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

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

Department of Chemistry, The University of Vermont, Burlington, VT 05403

* Jianing Li (jianing.li@uvm.edu, 82 University Place, Burlington, VT 05403)

Abstract: Recent advances in computer hardware and software, particularly the availability of machine learning libraries, allow the introduction of data-based topics such as machine learning into the Biophysical curriculum for undergraduate and/or graduate levels. However, there are many practical challenges of teaching machine learning to advanced-level students in the 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, pedagogical tools, and assessments of student learning, to develop the new methodology to incorporate the basis of machine learning 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 machine learning 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 machine learning algorithms. This work establishes a framework for future teaching approaches that unite machine learning 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, pedagogical tools, course design, computational biophysics, molecular biophysics.

1. 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/biophysical data and the potential power of these methods to predict molecular properties, which are important in driving the structure of biomolecules, 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 (QSARs).[1] Around the same time, the first method for encoding molecules into computer readable formats, in the form of Morgan fingerprints, was invented.[2] While different encoding mechanisms (e.g. SMILE strings[3-4] and its

derivatives[5]) 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. Later, the perceptron method, related to modern day artificial neural networks became popular to predict drug efficacy from the early 1970s to the 1990s.[6]

While 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 (BPS) 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, studying complex kinetics, force-field parameterization for simulations, etc.) 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 such revolution.

Biophysicists can determine the three-dimensional (3D) structures of proteins using experimental techniques like cryo-electron 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 the given amino acid sequence remains a core biophysical challenge, which has already involved enormous efforts[8-12] like supercomputer development (BlueGene[13] and Anton[14-15]), and novel citizen experiments (Folding@Home[16] and FoldIt[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 (a ML method) implemented to predict protein secondary structure within the SYBYL software program. 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 (2018).[18] In the assessment, the structure prediction results from AlphaFold were shown far more accurate than any that have come before in CASP series. Following this advance, a variety of ML methods were developed building off the success of AlphaFold,[19-20] 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 its applications within a biophysical elective course. This effort is outlined by first exploring biophysical data types with the students, 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. It is anticipated that this work will aid the development of teaching tools for educators to bring ML into the biophysics curriculum.

2. Scientific and Pedagogical 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 like 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 tuning of a small set of control variables, which are generally referred to as the hyperparameters. The power of ML (as described in the Introduction) and the increasing ease of use implies a necessity to include it in modern biophysical training, as has currently being done in other fields like chemistry.[21] It is crucial to provide the learning opportunity for future biophysicists to (1) understand the diverse tool kit that is ML methods outside of the well-publicized versions, (2) recognize their strengths and limitations, and (3) gain the knowledge/ability to apply them appropriately under different circumstances with the correct choice of hyperparameters.

The major pedagogical challenge arises from the apparent disconnect between the data science-heavy topic of ML and the more biological 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 biophysics-track undergraduate and graduate students. The course "Special Topics: Computational Chemistry, Biochemistry and Biophysics" (CHEM 267, 3 credits) offered in the 2019 Fall semester at the University of Vermont (UVM) was chosen due its students' diverse backgrounds, 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 their properties and reactive pathways, with a special focus on various molecules of biophysical interest. We selected three general topics (biochemical/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/class was 1 hour and 15 min, and the class met twice a week.

Students in the course had diverse training backgrounds and research interests, yet generally wanted to learn about how to use computers to aid chemical/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 emphasis on chemical problems, rather than a comprehensive overview of ML. The primary goals of this teaching approach were to (1) introduce students to the topics of biochemical/biophysical data and cheminformatics, (2) guide students through a project that uses ML for hands on experience, (3) encourage students to think like a data scientist, and (4) apply ML as a future biophysicist/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.

3. Materials and Methods.

At the beginning of the course, students learned the basic skills of computer modeling with commercial software programs like Maestro and Pymol (Schrödinger). They were also motivated after a tour to the supercomputing center, Vermont Advanced Computing Core (VACC) with the state-of-the-art GPU cluster *DeepGreen* at UVM. With these preparations, we approached the three topics of **(1) Biochemical/biophysical Data, (2) Cheminformatics, and (3) Basic Machine Learning** in ~8 lectures, before the introduction of traditional topics like quantum chemistry calculations and molecular simulations in the rest of the course. It is noteworthy that these topics were carefully organized and taught with our ultimate goal in mind, which was for students 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 three areas, including the tutorials of Simplified Molecular Input Line Entry System (**SMILES**, molecular structures) and **SMARTS** (chemical patterns), the tutorial of **RDKit**, which is an open source toolkit for cheminformatics, and finally the tutorial of **DeepChem**, which is a Python library for deep learning. All our teaching materials were accessible for students via our course management system Blackboard.

