TheBiophysicist

A Semester-Long Learning Path Teaching Computational Skills via Molecular Graphics in PyMOL

Magnus Kjaergaard[1,2*], Laura Skak Rasmussen[1], Johan Nygaard Vinther[1,2], Kasper Røjkjær Andersen[1], Ebbe Sloth Andersen[1,2], Esben Lorentzen[1], Søren S. Thirup[1], Daniel E. Otzen[1,2], and Ditlev Egeskov Brodersen[1*]

[1]Department of Molecular Biology and Genetics, Aarhus University, Aarhus, Denmark
[2]Interdisciplinary Nanoscience Center (iNANO), Aarhus University, Aarhus, Denmark

"*" corresponding authors

Received: 23 May 2022
Accepted: 14 October 2022
Published: 27 December 2022

ABSTRACT Structural biology describes biological processes at the molecular level and is an integral part of undergraduate study programs in molecular biosciences. Students are often fascinated by the visualizations created by molecular graphics software, which allow them to see the molecular world for the first time. Today, molecular visualization and structural analysis do not require expensive high-end computers but can be performed on the students' own laptops and are therefore highly suited for active learning approaches. We have designed a semester-long learning path that integrates molecular graphics and structural analysis using PyMOL into an undergraduate course in biomolecular structure and function. Compared to stand-alone PyMOL introductions, the semester-long learning path allows for an improved pedagogical design. The path progressively introduces more advanced functions in relevant scientific contexts and allows for spaced repetition. Advanced analysis functions in PyMOL are available only via the command line, so the learning path also teaches basic scripting and serves as an accessible introduction to computational thinking because a few lines of code can produce stunning results. Student surveys carried out at the end of the course suggest that the learning path supported the ability to perform structural analysis to a high degree. Moreover, a simulated exam showed that almost all students were able to carry out basic visualization tasks using PyMOL scripts, while three-quarters could undertake advanced structural analysis after following the course. In summary, integration of molecular graphics software with teaching of structural biochemistry allows a hands-on approach to analyzing molecular mechanisms and introduces biologically oriented students to computational thinking.

I. INTRODUCTION

The nearly 200,000 structural models in the Protein Data Bank (PDB) constitute a treasure trove for biochemistry. This resource will only get richer as the pace of structural biology accelerates due to advances in cryo-electron microscopy (1) and computational structure prediction (2, 3), which have provided structural models for whole proteomes (4). This implies that a structural understanding of biomolecules is no longer a specialized topic but rather an integral part of biochemistry. Consequently, all scientists working in the molecular biosciences will need to know how to analyze and evaluate structural models at some level. To accomplish this, an undergraduate degree program in the molecular biosciences should not only provide students with a theoretical understanding of the structural basis of biochemistry but also equip them with the tools necessary for structural analysis.

A. Scientific and pedagogical background

Structural analysis of biomolecules relies on molecular graphics software. Various programs, such as RasMOL (5), Swiss PDB Viewer (6), Jmol (7), and UCSF Chimera (8), have historically allowed users to visualize atomic coordinates in various representations and create stunning illustrations (9). These programs allow the user to navigate structures and focus on features of functional relevance. Advanced programs have built-in analysis tools used for interpretation of structural models, including measurement of interatomic distances, identification of hydrogen bonds and salt bridges, mapping of contacts between molecules, structural alignment of different models, and mapping of physical properties, such as surface charges, onto the structure. In the popular program PyMOL (10), most common functions are available via the graphical user interface (GUI). Advanced functions, however, require text commands that can be combined into elaborate analysis scripts using procedural programming (11, 12). The use of molecular graphics software is a key skill set that allows structural interpretation of molecular mechanisms and formulation of new hypotheses.

The constructivist theory of learning posits that knowledge cannot be passively transmitted from educator to learner but must be constructed anew in the mind of the learner. This has led to a range of "active learning" approaches, where the learners are guided through activities that allow them to rediscover the material for themselves (13). A large body of evidence shows that students learn better when they are active rather than passive (14), although students tend to underestimate the benefits due to the greater mental effort required for active learning (15). However, experimental structural biology is time consuming and requires expensive equipment and is thus difficult to implement in undergraduate classes. In contrast, molecular graphics software is run on standard laptops. Moreover, PyMOL is available at no cost for educators and students as an "Educational-use-only" version (https://pymol.org/edu). Therefore, structural analysis and visualization can be taught actively even in undergraduate classes with hundreds of students, using the students' own hardware. Moreover, exercises can be designed to simultaneously build structural understanding and acquire useful research skills. Such skills can readily be put to use in individual research projects by advanced undergraduate students. For example, PyMOL was recently used in an "Undergraduate Research Laboratory Experience" where students assigned functions to targets from the Protein Structure Initiative (16).

