The Biophysicist

Machine Learning in a Molecular Modeling Course for Chemistry, Biochemistry, and Biophysics Students

Jacob M. Remington[1], **Jonathon B. Ferrell**[1], **Marlo Zorman**[1], **Adam Petrucci**[1], **Severin T. Schneebeli**[1], **Jianing Li**[1,*]

[1]Department of Chemistry, University of Vermont, Burlington, VT 05405, USA

ABSTRACT Recent advances in computer hardware and software, particularly the availability of machine learning (ML) libraries, allow the introduction of data-based topics such as ML into the biophysical curriculum for undergraduate and graduate levels. However, there are many practical challenges of teaching ML to advanced level students in biophysics majors, who often do not have a rich computational background. Aiming to overcome such challenges, we present an educational study, including the design of course topics, pedagogic tools, and assessments of student learning, to develop the new methodology to incorporate the basis of ML in an existing biophysical elective course and engage students in exercises to solve problems in an interdisciplinary field. In general, we observed that students had ample curiosity to learn and apply ML algorithms to predict molecular properties. Notably, feedback from the students suggests that care must be taken to ensure student preparations for understanding the data-driven concepts and fundamental coding aspects required for using ML algorithms. This work establishes a framework for future teaching approaches that unite ML and any existing course in the biophysical curriculum, while also pinpointing the critical challenges that educators and students will likely face.

KEY WORDS machine learning; pedagogic tools; course design; computational biophysics; molecular biophysics

I. INTRODUCTION

Machine learning (ML), as a category of artificial intelligence (AI), includes a wide variety of methods and tools to train on a set of data and then create rules or knowledge from the data. In particular, biophysicists and chemists are interested in the applications to biochemical and biophysical data and the potential power of these methods to predict molecular properties, which are important in driving the structure of biomolecules and enzymatic activity between protein and substrate, among other macroscopic properties. The historical use of ML on molecules tracks to the very early days of computers in the 1960s, which mainly learned parameters in quantitative structure activity relationships (1). Around the same time, the first method for encoding molecules into computer-readable formats, in the form of Morgan fingerprints, was invented (2). Although different encoding mechanisms, for example, simplified molecular input line entry system (SMILE) strings and their derivatives, were driven by the need for chemical intuition, the development of ML techniques was done outside of biological sciences and then applied back to biochemical and biophysical problems (3–5). Later, the

"*" corresponding author

Received: 29 December 2019
Accepted: 20 May 2020
Published: 13 August 2020

perceptron method, related to modern day artificial neural networks became popular to predict drug efficacy from the early 1970s to the 1990s (6).

Although the earliest research focused on small molecules and generally emphasized drug discovery, this is not the only area for biophysicists to explore with ML techniques. According to a recent report about the Biophysical Society annual meetings (7), there is a fast-growing trend to adopt ML in a variety of biophysics-related fields ranging from computational (such as genetic mutational and sequence-based studies, feature detection and dimensional reduction of conformational spaces, the study of complex kinetics, and force field parameterization for simulations) to experimental techniques (such as analyses of different microscopy imaging techniques). In aggregate, ML and related applications can be revolutionary to biophysics. Recent progress in protein structure prediction illustrates an excellent example of this revolution.

Biophysicists can determine the 3-dimensional (3D) structures of proteins by using experimental techniques such as cryoelectron microscopy, nuclear magnetic resonance, and X-ray crystallography. However, these experiments are often lengthy and costly, depending on trials and errors. Thus, protein structure prediction with a given amino acid sequence remains a core biophysical challenge, which has already involved enormous efforts (8–12), such as supercomputer development (Blue Gene and Anton) and novel citizen experiments (Folding@home and Foldit) (13–17). In addition to the advance in computational power, the algorithm innovation is critical. As early as 1994, the first critical assessment of protein structure prediction (CASP) competition, an event held to encourage improvement of protein structure prediction algorithms, had an entry using a neural network (an ML method) implemented to predict protein secondary structure within the SYBYL software program (18). Notably, this introduction foretold the recent success of AlphaFold, which used cutting edge ML techniques to predict pairwise amino acid residue distance and achieve high accuracy in CASP13 (19). In the assessment, the structure prediction results from AlphaFold were shown to be far more accurate than any that have come before in the CASP series. Following this advance, a variety of ML methods were developed building off the success of AlphaFold (20, 21), which likely reflects the remarked difference made by ML to biophysics.

Acknowledging the growing impact of ML in the biophysical literature, we provide here an account of teaching ML principles and applications within a biophysical elective course. This effort is outlined by first exploring biophysical data types with the students and then focusing on cheminformatics, as it provides an interface between the chemical building blocks of biology and data input for computers, before finally introducing basic ML algorithms to the students. Along the way, we demonstrate concrete examples and provide our experience designing a case study of ML for the students to complete as a project. This work will aid the development of teaching tools for educators to bring ML into the biophysics curriculum.

