Biotechnology data analysis curriculum with simulations on virtual microorganisms

Ulf W. Liebal[1*], Rafael Schimassek[1], Iris Broderius[1], Nicole Maaßen[1], Lars M. Blank[1],

[1] iAMB - Institute of Applied Microbiology, ABBT, RWTH Aachen University, Aachen, Germany

* ulf.liebal@rwth-aachen.de

Abstract

Biotechnology experiences innovations in analytics and information processing. The amount of data as well as their complexity increases and new computational workflows are developed to extract knowledge. However, the speed of transformation outpaces the adaptation of biotechnology curricula and teaching strategies need to be developed to provide data analysis skills to biotechnologists. We have developed a simulator of virtual organisms (SILVIO) as the combination of various models of microbial properties to simulate experimental data generation. We used SILVIO to construct a computer-based educational workflow with selected key steps during strain characterization and recombinant expression of proteins. The educational workflow comes as a Jupyter Notebook with full explanatory text of the biotechnological information and experiment simulations via functions of SILVIO. The data analysis is performed by the students in Python or Excel. This educational workflow was independently implemented in two courses for biology and biotechnology Master students in video meetings with outbreak rooms. The concept of using virtual organism simulations that generate coherent results across different experiments can be used to construct consistent and motivating case stories for biotechnological projects.

- Biotech delivers computational skills
- Interdisciplinary students

Problem

- Bio-departments prefer experiments before data analysis
- realistic data analysis requires complex experiments
- complete and integrated representation of the biotech development cycle hard to show practically

Solution

- computer simulation covering important data acquisition steps

Methods

- Jupyter notebook for guided data analysis
- Python programming learning

- Cloud based service
- Evaluation of the teaching methods

Results

- tested with recombinant protein expression
- feedback evaluation

Outlook

- extension to metabolic simulations
- including fermentation and DSP
- training of ML skills

Biotechnological innovation is changing profoundly by increasing automation and sophisticated computer based data analysis. This ongoing shift needs to be represented in the biotechnology curriculum by enabling an interdisciplinary body of students to combine computational modelling and data analysis with understanding and optimizing microbial performance.

Author summary

The transition to a successful bioeconomy is fueled by the efficient engineering of microorganisms. During the engineering process, volumes of complex data are generated and require adequate integration and computational analysis. Thus, both decision making and judging how new experiments will facilitate strain development are difficult. Our aim in developing virtual microorganism simulations is to emulate biotechnology pipelines with realistic data to train data analysis skills and to raise awareness of the overall goal and the complexities at each step. The following modules are considered: (I) virtual organism, with genome and metabolism; (II) molecular biology, including sequencing, cloning, and analytics; (III) microbiology, with culturing and strain evolution; (IV) bioprocessing by modeling lab-scale fermentation processes. The simulated scenarios can not only be used to simulate biological processes to improve the flexibility of strain engineering by sandbox testing, but also to facilitate communication and to strengthen biotechnological education for a successful bioeconomy.

Introduction

Data literacy in biotechnology

Biotechnology is increasingly generating vast amounts of complex data that require advanced computational analysis [1]. This includes, among others, experimental techniques like multi-omics investigations [2] and computational modelling based approaches [3]. In parallel, data science provides new tools to facilitate data analysis in form of more accessible machine learning tools, for example via the Python SciKit learn library [4], and data analysis environments like Jupyter Notebooks [5], Galaxy [6], KBase [7], or KNIME [8]. Data analysis with Jupyter Notebooks in particular are becoming popular [9] and have been used to guide metabolomic data analysis [10], metabolic engineering [11, 12] or gene expression Liebal et al., 2021. However, the developments of analysis have rarely managed to receive adequate attention in the curriculum of biotechnology.

Increasing the share of data analysis and bioinformatics in the biology and biotechnology curricula are excellent means for developing critical 21st century skills by fostering inquiry-based interdisciplinary learning [13]. Because the topic can deviate from many students prime interests and skills, motivating and activating students is particularly important. There are a number of reports and guidelines to problem based learning and flipped classrooms in computational biology [13–16]. Also for teaching Jupyter Notebooks are popular [17, 18] and guidelines for their setup are available [19, 20]. A particularly useful tool for motivation can be *gamification* and *serious game* elements [21, 22] which contributes more strongly to positive self assessment compared to exam results [23].