There are several published examples of using PyMOL to analyze protein structures in a university study setting. Lineback and Jansma (17) used PyMOL to investigate the structures and homology between myoglobin and hemoglobin. Rigsby and Parker (18) also used hemoglobin as an example to explore ligand binding, while Allred *et al*. (19) used PyMOL to investigate the pH dependence of the interaction between immunoglobulin G and protein A. Finally, Simmons *et al*. (20) used PyMOL to investigate the impact of disease-associated mutations in 4 proteins. All these published teaching materials are intended to be used as single sessions, which resembles the way we originally taught the use of molecular graphics software. However, we observed several problems with this approach that we attribute to the single-session format. First, modern molecular graphics software has so many functions that students are easily overwhelmed by information if everything must be given in a single session. This left us with the unenviable choice of either omitting essential elements or moving at a pace that many students cannot follow. Second, introducing structural analysis in a single session typically also means limiting it to a single scientific context. It is rarely possible to use all the needed tools in a scientifically relevant way in a single session. Third, consolidation in long-term memory is greatly enhanced by spaced repetitions, where

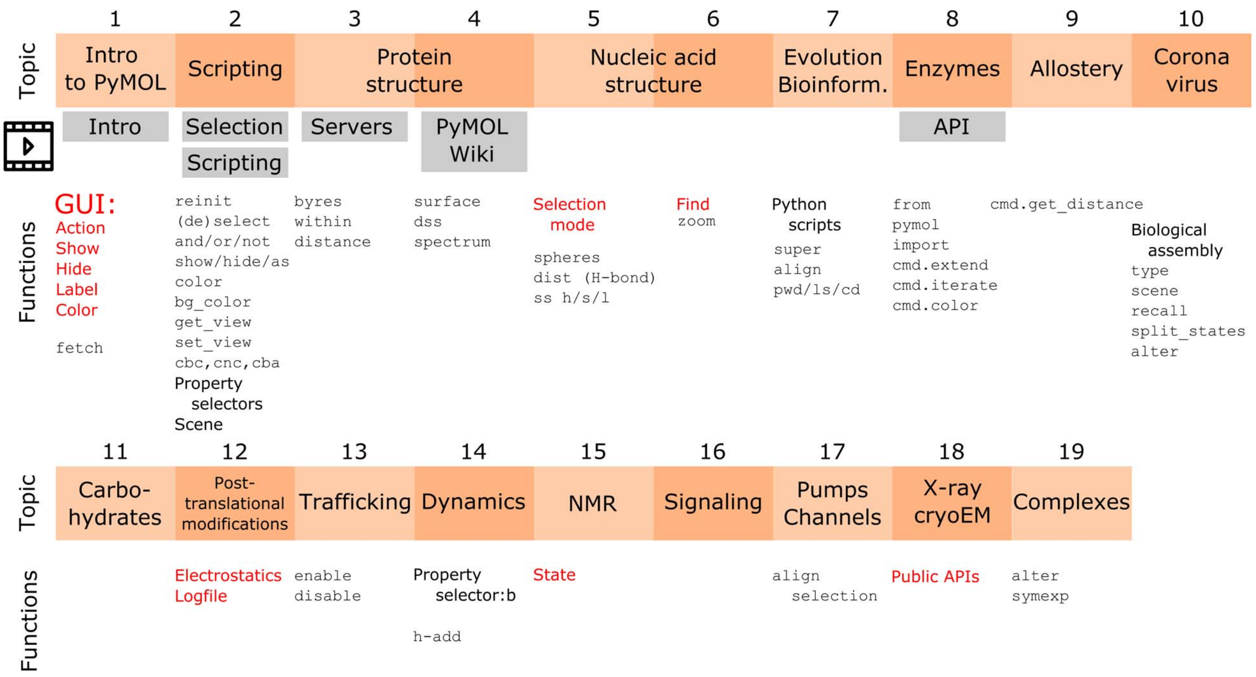

Fig 1. Overview of the PyMOL learning path. The learning path consists of 19 problem sets and 6 instructional videos organized to span a semester course in structural biochemistry. The course aims to progressively build PyMOL programming skills where functions are introduced in relevant scientific context. Red text indicates functions taught using the GUI, whereas black indicates functions taught using the command line. Courier font indicates commands used. The video icon indicates instructional videos paired to exercises.

material is actively recalled at a later date (21). This is not feasible in stand-alone sessions.