II. SCIENTIFIC AND PEDAGOGIC BACKGROUND

Thanks to many well-publicized examples, such as the defeat of human masters in the games of chess and Go, the success of self-driving cars, the vast improvements of language processing, and the success in what many consider an impossible task of protein structure prediction, ML methods have gained widespread popularity. This popularity, however, has not been adequately embraced by current biophysical education. On one hand, a diverse set of cutting edge ML tools has been made available to the public with the release of Tensorflow by Google, CNTK by Microsoft, and (py)Torch by Facebook. These tools are further supplemented by simpler and more intuitive libraries, such as SciKit-Learn. On the other hand, there are still often misconceptions, concerns, and suspicions about ML from scientists outside of computer science. Practically, the often less than transparent algorithms embedded in ML packages can require careful

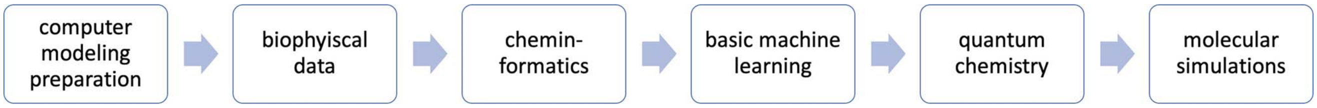

Fig 1. Flow chart of the course structure for a total of 26 lectures. Specifically, "preparation of computer modeling" for 2 lectures, 3 topics from "biochemical and biophysical data" to "basic ML" for 8 lectures, and "quantum chemistry" and "molecular simulations" for 8 lectures each.

tuning of a small set of control variables, which are generally referred to as the hyperparameters. The power of ML (as described in section I) and the increasing ease of use implies a necessity to include it in modern biophysical training, as is currently being done in other fields such as chemistry (22). It is crucial to provide the learning opportunity for future biophysicists to understand the diverse tool kit in ML methods outside of the well-publicized versions, recognize strengths and limitations, and gain the knowledge or ability to apply them appropriately under different circumstances with the correct choice of hyperparameters.

The major pedagogic challenge arises from the apparent disconnect between the data science–heavy topic of ML and the more biologic science–based curriculum. Rather than setting up a special topic course to only introduce ML, we experimented with incorporating ML material into the framework of a molecular modeling course, which is an elective for undergraduate and graduate students on a biophysics track. The course "Special Topics: Computational Chemistry, Biochemistry and Biophysics" (3 credits) offered in the 2019 fall semester at the University of Vermont (UVM) was chosen due to the diverse backgrounds of the students yet common interest in computational tools. The overall goal of the course is to provide students with methods on how to model different molecules in computers and how to calculate the properties and reactive pathways, with a special focus on various molecules of biophysical interest. We selected 3 general topics (biochemical and biophysical data, cheminformatics, and basic ML) to supplement existing topics in the course (such as molecular mechanics and quantum mechanics). The course included 12 students officially registered, 4 senior undergraduate students and 8 (mostly in their first year) graduate students. Each lecture or class was 1 h and 15 min, and the class met twice a week.

Students in the course had diverse training backgrounds and research interests. However, all of them generally wanted to learn about how to use computers to aid chemical or biophysics research. Because many of the students did not have an extensive background in coding and data science, we opted to focus on providing a practical introduction of ML with an emphasis on chemical problems, rather than a comprehensive overview of ML. The primary goals of this teaching approach were to introduce students to the topics of biochemical and biophysical data and cheminformatics, guide students through a project that uses ML for hands-on experience, encourage students to think like a data scientist, and apply ML as a future biophysicist or chemist. In the rest of this work, we discuss the design of topics, selection of teaching materials, and assessment, which may be useful for educators in biophysics and related fields.

III. MATERIALS AND METHODS

At the beginning of the course, students learned the basic skills of computer modeling with commercial software programs, such as Maestro and Pymol (Schrödinger). They were also motivated after a tour to the supercomputing center, Vermont Advanced Computing Core, with the state-of-the-art graphics processing unit cluster DeepGreen at UVM. With these preparations, we approached the 3 topics of biochemical and biophysical data, cheminformatics, and basic ML in about 8 lectures, before the introduction of traditional topics, such as quantum chemistry calculations and molecular simulations in the rest of the course (Fig 1). Note that these topics were

Table 1. Comparison between SMILES and SMARTS.

SMILES	SMARTS
SMILES describes molecules	SMARTS describes patterns
The resultant molecule of the SMILES string is subject to searching	The pattern described by the SMARTS string is matched against molecules
Atoms and bonds are specified in SMILES	Unspecified properties are not defined to be part of the pattern in SMARTS
All SMILES expressions are also valid SMARTS expressions	Most SMARTS expressions are not valid SMILES expressions

carefully organized and taught with our ultimate goal in mind: to critically understand the current strengths and limitations of ML methods and rationally grasp the real potential from the current hype surrounding ML.

We were aware of the challenge to find the most updated materials at the appropriate level from a textbook. Therefore, we adopted teaching materials from 3 areas, including the tutorials of SMILES (molecular structures) and SMILES arbitrary target specification (SMARTS; chemical patterns), the tutorial of RDKit, open-source tool kit for cheminformatics, and, finally, the tutorial of DeepChem, a Python library for deep learning. All teaching materials were accessible for students via our course management system Blackboard.

