

3-4-2016

Why should biochemistry students be introduced to molecular dynamics simulations—and how can we introduce them?

Donald E. Elmore
delmore@wellesley.edu

Follow this and additional works at: <http://repository.wellesley.edu/scholarship>

Version: Post-print

Recommended Citation

Elmore, D.E. (2016). Why should biochemistry students be introduced to molecular dynamics simulations—and how can we introduce them? *Biochemistry and Molecular Biology Education*, 2016, 44:118-123. DOI: 10.1002/bmb.20943

This Editorial is brought to you for free and open access by Wellesley College Digital Scholarship and Archive. It has been accepted for inclusion in Faculty Research and Scholarship by an authorized administrator of Wellesley College Digital Scholarship and Archive. For more information, please contact ir@wellesley.edu.

Why should biochemistry students be introduced to molecular dynamics simulations—and how can we introduce them?

Donald E. Elmore

Department of Chemistry and Biochemistry Program;

Wellesley College; Wellesley, MA 02481

Abstract

Molecular dynamics (MD) simulations play an increasingly important role in many aspects of biochemical research but are often not part of the biochemistry curricula at the undergraduate level. This article discusses the pedagogical value of exposing students to MD simulations and provides information to help instructors consider what software and hardware resources are necessary to successfully introduce these simulations into their courses. In addition, a brief review of the MD based activities in this issue and other sources are provided.

Keywords

molecular dynamics simulations, molecular modeling, biochemistry education, undergraduate education, computational chemistry

Published in *Biochemistry and Molecular Biology Education*, **2016**, 44:118-123.

Several years ago, I developed a course on the modeling of biochemical systems at Wellesley College, which was described in more detail in a previous issue of this journal [1]. As I set out to design that course, I decided to place a particular emphasis on molecular dynamics (MD) simulations. In full candor, this decision was likely related to my experience integrating these simulations into my own research, but I also believe it was due to some inherent advantages of exposing students to this particular computational approach. Notably, even as this initial biochemical molecular modeling course evolved into a more general (i.e. not solely biochemical) course in computational chemistry, a significant emphasis on MD has remained, emphasizing the central role I and my other colleague who teaches the current course felt it played in students' education. To a differing extent, MD simulations have played a notable role in courses in computational chemistry, biochemistry and pharmaceutical chemistry described by others in the literature [2-4].

In this article, I will reflect on our rationale for placing this significant focus on MD in our computational (bio)chemistry courses, along with the particular importance of this method for biochemistry students. Despite these compelling reasons, there are nonetheless challenges to successfully incorporating MD simulations into courses stemming from the necessary computer resources and the potential learning curve for students. Each of the subsequent articles in this special section of the journal addresses these challenges in a different manner. Here, I hope to outline other resources that may be helpful to other instructors hoping to integrate MD into their own courses.

Why integrate molecular dynamics simulations into biochemistry?

MD simulations—like many other computational methods—were traditionally seen as the purview of physical chemistry, and some may wonder whether how relevant the method is for inclusion into the biochemistry curriculum. In fact, many published examples of classroom or laboratory activities involving MD simulations have been explicitly developed for physical chemistry courses [5-7]. While the deeper understanding of the mathematical modeling of chemical systems developed in physical chemistry courses can be helpful to students in understanding MD, it isn't necessarily essential for developing a solid conceptual appreciation for the method. This is particularly true since MD largely utilizes ideas from classical mechanics (e.g. Newton's Law) to predict the motion of atoms and molecules over time. Practically, the Wellesley course in computational chemistry and the course utilizing the activity described in this issue by Spitznagle et al. do not have a physical chemistry prerequisite, and others have noted the ability to adjust their activities to other courses [5].

One reason I believe it is important to expose biochemistry students to MD methods stems from the increasingly prevalent role these simulations play in contemporary biochemical research. For several years after the first MD simulation of a protein was published in 1977 [8], MD simulations were the domain of relatively few specialists working with relatively rare computer resources. However, the application of MD simulations to an increasingly wide range of biological systems has exploded in the literature over the past fifteen years, as seen by the number of papers that explicitly refer to these simulations in either their title or abstract (Fig. 1). During the month of October

2015, six papers where MD simulations played a central role in the research appeared in the journal *Biochemistry* and five in the *Biophysical Journal*. MD simulations also were one of the central aspects of the work recognized by the 2013 Nobel Prize, which included Martin Karplus, one of the co-authors of the initial 1977 simulation of bovine pancreatic trypsin inhibitor [8].