To improve the pedagogical design, we adapted our existing teaching of practical PyMOL skills to second-year biochemistry students from a single session to a semester-long learning path (Fig 1). The learning path progresses from basic to advanced competences in structural analysis and introduces many tools currently used in research. In our experience, many students resist moving beyond the GUI, which prevents them from using advanced tools as well as automating repeated tasks using procedural scripts. For these reasons, we aimed to familiarize the students with the command line early on. We think PyMOL offers an excellent opportunity for teaching scripting to biology-focused science students; these are broadly applicable skills in the molecular biosciences and bioinformatics. Furthermore, we aim to support the development of a "computational thinking" mind-set. Computational thinking involves analyzing a problem and expressing its solution in machine-executable form, typically in the form of an algorithm that breaks the process down into simple sequential steps. Computational thinking is thus a skill set that transcends any specific scripting language. While the importance of scripting may be apparent to the instructors, it is not necessarily apparent for an undergraduate specializing in biology. A key pedagogical challenge is thus to design exercises where the students quickly become able to solve problems that are meaningful to them. A PyMOL script comprising a few lines of code can deliver stunning visualizations and allow the students to experience early successes. We thus think PyMOL offers a great opportunity for introducing the students to a computational mind-set that is transferable to other scientific contexts.

B. Teaching materials

The complete learning path consists of problem sets within 19 topics, each containing between 1 and 6 exercises, resulting in 53 exercises in total. Additionally, there are 6 instructional videos and a test used to evaluate the students' PyMOL acquired competences.

Figure 1 provides an overview of the teaching materials, including when new tools are introduced. The materials were originally in Danish but have been made available in English as a course pack via Figshare (https://figshare.com/s/0a55edec9b8e441b2813) or as supplementary information. The videos support different learning styles and illustrate concepts that are difficult to describe in writing, such as the use of the GUI. A set of advanced videos illustrates how to use scripts and implement custom-designed functions using the application programming interface and are deliberately kept below 10 minutes.

The 53 exercises that require PyMOL make up the core of the learning path. They cover a wide range of relevant biological topics and mechanisms and will fit into most introductory courses in biomolecular structure and function. The course starts by introducing the basic structure of proteins and nucleic acids, then progresses through important classes of biomolecules, such as enzymes, pumps, and channels. Additionally, the course deals with important regulatory concepts, such as post-translational modifications, trafficking, and structural dynamics, as well as essential structural techniques, including X-ray crystallography, cryogenic electron microscopy, nuclear magnetic resonance spectroscopy, and fluorescence. The students are also assigned reading materials describing the theoretical background of the topic, typically a chapter in a textbook, and solve other problems that do not require PyMOL. If this is material that is already covered, our materials could be used as a stand-alone PyMOL learning path. The materials are organized to follow the progression of scientific topics (Fig 1), and each topic is thus accompanied by between 1 and 6 exercises containing up to 13 questions. Questions get progressively more difficult within an exercise, which allows students at different academic levels to contribute.

PyMOL functions are introduced in a progressive manner such that the program is initially controlled through the GUI, after which the students are gradually encouraged to switch to the command line and subsequently scripts. Initially, the course focuses on visualization, followed by analysis, and finally touches on manipulation of structures during model building. Beyond the first sessions, only a few new tools are introduced in each session as relevant to the scientific context. For example, the structural alignment tool is introduced in the session on protein evolution. Key functions are repeatedly used throughout the learning path to reinforce long-term memory in students. New tools are explained in a short paragraph immediately before they are needed. Many of the exercises also include premade PyMOL scripts that demonstrate the power of scripting and allow the students to learn by example through the copying of relevant sections of the code. The students are also directed to use Internet resources, such as the RCSB Protein Databank (https://www.rcsb.org), the Protein Structure Classification Database, CATH (https://www.cathdb.info), UniProt (https://www.uniprot.org), and ProtParam (https://www.expasy.org/resources/protparam), and are instructed to find further guidance on PyMOL usage on PyMOLwiki (https://pymolwiki.org).

