Rediscovering the particle in a box: Machine learning regression analysis for hypothesis generation in physical chemistry lab

Elizabeth S. Thrall,[1]* Fernando Martinez Lopez,[2] Thomas J. Egg,[1] Seung Eun Lee,[2] Joshua Schrier,[1] Yijun Zhao[2]

[1]Department of Chemistry, Fordham University, The Bronx, NY 10458, United States

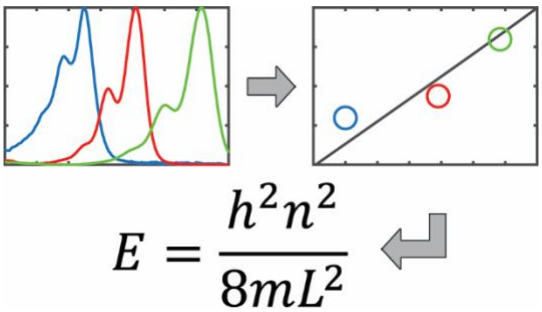
[2]Department of Computer and Information Sciences, Fordham University, New York, NY 10023, United States

*To whom correspondence should be addressed

ABSTRACT

Given the growing prevalence of computational methods in chemistry, it is essential that undergraduate curricula introduce students to these approaches. One such area is the application of machine learning (ML) techniques to chemistry. Here we describe a new activity that applies ML regression analysis to the common physical chemistry laboratory experiment on the electronic absorption spectra of cyanine dyes. In the classic version of this experiment, students collect experimental spectra and interpret them using the Kuhn free electron model, based on the quantum mechanical particle-in-a-box (PIB). Our new computational activity has students train regression models of increasing complexity to predict the wavelength of maximum absorption for different cyanine dyes using a set of 13 molecular features. In addition, the activity introduces methods for evaluating and interpreting regression models. Ultimately, students are prompted to use their regression analysis results to generate hypotheses for what molecular properties underlie cyanine dye absorption, leading them naturally to the PIB model. In this report, we provide a dataset, reference code implementations in Mathematica and Python notebooks, and an example lab protocol with an introduction to cyanine dyes and the ML techniques. This activity can be completed in a single 3-hour lab period by upper-level undergraduate students with relatively little prior programming experience. Although intended to complement the experimental measurement of cyanine dye spectra, this activity can also be performed on its own; alternatively, it can form the basis of more involved projects in a computational chemistry or ML course.

GRAPHICAL ABSTRACT



KEYWORDS

Upper-Division Undergraduate, Physical Chemistry, Computer-Based Learning, Computational Chemistry, Machine Learning, Spectroscopy, Cyanine

Introduction:

Some common physical chemistry laboratory experiments have been taught in largely the same manner for decades. Although these experiments illustrate fundamental concepts, they can be enhanced by supplementation with modern computational methods. This has the two-fold advantage of not requiring new instrumentation and of training students in programming and computational chemistry, which are increasingly recognized as key skills for chemists to possess.[1–7] One classic physical chemistry experiment explores the UV-visible absorption spectroscopy of cyanine dyes. Cyanine dyes contain a conjugated polymethine chain, consisting of alternating single and double carbon-carbon bounds, bounded by two nitrogen atoms, one of which is an iminium cation.[8] Due to the conjugated π-electron network, cyanine dyes have strong electronic absorption in the visible portion of the electromagnetic spectrum; the wavelength of maximum absorption (λ_{max}) shifts toward longer wavelengths as the polymethine chain length increases. In the classic experiment, students record absorption spectra of a series of cyanine dyes and use the Kuhn free electron model,[9] based on the one-dimensional particle-in-a-box (PIB), to explore the relationship between chain length and λ_{max}. This lab dates back at least 50 years,[10] and numerous variations or extensions have been reported in this *Journal*.[11–17] Its popularity derives from the use of inexpensive instrumentation and reagents as well as the direct connection to one of the first model Hamiltonians that students see in quantum chemistry. However, opportunities exist to connect this classic wet-lab activity to modern areas of computational chemistry.

Machine learning (ML) approaches are increasingly applied in many areas of chemistry research,[18–20] yet are still not widely incorporated into undergraduate chemistry curricula. These techniques use algorithms that can generalize from an example dataset to make predictions about new data. In ML regression analysis, a model is used to predict the numerical value of a dependent variable using one or more independent variables. Even simple regularized linear regression models suffice to predict chemical properties as diverse as molecular atomization energies,[21] molecular orbital energies,[22] and interatomic potentials,[23] or to analyze photocurrent spectroscopy experiments.[24] In fact, simple linear models built with an appropriate combination of input features are often better at extrapolating to novel examples.[25] Recent articles in this *Journal* have described activities to introduce chemistry students to ML techniques, including the use of ML classifier models to distinguish functional groups in IR spectra,[26] modeling the response of metal nanoparticle colorimetric sensors using neural networks,[27] chemometric analysis of wines,[28] and unsupervised clustering of FTIR and mass-spectrometry data for whisky, tea, and fruit.[29] In addition to teaching practical skills, these activities also implicitly teach students to be aware of limitations and possible failures of ML, including issues with data quantity and quality (*e.g.*, dataset imbalances, domain shifts) and effects on prediction quality. However, it is equally important to teach students how to use ML models not merely to make predictions but also as tools for determining underlying scientific hypotheses. Several recent reviews discuss trends in interpretable and explainable ML methods in chemistry.[30–32]

Here we report a computational activity, designed to be completed in a single 3-hour lab period, to complement the classic cyanine dye/PIB experiment for undergraduate physical chemistry students. In this activity, students apply ML regression analysis to predict λ_{max} for different cyanine dyes using a set of 13 molecular features. The analysis starts with simple linear regression, which chemistry students are likely to have seen previously, before introducing more advanced types of regression models, including multiple regression, regularized regression, and tree-based regression models. In addition to predicting λ_{max}, students analyze feature correlation and feature importance to gain insight into the model results. The final goal of the activity is for students to integrate the regression results with their prior chemistry knowledge to generate a hypothesis for what factors govern cyanine dye absorption and to identify the PIB as an appropriate model system. While our specific example of cyanine dye spectroscopy is not of significant practical importance, the types of model interpretation and feature selection methods taught in this

lab have been applied to a wide range of applied research problems, guiding the exploration of superhard materials,[33] antimicrobial conjugated oligoelectrolytes,[34] and halide perovskite crystal growth modifying additives.[35] Given its ubiquity in physical chemistry curricula, cyanine dye spectroscopy thus provides an opportunity to introduce these methods in a familiar context.

