TEACHING SYSTEMATIC, REPRODUCIBLE MODEL DEVELOPMENT USING SYNTHETIC BIOLOGY

Kate E. Dray, Kathleen S. Dreyer, Julius B. Lucks, and Joshua N. Leonard
Northwestern University • Evanston, IL 60208

INTRODUCTION

Synthetic biology is an emerging technical discipline that applies engineering concepts to design living systems to perform novel functions. One strategy for achieving this goal is using standard biological parts, such as DNA sequences and proteins, which are first characterized and then composed to create new functions such as genetically-encoded programs.[1] When implemented in a cell or cell-free system, such genetic programs can, as an example, direct the host to sense environmental or internal cellular cues, process such inputs, and then respond via regulation of the expression of one or more target genes. While synthetic biology employs concepts derived from electrical engineering, in which standard electrical parts are composed to build circuits,[2] the origins, open challenges, and potential applications of synthetic biology are closely aligned with chemical engineering.[3] Specifically, there is a need to develop systematic, scalable processes for the development, standardization, modularization, and production of cell-based devices with applications ranging from biomanufacturing to cell-based therapies.[3] Synthetic biology has matured towards a true engineering discipline by applying concepts and practices established in mature technical fields,[4] including the incorporation of mathematical modeling to facilitate understanding and enable prediction.[1, 5] Here we focus on models that employ ordinary differential equations (ODEs) to describe system dynamics, incorporating fundamental concepts taught in core chemical engineering subjects, including mass balances and reaction kinetics.

Mathematical modeling is widely used in chemical engineering to understand and predict the performance of chemical reactions and other complex processes, but new, holistic approaches are needed to train students in the process of model development.[6] This is a challenging goal, deriving in part from the diversity of expertise and skills needed to model effectively, with topics including intuition-guided iteration between model formulation and parameter estimation, investigation of multi-dimensional design spaces with unconstrained parameters, and effectively comparing experiments to simulations.[5] In addition, the model development process is often opaque, as many publications incorporating modeling focus on the final result while omitting description

Kate E. Dray *was a PhD candidate in the department of Chemical and Biological Engineering at Northwestern University at the time of this study, and she has since graduated. Her research focuses on systematizing the model development process in synthetic biology to improve rigor and reproducibility of the process. She has also developed and analyzed numerous dynamic models of synthetic biological systems designed to improve mechanistic understanding and drive future experimentation. ORCID: 0000-0002-2972-047X*

Kathleen S. Dreyer *is a PhD candidate in the department of Chemical and Biological Engineering at Northwestern University. Her research involves developing dynamic models of synthetic biological systems to predict outcomes of novel design choices. Her work focuses on automating the design process to expand predictive capability of models beyond that of intuition-guided manual design. She served as the Teaching Assistant in the computational biology course where the described educational unit was first implemented. ORCID: 0000-0002-2041-4118*

Julius B. Lucks, *PhD, is a Professor of Chemical and Biological Engineering at Northwestern University. He earned his PhD in Chemical Physics at Harvard University. His research focuses on elucidating the principles of RNA folding and function, and using these principles to create biotechnologies that benefit humanity. He is the PI on the first National Science Foundation Research Training Program in Synthetic Biology, aimed at developing new approaches to teach synthetic biology across disciplines. ORCID: 0000-0002-0619-6505*

Joshua N. Leonard, *PhD, is an Associate Professor of Chemical and Biological Engineering at Northwestern University. He earned his PhD in Chemical Engineering at the University of California-Berkeley. His research focuses on employing synthetic biology and computational systems biology to address unmet needs in biotechnology, including engineered cell and gene therapies. As a Charles Deering McCormick Professor of Teaching Excellence, his educational interests include enhancing critical thinking and making computational biology methods broadly accessible. ORCID: 0000-0003-4359-6126*