II. Results

A. Educational setting

The PyMOL learning path was designed for a course on biomolecular structure and function, which is a mandatory part of the third semester of the BS programs in molecular biology, molecular medicine, and medicinal chemistry at Aarhus University, Aarhus, Denmark. Overall, the course consists of lectures, classroom teaching, online learning tasks, and an experimental lab exercise. The course uses mainly the textbook *Biochemistry* by Berg *et al*. (22), supplemented with material from other sources. The learning path was implemented as part of 2-hour classroom sessions conducted twice a week, meeting in groups of 20–30 students led by a teaching assistant. The students were asked to attempt to solve the problem sets before arriving in class but were instructed that the difficulty was such that they were not expected to be able to finish all the problems themselves. In the classroom, each exercise was assigned to a group of 3 or 4 students who were given half an hour to prepare, aided by the teaching assistant, before presenting their solution to the rest of the class. This format allows for peer-to-peer learning while the instructor circulates to help groups that are stuck. The supervised preparation in small groups also serves to buttress struggling or insecure students by allowing them to whet their answers in a safe setting before presenting to the whole class. As a principle, all students must present part of a solved problem in every session. The instructor listens to the presentations and corrects errors or misunderstandings. Typically, each session would contain 1 exercise from the PyMOL learning path, while the remaining exercises focus on other aspects of the scientific topic that do not require structural analysis.

B. Evaluation of the students' perception

In the end-of-course survey, the students were asked 3 questions about the PyMOL learning path; 115 students returned the questionnaire out of 146 students participating in the final exam. Some students register for the course without attending classes or taking the exam. We think the number of students taking the final exam is the best proxy for the active student population and thus use these to calculate response rates. In principle, it is possible that some students filled the survey without taking the exam. However, we think this is unlikely to change the results by much. Our questionnaire thus had an estimated response rate of 79% for active students. The high response rate suggests that we capture opinions of a representative cross section of the students. First, the students were asked to assess how much time they had spent preparing for the PyMOL learning path per week (Fig 2A). The results show that students typically spent 2–4 hours per week on the learning path, which translates into 1–2 hours per exercise and suggests that most students made a serious attempt to solve the exercises. A small fraction of students report spending more than 8 hours per week on the PyMOL problems, which would be excessive. We think that this is misreporting the total preparation time spent on the course rather than just the PyMOL exercises. Nevertheless, in future iterations of the course, it might be useful to instruct the students not to get bogged down by single exercises.

Next, the students were asked to rate the degree to which the PyMOL learning path supported their ability to perform structural analysis (Fig 2B). Fifty percent of the students responded "very large" or "large," with most other students replying "some." Overall, most of the students felt that the exercises helped them analyze biomolecular structures, which was the central aim of the learning path. Finally, the students were asked to rate the degree to which the progression and the difficulty of the learning path were appropriate (Fig 2C). Thirty-three percent responded "very large" or "large," whereas most responded "some." This question did not distinguish between students who thought it was too easy or too hard. However, the questionnaire also allowed free text comments where several students indicated that they found the exercises too difficult or too time consuming. None stated that the exercises were too easy. In summary, this suggests that exercises were challenging but not to an extent that led the students to give up. The student-reported learning outcomes should be taken with a grain of salt, as the increased effort required for active learning makes students prefer passive approaches and to overestimate the learning outcomes of these relative to active approaches (15).

C. Evaluation of learning outcomes

We also asked the students to participate in an invigilated test that would evaluate their practical PyMOL skills after completing the course. Out of the 146 students who took the final exam, 123 students participated in this test, and again the high participation rate (84%) suggests that we tested a representative sample. The students were asked to solve 8 tasks individually within 45 minutes and provide their answers as a PyMOL script (Fig