Purpose of the Activity:

This experiment is intended to promote student understanding of both ML regression analysis and the electronic absorption of cyanine dyes, in addition to increasing student comfort in reading and modifying code in a computational notebook environment. The learning goals are that, by the end of the activity, students will be able to:

- Explain the workflow of ML regression analysis
- Describe different ML regression algorithms
- Apply different methods for evaluating regression model performance and feature importance
- Generate hypotheses for factors that govern cyanine dye absorption
- Read and modify code in a Mathematica or Python computational notebook

Methods:

Dataset:

Successful ML analysis requires a diverse dataset of cyanine dye spectra, which can be generated computationally. However, accurate calculation of the excited electronic states of cyanine dyes is challenging,[36] necessitating attention to the computational method, solvent interactions, and conformational degrees of freedom, and is typically performed using computationally-intensive time-dependent density functional theory (TD-DFT) calculations.[37] ML methods for excited states provide an alternative approach,[38] and deep-learning based methods can take into account conformational and solvent effects and reproduce high-quality TD-DFT calculations with trivial computational cost.[39,40] We have used these new methods to generate the

dataset used in this work. These tools can also be useful for possible student explorations, as the ML based calculations can be performed interactively.[41]

A dataset was generated by searching PubChem for entries whose name contains "cyanine"; 112 unique molecules remained after removing duplicates (including molecules that differ only by the counterion of the salt). We augmented this dataset by examining dyes with end groups separated by polyene linkers, mix-and-matching the end groups, and generating hypothetical molecules with diene, butadiene, and hexatriene linkers, resulting in a total of 147 unique molecules. The absorption maxima were computed with the UVVisML package,[39,42] using the default settings and with the solvent set as methanol to match typical experimental procedures. This package does not generate a full spectrum for the molecule, but only predicts λ_{max}, which is sufficient for our analysis. Thirteen different cheminformatics descriptors for each molecule were computed using Mathematica 13.2. The dataset deliberately includes several entries with large numbers of aromatic rings or anomalous sizes to serve the learning goal related to outlier identification. The complete dataset is available in CSV format at the GitHub repository for the project.[43]

Machine Learning:

Regression analysis seeks to predict a target value, the dependent variable or y, in terms of one or more features, the independent variable(s) or X. The most basic type of regression analysis is simple linear regression, which uses a single independent variable to predict the dependent variable. In this approach, the line of best fit is determined, usually by minimizing a cost function, such as the residual sum of squares (RSS), which measures the difference between the observed and predicted values. This approach can be generalized to a multiple linear regression using two or more independent variables to predict a single dependent variable. Although generally more accurate, because the model has more information available to make predictions, multiple regression analysis is prone to overfitting, in which the model becomes too specialized to the training data and fails to generalize well to new data. Regularized (or penalized) regression models include a penalty term to minimize the complexity of the resulting model and avoid overfitting. This activity explores two common regularized regression models, Ridge Regression and Lasso Regression. In addition, there are alternatives to linear regression models, including tree-based

regression models like Decision Tree and Random Forest, in which a tree structure is used to model the data.

Supervised ML algorithms, including regression models, require a training dataset for which the desired target values are already known; this dataset is used to adjust the model parameters to give the best agreement with the target values. It is important that this training dataset is representative of the data to which the resulting model will be applied, or else the model may generalize poorly to new data; for example, the training dataset should sample the full range of feature and target values. After model training, a separate dataset with known target values, the test dataset, is used to evaluate the ML model performance. A qualitative assessment of model performance can be obtained by plotting the actual values against the predicted values for the test dataset. In addition, the actual and predicted values for the test dataset can be used to generate quantitative performance metrics. This activity introduces four such metrics: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and coefficient of determination (R^2). MAE, MSE, and RMSE values closer to 0 indicate better model performance; for R^2, values closer to 1 indicate better model performance.

In addition to evaluating the performance of the regression models, this activity prompts students to analyze the correlation and importance of the different molecular features. In Pearson correlation coefficient analysis, the relationship between each of the 13 features and the target variable is determined. Values close to +1 or -1 indicate strong positive or negative correlation, whereas values close to 0 indicate no correlation. In addition to assessing the correlation of each feature with the target variable individually, the contribution of each feature to the overall multiple regression model is evaluated based on the importance of that feature in the resulting model. For linear models trained with normalized data, feature importance is assessed by examining the magnitude of the weight of each feature in the resulting model; larger weights indicate greater importance. For non-linear models, a model-agnostic approach called SHapley Additive exPlanations (SHAP)[44] is used to determine feature importance.

Code Implementation:

The Python implementation of this activity can be executed in Google Colaboratory or any Jupyter notebook environment; it was tested with Python version 3.10. It uses standard Python libraries available in the Google Colaboratory environment for working with datasets (pandas[45]),

performing numerical calculations (NumPy[46] and SciPy[47]), visualizing data (Matplotlib[48] and Seaborn[49]), performing machine learning analyses (scikit-learn[50]), and molecular visualization RDKit[51]). The Mathematica implementation used version 13.0 (but has also been tested on 13.2);[52] no additional libraries are required. Representative results shown here are taken from the Mathematica implementation, but similar results are obtained in Python. The latest versions of all notebooks are available on GitHub.[43]

Results:

Cyanine Dye Absorption Spectra:

The electronic absorption spectra of cyanine dyes feature strong maxima in the UV-visible wavelength range. For a series of dyes that share the same end group, λ_{max} shifts to longer wavelengths as the length of the polymethine chain increases. Figure 1A shows the structures of a series of common cyanine dyes, which differ only in chain length: 1,1′-diethyl-2,2′-cyanine, 1,1′-diethyl-2,2′-carbocyanine, and 1,1′-diethyl-2,2′-dicarbocyanine. The corresponding UV-visible absorption spectra (Figure 1B) reveal the expected shift in absorption, with λ_{max} increasing from approximately 525 nm to 605 nm to 707 nm as the chain length increases. In addition to the primary absorption peak, the spectra contain secondary peaks or shoulders at shorter wavelength due to vibronic transitions,[11] which are neglected in the simple Kuhn free electron model analysis.