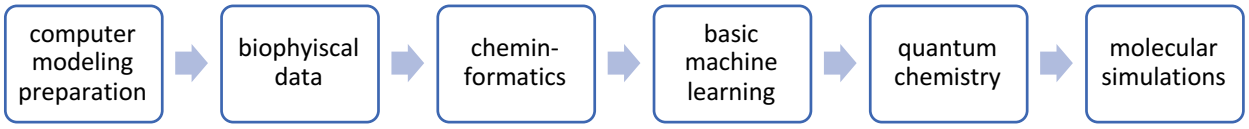

Figure 1 Flow chart of the course structure for a total of 26 lectures. Specifically, “preparation of computer modeling” for 2 lectures, the three topics from “biochemical/biophysical data” to “basic machine learning” for 8 lectures, “quantum chemistry” and “molecular simulations” for 8 lectures each.

(1) Biochemical/biophysical Data.

The primary goals of introducing Biochemical/biophysical Data were to help students understand (1) what biochemical/biophysical data to include, as well as (2) how to represent, store, and utilize biochemical/biophysical data. Up to September 2019, there are 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, there are a large number of properties, such as solubility, acidity, toxicity, phase transitions, etc., which affect its mechanism and function in biophysics. Thus, during the first lecture, we asked students to discuss the type and size of data which they generated from their teaching/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 toolkit.

(2) Cheminformatics.
SMILES is a line notation (a typographical 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 using rules that are straightforward extensions of SMILES. We started with a simple example molecule — phenol, due to the biophysical importance of phenolic compounds[22-24] 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 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).

Table 1 Comparison between SMILES and SMARTS.

SMILES	SMARTS
1. SMILES describes molecules; 2. The resultant molecule of the SMILES string is subject to searching; 3. Atoms and bonds are specified in SMILES; 4. All SMILES expressions are also valid SMARTS expressions.	1. SMARTS describes patterns; 2. The pattern described by the SMARTS string is matched against molecules; 3. Unspecified properties are not defined to be part of the pattern in SMARTS; 4. Most SMARTS expressions are not valid SMILES expressions.

Combining SMILES and SMARTS, we introduced several example applications, i.e. (1) substructure searching, (2) molecular similarity searching, and (3) molecular fingerprinting, in the context of RDKit. Short and simple Python scripts using the RDKit library were experimented with by the students and discussed in detail (Figures S1 and S2). 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 employed entailed signifying the structural similarity between heroin, morphine, and fentanyl, well-known opioids that act on the same opioid receptor (Figure 2). While morphine and heroin appear to have similar molecular structures, fentanyl appears quite distinct. We provided rational measurements of the similarity using the fingerprint similarity (numerical measurement, Figure S2) and the maximum common substructure (MCS, Figure S3), which further inspired students to think about the reasons why

these two 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 SI) for exercises and share their thoughts during the in-class discussions.