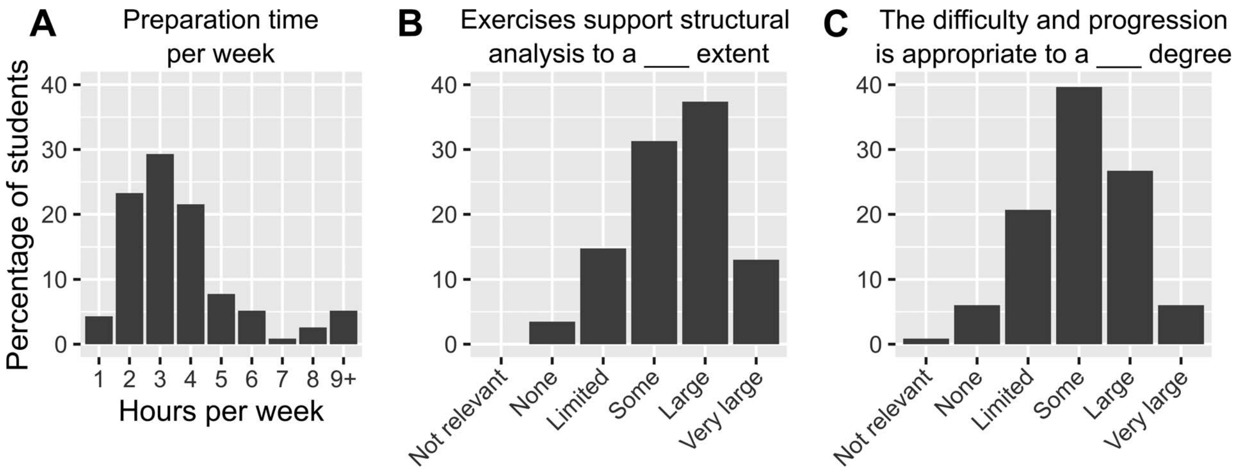

Fig 2. Survey of the students' perception of own learning. The students' experience of their own learning gain was evaluated as part of the overall course evaluation using an anonymous survey conducted via the LMS used for the course (Brightspace). The students were asked how much time they had spent on average preparing the PyMOL exercises (A) per week, which typically concerned 1 or 2 problem sets. Subsequently, the students were asked to assess to which extent the PyMOL learning path supported their ability to perform structural analysis (B) and to which extent the difficulty and progression was appropriate (C). The questionnaire was returned by 115 out of 146 students who took the final exam (79%).

3). This test format requires that the students produce an operational script and thus directly tests their computational skills and mind-set. During the test, the students had access to written resources and the Internet but were not allowed to communicate with one another. This format allows students to look up command syntax, but with an average of less than 6 minutes per question, there was little chance to learn to solve problems from scratch. The questions prompt the students to analyze 2 different crystal structures of the amino acid transporter LeuT (23, 24). As a starting point, the students were given a script that would load the required PDB files and create 8 empty scenes, 1 for each of the answers. The questions follow the progression of the learning path and were meant to become increasingly difficult (Fig 3A). The first 3 questions requested illustration of structural features such as the fold of the protein (Q1), nonprotein components (Q2), and selected residues (Q3). The next 3 questions requested common structural analyses, such as measuring the shortest distance between 2 residues (Q4), highlighting residues within 6 Å from some metal ions (Q5), and a structural alignment of 2 structures (Q6). The final 2 questions requested information to be mapped onto the structures via a color gradient, such as the B-factor of the crystal structure (Q7) and vacuum electrostatics (Q8). The answers were evaluated based on visual inspection of the resulting 8 scenes generated by the submitted scripts, and in cases of doubt, the raw scripts were inspected. Minor deviations from perfection, such as reversed color gradients or scripts that were nonfunctional due to wrong file extensions or omission of elements from the provided template, were tolerated.

The results show that practically all students (98%) manage to visualize the fold of the proteins and highlight nonprotein components after having completed the learning path. A large majority (86%) was also able to highlight specific residues. The test was conducted about 3 months after these functions were introduced in the course, which suggests that the students remember how to use of basic functions under realistic working conditions due to regular repetition. Furthermore, this shows that almost all students have a basic understanding of procedural scripting as implemented in PyMOL. The success rate dropped to 76% in Q4, which asked the students to measure the distance between 2 specific atoms. The syntax for this selection is more complex, as it involves both a residue number and a specific atom. The lowest