A number of virtual organism simulations allow precise phenotyic simulations of microorganisms. Models of organism are increasingly extending from the fundamental mathematical representation into programmatic environments [24]. Python is particularly popular because of the large community that supplies and maintains easy accessible packages to support general tasks from machine learning up to specific biological solutions [24]. An outstanding recent development of virtual organism simulations is *Vivarium* as a modular environment that enables the connection of multi-scale models for realistic simulations [25].

We present an educational workflow to teach data science in biotechnology. The data are generated by multi-model microorganism simulations to emulate results from laboratory experiments. These data are combined to

Materials and methods

The virtual organism simulator

The microbial phenotypes are simulated in Python with an ensemble of models called *SILVIO* (simulator of virtual organisms). SILVIO generates surrogate molecular and microbiological data without attempt to reproduce a target microorganism. Instead, at each start of SILVIO, a new virtual organism is initiated with distinct parameters and data that feels sufficiently realistic. The realism is integrated by perturbing measurements over a normal distribution and non-intuitive, nonlinear machine learning predictions. Figure CodeLogic shows the construction of the virtual organism class. The classes represent biological entities and functions define actions involved with classes.

- Python code development, classes philosophy

Model simulations

BioLabSim contains several models to simulate biological processes. These processes represent important microbial phenotypes that are element of a biotechnological characterization of microorganisms and include growth, the growth constant, DNA melting temperature and gene expression (Table 1). A logistic growth model based on the Verhulst equation is used to simulate the growth experiments to determine optimal growth temperature. The carrying capacity (K in Equation 1 of Table 1) for *E. coli* is randomly initiated between $[30 - 100]$ and for *P. putida* between [45-145] gCDW/l. The temperature dependence of the growth constant (r in Equation 1 of Table 1) is calculated via a normal distribution function with a solution of 1 at the optimal temperature (corresponding to mean value μ) with variance $\sigma = 5$. The optimal temperature (μ in normal distribution) is randomly initiated between [25-40]$^\circ C$.

Table 1. Models for biological processes.

Process	Model	Details
Growth	Verhulst logistic model $P(t) = \frac{K}{1+\left(\frac{K-P_0}{P_0}\right)e^{-rt}}$ (1)	$P(t)$: biomass concentration K: max. biomass, random $P_0 = 0.1$: initial biomass r: growth constant, temperature dependent
Growth constant	Normal distribution (2)	*mean*: optimal temperature, random *variance* $= 5$
DNA melting temperature	$T_m = 2\,(A+T) + 4\,(C+G)$ (3) $T_m = 81.5 - 16.6 + 0.41GC\% - \frac{600}{N_{Nt}}$ (4)	A, C, G, T: number of bases $GC\%$: GC content in % N_{Nt}: sequence length
Gene expression	Random-Forest Regression (5)	*E. coli*, *P. taiwanensis* promoter library Neves et al.

Empirical formulas are used to calculate the DNA melting temperature and the gene expression strength. The optimal primer length is randomly initiated between $16 - 28$ nucleotides [26]. Two equations are used to calculate the optimal melting temperature because for primers < 25 nucleotides the simple Equation 3 in Table 1 can be used and for larger primers Equation 4 in Table 1 is applicable. Gene expression is predicted based on the promoter sequence with 40 nucleotides (nt) upstream of the open reading frame. The regression is performed by random forest machine learning module Liebal et al., Frontiers which was trained with measurements of a sigma70 dependent synthetic promoter library expressed in *E. coli* and *P. taiwanensis* (manuscript in preparation).

Evaluation of the teaching methods

Results

General teaching setup

A pedagogical workflow was developed to convey biotechnological principles of recombinant expression within a modern data analysis environment. The steps included in the simulation are (1) choice of host organism and its characterization, (2) design and cloning of a promoter sequence, and (3) measurement of the final expression titer. To increase motivation for the students, we developed a story in which the students are tasked by a fictional biotech company to engineer a strain for protein subunit expression of the corona virus to supply a vaccine. The final titers depend only on the promoter sequence and the precision by which the students measure host growth properties. The principle learning outcomes are described in Table 2 and include biotechnological knowledge as well as computer science procedures. During the simulation different experiments are performed by using predefined functions appropriately. To stimulate parsimonious and effective use of experiments (here function calls) each experiment costs money and some experiments are time consuming.