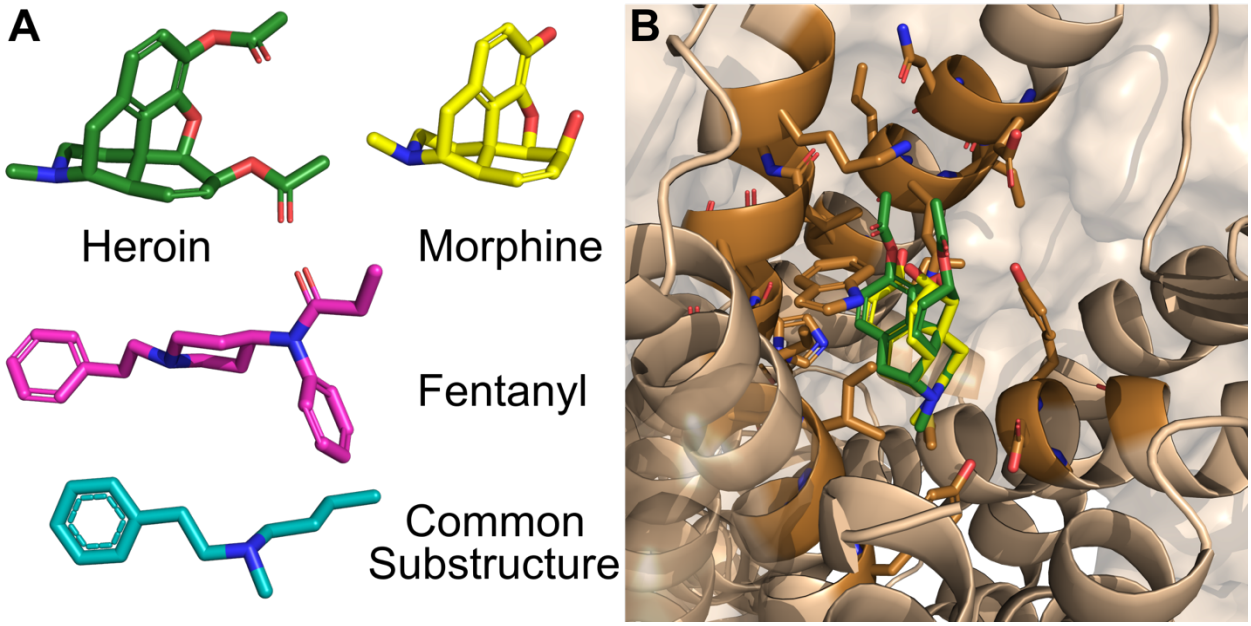

Figure 2 (A) Chemical structures of heroin (green carbons), morphine (yellow carbons), fentanyl (pink carbons), and the maximum common substructure (MCS, teal carbons) from the output of the example script S3. In each structure hydrogen atoms are neglected, oxygens red, and nitrogen atoms are green. **(B)** Heroin (green) and morphine (yellow) aligned to the morphinan antagonist bound state of the μ-opioid receptor (PDBID: 4DKL).[25]

Another aspect of cheminformatics, biochemical/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/biophysical data types early on to assist the students in recognizing that the abstract idea of a chemical species can be quantified into numerical 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 numerical representation, before moving onto molecular fingerprints[2] which instead quantify the presence of different functional groups. In other words, molecular fingerprints encode molecule structures into a series of binary digits that represent 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 which 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 taught to break apart a complex molecule into functional groups and to consider how the presence of groups affect its chemical properties, a concept that is fundamental to molecular fingerprints.

these two 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 SI) for exercises and share their thoughts during the in-class discussions.