Thus, it is clear that contemporary biochemists will need to have some familiarity with MD in order to appreciate the increasing number of research studies that incorporate these simulations. However, MD simulations also can play an important pedagogical role in helping students to appreciate the dynamic nature of biomolecules. Many writers have discussed the value of effective molecular visualization tools in biochemistry courses [9-11], and many modern biochemistry courses give students experience in viewing the structures of proteins, nucleic acids and other macromolecules using common molecular visualization software, such as Pymol [12], VMD [13], Rasmol [14] or Swiss PDB Viewer [15]. However, these visualization tools only provide a static view of molecular structure, such as a conformation of a protein captured in a crystal structure, and students can find it difficult to think about the amount of structural flexibility that occurs within these “stable” structures. One way to integrate this idea of molecular motion is through animations. While these animations can be valuable tools, some believe that students have the potential to learn more from a representation they are actively involved in creating instead of being passive viewers [16].

MD simulations provide a means through which students can be involved in directly creating the animations showing molecular motion. Although still outside common use, interactive MD simulations in which users directly “interact” with the forces and motions in a biological molecule through a virtual reality system are particularly intriguing for this purpose [17]. However, even more “conventional” MD approaches, such as those described in this issue and other published classroom activities, nonetheless allow students to participate directly in producing molecular “movies” portraying biomolecular dynamics. While these simulations have the potential for exposing students to overwhelming detail—for example, students can potentially “see” every water molecule in an image—there is some evidence that exposure to more complicated and sophisticated visualizations can enhance student learning [10, 11]. Another potential concern with molecular visualizations is that students may lack the appropriate background or visual language to appreciate or interpret some images in the ways desired by their instructors [16], and MD has the potential to help students think more closely about these representations. In fact, a few recent reports have provided evidence for the value of MD in different classroom settings. Burkholder et al. have described their use of MD simulations to help highlight the importance of intermolecular interactions in an introductory chemistry course [18]. More recently, Albrecht described how MD simulations were effective in helping students evaluate the feasibility of two Wittig reactions mechanisms, leading to an increase in student perceptions related to visualization in addition to other learning gains [5].

What computational resources are necessary to effectively teach molecular dynamics?

Software to perform and visualize MD simulations

Even an instructor who is convinced of the value of integrating MD simulations into the curriculum may have understandable concerns about being able to obtain the necessary computational resources to effectively expose students to these simulations. Thankfully, the expense of software and hardware has decreased over the years. In fact, it is frequently possible to use software that students can freely download for their own use after the completion of a course (Table 1). Many major MD software packages commonly used for research applications, such as Gromacs [19], NAMD [20] and Desmond [21] are freely available. While not free, two other commonly used software packages, Amber [22] and CHARMM [23], have relatively inexpensive academic licenses (e.g. ≈\$500-600), although that could serve as an impediment to a student's downloading software for their own computers for future use. While individual researchers have their personal preferences between these software packages, all generally incorporate similar methods, particularly in terms of the types of MD simulations one is likely to integrate into an undergraduate course setting. Another freely available package for MD simulations is Abalone [24]. While Abalone has fewer features and is therefore used less frequently for research applications, its relatively user-friendly interface could be amenable to some educational purposes, as discussed below.

MD simulations effectively produce a series of structures that represent the conformations of the simulated system over time, and these structures are saved into a trajectory file. Likely any course using MD simulations would wish to have students

view these trajectory files as molecular movies. The most commonly used software packages used to visualize these trajectories are also freely available, at least for educational use (Table 1). VMD is often used for this purpose as it has the ability to read in the trajectory files created during simulations in the formats produced by many different software packages [13]. Pymol [12] is also commonly used by researchers and has a somewhat more limited version freely available for educational use. While Pymol produces excellent images, and we use it extensively in our biochemistry courses at Wellesley to visualize PDB files, it is typically less straightforward to use for viewing trajectories. One can also use other freely available software, such as Swiss PDB Viewer [15] and Rasmol [14] to view individual conformations from a simulation, although these packages lack the ability to show trajectories as movies.