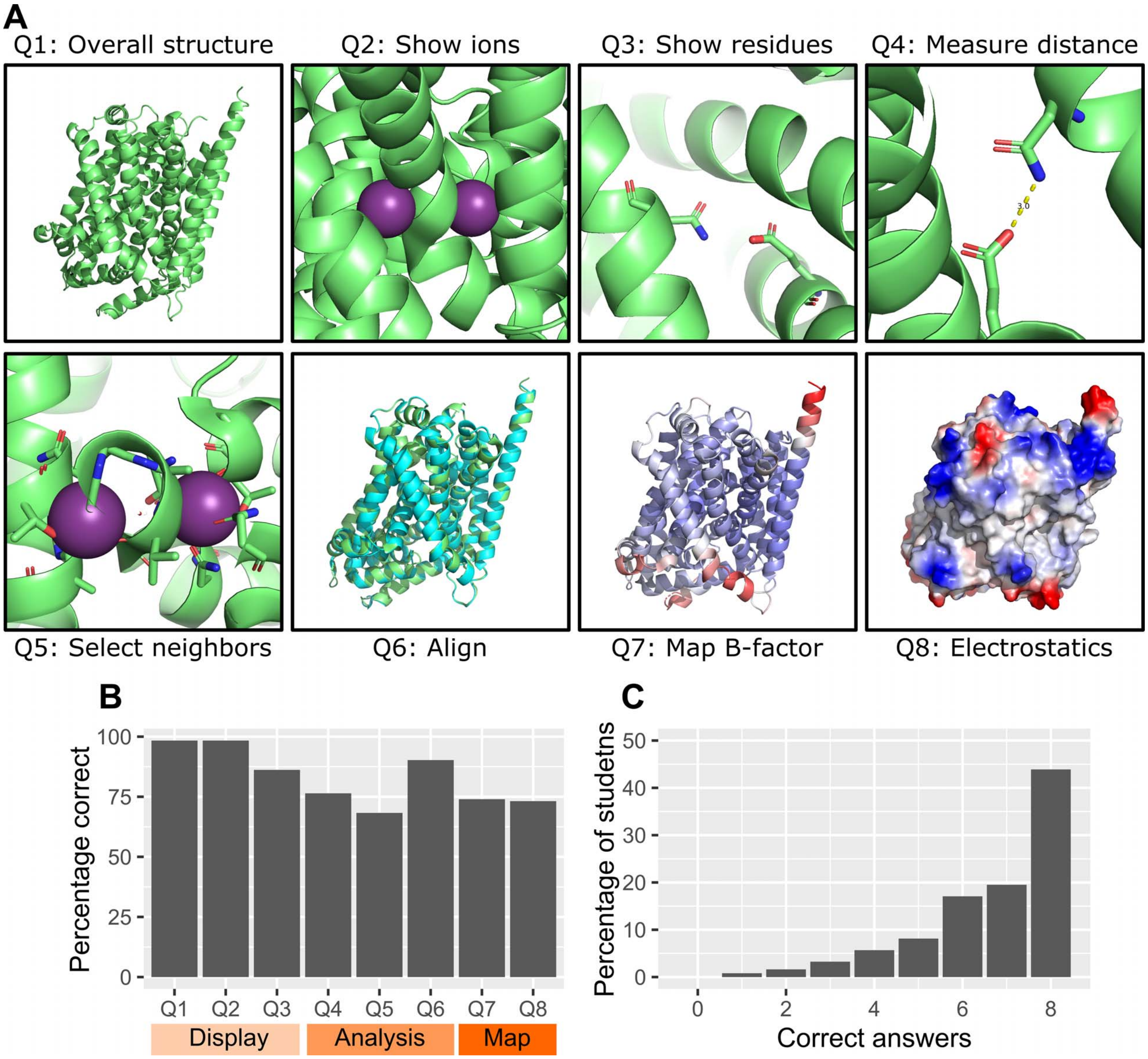

Fig 3. Evaluation of the students' PyMOL competences. The students were asked to complete 8 assignments (A) individually within 45 minutes using PyMOL and submit the answer as a PyMOL script. The questions test the students' ability to create customized displays of molecular features, perform basic structural analysis, and map specific information onto a structure. An example of a correct answer is shown in each panel. The answers were intended to become progressively harder. The answers were scored based on whether the students accomplished the main task, and minor flaws were tolerated. (B) shows the fraction of correct answers for each question, and (C) shows the distribution of correct answers per student. The test was taken by 123 out of 146 students who took the final exam (84%).

success rate (68%) was found for Q5, which asked the student to select and highlight residues within a custom distance of 6 Å. In the analysis section, 90% managed to perform a full-length structural alignment of 2 different structures. The last 2 questions requested structural properties to be displayed as color gradients (Q7: B factor visualized on the protein backbone; Q8: vacuum electrostatics plotted on a surface); this was managed by 74% and 73% of the students, respectively. In total, 44% of students answered all questions correctly, and 80% had at least 6 correct answers, suggesting that they managed to solve at least 1 advanced task. As many students solved all tasks, we may underestimate the level of the most accomplished students' learning. Anecdotally, we noticed that some of the students were highly motivated by the PyMOL scripting and moved beyond curriculum using online resources.