GAMES represents a direct application of the process-oriented thinking that is a key component of the chemical engineering curriculum

of the process used to arrive at the final model. As a result, it is often not clear as to how choices made in the development process affect the final model and how the computational tasks – such as model formulation, parameter estimation, parameter identifiability analysis, and model selection – are interconnected. Extensive prior work provides strategies for teaching important modeling tasks, such as developing differential equation-based models to describe systems of interest,[7-9] solving differential equations analytically and numerically,[9, 10] utilizing various coding languages to solve differential equations,[7, 9, 11, 12] parameter estimation with respect to experimental data,[8] and sensitivity analysis.[8] Most prior work has been designed for use with MATLAB®, a commercially available programing and numerical analysis platform. While each of these individual tasks is vital to the model development process, there exists a need for educational materials focused on providing a holistic, conceptual framework of model development and analysis using an accessible, open-source coding language. Such a framework could provide a theoretical roadmap for the entire model development process by teaching both the theory behind each computational method in the process and how methods can be iterated upon to arrive at a final model.

To address challenges in executing and communicating the model development process and to lower the barrier to entry for new modelers, we recently published a tutorial describing a systematic model development workflow for rigorous characterization of genetic programs called GAMES (a workflow for the Generation and Analysis of Models for Exploring Synthetic systems).[5] GAMES provides a conceptual framework for understanding, applying, and linking each computational method involved in the model development process using example code written in an open-source coding language, Python®, and is therefore well suited to be expanded to develop an educational unit designed to teach the entire model development process. Further, GAMES represents a direct application of the process-oriented thinking that is a key component of the chemical engineering curriculum, thus complementing the existing undergraduate curriculum in chemical engineering and addressing the need for teaching synthetic biology to "career chemical engineers" by incorporating synthetic biology concepts into existing courses.[13] Since GAMES applies concepts taught in core chemical engineering reaction kinetics courses, the concepts and approaches employed could be used to develop and solve models describing classical chemical engineering processes as well, which is a vital component of chemical engineering education.[10] Thus, GAMES can help students to understand and deploy systematic and reproducible model development for any ODE-based model.

Our core thesis is that teaching a structured, systematic approach to the process of ODE model development in synthetic biology will enable students to understand key concepts and apply these concepts to new modeling problems. To this end, we developed an educational unit consisting of a two-day lecture series and a homework set, focused on an audience of advanced undergraduate and graduate students. Following implementation of the unit in a computational biology course at Northwestern University, we identified key areas for improvement for future iterations of the course. Overall, the goals for this unit are to enable students to: (1) understand key steps and considerations in the model development process at a conceptual level, rather than relying exclusively on automated model analysis packages; (2) practice implementing key methods and interpreting the results; and (3) gain exposure to diverse applications of chemical engineering thinking. These outcomes align with those that are known to positively impact student experiences.[14-16] By mapping core chemical engineering topics to new application spaces, this tool may be especially useful when biology-focused chemical engineering curriculum options are limited. This training also seeks to prepare students for future research opportunities focused on mathematical modeling in synthetic biology. Here, we introduce instructors to the GAMES workflow and describe strategies for implementing associated pedagogical tools in a core or elective chemical engineering course.

METHODS

GAMES Conceptual Workflow

The GAMES workflow describes the model development process as a set of five interconnected tasks, or modules (Figure 1a). After defining the system of interest, the modeler first initializes the model development process in Module 0 by identifying a modeling objective, collecting or otherwise obtaining training data, and formulating a base case model, which represents a series of biological, mechanistic hypotheses proposed to describe how the system functions. Before estimating the values of the unknown parameters (which are often kinetic rate constants), the modeler must first evaluate their parameter estimation method in Module 1 to ensure that it is appropriate for their parameter estimation problem (defined by the model equations, free parameters, and training data). After ensuring that the parameter estimation method (PEM) is appropriate, the modeler can then use the PEM to estimate parameters using training data in Module 2. Next, in Module 3, the modeler determines whether their parameters are well-constrained or identifiable. If any parameters are unidentifiable, they must be refined via either model reduction or experimental design. Finally, in Module 4, the modeler compares any competing models and selects the best option. Different iterations of the modules in the workflow can be used to accomplish a variety of key modeling goals, such as parameter estimation, model refinement, experimental design, model reduction, model selection, and model validation (Figure 1b).[5]