The teaching unit was conducted as part of an otherwise lab-based biotech training in genetic manipulation and fermentation. The students were in their first year of a Master in biology and biotechnology at the RWTH Aachen and the Westfälische

Hochschule. The course was conducted as a Zoom-Meeting and started in the first 30 Minutes with an introduction and a quick presentation of the simulation including the solution to all steps. Then, participants were allotted to two-person groups in breakout rooms to work autonomously for ∼2h on solving the simulation with regular (∼20 minutes) contact with a supervisor. The participants managed to test at least one and some even up to four promoter sequences for vaccine expression. In the last ∼20 minutes, all participants joined a final conclusion round, during which the statistical relationship of promoter sequence and expression was examined and the winning vaccination producer was identified.

Table 2. Principle learning outcomes of the *Recombinant Expression Workflow* in BioLabSim.

Informative	Growth and biomass are strain specific Unknown factors impact cloning efficiencies Promoter architecture of -35 and -10 boxes
Performative	Growth curve analysis with linear optimization Calculation of DNA melting temperature Variable assignment in Python

Data analysis in recombinant expression

The recombinant engineering workflow is simulated with four selected steps and data analysis tasks (see Table 3). The users start by choosing a host bacterium, predefined as either *E. coli* or *P. putida*. Both are popular bacterial hosts and related to important pathogens. The host choice affects the possible maximum biomass concentration and the predictor for the promoter activity (see Methods for details). Along with the host, the user also decides how much money is invested to the laboratory equipment. An increasing investment correlates with decreasing experiment failures, based on a saturating curve. We have provided the users with 10,000 EUR starting capital of which 10-20% are optimally used for the equipment.

Table 3. Computational steps of the *Recombinant Expression Workflow* in BioLabSim. Each step involves a set of different activities and challenges.

Step	Activity	Challenge
1. Host Choice	•Read general introduction	•Python variable assignment
2. Host characterization	•Analyse Excel data •Estimate growth curve parameters	•Variance in biomass concentration •Growth experiment fails
3. Promoter design	•Learn importance of promoter boxes •Calculate DNA melting temperature •Derive primer complementary to promoter	•Optimal primer length unknown
4. Evaluation	•Judge effectiveness of promoter •Test impact of GC-content on expression	•Multiple rounds of promoter design

Host characterization

In the second step, the user identifies the optimal growth temperature along with the associated growth rate and biomass. The temperature is randomly initiated between 20-40° C, the growth rate at the optimal temperature is equal to 1 /h and biomass is randomly initiated between 30-100 gCDW/L (*E. coli*) and 45-145 gCDW/L (*P. putida*) in each simulation. To identify the optimal temperature, the user invokes an experiment

function (`Make_TempGrowthExp`) and inputs a vector with the test temperatures. The experiment for each temperature costs 100 EUR and takes ∼20 s simulation time. With a low probability of ∼10%, depending on investment to equipment, the growth fails and stays at the inocculum level. The outcome of this experiment is a csv-file and related to the format of the GrowthProfiler (Enzyscreen). During the course the students could choose data analysis in Excel or Python. In Excel, the data has to be correctly imported, followed by logarithmic operation and visualization. The experimental column with the highest slope is then be subjected to linear regression within the linear regime of the logarithmic data to determine the growth rate and the average biomass during stationary phase in the original data. We aimed to engage the students with scripting and provided guidance with Python code for the data analysis. The tasks was separated into two cell blocks, the first served for visualization (Figure 1) and the second for linear regression to extract growth rate and biomass. The code lines in each cell block were disordered in form of a Parsons-Puzzle. As in the Excel procedure, the user identifies the optimal temperature and extracts the associated growth rate and biomass. Their correct assessment is tested in the last experiment of the fermentation.