Cyanine Dye Computational Dataset:

To validate our UVVisML dataset, we compared experimental λ_{max} values to those calculated with UVVisML for 11 different cyanine dyes and found good agreement (Figure 1C; R^2 = 0.88), justifying this computational approach for the full dataset of 147 molecules. Six randomly selected molecules are shown in Figure 2 to illustrate the diversity of end groups, conjugated chain lengths, and presence of other functional groups (such as azides or alkynes).

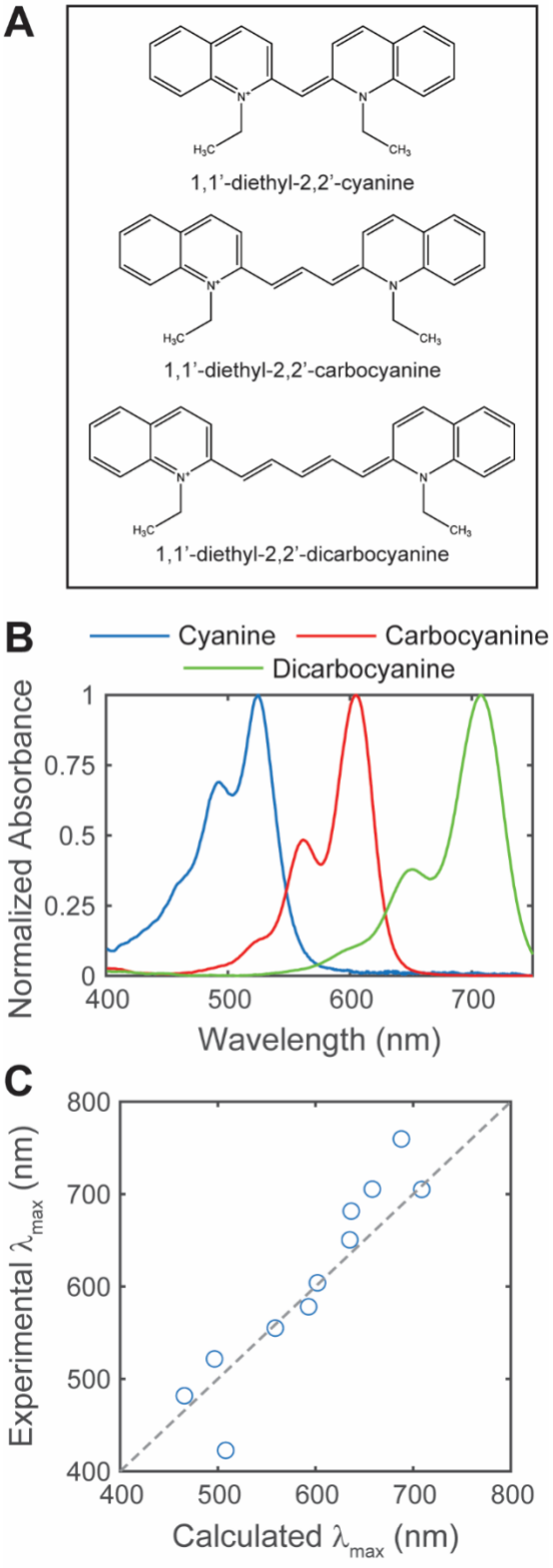


Figure 1. (A) Structures of three cyanine dyes with the same end groups but different conjugated chain lengths. (B) Experimental UV-vis absorption spectra for the three dyes in (A). (C)

Comparison of experimental and calculated λ_{max} values for 11 different cyanine dyes. Dashed gray line: $x = y$.