A. Biochemical and biophysical data

The primary goals of introducing biochemical and biophysical data were to help students understand what biochemical and biophysical data to include, as well as how to represent, store, and use biochemical and biophysical data. Up to September 2019, there were 96 million compounds in PubChem and 76 million in ChemSpider. Modern drug discovery projects may have to examine millions of compounds to find an active one. Associated with each compound are a large number of properties, such as solubility, acidity, toxicity, and phase transitions, affecting mechanism and function in biophysics. Thus, during the first lecture, we asked students to discuss the type and size of data that they generated from teaching or research labs, as well as how the data were stored. After encouraging students to think about how to search for compounds by name, molecular formulae, structures, and other features in compound databases, we introduced an overview of the SMILES language, the SMARTS pattern, and the RDKit tool kit.

B. Chemoinformatics

SMILES is a line notation (a typographic method using printable characters) for entering and representing molecules and reactions (3, 4). SMILES represents a universal, comprehensive chemical nomenclature, which shows the molecular structure in a string, facilitating data storage and efficient searching. As it is commonly used in cheminformatics and compound databases, we introduced the rules to represent atoms and molecules, with special efforts to explain the representation of stereochemistry due to its importance in chemistry and biology (i.e., double bonds and chiral carbon centers). To enhance learning, sufficient examples and exercises were provided during and after each lecture.

Following the introduction of SMILES, we demonstrated how SMARTS is useful for substructure searching. SMARTS is a language that allows users to specify substructures by using rules that are straightforward extensions of SMILES. We started with a simple example molecule, phenol, due to the biophysical importance of phenolic compounds (23–25) in the regulation of lipid and protein activities. To search in a database for phenol-containing structures, one would use the SMARTS string [OH]c1ccccc1. For a flexible and efficient substructure search, the basic rules of SMARTS were introduced to students. To enhance the learning effects, we also discussed the comparison between SMILES and SMARTS (Table 1).

Combining SMILES and SMARTS, we introduced several example applications (i.e., substructure searching, molecular similarity

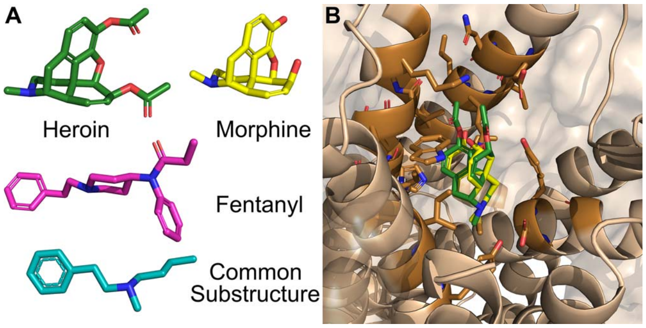

Fig 2. (A) Chemical structures of heroin (green carbons), morphine (yellow carbons), fentanyl (pink carbons), and the MCS (teal carbons) from the output of the example script (Supplemental Fig S3). In each structure, hydrogen atoms are neglected, oxygen atoms are red, and nitrogen atoms are green. (B) Heroin (green) and morphine (yellow) aligned to the morphinan antagonist bound state of the μ-opioid receptor (37).

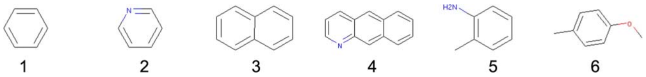

Fig 3. Image (generated by the RDkit tool) of a compound set used to demonstrate the concept of molecular fingerprinting.

searching, and molecular fingerprinting) in the context of RDKit. Short and simple Python scripts (Supplemental Figs S1 and S2) using the RDKit library were experimented with by the students and discussed in detail. To better engage students, we created several real-life examples, with the biophysical background introduced along with the technical ML details. One of the examples entailed signifying the structural similarity between heroin, morphine, and fentanyl, well-known opioids that act on the same opioid receptor (Fig 2). Although morphine and heroin appear to have similar molecular structures, fentanyl appears quite distinct. We provided rational measurements of the similarity by using the fingerprint similarity (numeric measurement; Supplemental Fig S2) and the maximum common substructure (MCS, Supplemental Fig S3), which further inspired students to think about the reasons why these 2 compounds act on the same receptor protein, as well as the deeper reason for the ongoing fentanyl crisis. Students were encouraged to modify the scripts (provided in the Supplemental Material) for exercises and share thoughts during the in-class discussions.