GAMES Software Package

To support the use of the GAMES, we developed a freely available Python-based software package and deposited it on GitHub®. This software manifests the concepts in each module through practical implementations of each method. The software package is modular, such that the user must simply modify only two files to run the workflow for any new system of interest: (1) the .py file, which includes a problem-specific model class, containing information such

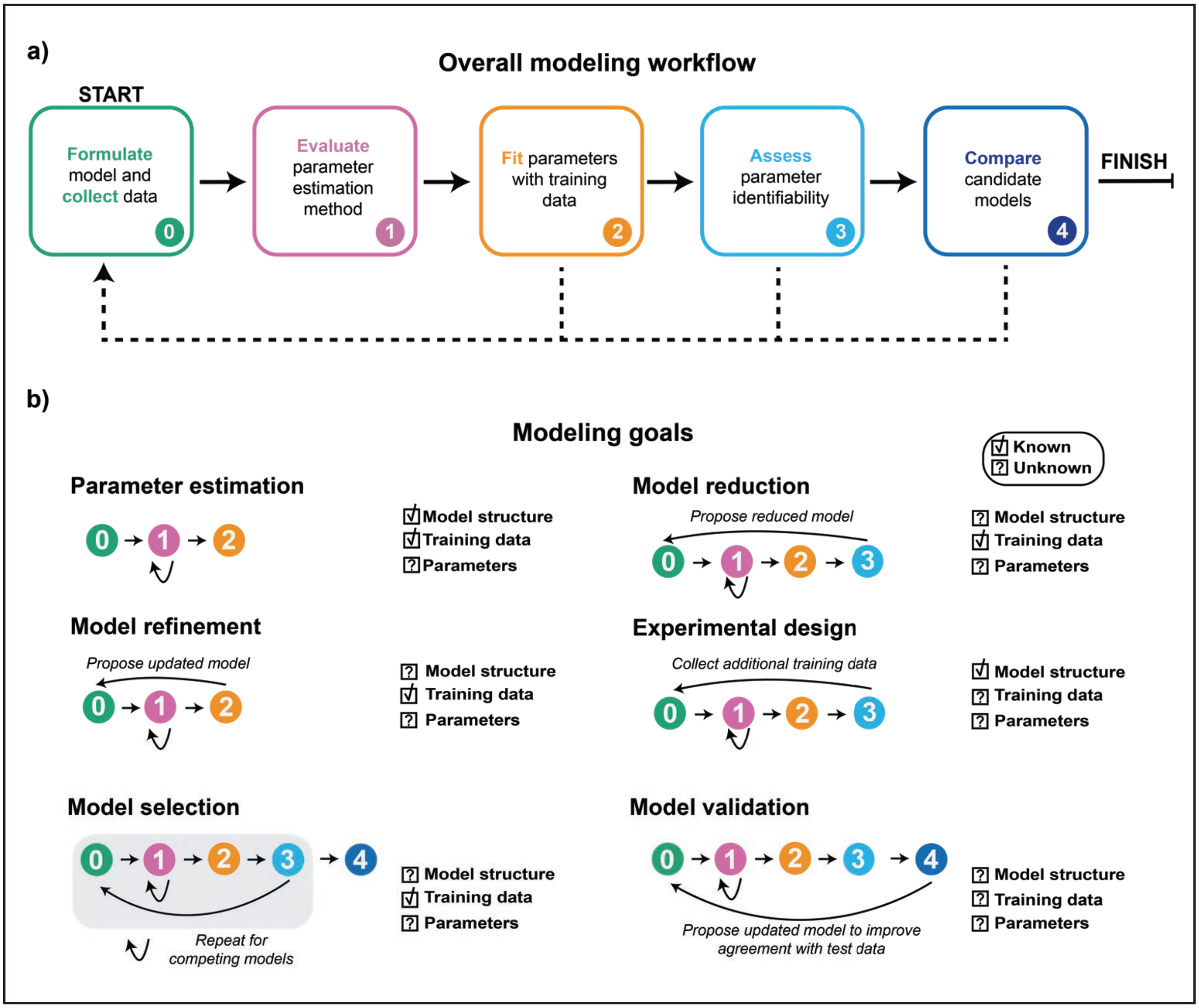

Figure 1. *Overview of the GAMES workflow. (a) GAMES describes the model development process as a set of 5 interconnected tasks, or modules. The overall modeling workflow is iterative, as represented by the dotted lines. (b) The modules described in GAMES can be iterated upon in different orders to accomplish a variety of modeling goals, each of which is important in the development of mathematical models. Reprinted with permission from 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. 11(2):1009-1029. DOI: 10.1021/acssynbio.1c00528. Copyright 2022 American Chemical Society.*