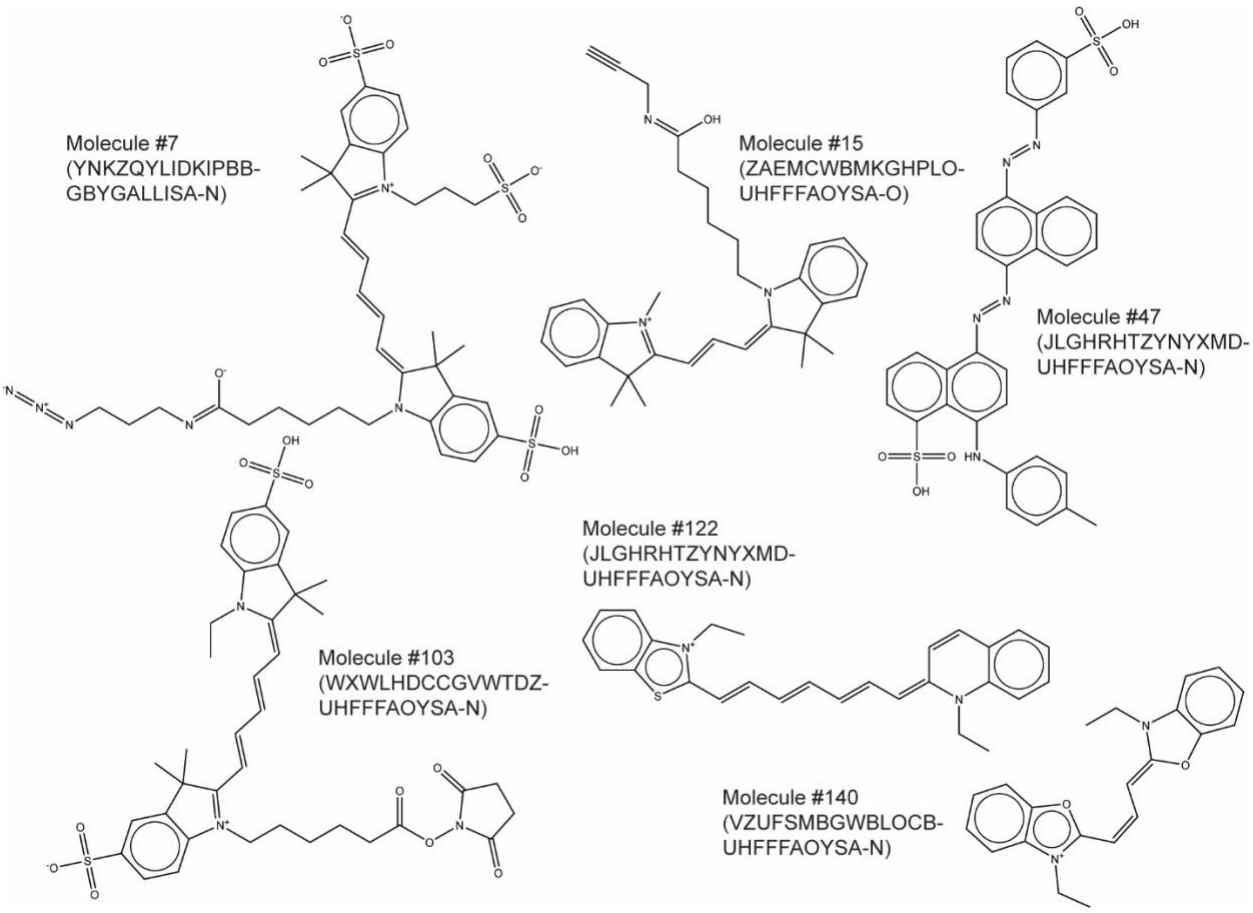


Figure 2. Six randomly selected molecules from the computational cyanine dataset with corresponding InChIKey identifiers in parentheses.

Regression Models:

Students first train simple linear regression models using a single feature to predict λ_{max}. By comparing the actual and predicted λ_{max} values for different features in a parity plot, students can begin to develop intuition for which molecular characteristics have better predictive power. For example, molecular mass (Figure 3A) does not predict λ_{max} as effectively as conjugated linker length (Figure 3B). In addition to a graphical assessment of model performance, students compute various quantitative performance metrics for the different models. A comparison of these metrics for the molecular mass and linker length models is consistent with the graphical assessment; for

example, R^2 = -0.02 for molecular mass vs. R^2 = 0.23 for linker length. Performance metrics for simple linear regression models with the 13 different features are shown in Table S1.

After testing different simple linear regression models, students then train multiple regression models that incorporate all 13 molecular features as independent variables. An unpenalized multiple linear regression model gives better performance than any simple linear regression model for this dataset (Figure 3C; R^2 = 0.48). In addition, students test two different regularized regression models, Ridge and Lasso, with different regularization penalties. These models provide small improvements on the unpenalized multiple linear regression model as the regularization penalty is increased (corresponding to larger values of the penalty parameter α). Introduction of regularized models helps illustrate the concept of model overfitting by showing that a simpler model can sometimes generalize better to the test data. Performance metrics for unpenalized and regularized multiple linear regression models are shown in Table S2. As an optional extension, students test two tree-based regression models, Decision Tree and Random Forest. For this dataset, Random Forest gives the best performance (R^2 = 0.66). Performance metrics for tree-based models are shown in Table S3.

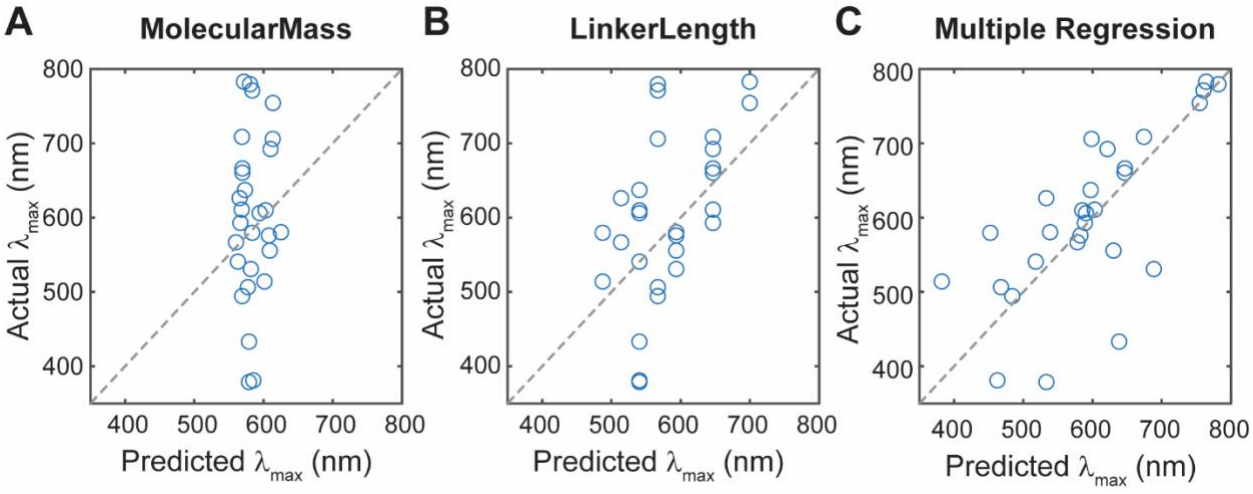


Figure 3. Actual vs. predicted λ_{max} values for (A) simple linear regression using the MolecularMass feature, (B) simple linear regression using the LinkerLength feature, and (C) multiple linear regression. Dashed gray line: $x = y$.

Feature Correlation and Feature Importance Analysis:

To study what molecular features govern cyanine dye absorption, students analyze feature correlation and feature importance. Pearson correlation coefficient analysis reveals the correlation between each individual feature and λ_{max}. As expected, several related features, including the linker length and the longest conjugated π chain, are among the most highly correlated with λ_{max} (Figure 4A). Students should identify the extent of the conjugated π-electron network as the physically relevant feature that governs cyanine dye absorption. In a complementary approach, students analyze feature importance in the multiple linear regression model by looking at the different coefficient weights (Figure 4B). This analysis reveals which features contribute the most to the model, although it can be complicated by correlations between the features. As a result, the features with the strongest individual correlation do not have the highest contributions to the model. In an optional extension of the activity, students use an alternative method, SHapley Additive exPlanations (SHAP), to analyze feature importance; this advanced approach can also be applied to non-linear models, like Regression Tree and Random Forest. SHAP analysis for the Decision Tree model is shown in Figure S1.