Simplified user interfaces for MD software

While instructors may be able to freely obtain MD software, most of the packages listed above lack a native graphical user interface (GUI). Instead, these programs typically work by a combination of text file scripts and programs run from the Unix/Linux command line. Thus, learning how to use the software can present a significant learning curve to students (and instructors). While students usually are technologically savvy in many aspects, in our experience relatively few chemistry and biochemistry students have significant experience working on computers outside a modern Windows/Mac based environment. In our upper-level computational biochemistry and chemistry courses at Wellesley, we have decided that there is a value to having students learn the native interface of a software package. While this decision makes sense in the context of a full

course on computational chemistry, it may not be feasible from the perspective of implementing a smaller-scale MD-based activity into a broader biochemistry course.

One way to overcome this is to use a program with an easier to use native interface, such as Abalone, which is used in the activities described by Rodrigues et al. in this issue.

While Abalone lacks some of the capacities of other software packages, it can work well for some applications. Many commercially available software packages also include more user-friendly interfaces, although those can have more significant licensing fees.

Over the past few years, researchers have increasingly developed more user-friendly interfaces that can be used with other free or inexpensive MD packages discussed above.

For example, VMD can be used as a GUI to set up and analyze MD simulations in NAMD. Having this interface significantly decreases the learning curve for learning NAMD, as demonstrated in the integration of MD simulations into a single lab period of a biochemistry course described by Chiang et al [25]. A recent report has also described the creation of a freely available plugin that allows Pymol to be used as GUI to setup simulations in Gromacs [26]. Miller et al. have also described the creation of the CHARMMing portal, which provides a web-based front end for setting up and analyzing simulations in CHARMM [27, 28]. Notably, they have also developed a number of activities as part of CHARMMing that could be useful to students and integrated into courses. Another potentially useful resource is CHARMM-GUI, which provides an interactive web-based interface that can produce input files for CHARMM, GROMACS, NAMD and AMBER [29]. While these resources can be very useful for developing

course activities and for some research applications, these interfaces typically lack the ability to access the full capacities of MD software packages.

Hardware resources

Integrating MD simulations into a course also requires sufficient computer resources to run simulations. Modern desktop computers now have sufficient processing power to perform most simulations that would be relevant for courses. For example, in some semesters all the MD simulations in our computational modeling courses were performed on a multi-processor Macintosh workstation [1]. While students can potentially perform some simulations on their own laptop computers, we have typically found it easier to install and maintain the appropriate MD packages on departmentally managed computers. However, students do routinely log on to these centralized computer resources through their own laptops. Students frequently visualize structures and trajectories on their own machines using freely available software packages described above.

In cases where institutional computer resources are not sufficient for the desired MD activities, instructors can also consider procuring computational resources from another source. An excellent option for this is through XSEDE (Extreme Science and Engineering Discovery Environment), which has a specific allocation process for instructors wishing to use supercomputer resources in classes. One can learn more about these educational allocations at: portal.xsede.org/allocations-overview#writing-startupeducation. These allocations can provide significant amounts of computer time and could be particularly helpful in teaching somewhat larger numbers of students.

Analogous resources are available in other countries, such as SURFSara in the Netherlands, which was utilized in the activities described by Rodrigues et al. in this issue. Some commercial cloud computing options, such as Amazon EC2, may also be financially feasible for course-based simulations, particularly if simulations could be done within the free trial timeframe. Students could also run short simulations through MDWeb, where one can perform short MD simulations (e.g. <400 ps) through a web account [30]. Although these simulations are very short, they still could be used to give students an initial idea of the extent of molecular motions that occur on a relatively short timescale.