III. Discussion

Computer-based structural analysis is a key practical skill in modern biochemistry. The lack of requirements for specialized equipment makes computer-based analysis an attractive platform for active learning approaches in large classes. Here, we have redesigned our teaching of the molecular graphics software PyMOL to a semester-long learning path to integrate practical skills into relevant scientific contents. By distributing the exercises throughout the semester, it was possible to divide the learning into manageable chunks and introduce spaced repetition. We covered many more functions than in previous years, so our quantitative evaluation cannot be immediately compared to a control group. Our evaluation indicates that after completing the course, most students mastered many advanced functions, including analysis functions typically used in functional interpretation in research labs. The gradual learning path thus likely allowed the students to achieve a higher level of proficiency than was possible in a single-session format.

A semester-long PyMOL path requires a great deal of time and may therefore be more difficult to incorporate in all educational settings. However, as the teaching of the computer analysis is tightly integrated with the learning goals of the overall course, it may not require much more instructional time but rather a change in the nature of the teaching activities to emphasize student active learning and allow independent exploration of structural biology. While the sessions form part of a learning path, they can also work as independent modules. As specialized functions are introduced when they are needed, it is to some extent possible to excise individual exercises to adapt the path to a different curriculum. We thus share the materials of the learning path as supplementary material in the hope that they will be useful to others.

Students enter the course with a wide range skills and attitudes to computers. From the onset, some students already know how to code and may even be more computer savvy than the instructors. Others are only superficially familiar with computing and may have previous negative experiences with coding. The preexisting skills of the students thus vary more than usual in molecular biosciences, which could explain the wide range of different reported preparation times (Fig 2A) and perception of the difficulty (Fig 2C). This poses a challenge for course design: How do you motivate advanced students without losing the students who struggle? We thus think it is important to teach the use of the GUI in parallel to scripting. The GUI lowers the barrier for accessing basic tools but does not allow for all advanced functions. Furthermore, a self-paced learning path supports struggling students with gradual progression and manageable information chunks while allowing more experienced students to skim basic exercises before independently moving beyond the confines of the course. The course also serves as an intuitive way of introducing computational thinking in undergraduate classes. Practically all students managed to produce a PyMOL script and thus have at least a rudimentary understanding of procedural scripting. The learning path thus serves as a primer to scripting that is used broadly in the molecular biosciences in, for example, automation and data analysis as well as bioinformatics.

SUPPLEMENTAL MATERIAL

Course pack and videos are available at: https://doi.org/10.35459/tbp.2022.000219.S1.

AUTHOR CONTRIBUTIONS

All authors contributed to designing teaching materials. The learning outcomes were analyzed by MK and DEB. The article was drafted by MK and DEB with input from all authors.

ACKNOWLEDGMENT

This work was supported by an EduIT grant from the Faculty of Natural Sciences at Aarhus University.

REFERENCES

1. Kuhlbrandt, W. 2014. The resolution revolution. *Science* 343:1443–1444. https://doi.org/10.1126/science.1251652.
2. Jumper, J. R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596:583–589. https://doi.org/10.1038/s41586-021-03819-2.