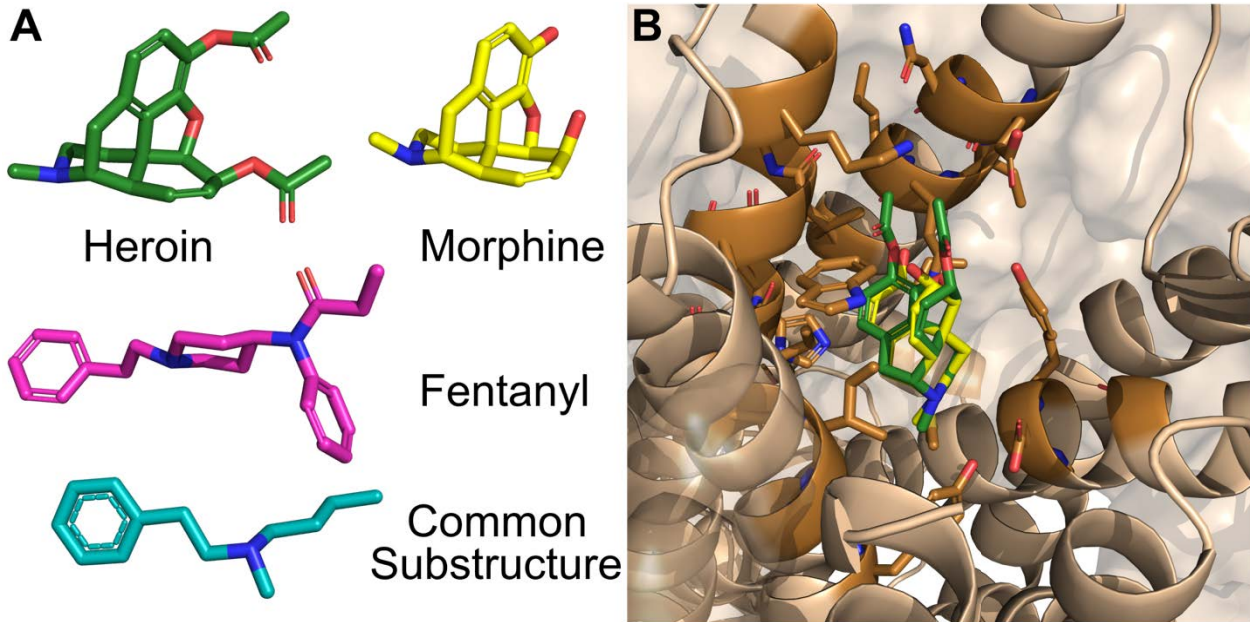

Figure 2 (A) Chemical structures of heroin (green carbons), morphine (yellow carbons), fentanyl (pink carbons), and the maximum common substructure (MCS, teal carbons) from the output of the example script S3. In each structure hydrogen atoms are neglected, oxygens red, and nitrogen atoms are green. **(B)** Heroin (green) and morphine (yellow) aligned to the morphinan antagonist bound state of the μ-opioid receptor (PDBID: 4DKL).[25]

Another aspect of cheminformatics, biochemical/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/biophysical data types early on to assist the students in recognizing that the abstract idea of a chemical species can be quantified into numerical 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 numerical representation, before moving onto molecular fingerprints[2] which instead quantify the presence of different functional groups. In other words, molecular fingerprints encode molecule structures into a series of binary digits that represent 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 which 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 taught to break apart a complex molecule into functional groups and to consider how the presence of groups affect its chemical properties, a concept that is fundamental to molecular fingerprints.

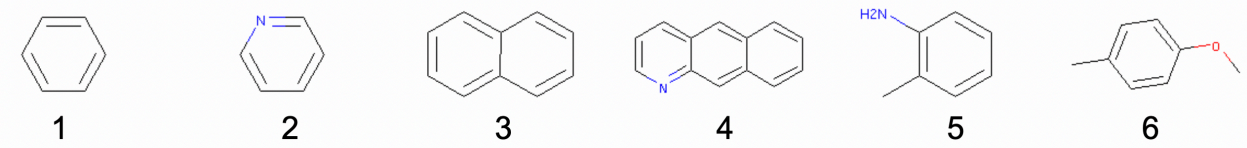

Figure 3 Image (generated by the rdkit tool) of a compound set used to demonstrate the concept of molecular fingerprinting.

It must be pointed out that these 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 pre-selected biophysical molecules (e.g. the natural amino acids, nucleic acids, or key intermediates involved in a biosynthetic pathway like glycolysis). Based on the results reported in our paper, we envision that having the students design a minimal number of keys for a fingerprint that captures the similarities among the amino acids will further foster the abilities to generalize the fundamental concepts explored. A detailed example for this process (illustrated for five amino acids) is provided in Section 4 of the SI.

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

mol. no. \ FP keys	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

(3) A case study of Machine Learning.

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 a ML algorithm can introduce greatly enhanced flexibility for determining relationships between the input/output data, compared to 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. For example, we addressed questions like (please see the ML "cheat sheet" for additional information provided to the students): (i) What is supervised and unsupervised learning? (ii) What is the distinction between artificial intelligence and ML? (iii) 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 to 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 a ML algorithm), were to try different approaches to solving a problem and to learn mostly on their own which of the many available models is best able capture patterns present in their data. Overall, the primary aim of our discovery-based approach toward ML was to empower students with better intuition — rather than with the often high-level and abstract mathematical representations of ML models — that are often presented in a more classical lecture/presentation style class.

Aqueous solubility is a key physical property for biophysicists because solubility affects the uptake/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 paper by Delaney[27] contains 2874 measured solubilities. We prepared the data file in the simple csv format, with the first few lines of the file shown to the students. With data from the last two fields labeled as "smiles" and "measured log solubility in mols per litre", we constructed our ML model, with the Python script provided in the supporting materials.

While there are many different ML algorithms, we chose the random forest (RF) model to learn the structure-solubility relationship from the molecules/compounds in the train set. The algorithm of RF was explained in detail during the class (Figure S4). Briefly, a random forest model is composed of multiple decision trees. These decision trees are trained on pre-classified input data, in this case the chemical solubilities along with SMILES strings for many molecules. This allows the trees to learn some heuristics from the input data and "decide" what is the correct class to be in. A random forest then polls all of the individual trees and takes the most popular classification as the correct answer. Then we (1) prepared the data set (featurization, splitting, etc.), (2) fit simple learning models to our train data and evaluate the model on the validation set to determine its predictive power, (3) constructed stronger models and optimize hyperparameters, and (4) 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.