as the ODEs, normalization strategy, and plotting functions; and (2) the .json configuration file, which includes information related to hyperparameters necessary for each method and defines free parameter values and bounds. The software package is available on GitHub at https://github.com/leonardlab/GAMES. The version of the GAMES software package necessary for the homework set presented here can be found at https://github.com/leonardlab/GAMES_education.

Installation requires the use of various Python tools, including pyenv [17] and Poetry [18], to ensure that the simulation environment and package dependencies are properly implemented, both of which are vital for promoting code reproducibility, which is a major problem in computational modeling of biological systems.[19-21] As installation of software is a vital step in the use of any software package obtained from GitHub, the installation of GAMES provides a useful opportunity for students to understand Python environments and to practice execution of installation instructions. We highly recommend that the instructor and/or teaching assistants for the course familiarize themselves with the installation protocol before distributing the assignment and that an in-class or office hours session be dedicated towards troubleshooting any installation roadblocks.

EDUCATIONAL METHODOLOGY AND MATERIALS

Overview

The educational unit is structured to address overall learning objectives through an interconnected two-day lecture series and homework set. The overarching goal of the educational unit is to introduce model development concepts and to increase familiarity with modeling synthetic biological systems using both conceptual and experiential pedagogical materials. A key choice made in defining learning objectives was to focus on practicing interpretation and analysis of results in each step of the workflow, rather than on writing code or implementing computational methods. The unit was implemented in the spring 2022 iteration of the course “Computational Biology: Design and Analysis of Living Systems” at Northwestern University. Students in this course comprised 13 undergraduate and PhD students in both the Chemical Engineering and Integrated Biological Sciences departments. Incoming students were expected to have basic proficiency in Python (or some related programming language), along with a general knowledge of mass balances and reaction kinetics (which were introduced earlier in the course), such that students could understand how the ODEs in the GAMES tutorial were formulated from first principles. The amount of coding expertise required to complete the homework set is, by design, minimal. Instead, the assignment focuses on prompting thoughtful interpretation of GAMES-generated results. Similarly, the course instructor and/or teaching assistant should have sufficient proficiency in Python to facilitate resolution of difficulties with executing and modifying code, but de novo writing in Python is not required.

Lecture Series

The lectures provide a conceptual introduction to each module of the GAMES workflow,[5] using the same proof-of-principle example and topical sequence as the tutorial, to demonstrate each milestone in the process. The lectures were presented as two consecutive 50-minute class meetings. The lecture slides are freely available for download at Zenodo.[22] Both lectures were held in-person with a Zoom® option and were recorded. Learning objectives were provided at the beginning of each day (Table 1). The first lecture focused on introducing each module of GAMES in a linear, idealized manner such that iteration between modules was unnecessary. As model development is rarely linear in practice, the second lecture leveraged the students’ basic understanding of each module to elaborate the workflow to a more realistic, nonlinear form that incorporates the practically unavoidable iteration between modules. In the second lecture additional practical information not included in the tutorial was presented. This information included a discussion of cross-validation methods for model validation along with a series of lessons learned from applying GAMES to complex biological systems beyond the example presented in the tutorial.

Homework Set

The homework set guides students through hands-on participation in the model development process by introducing a new example designed to be explored using the GAMES code. Students were given one week to complete the homework set, with extensions available if necessary. Learning objectives were provided alongside each homework problem. Homework solutions are omitted from this manuscript, but solutions are available to educators upon request via e-mail to j-leonard@northwestern.edu. Two one-hour office hour blocks with the course TA were offered to help students with this set. Asynchronous help was available via the course Slack® workspace. This version of the homework set has been updated to use an improved, refactored version of the GAMES code and is therefore slightly different than the homework set used in the spring 2022 implementation of our educational unit. The differences relate only to the computational implementation of the GAMES workflow; all learning objectives and conceptual framings remain the same. The text for the homework set is available at Zenodo.[22]

TABLE 1
Learning objectives for the GAMES lecture series.