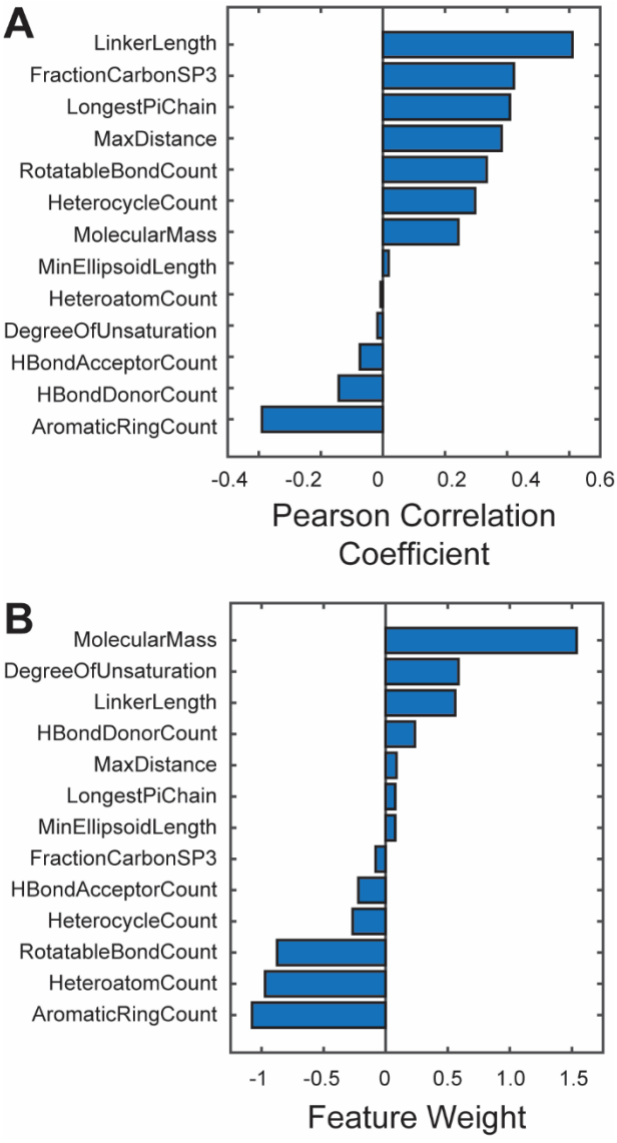


Figure 4. (A) Pearson correlation coefficient and (B) coefficient weight analysis for the 13 molecular features.

Implementation:

This activity was tested in the Fall 2022 and Spring 2023 semesters in physical chemistry lab courses at Fordham University and Whitman College, with class sizes of 9 and 4. It is designed for an upper-level physical chemistry lab course running either concurrent with or following a semester of quantum chemistry. It is implemented in the form of a computational notebook, which combines explanatory text, hyperlinks to other resources, executable and editable code blocks, and

output figures and tables. In addition to scaffolding the analysis for students, computational notebooks can also be easily shared with instructors for troubleshooting or assessment. The notebook is divided into two parts: Part I introduces simple and multiple linear regression, regularized regression, and feature correlation and importance analysis, and can be completed in a single 3-hour lab period. Part II includes more advanced topics, including feature engineering, tree models, and SHAP analysis. Part I forms the core of the activity, with the topics in Part II intended for more advanced students or for classes where a second lab period is available. There are prompts throughout the notebook for students to write new blocks of code to accomplish specific tasks, such as performing the analysis with different models or parameters. To facilitate adoption by instructors, we have created an instructor version of the notebook, which includes the solutions to these coding tasks.

The computational notebooks are provided in both Python and Mathematica. There are several options for executing Python notebooks, including the free web-based Google Colaboratory platform or a Jupyter Notebook application. We recommend the use of Google Colaboratory to ensure that the necessary Python packages are compatible and up to date. For expert users who wish to run the notebooks locally, we have included a pip "requirements.txt" file in the GitHub repository[43] that specifies appropriate package versions. However, we note that other Python compatibility issues may still arise. The Mathematica notebook can be executed in either the desktop or online version of Mathematica; although Mathematica is not free, many institutions have a campus license. The two versions have the same content and structure, except for minor differences due to the use of different programming languages. Thus, instructors may choose which version to use based on student background or on the programming language used in their course or department. Writing the programs to perform basic chemical data analysis tasks like these might soon be handled using interactive large language models (LLMs) such as GPT-3,[53,54] and so the most important goal is to help students to think critically about reading, modifying, and debugging code, independent of language.

In the Fordham University physical chemistry lab course, we conducted anonymous pre- and post-lab surveys to assess student background and student impressions of the activity. Based on the pre-lab survey, almost all respondents (7/8) had heard of ML and regression analysis, but fewer than half (3/8) were aware of specific applications of ML inside or outside of chemistry. A few students (3/8) had taken a previous computer programming class, but the physical chemistry

course was the first exposure to programming in general, and to Mathematica specifically, for most students. Although the post-lab survey response rate was lower, more than half of students Agreed or Strongly Agreed (on a 5-point Likert scale from Strongly Disagree to Strongly Agree) that they understood the basic steps in an ML regression task, some of the factors that affect regression analysis, and possible applications of ML to chemistry. Most students also Agreed or Strongly Agreed that they were more able to read and write Mathematica code after completing the activity.

From the instructor perspective, the activity was broadly successful in meeting its learning goals. Almost all students at both Fordham University and Whitman College completed the core steps in the exercise in the allotted time, with some students moving on to the advanced topics. Likewise, all lab groups generated reasonable hypotheses for the underlying chemical principles that govern cyanine dye absorption and, with some instructor assistance, identified the PIB as an appropriate quantum mechanical model for the system. However, some students struggled with the programming aspect of the activity. At Fordham University, the exercise was performed early in the first semester of the physical chemistry lab course, and despite an introduction and opportunities to practice in the lecture course, students were still developing their proficiency in Mathematica. Instructors may wish to implement this activity with students who have had a longer exposure to programming or else to provide greater scaffolding for the programming components of the activity.