Another aspect of cheminformatics as well as biochemical and biophysical data types is at the apex of the challenge presented by teaching ML to students with relatively little data science knowledge. To approach this challenge, we chose to include course presentations on various biochemical and biophysical data types early on to assist the students in recognizing that the abstract idea of a chemical species can be quantified into numeric data. We started with basic 3D structural data file formats (e.g., .xyz and .pdb), as the spatial coordinates of a molecule represent arguably its most obvious numeric representation, before moving onto molecular fingerprints (2), which instead quantify the presence of different functional groups. In other words, molecular fingerprints encode molecular structures into a series of binary digits that represents the presence or absence of particular substructures (the so-called keys). Examples explored by the students are shown in Table 2 and Figure 3. Molecular fingerprints were chosen as they appeal to the chemical intuition that students develop in other chemistry courses, such as general, bio-, or organic chemistry (which are often part of a biophysical curriculum). In these courses, students are

Table 2. Design of simple molecular fingerprints. The fingerprints for the molecules shown in Figure 3 (numbered 1 to 6) are generated with the listed fingerprint keys.

	Fingerprint keys					
Molecular number	cccccc	[N,n,O,o]	[NX3]	Ncccccc	CaaaaO	c(c)(c)c
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	1	0	0	0	0	1
4	1	1	0	0	0	1
5	1	1	1	1	0	0
6	1	1	0	0	1	0

taught to break apart a complex molecule into functional groups and to consider how the presence of each group affects the molecule's properties, a concept that is fundamental to molecular fingerprints.

Note that the example molecules were selected arbitrarily for proof of concept in the reported practice. In a future biophysics course, an instructor should pick more biophysically relevant examples and exercises. For instance, after showing the example illustrated in Table 2 and Figure 3 in class, we recommend carrying out a student assignment to design new molecular fingerprints (by focusing mainly on the fingerprint keys) for a group of preselected biophysical molecules (e.g., the natural amino acids, nucleic acids, or key intermediates involved in a biosynthetic pathway such as glycolysis). On the basis of the results reported in our article, we envision that having students design a minimal number of keys for a fingerprint that captures the similarities among amino acids will further foster the abilities to generalize the fundamental concepts explored. A detailed example for this process (illustrated for 5 amino acids) is provided in section 4 of the Supplemental Material.

C. A case study of ML

To introduce students to the concept of ML, we started by building on the students' prior understanding of regression analysis and calibration curves. However, as a key distinction to regression analysis, we explained to the students early on that the choice of an ML algorithm can introduce greatly enhanced flexibility for determining relationships between the input and output data, compared with the relatively simple model functions commonly used for regression analysis. Furthermore, at the start of class, we also provided a brief overview of ML, as well as a number of nomenclature distinctions to help the students further explore ML on their own time. For example, we addressed questions such as in the following (see the ML "cheat sheet" in the Supplemental Material for additional information provided to the students): (a) What is supervised and unsupervised learning? (b) What is the distinction between AI and ML? (c) What are the different categories and applications of ML (26)? Overall, we strongly encourage the teaching of ML principles through the lens of explorative learning, instead of directly lecturing students on the benefits of individual models. To meet this aim, we approached and focused this learning module on a case study of aqueous solubility prediction. Teaching ML principles with explorative learning was chosen as the students (much like in an ML algorithm) were to try different approaches to solving a problem and to learn mostly independently which of the many available models is best able capture patterns present in the data. Overall, the primary aim of our discovery-based approach toward ML was to empower students with better intuition, rather than with the high-level and abstract mathematic representations of ML models that are often presented in a more classic lecture or presentation-style class.

Aqueous solubility is a key physical property for biophysicists because solubility affects the uptake and/or distribution of biologically active compounds. The ability of a compound to partition into different components of the cell influences what targets it can reach and ultimately affects its potential efficacy. Accurate equilibrium solubility determination is a time-consuming experiment, and it is useful to be able to assess solubility in the absence of a physical sample. With ML, it is viable to develop a simple method for estimating the aqueous solubility of a compound directly from its structure. The data set provided by an early article by Delaney (27) contains 2,874 measured solubilities. We prepared the data file in the simple comma-separated value format, with the first few lines of the file shown to the students. With data from the last 2 fields labeled as "SMILES" and "measured log solubility in moles per liter," we constructed our ML model, with the Python script provided in the Supplemental Material.

Although there are many different ML algorithms, we chose the random forest (RF) model to learn the structure–solubility relationship from the molecules and compounds in the training set. The algorithm of RF was explained in detail during the class (Supplemental Fig S4). Briefly, the RF model is composed of multiple decision trees. These decision trees are trained on preclassified input data (in this case, the chemical solubilities along with SMILES strings for many molecules), which allows the trees to learn some heuristics from the input data and "decide" what is the correct class to be in. The RF then polls all of the individual trees and takes the most popular classification as the correct answer. Then, we prepared the data set (e.g., featurization, splitting), fitted simple learning models to our training data and evaluated the model on the validation set to determine its predictive power, constructed stronger models and optimized hyperparameters, and made the predictions. With the detailed introduction of the breakdown, students could see an example of how each stage of the ML process ultimately affects the predictive power.

IV. RESULTS AND DISCUSSION

A. Assessing the learning effect with a student competition to design an ML method to predict pK_a

The pH across different tissues and organelles can vary greatly from 4 to 8 and is an important property that affects the intrinsic activity and self-assembly behavior of biologic molecules (28, 29). The choice of pK_a for this competition was influenced by the accessible data and the biophysical importance.