Day 1	Day 2
Understand general challenges for developing models in synthetic biology	Analyze parameter profile likelihood results (PPL) and understand how results can motivate refinement of parameter identifiability
List the key steps in model development according to the GAMES workflow	Compare scenarios using model reduction or experimental design to refine parameter identifiability
Understand the importance of defining a modeling objective at the outset of a modeling quest	Define metrics for comparing competing models
Understand the importance of evaluating a parameter estimation method before interpreting parameter estimation results	Explain various nonlinear iterations between modules that may be necessary in practice
Explain the base case parameter estimation method used in the GAMES workflow	Understand challenges with applying the model development workflow to more complex synthetic biological systems
Understand the concept of parameter identifiability and why it is important for model development and analysis	

The topics of each question are as follows:

1. Introduction to GAMES
2. Model formulation, normalization, and parameter estimation
3. Evaluation of parameter estimation method
4. Iteration between model development and parameter estimation
5. Parameter identifiability analysis and refinement

IMPLEMENTATION AND OPPORTUNITIES FOR IMPROVEMENT

Students who participated in our educational unit gained increased interest and understanding of the model development process in synthetic biology. A core part of the curriculum of the course in which this unit was implemented is guiding students through the process of proposing, reviewing, revising, and executing an independent research project using methods and approaches introduced in this course. Some students chose to incorporate ODE modeling via the GAMES framework, suggesting that our educational unit provided students with the desire and ability to apply the concepts presented here to new problems of interest. This material was successfully employed by both undergraduate and graduate students with varying levels of computational and biological experience, suggesting that the educational unit could be implemented for a range of different academic levels and proficiencies. While we implemented this unit in a computational biology course, the unit could be implemented as part of the traditional chemical engineering curriculum in a reaction kinetics class to expose students to computational modeling, synthetic biology, and Python.

Through this initial offering, we identified several areas for improvement for the next iteration of this education unit based on our own experience as course instructors (a formal evaluation of student learning outcomes is planned for future offerings of this course). Students would likely benefit from having more time to complete the homework assignment; two weeks would be a better timescale to enable deeper engagement with several new and sophisticated topics. This recommendation is based upon the general observation that students were more comfortable describing the general use of novel concepts (e.g., parameter profile likelihood) than in applying these methods to interpret specific model development scenarios. We see this as an opportunity to help students achieve deeper levels of learning (i.e., moving from Understanding to Analyzing and Interpreting in the Revised Bloom's Taxonomy [23]). It would perhaps be useful to spread the GAMES lecture series content across three days to ensure more time to present and probe understanding of core concepts introduced in this unit. Based on the questions asked in office hours, we recommend a live demonstration of the code to prepare students to complete the homework set. This could be completed either on the last day of lectures or interspersed throughout the lecture series as each module is introduced. We suggest including a general introduction/refresher on ODE model formulation focused on converting a biological hypothesis to a testable mathematical implementation based on ODEs – this topic is not covered in the

GAMES tutorial but is a central part of the model development process. Depending on the course, such an introduction may already be included in the curriculum. Otherwise, existing resources may help serve this role.[24, 25]

DIVERSITY, EQUITY, AND INCLUSION CONSIDERATIONS

A key shared goal of writing the GAMES tutorial[5] and disseminating the educational materials described here is to increase student access to the training and resources required to lead cutting-edge research involving computational modeling in general, and particularly in synthetic biology. We anticipate that these resources could make such training more accessible at institutions that currently provide limited opportunities in computational biological modeling or synthetic biology. Since synthetic biology curricula are most commonly implemented at large research institutions in the USA and the EU,[13] this unit could help to overcome current limitations on access to such training for students who do not attend these institutions. Implementing this educational unit may also increase student awareness and interest in synthetic biology as an application of chemical engineering. We employed the freely available Python language and disseminated these tools for free to reduce financial barriers in order to promote equity. Finally, the approaches embodied in these educational materials are designed to engage students with a diversity of learning styles and to follow inclusive teaching best practices.[26-28] Key choices include using visual and audio presentation of step-by-step examples, communication of specific learning objectives, integration of written self-directed instruction, distribution of slides to students in advance of lectures, and recording of presented material for future reference. Overall, we hope that these resources will help educators to engage a broader swath of students who might be interested in computational biology and synthetic biology.