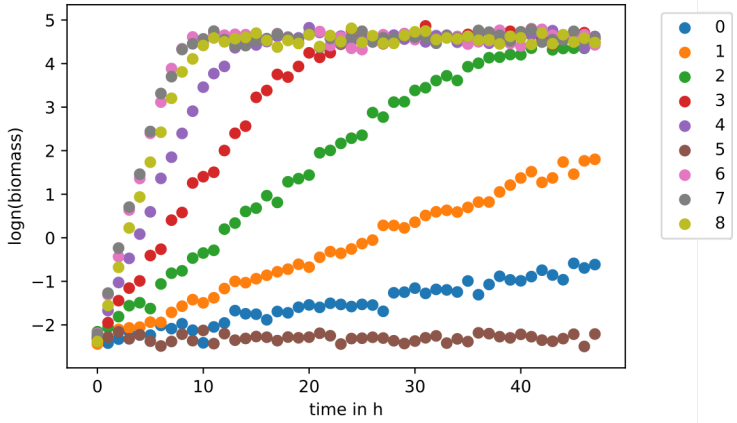

Fig 1. Example experiments to determine optimal growth temperature. Starting with 22° C to 38° C nine temperatures separated by 2° C were measured, which takes about 3 min simulation time. Experiment 7 (36° C) displayed the strongest growth increase until ∼7 h. Experiment 5 failed, with an expected growth rate in the order of experiment 8. The visualization was performed with the code lines to be ordered by the user.

Promoter construction and cloning

After characterizing the host organism, the user is tasked to identify a suitable promoter sequence along with cloning into the host. The key learning objectives are knowledge about the promoter boxes that control prokaryotic gene expression, computation of DNA melting temperature and base pair complements and appreciation of the fickleness of cloning. The users are provided with a 40 nt sequence reference:

GCCCAXXXXXXAXGCXXXCXCGTXXXGGXXXXXXTGCACG,

The X represent positions which the user has to replace with standard nucleotides. The first and last six X are the box positions and the optimal composition is explained in the accompanying text [27]. After the user has replaced the X with nucleotides, the primer

have to be developed. The primers start with the start of the promoter but the optimal primer length is unknown and randomly initiated between 16-28 nt. The user then calculates the melting temperature according to the basic formula given in Equation 3 in Table 1. The cloning experiment has a higher error rate and identical configurations will lead to different results with relatively frequent cloning failures. The cloning process is time consuming and weakly predictive and thus mirrors the process of reality. The students are motivated to try several versions of promoters to improve expression.

Simulation results and cross-connection

Following the successful design of promoters, the users measure vaccine expression values, examine correlations of GC content and expression and identify the most effective host. The learning objectives are that multiple promoters need to be constructed to find a good one, and that the GC-content is not predictive of expression strength. The users first measure the promoter strength of each promoter construct. This evaluation is based on a Random Forest model trained on promoter libraries of *E. coli* and *P. putida* as described in the Materials and Methods section [2]. The final experiment function (`Make_ProductionExperiment`) reports the expression production only if the host characterization information of optimal temperature, biomass and growth rate are within 10% of the correct values. As each group will only have tested 1-4 promoter variants, combining the results across the groups can lead to further insight. We have chosen to let the students identify correlations between GC content and expression strength. We constructed an online audience response system at ARS NOVA that allowed Hot-map feedback when clicking on squares overlaid to a diagram of GC content versus expression strength as if Figure 2 would be blank. ARS NOVA has meanwhile been converted into a commercial product and the Hot-map function was discontinued altogether. The final outcome of three courses is show in Figure 2. Apparently, there is no direct correlation among GC content and expression. Based on the most effective promoter, a winner was determined with the strongest vaccine production rate.