This exercise was implemented as a one-day module to complement a separate experimental measurement of cyanine dye absorption spectra, but it could be adapted to other contexts. For example, in a computational chemistry or cheminformatics course, students could compute their own spectra using TD-DFT or generate novel molecular features to use in the regression analysis, connecting the core regression activity to more advanced computational chemistry topics. The activity could instead be simplified to make it accessible to introductory chemistry students by creating a streamlined version of the computational notebook without the programming tasks; in this case, the goal would be for students to gain experience working with a computational notebook and to understand the basic workflow of the regression analysis, but not to generate their own computer code. This simplified activity could serve as a stepping-stone to more advanced computational chemistry and programming content later in the course or in subsequent courses. At Fordham University, the regression analysis and experimental measurements were part of a three-day sequence, with the third lab period dedicated to electronic

structure calculations of cyanine dye molecular orbitals and absorption spectra. Alternatively, this activity could be performed as a purely computational exercise without an experimental component.

Conclusion:

Here we report a new computational activity designed to introduce students to ML regression analysis in the context of the classic physical chemistry lab experiment on the absorption spectra of cyanine dyes. Student and instructor feedback indicate that this exercise was successful in introducing students to ML regression analysis and in strengthening student programming skills. Supplementing classic experiments with activities that expose students to modern research tools can help instructors balance the competing demands of teaching the physical chemistry canon while also ensuring that curricula stay relevant and engaging.

ASSOCIATED CONTENT

The Supporting Information is available on the ACS Publications website at DOI: 10.1021/acs.jchemed.XXXXXXX.

Representative results of the activity. (PDF)

Student handout containing background information, an example protocol, and discussion questions. (DOCX, PDF)

Notebook implementation of the regression analysis in Python and Mathematica. (IPYNB, NB)

AUTHOR INFORMATION

Corresponding Author

Elizabeth S. Thrall – Department of Chemistry, Fordham University, The Bronx, New York 10458, United States; orcid.org/0000-0002-7670-3939

*Email: ethrall@fordham.edu

Authors

Fernando Martinez Lopez – Department of Computer and Information Sciences, Fordham University, New York, New York 10023, United States; orcid.org/0009-0007-2208-2691

Thomas J. Egg – Department of Chemistry, Fordham University, The Bronx, New York 10458, United States; orcid.org/0009-0003-1651-4751

Seung Eun Lee – Department of Computer and Information Sciences, Fordham University, New York, New York 10023, United States

Joshua Schrier – Department of Chemistry, Fordham University, The Bronx, New York 10458, United States; orcid.org/0000-0002-2071-1657

Yijun Zhao – Department of Computer and Information Sciences, Fordham University, New York, New York 10023, United States; orcid.org/0000-0003-2424-5988

ACKNOWLEDGMENTS

We thank Mark Hendricks for testing this activity in his physical chemistry lab course. We particularly thank the physical chemistry lab students at Fordham University and Whitman College for their participation in and feedback on this experiment. We gratefully acknowledge Clara Victorio for providing the experiment spectra used in Figure 1B and James McCormick and Stuart Winikoff for providing the experimental λ_{max} values used in Figure 1C. Funding for this project was provided by the Fordham University Faculty of Arts and Sciences through the Faculty Interdisciplinarity Grant program, the National Science Foundation (PHYS-22265110), and the Henry Dreyfus Teacher-Scholar Award (TH-14-010). We also thank the MERCURY Consortium for providing computational resources, supported by National Science Foundation (CNS-2108427).

REFERENCES

(1) *Using Computational Methods To Teach Chemical Principles*; Grushow, A., Reeves, M. S., Eds.; American Chemical Society, Series Ed.; ACS Symposium Series; American Chemical Society: Washington, DC, 2019; Vol. 1312. https://doi.org/10.1021/bk-2019-1312.

(2) *Teaching Programming across the Chemistry Curriculum*; Ringer McDonald, A., Nash, J. A., Eds.; American Chemical Society, Series Ed.; ACS Symposium Series; American Chemical Society: Washington, DC, 2021; Vol. 1387. https://doi.org/10.1021/bk-2021-1387.

(3) DeVore, T. C. Introducing Quantum Calculations into the Physical Chemistry Laboratory. In *ACS Symposium Series*; Grushow, A., Reeves, M. S., Eds.; American Chemical Society: Washington, DC, 2019; Vol. 1312, pp 109–125. https://doi.org/10.1021/bk-2019-1312.ch009.

(4) Whitnell, R. M.; Reeves, M. S. Process Oriented Guided Inquiry Learning Computational Chemistry Experiments: Revisions and Extensions Based on Lessons Learned from