4. Results and Discussion.

4.1 Assessing the learning effect with a student competition to design a ML method to predict *p*K_a. The *p*H 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 biological molecules. [28-29] The choice of *p*K_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/techniques are introduced.[30-31] To provide data and evidence for future course improvement and

implementation, we designed a student activity and a survey. Firstly, 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 a ML model that would best predict the pK_a values of ten undisclosed molecules, revealed on the competition day by the instructor. This motivated the students to think creatively in attempts to design superior algorithms. That said, the inherent complexity of the task and the students' relative unfamiliarity with programming encouraged cooperation and the exchange of advice. The class was split in two 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 their programs on a laptop.

In a period of three weeks, both teams adopted similar approaches. The team programmers familiarized themselves with the free-to-use Python application programming interfaces (APIs) 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 based on their functional groups. The other members of each team were tasked with collecting data in the form of molecules, their SMILES strings, and pK_a values in DMSO. Teams used databases (e.g. the Bordwell pK_a Table) and literature[34-35] to ultimately compile 200-400 data points (which are experimentally determined pK_a values of organic molecules in DMSO and the corresponding SMILES strings, see the SI for detail). Team programmers used their 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 their chosen fingerprints for the ten test compounds before running their ML algorithms to predict the pK_a values. The results of the two groups are shown (Figure 4 and 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 their training data sets, having the students report summary statistics for pK_a from their training sets (mean, range, etc.) would better allow them to asses why their 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 their training sets. This would allow students to recognize that the predictive power of their model is limited by how similar molecules in the test set are to molecules the ML algorithm was trained on.

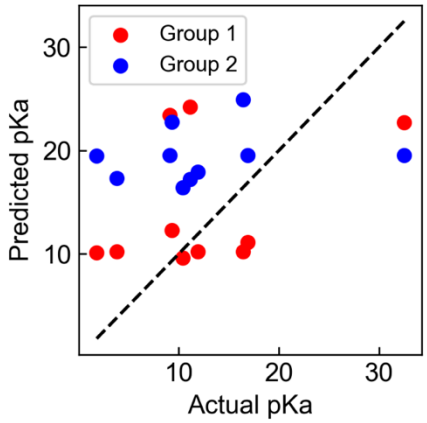

Figure 4 A scatter plot of pK_a values for the ten test compounds chosen by the teacher with actual values from literature[21-23] on the x-axis and predicted values on the y-axis for the two groups respectively. The line y = x is plotted as a dashed line.

The students found that a critical stage of the hands-on application of the ML algorithm was hyperparameterization. From a biophysical/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 in-lecture focus to the process of parametrization, the phenomena of under-fitting and over-fitting, and the purpose of random variables inherent to random forest algorithms would improve both an understanding of fundamental ML and its relevance to biophysics.

4.2 Assessing student experience with ML using a student survey.

Following the completion of the competition students were asked to respond to three questions: (1) What did you learn from the pK_a ML project and related lectures? (2) What did you like most when working on the project, and are you going to read or study the related topics? (3) What improvement can we implement for teaching ML and related topics in the future. These questions were formulated respectively 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. A graphical summary of students' responses in this survey is shown in Table 3.

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

Learning Outcomes	Interest in ML	Future Suggestions
• Predictive Power. • Better Data = Better Predictions. • Human Choice of Input Affects Machine Output.	▪ Multitude of Applications. ▪ More Machine Learning Algorithms. ▪ Coding and Computer Science.	♦ More Coding Examples. ♦ More Preliminary ML Assignments. ♦ Better Distribution of Workload in Groups.

The most common response to question (1) centered on how the students gained recognition that ML allows predictions to be made from a large, high quality dataset. Furthermore, they recognized similarities between ML algorithms and calibration curves they were already familiar with. Importantly, students reported two fundamental aspects of successful ML applications: larger training datasets 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 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 parameterization work to be done on the human end*". We believe that such response highlights the fact that ultimately humans still control a 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/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 the question about the students' interest (2), the most common responses were about how they wanted to apply ML algorithms to more applications like their own research, drug discovery, quantum computing, and even biophysical/chemical structure prediction. Students also suggested that they wanted 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 which should be explored further. Finally, even though students struggled with the programming (as evident by responses to question 3 below), they actually enjoyed the coding which they were able to do, wanted to learn a programming language, and even showed interest in taking a computer science course where they could learn more.

To the final question (3) student criticism generally fell into two 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 programming background. As a result, mostly only team programmers worked on the ML code. The true significance of the programming portion is to introduce students to the structure of computational biophysics/chemistry code, not to teach them to design such 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 programming models themselves clarified 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.