CONCLUSIONS

Here we present an educational unit focused on teaching systematic, reproducible model development through the lens of synthetic biology. As a result of the unit, we anticipate that students will become more aware of, interested in, and comfortable with synthetic biology, computational modeling, and the intersection between the two. We expect that students will be able to apply the general skills and concepts learned here to other parts of the chemical engineering curricula, including reaction kinetics, process modeling, and engineering design courses. A key innovation of this educational unit vis-à-vis prior work is the focus on providing a holistic, conceptual framework for the model development process. Such a focus on process is vital in chemical engineering education and for building technical literacy as to how modeling supports a variety of applications. As the field of chemical engineering evolves to keep apace of new challenges and needs facing society,[29, 30] this educational unit exemplifies the opportunity to integrate new and exciting applications, such as synthetic biology and computational modeling, into the chemical engineering curriculum.

ACKNOWLEGEMENTS

This work was supported by the NSF NRT-URoL Synthesizing Biology Across Scales training program under award NSF 2021900 (JBL), and by the National Institute of Biomedical Imaging and Bioengineering of the NIH under award number 1R01EB026510 (JNL).

REFERENCES

1. Dray KE, Edelstein HI, Dreyer KS, and Leonard JN (2021) Control of mammalian cell-based devices with genetic programming. *Curr Opin Syst Biol*. 28. DOI: 10.1016/j.coisb.2021.100372.
2. Purnick PE and Weiss R (2009) The second wave of synthetic biology: From modules to systems. *Nat Rev Mol Cell Biol*. 10(6):410-22. DOI: 10.1038/nrm2698.
3. Luo Y, Lee JK, and Zhao H (2013) Challenges and opportunities in synthetic biology for chemical engineers. *Chem Eng Sci*. 103. DOI: 10.1016/j.ces.2012.06.013.
4. Cheng AA and Lu TK (2012) Synthetic biology: An emerging engineering discipline. *Annu Rev Biomed Eng*. 14:155-78. DOI: 10.1146/annurev-bioeng-071811-150118.
5. 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*. 11(2):1009-1029. DOI: 10.1021/acssynbio.1c00528.
6. Elnashaie SSEH, Alhabdan FM, and Elshishini SS (1993) The vital role of mathematical modelling in chemical engineering education. *Mathematical and Computer Modelling*. 17(3):3-11. DOI: 10.1016/0895-7177(93)90034-v.
7. Li X and Huang Z (2017) An inverted classroom approach to educate MATLAB in chemical process control. *Education for Chemical Engineers*. 19:1-12. DOI: 10.1016/j.ece.2016.08.001.
8. Molina R, Orcajo G, and Martinez F (2019) KBR (Kinetics in Batch Reactors): A MATLAB-based application with a friendly Graphical User Interface for chemical kinetic model simulation and parameter estimation. *Education for Chemical Engineers*. 28:80-89. DOI: 10.1016/j.ece.2018.11.003.
9. Golman B (2016) Transient kinetic analysis of multipath reactions: An educational module using the IPython software package. *Education for Chemical Engineers*. 15:1-18. DOI: 10.1016/j.ece.2015.12.002.
10. Gossage JL, Yaws CL, Chen DH, Li K, Ho TC, Hopper J, and Cocke DL (2001) Integrating best practice pedagogy with computer-aided modeling and simulation to improve undergraduate chemical engineering education *Proceedings Proceedings ASEE Annual Conference & Exposition*.
11. Geng J, Chen K, Wang N, Ling S, Guo M, and Huang Z (2019) Comparison of R and Simulink in educating high school students with ODE modeling skills. *Chemical Engineering Education*. 53(2):121-129.