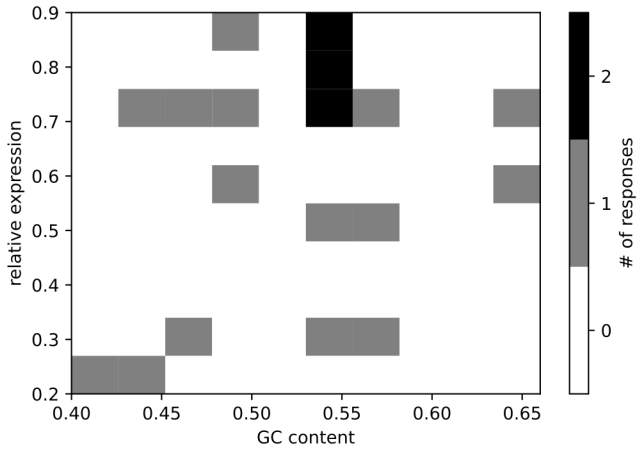

Fig 2. Joint simulation result of all groups with the relationship of GC-content to expression strength. The plot is an example and not the actual outcome of the class implementation. It shows the potential to combine results from all groups to arrive at new knowledge when the effort of all groups is integrated.

Course feedback evaluation

A feedback evaluation was performed to assess the degree of skill acquisition, performance reflection and simulation usability. The evaluation was requested from three independent groups with 25 students in total with 16 full responses ($\sim 64\%$). The feedback is based on Name of Feedback Scheme and refs and the detailed analysis is provided as supplementary. In the feedback, the students responded to questions with a slider which could be drawn to a negative and a positive side. The responses were quantified on a range 0-5, with 2.5 representing an indifferent result. The students started with low skills about programming and computational data analysis and over 75% reported new skill acquisition (average acquisition skill rating > 3). And although the students were working in a mostly unfamiliar environment, over 75% felt subjectively competent during the tasks (rating ¿3), and evaluated the Jupyter Notebook as user friendly. One critic related to the use of multiple programs such as the browser and Zoom, which was reported impractical without two monitors.

Discussion & Conclusion

We have developed a multi-model organism simulation to generate consistent experimental data for bacterial growth and gene expression. The experiments were combined as a Jupyter Notebook for teaching recombinant expression principles and data analysis to biology and biotechnology master students.

- cloning management with pyDNA (Ferreira et al., 2015)
- PCR trainer (Fellerman et al., 2018, 2019)
- Design-Build-Test-Learn ()
- Difference to scientific whole cell simulations (Karr et al. 2012, Skalnik et al. 2021, vivarium)
- biotechnology process models (Narayanan et al. 2020)

Supporting information

Acknowledgments

We thank Frank Eiden and Jonathan Sturm for support in the course implementation. We are gratefull for the support of course evaluation by the RWTH CLS with Alina Vogelsang, Philipp Weyers and Malte Persike. We thank the students for participating in the courses.

References

1. Oliveira AL. Biotechnology, Big Data and Artificial Intelligence. Biotechnol J. 2019;14(8):1800613. doi:10.1002/biot.201800613.
2. Liebal UW, Köbbing S, Blank LM. Exp2Ipynb: A general machine-learning workflow for the analysis of promoter libraries. bioRxiv. 2020; p. 2020.12.14.422740. doi:10.1101/2020.12.14.422740.

3. Fang X, Lloyd CJ, Palsson BO. Reconstructing organisms in silico: genome-scale models and their emerging applications. Nat Rev Microbiol. 2020; p. 1–13. doi:10.1038/s41579-020-00440-4.

4. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in {P}ython. J Mach Learn Res. 2011;12:2825–2830.

5. Kluyver T, Ragan-Kelley B, Perez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. In: Proc. 20th Int. Conf. Electron. Publ. ELPUB 2016. IOS Press; 2016. p. 87–90. Available from: https://ebooks.iospress.nl/doi/10.3233/978-1-61499-649-1-87.

6. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Čech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46(W1):W537–W544. doi:10.1093/NAR/GKY379.

7. Arkin AP, Cottingham RW, Henry CS, Harris NL, Stevens RL, Maslov S, et al. KBase: The United States Department of Energy Systems Biology Knowledgebase. Nat Biotechnol. 2018;36(7):566. doi:10.1038/nbt.4163.

8. Berthold MR, Cebron N, Dill F, R G, Kötter T, Meinl T, et al. KNIME - the Konstanz information miner. ACM SIGKDD Explor Newsl. 2009; p. 58–61. doi:10.1145/1656274.1656280.

9. Perkel JM. Why Jupyter is data scientists' computational notebook of choice. Nature. 2018;563(7729):145–146. doi:10.1038/D41586-018-07196-1.