Resources for developing MD activities in this issue

In developing computational modeling courses at Wellesley, I was struck by the relative absence of computer lab activities in the educational literature performing MD or other methods to consider the dynamic behavior of biomolecular motions. The websites for most MD packages typically link to tutorials; some of these are quite good and can provide a starting point for developing a class activity. However, there is a value in having access to activities that have been successfully implemented in an actual class environment. Thus, a goal of this special section was to collect together a series of activities that could be adapted by instructors hoping to integrate MD simulations and other related methods into their own classes. The four articles included target different audiences and skills, highlighting how simulations could be useful at different levels of the curriculum. Links to resources, such as course handouts, are provided with each contribution.

In the first article, Lundquist et al. describe the development of VMDlite, a simplified version of VMD created to introduce high school students to the visualization of biological molecules. These visualization activities include giving students the opportunity to view trajectories from MD simulations in order to provide insight into molecular motions. Considering these structures during high school will help students better understand molecular principles in addition to teaching them an important software skill they can ultimately apply during subsequent college courses. Activities in VMDlite may also be relevant to some introductory-level college students.

Spitznagel et al. write about a series of activities that provide students with insight into the conceptual background of MD through a coding exercise followed by the opportunity to perform simulations on a real peptide using a more complete MD package. These activities were integrated into a course on Modern Physics in Biology that is taken primarily by students in biologically oriented major. The only prerequisite for this course is one year of physics coursework, making it potentially open to students relatively early in their college career.

The contribution from Hati and Bhattacharyya describes an upper-level project-based laboratory course in biophysical chemistry that incorporates several aspects of molecular visualization and computation. This is a capstone course taken by senior biochemistry and molecular biology majors, most of whom have already completed two semesters of biochemistry. Instead of using MD simulations, this course shows students how to utilize

another approach, normal model analyses, to consider the conformational dynamics of proteins. Students in this course also have the opportunity to consider how to use these calculations along with other approaches, such as homology modeling, during a multi-week project considering an active research question.

Finally, Rodrigues et al. discuss the design of a series of course activities designed to give advanced undergraduate students experience in setting up, running and analyzing MD simulations. These activities are part of a larger course on molecular modeling. While they have been successfully implemented in the undergraduate curriculum at Utrecht University, they could easily be adapted for graduate students or as training exercises for students starting to utilize MD simulations as part of their independent research.

It has been a pleasure working with these authors on this section, and reading their contributions have given me new ideas and inspiration for thinking about the incorporation of MD simulations and other computational methods into my courses. I hope that others find their work as a useful springboard for enhancing the role of computer modeling in the curriculum at their own institutions.

Acknowledgements

I would like to express my appreciation to Mala Radhakrishnan, my colleague at Wellesley who has also taught Computational Chemistry, for many helpful discussions about the teaching of computational modeling that have helped to shape this article. I also thank the Wellesley Educational Research and Development Committee for their consistent support in obtaining the necessary computer resources to teach our computational courses. D. E. E. is a Henry Dreyfus Teacher-Scholar.

Tables

Table 1: Examples of free or low-cost software packages for performing MD simulations¹

Software package	Cost	Website
Desmond	Free	www.deshawresearch.com/resources_desmond.html
Gromacs	Free	www.gromacs.org
NAMD	Free	www.ks.uiuc.edu/Research/namd
Abalone	Free	www.biomolecular-modeling.com/Abalone
AMBER	\$500	ambermd.org
CHARMM	\$600	www.charmm.org