4.3 Transferability of our materials into other courses in a biophysics curriculum.

Based on 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, for either the undergraduate or the graduate level.

(1) For instance, all the small molecule examples are directly applicable to the core components in an undergraduate curriculum like general chemistry, organic chemistry, and introductory molecular biology/biophysics.

(2) With preparation of simple Python scripting, the breadth and/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/function prediction may be suitable to incorporate into advanced biophysical courses that discuss macromolecular structures and functions.

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

5. 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, pedagogical 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 that 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, outcome-based course design, etc.). In summary, this work establishes a framework for future teaching approaches that unite ML and any course in the existing biophysical curriculum, and also identifies critical challenges during teaching and learning.

Author Contributions

J.L. conceived the study and taught the course. S.T.S. and J.M.R. co-taught the course. J.L., J.M.R., and J.B.F. analyzed the data and all the authors together wrote the manuscript.

Acknowledgements

We thank Andrea Elledge, Jim Lawson, and Andy Evans from VACC for supporting this study and Prof. Christopher Landry, and Prof. Rory Waterman at the University of Vermont for helpful discussions. J.L. was partially supported by the ACS PRF award (58219-DNI) and an NSF CAREER award (CHE-1945394); S.T.S. was partially supported by an NSF CAREER award (CHE-1848444); J.R. was supported by the NIH grant (R01GM129431).

References

1. Hansch, C.; Fujita, T., p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *Jounral of the American Chemical Society* **1964,** *86* (8), 1616-1626.
2. Morgan, H. L., The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation* **1965,** *5* (2), 107-113.
3. Weininger, D., SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **1988,** *28* (1), 31-36.
4. Weininger, D.; Weininger, A.; Weininger, J. L., SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences* **1989,** *29* (2), 97-101.
5. Hanson, R. M., Jmol SMILES and Jmol SMARTS: specifications and applications. *Journal of Cheminformatics* **2016,** *8* (1), 50.
6. Hiller, S. A.; Golender, V. E.; Rosenblit, A. B.; Rastrigin, L. A.; Glaz, A. B., Cybernetic methods of drug design. I. Statement of the problem--the perceptron approach. *Computers and Biomedical Research, an International Journal* **1973,** *6* (5), 411-21.
7. Society, B., BPS2019 - Playing catch with machine learning trends. 2019; Vol. 2020.
8. Li, J.; Abel, R.; Zhu, K.; Cao, Y.; Zhao, S.; Friesner, R. A., The VSGB 2.0 model: A next generation energy model for high resolution protein structure modeling. *Proteins: Structure, Function, and Bioinformatics* **2011,** *79* (10), 2794-2812.
9. Zhao, S.; Zhu, K.; Li, J.; Friesner, R. A., Progress in super long loop prediction. *Proteins: Structure, Function, and Bioinformatics* **2011,** *79* (10), 2920-2935.
10. Kryshtafovych, A.; Schwede, T.; Topf, M.; Fidelis, K.; Moult, J., Critical assessment of methods of protein structure prediction (CASP)—Round XIII. *Proteins: Structure, Function, and Bioinformatics* **2019,** *87* (12), 1011-1020.
11. Yang, J.; Anishchenko, I.; Park, H.; Peng, Z.; Ovchinnikov, S.; Baker, D., Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* **2020,** *117* (3), 1496.
12. Dill, K. A.; MacCallum, J. L., The Protein-Folding Problem, 50 Years On. *Science* **2012,** *338* (6110), 1042.