It is often key but challenging to assess the learning outcomes when new concepts or techniques are introduced (30, 31). To provide data and evidence for future course improvement and implementation, we designed a student activity and a survey. First, the activity was formatted as a competition, with bonus points for the final grade as the prize. At the beginning of the ML topic, students were asked to collect training data and prepare an ML model that would best predict the pK_a values of 10 undisclosed molecules, revealed on the competition day by the instructor. This motivated the students to think creatively in attempts to design superior algorithms. Thus, the inherent complexity of the task and the students' relative unfamiliarity with programming encouraged cooperation and the exchange of advice. The class was split into 2 teams, with a nearly even distribution of graduate and undergraduate students. Each team was led by a student with programming experience. Given the limited data size, students were able to run the programs on a laptop.

In 3 weeks, both teams adopted similar approaches. The team programmers familiarized themselves with the free-to-use Python application programming interfaces, including RDKit (32) for cheminformatics and SciKit-Learn (33) for ML. They adapted example scripts from online ML tutorials to produce functioning models. This involved converting SMILES to readable formats and one-hot encoding chemicals on the basis of the functional groups. The other members of each team were tasked with collecting data in the form of molecules, the SMILES strings, and pK_a values in dimethyl sulfoxide (DMSO). Teams used databases (e.g., the Bordwell pK_a table; 34) and literature (35, 36) to ultimately compile 200 to 400 data points (which are experimentally determined pK_a values of organic molecules in DMSO and the corresponding SMILES strings; see the Supplemental Material for details). Team programmers used the respective data sets to optimize parameters for pK_a prediction.

At competition time during the last class for the ML topic, the students first computed the chosen fingerprints for the 10 test compounds before running the ML algorithms to predict the pK_a values. The results of the 2 groups are shown (Fig 4 and Supplemental Table S1), which demonstrate the challenges of the ML models designed by the students. Group 1 had the greatest range in predicted pK_a values from a minimum of 10 to a maximum of 24, while the pK_a group 2 only had a range of 17 to 25. Neither group was able to correctly capture the high and low pK_a values. As ML models are only as good as the training data sets, having the students report summary statistics for pK_a from

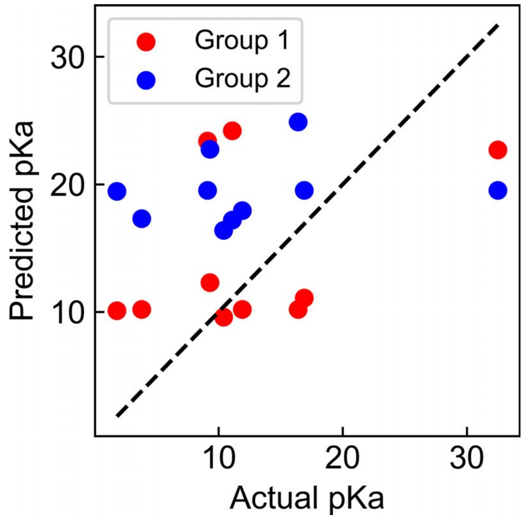

Fig 4. A scatter plot of pK_a values for the 10 test compounds chosen by the teacher with actual values from literature (22–24) on the *x* axis and predicted values on the *y* axis for the 2 groups respectively. The line $y = x$ is plotted as a dashed line.

these training sets (e.g., mean, range) would better allow them to assess why these models failed to predict the test compounds. Another aspect to improve this project would be to compute the maximum molecular similarity of each test compound with all other compounds of the training sets. This would allow students to recognize that the predictive power of the model is limited by how similar molecules in the test set are to molecules the ML algorithm was trained on.

The students found that a critical stage of the hands-on application of the ML algorithm was hyperparameterization. From a biophysical or chemical standpoint, this is simultaneously often the most intriguing and difficult aspect of the computational technique. For example, in contrast with chemical intuition, it was observed by students during the project that removing consideration of alcohols and ketones improved pK_a predictions. Similarly, the inclusion of thioketones improved results. Therefore, directing the in-lecture focus to the process of parametrization, the phenomena of underfitting and overfitting, and the purpose of random variables inherent to RF algorithms would improve both an understanding of fundamental ML and the relevance to biophysics.

B. Assessing student experience with ML by using a student survey

Following the completion of the competition, students were asked to respond to 3 questions: (a) What did you learn from the pK_a ML project and related lectures? (b) What did you like most when working on the project, and are you going to read or study the related topics? (c) What improvement can we implement for teaching ML and related topics in the future? These questions were formulated to provide insight into the effectiveness of the teaching approach, the interest the students had in the topic, and future improvements that could be made, respectively. A graphic summary of students' responses in this survey is shown in Table 3.