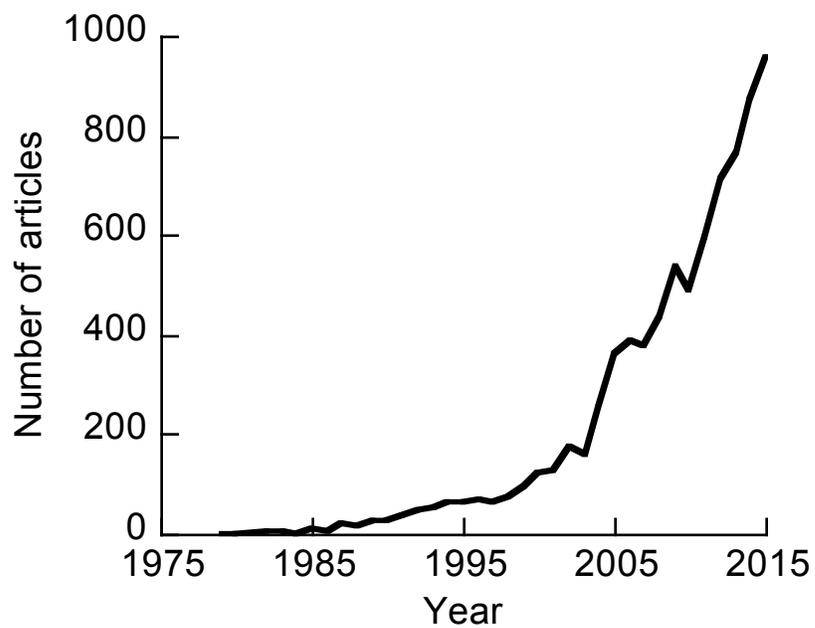
¹ Costs given for software used for educational purposes at an academic institution; licenses for research use may be more expensive in some cases. List of software packages is not intended to be exhaustive, as other resources are available for these purposes.

Table 2: Examples of free software packages for visualizing molecular structures and trajectories from MD simulations¹

Software package	Can view MD trajectory	Website
VMD	Yes	www.ks.uiuc.edu/Research/vmd
Pymol	Yes	pymol.org
Swiss PDB Viewer	No	spdbv.vital-it.ch
Rasmol	No	rasmol.org

¹ While all these packages are free for educational use at academic institutions, licenses for research use may not be free in some cases. List of software packages is not intended to be exhaustive, as other resources are available for these purposes.

Figure 1: Number of journal articles from 1975-2015 in PubMed including the terms molecular dynamic simulation, molecular dynamics simulation or MD simulation in either the title or abstract. Search performed on November 5, 2015.



References

- [1] D. E. Elmore, R. C. Guayasamin, M. E. Kieffer (2010) A series of molecular dynamics and homology modeling computer labs for an undergraduate molecular modeling course, *Biochemistry and molecular biology education : a bimonthly publication of the International Union of Biochemistry and Molecular Biology*, 38, 216-223.
- [2] Z. L. Gasyna, S. A. Rice (1999) Computational chemistry in the undergraduate chemistry curriculum: Development of a comprehensive course formula, *J Chem Educ*, 76, 1023-1029.
- [3] M. J. Ramos, P. A. Fernandes, A. Melo (2004) Modeling chemical and biological systems: A successful course for undergraduate students, *J Chem Educ*, 81, 72-75.
- [4] C. S. Tsai (2007) Using computer applications and online resources to teach and learn pharmaceutical chemistry, *J Chem Educ*, 84, 2019-2023.
- [5] B. Albrecht (2014) Computational chemistry in the undergraduate laboratory: A mechanistic study of the Wittig reaction, *J Chem Educ*, 91, 2182-2185.
- [6] E. Saiz, M. P. Tarazona (1997) Molecular dynamics and the water molecule: An introduction to molecular dynamics for physical chemistry students, *J Chem Educ*, 74, 1350-1354.
- [7] O. F. Speer, B. C. Wengerter, R. S. Taylor Molecular dynamics simulations of simple liquids, *J Chem Educ*, 81, 1330-1332.
- [8] J. A. McCammon, B. R. Gelin, M. Karplus (1977) Dynamics of folded proteins, *Nature*, 267, 585-590.
- [9] D. R. Canning, J. R. Cox (2001) Teaching the structural nature of biological molecules: Molecular visualization in the classroom and in the hands of students., *Chem Educ Res Pract*, 2, 102-122.
- [10] J. Jenkinson, G. McGill (2012) Visualizing protein interactions and dynamics: evolving a visual language for molecular animation, *CBE life sciences education*, 11, 103-110.
- [11] L. A. Tibell, C. J. Rundgren (2010) Educational challenges of molecular life science: Characteristics and implications for education and research, *CBE life sciences education*, 9, 25-33.
- [12] The PyMOL Molecular Graphics System (2015) Version 1.7.4. Schrödinger, LLC.
- [13] W. Humphrey, A. Dalke, K. Schulten (1996) VMD: visual molecular dynamics, *Journal of molecular graphics*, 14, 33-38, 27-38.
- [14] R. Sayle, E. J. Milner-White (1995) RasMol: Biomolecular graphics for all, *TIBS*, 20, 374.
- [15] N. Guex, M. C. Peitsch (1997) SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling, *Electrophoresis*, 18, 2714-2723.
- [16] L. L. Jones, K. D. Jordan, N. A. Stillings (2005) Molecular visualization in chemistry education: The role of multidisciplinary collaboration, *Chem Educ Res Pract*, 6, 136-149.
- [17] J. E. Stone, A. Kohlmeyer, K. L. Vandivort, K. Schulten, in: R. Boyle, B. Parvin, D. Koracin, R. Chung, H. M. Hussain, K.-H. Tan, R. Crawfis, D. Thalmann, D. Kao, L. Avila (Eds.) (2010) *Advances in Visual Computing*, vol. 6454, Springer-Verlag, Berlin, 2010, pp. 382-393.