10. Mendez KM, Pritchard L, Reinke SN, Broadhurst DI. Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. Metabolomics. 2019;15(10):125. doi:10.1007/s11306-019-1588-0.

11. Birkel GW, Ghosh A, Kumar VS, Weaver D, Ando D, Backman TWH, et al. The JBEI quantitative metabolic modeling library (jQMM): a python library for modeling microbial metabolism. BMC Bioinformatics. 2017;18(1):205. doi:10.1186/s12859-017-1615-y.

12. Cardoso JGR, Jensen K, Lieven C, Hansen ASL, Galkina S, Beber M, et al. Cameo: A Python Library for Computer Aided Metabolic Engineering and Optimization of Cell Factories. ACS Synth Biol. 2018;7(4):1163–1166. doi:10.1021/ACSSYNBIO.7B00423.

13. Fahnert B. Be prepared – Learning for the future. FEMS Microbiol Lett. 2019;366(16). doi:10.1093/femsle/fnz200.

14. Compeau P. Establishing a computational biology flipped classroom. PLOS Comput Biol. 2019;15(5):e1006764. doi:10.1371/journal.pcbi.1006764.

15. Jung H, Ventura T, Chung JS, Kim WJ, Nam BH, Kong HJ, et al. Twelve quick steps for genome assembly and annotation in the classroom. PLoS Comput Biol. 2020;16(11):e1008325. doi:10.1371/journal.pcbi.1008325.

16. Dillon MR, Bolyen E, Adamov A, Belk A, Borsom E, Burcham Z, et al. Experiences and lessons learned from two virtual, hands-on microbiome bioinformatics workshops. PLOS Comput Biol. 2021;17(6):e1009056. doi:10.1371/JOURNAL.PCBI.1009056.

17. Suárez A, Alvarez-Feijoo MA, Fernández González R, Arce E. Teaching optimization of manufacturing problems via code components of a Jupyter Notebook. Comput Appl Eng Educ. 2018;26(5):1102–1110. doi:10.1002/cae.21941.

18. Seddighi M, Allanson D, Rothwell G, Takrouri K. Study on the use of a combination of IPython Notebook and an industry-standard package in educating a CFD course. Comput Appl Eng Educ. 2020;28(4):952–964. doi:10.1002/cae.22273.

19. Rule A, Birmingham A, Zuniga C, Altintas I, Huang SC, Knight R, et al. Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. PLOS Comput Biol. 2019;15(7):e1007007. doi:10.1371/journal.pcbi.1007007.

20. Brown NCC, Wilson G. Ten quick tips for teaching programming. PLOS Comput Biol. 2018;14(4):e1006023. doi:10.1371/journal.pcbi.1006023.

21. Brown CL, Comunale MA, Wigdahl B, Urdaneta-Hartmann S. Current climate for digital game-based learning of science in further and higher education. FEMS Microbiol Lett. 2018;365(21):237. doi:10.1093/FEMSLE/FNY237.

22. Koivisto J, Hamari J. The rise of motivational information systems: A review of gamification research. Int J Inf Manage. 2019;45:191–210. doi:10.1016/j.ijinfomgt.2018.10.013.

23. McEvoy JP. Interactive problem-solving sessions in an introductory bioscience course engaged students and gave them feedback, but did not increase their exam scores. FEMS Microbiol Lett. 2017;364(18):182. doi:10.1093/FEMSLE/FNX182.

24. Lubbock ALR, Lopez CF. Programmatic modeling for biological systems. Curr Opin Syst Biol. 2021;doi:10.1016/j.coisb.2021.05.004.

25. Agmon E, Spangler RK, Skalnik CJ, Poole W, Peirce SM, Morrison JH, et al. Vivarium: an interface and engine for integrative multiscale modeling in computational biology. bioRxiv. 2021;doi:10.1101/2021.04.27.441657.

26. Chuang LY, Cheng YH, Yang CH. Specific primer design for the polymerase chain reaction; 2013. Available from: `https://link.springer.com/article/10.1007/s10529-013-1249-8`.

27. Paget M. Bacterial sigma factors and anti-sigma factors: structure, function and distribution. Biomolecules. 2015;5(3):1245–1265. doi:10.3390/biom5031245.