences long.*

b. *One way to determine whether a parameter is identifiable is the parameter profile likelihood. Generally, this method requires parallelization of computational tasks to run in a reasonable amount of time, so we have provided hypothetical example results for the purpose of this problem set (Figure 2). Based on the PPL results (Figure 2a and 2b), classify g_1 and g_2 as identifiable, structurally unidentifiable, or practically unidentifiable and justify your response. Your response should be ~1 sentence long.*

c. *The PPL can be used to investigate the causes of unidentifiable parameters by plotting the re-optimized parameter values for each data point in each PPL plot (Figure 2c and 2d). Based on the parameter relationship plots, what is the source of the unidentifiability associated with g_1 and g_2? Your response should be ~2-3 sentence long.*

d. *Propose a model reduction strategy that addresses the limitation identified in 5b and 5c. Explain in words what the PPL may look like for the remaining free parameter(s) after model reduction (focus on a qualitative description of the PPL). Your response should be ~2 sentences long*

   In this case, we limited refinement of parameter identifiability to model reduction, but refinement can also be accomplished via experimental design by adding additional training data to constrain a previously unidentifiable parameter. Consider a scenario in which you would use the model of the synTF cascade presented here to predict the effect of adding in a synTF1 transcriptional inhibitor. In this case, if g_1 and g_2 are still unidentifiable (meaning that the training data are not affected by changes in either parameter), but the predictions are affected by changes in both individual parameters, would model reduction or experimental design be a more appropriate choice to refine the model? Justify your response and explain the possible consequences of pursuing the less appropriate option. Your response should be ~2-3 sentences long.

REFERENCES

1. Donahue PS, Draut JW, Muldoon JJ, Edelstein HI, Bagheri N, and Leonard JN (2020) The COMET toolkit for composing customizable genetic programs in mammalian cells. *Nat Commun*. 11(1):779. DOI: 10.1038/s41467-019-14147-5.
2. Dray KE, Muldoon JJ, Mangan NM, Bagheri N, and Leonard JN (2022) GAMES: A Dynamic Model Development Workflow for Rigorous Characterization of Synthetic Genetic Systems. *ACS Synth Biol*. DOI: 10.1021/acssynbio.1c00528.