- [18] P. R. Burkholder, G. H. Purser, R. S. Cole Using molecular dynamics simulation to reinforce student understanding of intermolecular forces, *J Chem Educ*, 85, 1071-1077.
- [19] S. Pronk, S. Pall, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, E. Lindahl (2013) GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit, *Bioinformatics*, 29, 845-854.
- [20] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, K. Schulten (2005) Scalable molecular dynamics with NAMD, *Journal of computational chemistry*, 26, 1781-1802.
- [21] K. J. Bowers, E. Chow, H. Xu, R. O. Dror, M. P. Eastwood, B. A. Gregersen, J. L. Klepeis, I. Kolossváry, M. A. Moraes, F. D. Sacerdoti, J. K. Salmon, Y. Shan, D. E. Shaw (2006) Scalable algorithms for molecular dynamics simulations on commodity clusters, *Proceedings ACM/IEEE Conference on Supercomputing (SC06)*.
- [22] Author (2015) AMBER 2015. University of California, San Francisco.
- [23] B. R. Brooks, C. L. Brooks, 3rd, A. D. Mackerell, Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, M. Karplus (2009) CHARMM: the biomolecular simulation program, *Journal of computational chemistry*, 30, 1545-1614.
- [24] Abalone. (2015) Agile Molecule.
- [25] H. Chiang, L. C. Robinson, C. J. Brame, T. C. Messina (2013) Molecular mechanics and dynamics characterization of an in silico mutated protein: a stand-alone lab module or support activity for in vivo and in vitro analyses of targeted proteins, *Biochemistry and molecular biology education : a bimonthly publication of the International Union of Biochemistry and Molecular Biology*, 41, 402-408.
- [26] T. Makarewicz, R. Kazmierkiewicz (2013) Molecular dynamics simulation by GROMACS using GUI plugin for PyMOL, *Journal of chemical information and modeling*, 53, 1229-1234.
- [27] B. T. Miller, R. P. Singh, J. B. Klauda, M. Hodoscek, B. R. Brooks, H. L. Woodcock, 3rd (2008) CHARMMing: a new, flexible web portal for CHARMM, *Journal of chemical information and modeling*, 48, 1920-1929.
- [28] B. T. Miller, R. P. Singh, V. Schalk, Y. Pevzner, J. Sun, C. S. Miller, S. Boresch, T. Ichiye, B. R. Brooks, H. L. Woodcock, 3rd (2014) Web-based computational chemistry education with CHARMMing I: Lessons and tutorial, *PLoS computational biology*, 10, e1003719.
- [29] S. Jo, T. Kim, V. G. Iyer, W. Im (2008) CHARMM-GUI: a web-based graphical user interface for CHARMM, *Journal of computational chemistry*, 29, 1859-1865.
- [30] A. Hospital, P. Andrio, C. Fenollosa, D. Cicin-Sain, M. Orozco, J. L. Gelpi (2012) MDWeb and MDMoby: an integrated web-based platform for molecular dynamics simulations, *Bioinformatics*, 28, 1278-1279.