3. Serpell, L. C., S. E Radford, and D. E Otzen. 2021. AlphaFold: a special issue and a special time for protein science. *J Mol Biol* 433:167231. https://doi.org/10.1016/j.jmb.2021.167231.
4. Tunyasuvunakool, K., J. Adler, Z. Wu, T. Green, M. Zielinski, A. Žídek, A. Bridgland, A. Cowie, C. Meyer, A. Laydon, S. Velankar, G. J. Kleywegt, A. Bateman, R. Evans, A. Pritzel, M. Figurnov, O. Ronneberger, R. Bates, S. A. A. Kohl, A. Potapenko, A. J. Ballard, B. Romera-Paredes, S. Nikolov, R. Jain, E. Clancy, D. Reiman, S. Petersen, A. W. Senior, K. Kavukcuoglu, E. Birney, P. Kohli, J. Jumper, and D. Hassabis. 2021. Highly accurate protein structure prediction for the human proteome. *Nature* 596:590–596. https://doi.org/10.1038/s41586-021-03828-1.
5. Sayle, R. A., and E. J. Milner-White. 1995. RASMOL: biomolecular graphics for all. *Trends Biochem Sci* 20:374–376. https://doi.org/10.1016/S0968-0004(00)89080-5.
6. Guex, N., and M. C. Peitsch. 1997. SWISS-MODEL and the Swiss-Pdb viewer: an environment for comparative protein modeling. *Electrophoresis* 18:2714–2723. https://doi.org/10.1002/elps.1150181505.
7. Jmol: an open-source java viewer for chemical structures in 3D. 2010. Accessed 22 November 2022. http://www.jmol.org/.
8. Pettersen, E. F., T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. 2004. UCSF Chimera—a visualization system for exploratory research and analysis. *J Comput Chem* 25:1605–1612. https://doi.org/10.1002/jcc.20084.
9. Berry, C., and M. D. Baker. 2010. Inside protein structures: teaching in three dimensions. *Biochem Mol Biol Educ* 38:425–429. https://doi.org/10.1002/bmb.20434.
10. Schrödinger, LLC. 2015. The PyMOL Molecular Graphics System. Accessed 22 November 2022. https://pymol.org/.
11. Mooers, B. H. M. 2016. Simplifying and enhancing the use of PyMOL with horizontal scripts: PyMOL with horizontal scripts. *Protein Sci* 25:1873–1882. https://doi.org/10.1002/pro.2996.
12. Mooers, B. H. M., and M. E. Brown. 2021. Templates for writing PyMOL scripts. *Protein Sci* 30:262–269. https://doi.org/10.1002/pro.3997.
13. Hartikainen, S., H. Rintala, L. Pylväs, and P. Nokelainen. 2019. The concept of active learning and the measurement of learning outcomes: a review of research in engineering higher education. *Educ Sci* 9: 276. https://doi.org/10.3390/educsci9040276.
14. Freeman, S., S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proc Natl Acad Sci* 111:8410–8415. https://doi.org/10.1073/pnas.1319030111.
15. Deslauriers, L., L. S. McCarty, K. Miller, K. Callaghan, and G. Kestin. 2019. Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proc Natl Acad Sci* 116:19251–19257. https://doi.org/10.1073/pnas.1821936116.
16. Craig, P. A. 2017. A survey on faculty perspectives on the transition to a biochemistry course-based undergraduate research experience laboratory: transition to a CURE. *Biochem Mol Biol Educ* 45:426–436. https://doi.org/10.1002/bmb.21060.
17. Lineback, J. E., and A. L. Jansma. 2019. PyMOL as an instructional tool to represent and manipulate the myoglobin/hemoglobin protein system. *J Chem Educ* 96:2540–2544. https://doi.org/10.1021/acs.jchemed.9b00143.
18. Rigsby, R. E., and A. B. Parker. 2016. Using the PyMOL application to reinforce visual understanding of protein structure: PyMOL application to understand protein structure. *Biochem Mol Biol Educ* 44:433–437. https://doi.org/10.1002/bmb.20966.
19. Roche Allred, Z. D., H. Tai, S. L. Bretz, and R. C. Page. 2017. Using PyMOL to explore the effects of pH on noncovalent interactions between immunoglobulin G and protein A: a guided-inquiry biochemistry activity. *Biochem Mol Biol Educ* 45:528–536. https://doi.org/10.1002/bmb.21066.
20. Simmons, A. D., T. K. T. Nguyen, J. L. Follis, and A. Ribes-Zamora. 2014. Using a PyMOL activity to reinforce the connection between genotype and phenotype in an undergraduate genetics laboratory. *PLoS One* 9:e114257. https://doi.org/10.1371/journal.pone.0114257.
21. Dunlosky, J., K. A. Rawson, E. J. Marsh, M. J. Nathan, and D. Willingham. 2013. Improving students' learning with effective learning techniques: promising directions from cognitive and educational psychology. *Psychol Sci Public Interest* 14:4–58. https://doi.org/10.1177/1529100612453266.
22. Berg, J. M., J. L. Tymoczko, G. J. Gatto Jr., and L. Stryer. 2019. L. Biochemistry. 9th edition. Macmillan, New York.
23. Zhou, Z., J. Zhen, N. K. Karpowich, R. M. Goetz, C. J. Law, M. E. A. Reith, and D.-N. Wang. 2007. LeuT-desipramine structure reveals how antidepressants block neurotransmitter reuptake. *Science* 317:1390–1393.
24. Singh, S. K., A. Yamashita, and E. Gouaux. 2007 Antidepressant binding site in a bacterial homologue of neurotransmitter transporters. *Nature* 448:952–956. https://doi.org/10.1038/nature06038.