12. Inguva P, Bhute VJ, Cheng TNH, and Walker PJ (2021) Introducing students to research codes: A short course on solving partial differential equations in Python. *Education for Chemical Engineers*. 36:1-11. DOI: 10.1016/j.ece.2021.01.011.
13. Hall GM and Howe J (2012) The impact of synthetic biology in chemical engineering—Educational issues. *Education for Chemical Engineers*. 7(2):e51-e55. DOI: 10.1016/j.ece.2012.02.002.
14. Madan CaT, Braden (2013) The benefits of undergraduate research: The student's perspective. *The Mentor: An Academic Advising Journal*. DOI: 10.26209/mj1561274.
15. Russell SH, Hancock MP, and McCullough J (2007) The pipeline. Benefits of undergraduate research experiences. *Science*. 316(5824):548-9. DOI: 10.1126/science.1140384.
16. Gillett JE (2001) Chemical engineering education in the next century. *Chemical Engineering & Technology*. 24(6):561-570. DOI: 10.1002/1521-4125(200106)24:6<561::Aid-ceat561>3.0.Co;2-x
17. pyenv. https://github.com/pyenv/pyenv accessed November 30, 2022.
18. Poetry. https://python-poetry.org/ accessed November 30, 2022.
19. Mukherjee S, Almanza A, and Rubio-González C (2021) Fixing dependency errors for Python build reproducibility, *Proceedings 30th ACM SIGSOFT International Symposium on Software Testing and Analysis.* DOI: 10.1145/3460319.3464797.
20. Papin JA, Mac Gabhann F, Sauro HM, Nickerson D, and Rampadarath A (2020) Improving reproducibility in computational biology research. *PLoS Comput Biol*. 16(5):e1007881. DOI: 10.1371/journal.pcbi.1007881.
21. Tiwari K, Kananathan S, Roberts MG, Meyer JP, Sharif Shohan MU, Xavier A, Maire M, Zyoud A, Men J, Ng S, Nguyen TVN, Glont M, Hermjakob H, and Malik-Sheriff RS (2021) Reproducibility in systems biology modelling. *Mol Syst Biol*. 17(2):e9982. DOI: 10.15252/msb.20209982.
22. Dray KE, Dreyer KS, Lucks JB, and Leonard JN (2023) leonardlab/GAMES education: supplementary material. *Zenodo*. DOI: 10.5281/zenodo.7935134.
23. Anderson LW and Krathwohl DR (2021) *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Longman, New York, NY.
24. Endler L, Rodriguez N, Juty N, Chelliah V, Laibe C, Li C, and Le Novere N (2009) Designing and encoding models for synthetic biology. *J R Soc Interface*. 6 Suppl 4:S405-17. DOI: 10.1098/rsif.2009.0035.focus.
25. Cedersund G and Roll J (2009) Systems biology: Model based evaluation and comparison of potential explanations for given biological data. *FEBS J*. 276(4):903-22. DOI: 10.1111/j.1742-4658.2008.06845.x.
26. Keller J and Lyndgaard K (2017) A brief taxonomy of inclusive pedagogies: What faculty can do differently to teach more inclusively. *Headwaters*. 30(1):64-82.
27. Zumbrunn S, McKim C, Buhs E, and Hawley LR (2014) Support, belonging, motivation, and engagement in the college classroom: A mixed method study. *Instructional Science*. 42(5):661-684.
28. Wankat PC and Bullard LG (2016) The future of engineering education–Revisited. *Chemical Engineering Education*. 50(1):19-28.
29. Th M, Schaer E, Abildskov J, Feise H, Glassey J, Liauw M, Ó'Súilleabháin C, and Wilk M (2022) The importance/role of education in chemical engineering. *Chemical Engineering Research and Design*. 187:164-173. DOI: 10.1016/j.cherd.2022.08.061.
30. Engineering NAo, National Academies of Sciences E, and Medicine (2022) *New Directions for Chemical Engineering*. The National Academies Press, Washington, DC. ❒