Implementation. In *ACS Symposium Series*; Grushow, A., Reeves, M. S., Eds.; American Chemical Society: Washington, DC, 2019; Vol. 1312, pp 65–77. https://doi.org/10.1021/bk-2019-1312.ch006.
(5) Martin, W. R.; Ball, D. W. Using Computational Chemistry to Extend the Acetylene Rovibrational Spectrum to C 2 T 2. In *ACS Symposium Series*; Grushow, A., Reeves, M. S., Eds.; American Chemical Society: Washington, DC, 2019; Vol. 1312, pp 93–107. https://doi.org/10.1021/bk-2019-1312.ch008.
(6) Hu, D.; Ahn, J. N.; Lakatos, A.; Bello, J.; McTague, J.; Foley, J. J. Integrating Programming to Reinforce Quantum Mechanical Principles in Physical Chemistry. In *ACS Symposium Series*; Ringer McDonald, A., Nash, J. A., Eds.; American Chemical Society: Washington, DC, 2021; Vol. 1387, pp 89–105. https://doi.org/10.1021/bk-2021-1387.ch007.
(7) Magers, D. B.; Chávez, V. H.; Peyton, B. G.; Sirianni, D. A.; Fortenberry, R. C.; Ringer McDonald, A. PSI4EDUCATION: Free and Open-Source Programing Activities for Chemical Education with Free and Open-Source Software. In *ACS Symposium Series*; Ringer McDonald, A., Nash, J. A., Eds.; American Chemical Society: Washington, DC, 2021; Vol. 1387, pp 107–122. https://doi.org/10.1021/bk-2021-1387.ch008.
(8) Shindy, H. A. Fundamentals in the Chemistry of Cyanine Dyes: A Review. *Dyes and Pigments* **2017**, *145*, 505–513. https://doi.org/10.1016/j.dyepig.2017.06.029.
(9) Kuhn, H. A Quantum-Mechanical Theory of Light Absorption of Organic Dyes and Similar Compounds. *The Journal of Chemical Physics* **1949**, *17* (12), 1198–1212. https://doi.org/10.1063/1.1747143.
(10) Gerkin, R. E. A Molecular Spectral Corroboration of Elementary Operator Quantum Mechanics. *J. Chem. Educ.* **1965**, *42* (9), 490. https://doi.org/10.1021/ed042p490.
(11) Burkhart, R. D.; Howells, P. N. Collecting and Manipulating Digital Data in an Experiment on Electronic Spectroscopy. *J. Chem. Educ.* **1979**, *56* (4), 249. https://doi.org/10.1021/ed056p249.
(12) Farrell, J. J. The Absorption Spectra of a Series of Conjugated Dyes: Determination of the Spectroscopic Resonance Integral. *J. Chem. Educ.* **1985**, *62* (4), 351. https://doi.org/10.1021/ed062p351.
(13) Moog, R. S. Determination of Carbon-Carbon Bond Length from the Absorption Spectra of Cyanine Dyes. *J. Chem. Educ.* **1991**, *68* (6), 506. https://doi.org/10.1021/ed068p506.
(14) Bahnick, D. A. Use of Huckel Molecular Orbital Theory in Interpreting the Visible Spectra of Polymethine Dyes: An Undergraduate Physical Chemistry Experiment. *J. Chem. Educ.* **1994**, *71* (2), 171. https://doi.org/10.1021/ed071p171.
(15) Shalhoub, G. M. Visible Spectra Conjugated Dyes: Integrating Quantum Chemical Concepts with Experimental Data. *J. Chem. Educ.* **1997**, *74* (11), 1317. https://doi.org/10.1021/ed074p1317.
(16) Autschbach, J. Why the Particle-in-a-Box Model Works Well for Cyanine Dyes but Not for Conjugated Polyenes. *J. Chem. Educ.* **2007**, *84* (11), 1840. https://doi.org/10.1021/ed084p1840.
(17) Mansell, A. A Small Program to Extend the Conjugated Dyes Particle in a Box Experiment. *J. Chem. Educ.* **2023**, acs.jchemed.3c00171. https://doi.org/10.1021/acs.jchemed.3c00171.
(18) Janet, J. P.; Kulik, H. J. *Machine Learning in Chemistry*; ACS In Focus; American Chemical Society: Washington, DC, USA, 2020. https://doi.org/10.1021/acs.infocus.7e4001.

(19) Baum, Z. J.; Yu, X.; Ayala, P. Y.; Zhao, Y.; Watkins, S. P.; Zhou, Q. Artificial Intelligence in Chemistry: Current Trends and Future Directions. *J. Chem. Inf. Model.* **2021**, *61* (7), 3197–3212. https://doi.org/10.1021/acs.jcim.1c00619.
(20) Yano, J.; Gaffney, K. J.; Gregoire, J.; Hung, L.; Ourmazd, A.; Schrier, J.; Sethian, J. A.; Toma, F. M. The Case for Data Science in Experimental Chemistry: Examples and Recommendations. *Nat Rev Chem* **2022**, *6* (5), 357–370. https://doi.org/10.1038/s41570-022-00382-w.
(21) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108* (5), 058301. https://doi.org/10.1103/PhysRevLett.108.058301.
(22) Stuke, A.; Todorović, M.; Rupp, M.; Kunkel, C.; Ghosh, K.; Himanen, L.; Rinke, P. Chemical Diversity in Molecular Orbital Energy Predictions with Kernel Ridge Regression. *J. Chem. Phys.* **2019**, *150* (20), 204121. https://doi.org/10.1063/1.5086105.
(23) Xie, S. R.; Rupp, M.; Hennig, R. G. Ultra-Fast Interpretable Machine-Learning Potentials. arXiv October 1, 2021. http://arxiv.org/abs/2110.00624 (accessed 2023-05-19).
(24) Piccioni, A.; Vecchi, P.; Vecchi, L.; Grandi, S.; Caramori, S.; Mazzaro, R.; Pasquini, L. Distribution of Relaxation Times Based on Lasso Regression: A Tool for High-Resolution Analysis of IMPS Data in Photoelectrochemical Systems. *J. Phys. Chem. C* **2023**, *127* (17), 7957–7964. https://doi.org/10.1021/acs.jpcc.3c00770.
(25) Muckley, E. S.; Saal, J. E.; Meredig, B.; Roper, C. S.; Martin, J. H. Interpretable Models for Extrapolation in Scientific Machine Learning. *Digital Discovery* **2023**, 10.1039.D3DD00082F. https://doi.org/10.1039/D3DD00082F.
(26) Thrall, E. S.; Lee, S. E.; Schrier, J.; Zhao, Y. Machine Learning for Functional Group Identification in Vibrational Spectroscopy: A Pedagogical Lab for Undergraduate Chemistry Students. *J. Chem. Educ.* **2021**, *98* (10), 3269–3276. https://doi.org/10.1021/acs.jchemed.1c00693.
(27) Revignas, D.; Amendola, V. Artificial Neural Networks Applied to Colorimetric Nanosensors: An Undergraduate Experience Tailorable from Gold Nanoparticles Synthesis to Optical Spectroscopy and Machine Learning. *J. Chem. Educ.* **2022**, *99* (5), 2112–2120. https://doi.org/10.1021/acs.jchemed.1c01288.
(28) Lafuente, D.; Cohen, B.; Fiorini, G.; García, A. A.; Bringas, M.; Morzan, E.; Onna, D. A Gentle Introduction to Machine Learning for Chemists: An Undergraduate Workshop Using Python Notebooks for Visualization, Data Processing, Analysis, and Modeling. *J. Chem. Educ.* **2021**, *98* (9), 2892–2898. https://doi.org/10.1021/acs.jchemed.1c00142.
(29) St James, A. G.; Hand, L.; Mills, T.; Song, L.; Brunt, A. S. J.; Bergstrom Mann, P. E.; Worrall, A. F.; Stewart, M. I.; Vallance, C. Exploring Machine Learning in Chemistry through the Classification of Spectra: An Undergraduate Project. *J. Chem. Educ.* **2023**, *100* (3), 1343–1350. https://doi.org/10.1021/acs.jchemed.2c00682.
(30) Dybowski, R. Interpretable Machine Learning as a Tool for Scientific Discovery in Chemistry. *New J. Chem.* **2020**, *44* (48), 20914–20920. https://doi.org/10.1039/D0NJ02592E.
(31) Ghiringhelli, L. M. Interpretability of Machine-Learning Models in Physical Sciences. **2021**. https://doi.org/10.48550/ARXIV.2104.10443.
(32) Oviedo, F.; Ferres, J. L.; Buonassisi, T.; Butler, K. T. Interpretable and Explainable Machine Learning for Materials Science and Chemistry. *Acc. Mater. Res.* **2022**, *3* (6), 597–607. https://doi.org/10.1021/accountsmr.1c00244.