The most common response to question (a) centered on how the students gained recognition that ML allows predictions to be made from a large, high-quality data set. Furthermore, the students recognized similarities between ML algorithms and calibration curves they were already familiar with. Importantly, students reported 2 fundamental aspects of successful ML applications: larger training data sets increase the effectiveness of ML and the choice of the inputs (fingerprint keys) affects the accuracy of ML. Because ML technologies are becoming more widespread in our society and there is a sense that they are a "black box" and outside of human control, one response by a student was extraordinarily appropriate: "I thought it would be straightforward and automatic; instead, there was a lot of param-

Table 3. Summary of the students' responses to the 3-question survey about the pK_a prediction using the ML project.

Learning outcomes	Interest in ML	Future suggestions
Predictive power	Multitude of applications	More coding examples
Better data = better predictions	More ML algorithms	More preliminary ML assignments
Human choice of input affects machine output	Coding and computer science	Better distribution of workload in groups

eterization work to be done on the human end." We believe that such a response highlights that ultimately humans still control an ML algorithm, and it further emphasizes the educational utility of providing a hands-on ML project to future chemists who may move into a workforce in which successful implementation or understanding of ML will be advantageous. In particular, teaching ML in this hands-on manner helps demystify the black box nature of ML.

In response to question (b) about the students' interest, the most common responses were about how they wanted to apply ML algorithms to more applications such as their own research, drug discovery, quantum computing, and even biophysical or chemical structure prediction. Students also suggested that they would like to learn more about the different types of ML algorithms themselves. The genuine excitement about the topic suggests that this is a promising area of biophysical education research that should be explored further. Finally, even though students struggled with the programming (as evident by responses to question (c) in the following), they actually enjoyed the coding that they were able to do, wanted to learn a programming language, and even showed interest in taking a computer science course in which they could learn more.

To the final question (c), student criticism generally fell into 2 categories: programming preparation and group assignments. Overall, most students felt that they did not have the programming knowledge necessary to prepare ML models. This was due, in part, to it not being made clear that groups could modify RDKit and DeepChem example codes covered in class but mostly due to a general lack of a programming background. As a result, mostly team programmers worked on the ML code. The true significance of the programming portion is to introduce students to the structure of computational biophysics or chemistry code, not to teach them to design such a code. Class examples that highlight crucial lines of code and worksheets that involve filling in certain keywords or comment blocks along an example script would help highlight important processes and more efficiently ingrain fundamental ML ideas. Applying these concepts to the project could make it more accessible to those with little or no programming background.

It was suggested by students to include more assignments before the project, as they found that the process of programming models themselves clarified key ML concepts. The suggestions of more assignments emphasize new opportunities for this framework to be further embedded in a biophysics curriculum, where initial ML assignments and concepts can be taught along with initial biophysics concepts. For example, while introducing amino acids and which ones are considered charged, students could be simultaneously taught to train a simple ML algorithm to predict which amino acids are charged by giving it sequences and the total charges of those sequences. The methods of ML can be increased in complexity along with predicting more complex properties, such as alpha helicity, thus allowing courses to be properly adjusted to student abilities, as well as giving a broad method for this framework to be embedded in a variety of biophysics courses. This highlights the importance of hands-on experience, a theme applicable to teaching computational biophysics and teaching more generally.

C. Transferability of our materials to other courses in a biophysics curriculum

On the basis of our findings and the molecular focus in the design, it is viable to teach ML in most existing core or elective courses in a biophysics curriculum, at either the undergraduate or the graduate level.

(a) For instance, all the small molecule examples are directly applicable to the core components in an undergraduate curriculum, such as general chemistry, organic chemistry, and introductory molecular biology and biophysics.
(b) With preparation of simple Python scripting, the breadth or depth can be readily increased for a graduate-level course with

more profound discussions about various ML methods and applications. For example, the neural networks for protein structure or function prediction may be suitable to incorporate into advanced biophysical courses that discuss macromolecular structures and functions.

(c) The materials used in this work can inspire further development of course resources to introduce biophysical lab techniques, such as spectroscopy and microscopes. In practice, it will be critical to determine the suitable level of depth for teaching the theory behind ML, for example, with consideration given to the learning goals of the specific course, as well as to student preparation and interests. Further, it may be helpful to use outcome-based design (31), which sets teaching or learning goals early in the course and allows for timely adjustments during the actual teaching practice.

V. CONCLUSION

Aiming to overcome the challenge of teaching ML to students in biophysics and related fields, we describe an educational study, including the design of data and ML-related topics in an existing biophysics elective course, pedagogic tools, and assessments of student learning, to develop the new methodology to teach the basis of ML and engage students in exercises to solve chemical problems with some biophysical applications. Direct assessment of the learning effect with a student competition allowed students to recognize the predictive power and limitations of current ML methods. Indirect assessment with a simple, effective student survey revealed the importance of student preparations and hands-on experience for the teaching and learning of ML. These assessments provide new directions to implement changes for our future practice (e.g., computational labs and outcome-based course design). In summary, this work establishes a framework for future teaching approaches that unite ML and any course in the existing biophysical curriculum, while also identifying critical challenges during teaching and learning of this important topic.