13. G., A., An Overview of the Blue Gene/L System Software Organization. *Euro-Par 2003 Parallel Processing* **2003,** *790*, 543-555.
14. Shaw, D.; Deneroff, M.; Dror, R.; Kuskin, J.; Larson, R.; Salmon, J.; Young, C.; Batson, B.; Bowers, K.; Chao, J.; al., e., Anton, a Special-Purpose Machine for Molecular Dynamics Simulation. *Commun. ACM* **2008,** *51* (7), 91.
15. Shaw, D. E.; Grossman, J. P.; Bank, J. A.; Batson, B.; Butts, J. A.; Chao, J. C.; Deneroff, M. M.; Dror, R. O.; Even, A.; ., C. H. F.; et al., 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*, IEEE Press: New Orleans, Louisana, 2014; pp 41–53.
16. Beberg, A. L.; Ensign, D. L.; Jayachandran, G.; Khaliq, S.; Pande, V. S., Folding@home: Lessons from Eight Years of Volunteer Distributed Computing. *Ipdps '09* **2009**, 1–8.
17. Kleffner, R.; Flatten, J.; Leaver-Fay, A.; Baker, D.; Siegel, J. B.; Khatib, F.; Cooper, S., Foldit Standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics* **2017,** *33* (17), 2765-2767.
18. Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W. R.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Crossan, S.; Kohli, P.; Jones, D. T.; Silver, D.; Kavukcuoglu, K.; Hassabis, D., Improved protein structure prediction using potentials from deep learning. *Nature* **2020,** *577* (7792), 706-710.
19. AlQuraishi, M., End-to-End Differentiable Learning of Protein Structure. *Cell Systems* **2019,** *8* (4), 292-301.e3.
20. Billings, W. M.; Hedelius, B.; Millecam, T.; Wingate, D.; Corte, D. D., ProSPr: Democratized Implementation of Alphafold Protein Distance Prediction Network. *bioRxiv* **2019**, 830273.
21. Joss, L.; Müller, E. A., Machine Learning for Fluid Property Correlations: Classroom Examples with MATLAB. *Journal of Chemical Education* **2019,** *96* (4), 697-703.
22. Dinis, T. C.; Maderia, V. M.; Almeida, L. M., Action of phenolic derivatives (acetaminophen, salicylate, and 5-aminosalicylate) as inhibitors of membrane lipid peroxidation and as peroxyl radical scavengers. *Arch Biochem Biophys* **1994,** *315* (1), 161-9.
23. Ishtikhar, M.; Ahmad, E.; Siddiqui, Z.; Ahmad, S.; Khan, M. V.; Zaman, M.; Siddiqi, M. K.; Nusrat, S.; Chandel, T. I.; Ajmal, M. R.; Khan, R. H., Biophysical insight into the interaction mechanism of plant derived polyphenolic compound tannic acid with homologous mammalian serum albumins. *International journal of biological macromolecules* **2018,** *107* (Pt B), 2450-2464.
24. Kim, Y. A.; Gaidin, S. G.; Tarahovsky, Y. S., The Influence of Simple Phenols on Collagen Type I Fibrillogenesis in vitro. *Biophysics* **2018,** *63* (2), 162-168.
25. Manglik, A.; Kruse, A. C.; Kobilka, T. S.; Thian, F. S.; Mathiesen, J. M.; Sunahara, R. K.; Pardo, L.; Weis, W. I.; Kobilka, B. K.; Granier, S., Crystal structure of the μ-opioid receptor bound to a morphinan antagonist. *Nature* **2012,** *485* (7398), 321-326.
26. SAS Which machine learning algorithm should I use? https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/. .
27. Delaney, J. S., ESOL: estimating aqueous solubility directly from molecular structure. *J Chem Inf Comput Sci* **2004,** *44* (3), 1000-5.

28. Ye, Z.; Zhang, H.; Luo, H.; Wang, S.; Zhou, Q.; Du, X.; Tang, C.; Chen, L.; Liu, J.; Shi, Y. K.; Zhang, E. Y.; Ellis-Behnke, R.; Zhao, X., Temperature and pH effects on biophysical and morphological properties of self-assembling peptide RADA16-I. *Journal of peptide science : an official publication of the European Peptide Society* **2008,** *14* (2), 152-62.
29. Shahul Hameed, U. F.; Liao, C.; Radhakrishnan, A. K.; Huser, F.; Aljedani, S. S.; Zhao, X.; Momin, A. A.; Melo, F. A.; Guo, X.; Brooks, C.; Li, Y.; Cui, X.; Gao, X.; Ladbury, J. E.; Jaremko, Ł.; Jaremko, M.; Li, J.; Arold, S. T., H-NS uses an autoinhibitory conformational switch for environment-controlled gene silencing. *Nucleic. Acids. Res.* **2018,** *47* (5), 2666-2680.
30. Ferrell, J. B.; Campbell, J. P.; McCarthy, D. R.; McKay, K. T.; Hensinger, M.; Srinivasan, R.; Zhao, X.; Wurthmann, A.; Li, J.; Schneebeli, S. T., Chemical Exploration with Virtual Reality in Organic Teaching Laboratories. *Journal of Chemical Education* **2019,** *96* (9), 1961-1966.
31. Towns, M. H., Developing Learning Objectives and Assessment Plans at a Variety of Institutions: Examples and Case Studies. *Journal of Chemical Education* **2010,** *87* (1), 91-96.
32. *RDKit: Open-source cheminformatics*.
33. Fabian Pedregosa, G. V., Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay, Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011,** *12*, 2825-2830.
34. Li, J.; Liu, L.; Fu, Y.; Guo, Q.-X., What are the pKa values of organophosphorus compounds? *Tetrahedron* **2006,** *62* (18), 4453-4462.
35. Shen, K.; Fu, Y.; Li, J.; Liu, L.; Guo, Q.-X., What are the p K a values of C–H bonds in aromatic heterocyclic compounds in DMSO? *Tetrahedron* **2007,** *63*, 1568-1576.