(33) Mansouri Tehrani, A.; Oliynyk, A. O.; Parry, M.; Rizvi, Z.; Couper, S.; Lin, F.; Miyagi, L.; Sparks, T. D.; Brgoch, J. Machine Learning Directed Search for Ultraincompressible, Superhard Materials. *J. Am. Chem. Soc.* **2018**, *140* (31), 9844–9853. https://doi.org/10.1021/jacs.8b02717.
(34) Tiihonen, A.; Cox-Vazquez, S. J.; Liang, Q.; Ragab, M.; Ren, Z.; Hartono, N. T. P.; Liu, Z.; Sun, S.; Zhou, C.; Incandela, N. C.; Limwongyut, J.; Moreland, A. S.; Jayavelu, S.; Bazan, G. C.; Buonassisi, T. Predicting Antimicrobial Activity of Conjugated Oligoelectrolyte Molecules via Machine Learning. *J. Am. Chem. Soc.* **2021**, *143* (45), 18917–18931. https://doi.org/10.1021/jacs.1c05055.
(35) Hartono, N. T. P.; Ani Najeeb, M.; Li, Z.; Nega, P. W.; Fleming, C. A.; Sun, X.; Chan, E. M.; Abate, A.; Norquist, A. J.; Schrier, J.; Buonassisi, T. Principled Exploration of Bipyridine and Terpyridine Additives to Promote Methylammonium Lead Iodide Perovskite Crystallization. *Crystal Growth & Design* **2022**, *22* (9), 5424–5431. https://doi.org/10.1021/acs.cgd.2c00522.
(36) Le Guennic, B.; Jacquemin, D. Taking Up the Cyanine Challenge with Quantum Tools. *Acc. Chem. Res.* **2015**, *48* (3), 530–537. https://doi.org/10.1021/ar500447q.
(37) Ilieva, S.; Kandinska, M.; Vasilev, A.; Cheshmedzhieva, D. Theoretical Modeling of Absorption and Fluorescent Characteristics of Cyanine Dyes. *Photochem* **2022**, *2* (1), 202–216. https://doi.org/10.3390/photochem2010015.
(38) Westermayr, J.; Marquetand, P. Machine Learning for Electronically Excited States of Molecules. *Chem. Rev.* **2021**, *121* (16), 9873–9926. https://doi.org/10.1021/acs.chemrev.0c00749.
(39) Greenman, K. P.; Green, W. H.; Gómez-Bombarelli, R. Multi-Fidelity Prediction of Molecular Optical Peaks with Deep Learning. *Chem. Sci.* **2022**, *13* (4), 1152–1162. https://doi.org/10.1039/D1SC05677H.
(40) Joung, J. F.; Han, M.; Hwang, J.; Jeong, M.; Choi, D. H.; Park, S. Deep Learning Optical Spectroscopy Based on Experimental Database: Potential Applications to Molecular Design. *JACS Au* **2021**, *1* (4), 427–438. https://doi.org/10.1021/jacsau.1c00035.
(41) Deep4Chem. http://deep4chem.korea.ac.kr/predict.
(42) Greenman, K. P.; Green, W. H.; Gomez-Bombarelli, R. UVVisML, 2022. https://doi.org/10.5281/ZENODO.5986671.
(43) MLforPChem/MLcyaninedye. https://github.com/elizabeththrall/MLforPChem/tree/main/MLcyaninedye.
(44) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; Vol. 30.
(45) McKinney, W. Pandas: A Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing* **2011**, *14* (9), 1–9.
(46) Oliphant, T. E. *A Guide to NumPy*; Trelgol Publishing: USA, 2006; Vol. 1.
(47) SciPy 1.0 Contributors; Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P. SciPy 1.0: Fundamental Algorithms for

Scientific Computing in Python. *Nat Methods* **2020**, *17* (3), 261–272. https://doi.org/10.1038/s41592-019-0686-2.
(48) Barrett, P.; Hunter, J. D.; Miller, T.; Hsu, J. Matplotlib -- A Portable Python Plotting Package. In *Astronomical data analysis software and systems XIV*; 2005; Vol. 347, p 91.
(49) Waskom, M. Seaborn: Statistical Data Visualization. *JOSS* **2021**, *6* (60), 3021. https://doi.org/10.21105/joss.03021.
(50) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12* (85), 2825–2830.
(51) RDKit: Open-source cheminformatics. https://www.rdkit.org. .
(52) Wolfram Research. Mathematica, Version 13.0. https://www.wolfram.com/mathematica. .
(53) White, A. D.; Hocky, G. M.; Gandhi, H. A.; Ansari, M.; Cox, S.; Wellawatte, G. P.; Sasmal, S.; Yang, Z.; Liu, K.; Singh, Y.; Peña Ccoa, W. J. Assessment of Chemistry Knowledge in Large Language Models That Generate Code. *Digital Discovery* **2023**, *2* (2), 368–376. https://doi.org/10.1039/D2DD00087C.
(54) Tu, X.; Zou, J.; Su, W. J.; Zhang, L. What Should Data Science Education Do with Large Language Models? arXiv July 7, 2023. http://arxiv.org/abs/2307.02792 (accessed 2023-07-21).