SUPPLEMENTAL MATERIAL

Further information on the design of the student competition, student survey analysis, example Python scripts, and machine learning resources and tutorials, with additional figures and tables, is available at: https://doi.org/10.35459/tbp.2019.000140.S1.

ACKNOWLEDGMENTS

We thank Andrea Elledge, Jim Lawson, and Andy Evans from the Vermont Advanced Computing Core for supporting this study as well as Professors Christopher Landry and Rory Waterman at the University of Vermont for helpful discussions. JL was partially supported by an ACS PRF award (58219-DNI) and a National Science Foundation (NSF) CAREER award (CHE-1945394); STS was partially supported by an NSF CAREER award (CHE-1848444); and JR was supported by a National Institutes of Health grant (R01GM129431).

AUTHOR CONTRIBUTIONS

JL conceived the study and taught the course. STS and JMR cotaught the course. JL, JMR, and JBF analyzed the data, and all authors, including MZ and AP, wrote the manuscript.

REFERENCES

1. Hansch, C., and T. Fujita. 1964. p-σ-π Analysis. A method for the correlation of biological activity and chemical structure. *J Am Chem Soc* 86(8):1616–1626.
2. Morgan, H. L. 1965. The generation of a unique machine description for chemical structures—a technique developed at Chemical Abstracts Service. *J Chem Doc* 5(2):107–113.
3. Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 28(1):31–36.
4. Weininger, D., A. Weininger, and J. L. Weininger. 1989. SMILES. 2. Algorithm for generation of unique SMILES notation. *J Chem Inf Comput Sci* 29(2):97–101.
5. Hanson, R. M. 2016. Jmol SMILES and Jmol SMARTS: specifications and applications. *J Cheminform* 8(1):50.
6. Hiller, S. A., V. E. Golender, A. B. Rosenblit, L. A. Rastrigin, and A. B. Glaz. 1973. Cybernetic methods of drug design. I. Statement of the problem—the perceptron approach. *Comput Biomed Res* 6(5):411–421.
7. The Biophysical Society. 2019. BPS2019—playing catch with machine learning trends. Vol. 2020.
8. Li, J., R. Abel, K. Zhu, Y. Cao, S. Zhao, and R. A. Friesner. 2011. The VSGB 2.0 model: a next generation energy model for high resolution protein structure modeling. *Proteins* 79(10):2794–2812.
9. Zhao, S., K. Zhu, J. Li, and R. A. Friesner. 2011. Progress in super long loop prediction. *Proteins* 79(10):2920–2935.
10. Kryshtafovych, A., T. Schwede, M. Topf, K. Fidelis, and J. Moult. 2019. Critical assessment of methods of protein structure prediction (CASP)—round XIII. *Proteins* 87(12):1011–1020.
11. Yang, J., I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, and D. Baker. 2020. Improved protein structure prediction using predicted interresidue orientations. *Proc Natl Acad Sci U S A* 117(3):1496.
12. Dill, K. A., and J. L. MacCallum. 2012. The protein-folding problem, 50 years on. *Science* 338(6110):1042–1046.
13. Almási, G., R. Bellofatto, J. Brunheroto, C. Caşcaval, J. G. Castaños, L. Ceze, P. Crumley, C. C. Erway, J. Gagliano, D. Lieber, X. Martorell, J. E. Moreira, A. Sanomiya, and K. Strauss. An overview of the Blue Gene/L system software organization. In Proceedings of the 9th International Euro-Par Conference on

Parallel Processing. Klagenfurt, Austria, 26–29 August 2003. Springer, Berlin, pp. 543–555.