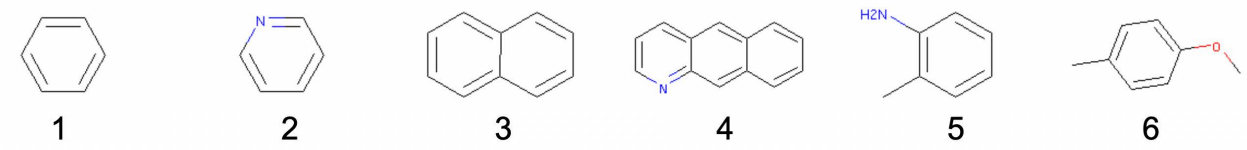

Figure 3 Image (generated by the rdkit tool) of a compound set used to demonstrate the concept of molecular fingerprinting.

It must be pointed out that these 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 pre-selected biophysical molecules (e.g. the natural amino acids, nucleic acids, or key intermediates involved in a biosynthetic pathway like glycolysis). Based on the results reported in our paper, we envision that having the students design a minimal number of keys for a fingerprint that captures the similarities among the amino acids will further foster the abilities to generalize the fundamental concepts explored. A detailed example for this process (illustrated for five amino acids) is provided in Section 4 of the SI.

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

mol. no. \ FP keys	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

(3) A case study of Machine Learning.

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 a ML algorithm can introduce greatly enhanced flexibility for determining relationships between the input/output data, compared to 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. For example, we addressed questions like (please see the ML "cheat sheet" for additional information provided to the students): (i) What is supervised and unsupervised learning? (ii) What is the distinction between artificial intelligence and ML? (iii) 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 to students on the benefits of individual models. To meet this aim, we approached and focused this learning

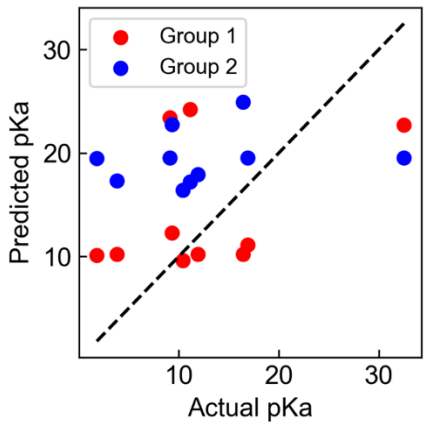

Figure 4 A scatter plot of *p*K_a values for the ten test compounds chosen by the teacher with actual values from literature[21-23] on the x-axis and predicted values on the y-axis for the two groups respectively. The line y = x is plotted as a dashed line.

The students found that a critical stage of the hands-on application of the ML algorithm was hyperparameterization. From a biophysical/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 *p*K_a predictions. Similarly, the inclusion of thioketones improved results. Therefore, directing in-lecture focus to the process of parametrization, the phenomena of under-fitting and over-fitting, and the purpose of random variables inherent to random forest algorithms would improve both an understanding of fundamental ML and its relevance to biophysics.

4.2 Assessing student experience with ML using a student survey.

Following the completion of the competition students were asked to respond to three questions: (1) What did you learn from the *p*K_a ML project and related lectures? (2) What did you like most when working on the project, and are you going to read or study the related topics? (3) What improvement can we implement for teaching ML and related topics in the future. These questions were formulated respectively 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. A graphical summary of students' responses in this survey is shown in Table 3.

Table 3. Summary of the students' responses to the three-question survey about the *p*K_a prediction using ML project.

Learning Outcomes	Interest in ML	Future Suggestions
• Predictive Power. • Better Data = Better Predictions. • Human Choice of Input Affects Machine Output.	▪ Multitude of Applications. ▪ More Machine Learning Algorithms. ▪ Coding and Computer Science.	♦ More Coding Examples. ♦ More Preliminary ML Assignments. ♦ Better Distribution of Workload in Groups.