14. Shaw, D., M. Deneroff, R. Dror, J. Kuskin, R. Larson, J. Salmon, C. Young, B. Batson, K. Bowers, J. Chao, M. P. Eastwood, J. Gagliardo, J. P. Grossman, C. R. Ho, D. Ierardi, I. Kolossváry, J. L. Klepeis, T. Layman, C. McLeavey, M. A. Moraes, R. Mueller, E. C. Priest, Y. Shan, J. Spengler, M. Theobald, B. P. Towles, and S. C. Wang. 2008. Anton, a special-purpose machine for molecular dynamics simulation. *Commun ACM* 51(7):91.
15. Shaw, D. E., J. P. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton, A. Forte, J. Gagliardo, G. Gill, B. Greskamp, C. R. Ho, D. J. Ierardi, L. Iserovich, J. S. Kuskin, R. H. Larson, T. Layman, L. Lee, A. K. Lerer, C. Li, D. Killebrew, K. M. Mackenzie, S. Y. Mok, M. A. Moraes, R. Mueller, L. J. Nociolo, J. L. Peticolas, T. Quan, D. Ramot, J. K. Salmon, D. P. Scarpazza, U. B. Schafer, N. Siddique, C. W. Snyder, J. Spengler, P. T. P. Tang, M. Theobald, H. Toma, B. Towles, B. Vitale, S. C. Wang, and C. Young. Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. New Orleans, LA, November 2014. IEEE Press, New York, pp. 41–53.
16. Beberg, A. L., D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande. Folding@home: lessons from eight years of volunteer distributed computing. In Proceedings of the 2009 IEEE International Parallel & Distributed Processing Symposium. Rome, Italy, May 2009. IEEE, New York, pp. 1–8.
17. Kleffner, R., J. Flatten, A. Leaver-Fay, D. Baker, J. B. Siegel, F. Khatib, and S. Cooper. 2017. Foldit Standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics* 33(17):2765–2767.
18. Moult, J., J. T. Pedersen, R. Judson, and K. Fidelis. 1995. A large-scale experiment to assess protein structure prediction methods. *Proteins* 23(3):ii–iv.
19. Senior, A., W. R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis. 2020. Improved protein structure prediction using potentials from deep learning. *Nature* 577(7792):706–710.
20. AlQuraishi, M. 2019. End-to-end differentiable learning of protein structure. *Cell Syst* 8(4):292–301.e3.
21. Billings, W. M., B. Hedelius, T. Millecam, D. Wingate, and D. D. Corte. 2019. ProSPr: democratized implementation of Alphafold protein distance prediction network. *bioRxiv* 2019:830273.
22. Joss, L., and E. A. Müller. 2019. Machine learning for fluid property correlations: classroom examples with MATLAB. *J Chem Educ* 96(4):697–703.
23. Dinis, T. C., V. M. Maderia, and L. M. Almeida. 1994. Action of phenolic derivatives (acetaminophen, salicylate, and 5-aminosalicylate) as inhibitors of membrane lipid peroxidation and as peroxyl radical scavengers. *Arch Biochem Biophys* 315(1):161–169.
24. Ishtikhar, M., E. Ahmad, Z. Siddiqui, S. Ahmad, M. V. Khan, M. Zaman, M. K. Siddiqi, S. Nusrat, T. I. Chandel, M. R. Ajmal, and R. H. Khan. 2018. Biophysical insight into the interaction mechanism of plant derived polyphenolic compound tannic acid with homologous mammalian serum albumins. *Int J Biol Macromol* 107(Pt. B):2450–2464.
25. Kim, Y. A., S. G. Gaidin, and Y. S. Tarahovsky. 2018. The influence of simple phenols on collagen type I fibrillogenesis in vitro. *Biophysics* 63(2):162–168.
26. SAS Institute Inc. 2017. Which machine learning algorithm should I use? Accessed 1 August 2019. https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/.
27. Delaney, J. S. 2004. ESOL: estimating aqueous solubility directly from molecular structure. *J Chem Inf Comput Sci* 44(3):1000–1005.
28. Ye, Z., H. Zhang, H. Luo, S. Wang, Q. Zhou, X. Du, C. Tang, L. Chen, J. Liu, Y. K. Shi, E. Y. Zhang, R. Ellis-Behnke, and X. Zhao. 2008. Temperature and pH effects on biophysical and morphological properties of self-assembling peptide RADA16-I. *J Pept Sci* 14(2):152–162.
29. Shahul Hameed, U. F., C. Liao, A. K. Radhakrishnan, F. Huser, S. S. Aljedani, X. Zhao, A. A. Momin, F. A. Melo, X. Guo, C. Brooks, Y. Li, X. Cui, X. Gao, J. E. Ladbury, Ł. Jaremko, M. Jaremko, M. J. Li, and S. T. Arold. 2018. H-NS uses an autoinhibitory conformational switch for environment-controlled gene silencing. *Nucleic Acids Res* 47(5):2666–2680.
30. Ferrell, J. B., J. P. Campbell, D. R. McCarthy, K. T. McKay, M. Hensinger, R. Srinivasan, X. Zhao, A. Wurthmann, J. Li, and S. T. Schneebeli. 2019. Chemical exploration with virtual reality in organic teaching laboratories. *J Chem Educ* 96(9):1961–1966.
31. Towns, M. H. 2010. Developing learning objectives and assessment plans at a variety of institutions: examples and case studies. *J Chem Educ* 87(1):91–96.
32. Landrum, G. RDKit: Open-source cheminformatics. Accessed 1 August 2019. http://www.rdkit.org.
33. Fabian Pedregosa, G. V., A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. 2011. SciKit-Learn: machine learning in Python. *J Mach Learn Res* 12(2011):2825–2830.
34. Bordwell, F. G. 1988. Equilibrium acidities in dimethyl sulfoxide solution. *Acc Chem* 21(12):456–463.
35. Li, J., L. Liu, Y. Fu, and Q.-X. Guo. 2006. What are the pK_a values of organophosphorus compounds? *Tetrahedron* 62(18):4453–4462.
36. Shen, K., Y. Fu, J. Li, L. Liu, and Q.-X. Guo. 2007. What are the pK_a values of C–H bonds in aromatic heterocyclic compounds in DMSO? *Tetrahedron* 63(7):1568–1576.
37. RCSB Protein Data Bank (entry number 4DKL). Accessed 3 August 2019. https://www.rcsb.org/structure/4DKL.