The Biophysicist

A Computer Laboratory for the Calculation of the Heat Capacity of a Peptide Unfolding Transition

Lau Blom Grøndahl[1], Fabian Schuhmann[2], Dennis Bruhn[1], Vikas Dubey[1], Ilia A. Solov'yov[3], Himanshu Khandelia[1,*]

[1]PHYLIFE: Physical Life Science, Department of Physics, Chemistry and Pharmacy, University of Southern Denmark, Odense, Denmark
[2]Niels Bohr International Academy, Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark
[3]Institute of Physics, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany

ABSTRACT We present the implementation of a computer simulation laboratory for the calculation of the heat capacity of the folding-unfolding pseudo phase transition of a polyalanine Ala_{30} α-helix, using molecular dynamics simulations. Unlike conventional calorimetry experiments, which provide a macroscopic interpretation of the heat capacity, the computer laboratory additionally offers a molecular perspective of the energy exchange processes that lead to the peak of the heat capacity at the phase transition, as well as the statistical mechanical interpretation of the phase transition. The exercise is targeted toward 1st- or 2nd-year students of biophysics in the context of an undergraduate physical chemistry course. We use a graphical web interface to overcome the steep learning barriers associated with command line interactions in commonly used molecular dynamics protocols. The web interface (VIKING) also handles communication with computing hardware in the background, so that students with limited or no programming experience can launch and analyze the simulations easily. A collaboration between the VIKING support team and responsible teachers is required to set up VIKING. We also present a solution in which simulations can be launched on students' own computers, if some command line knowledge can be expected. The laboratory can also be repurposed for students of various backgrounds (physics, chemistry, biology, or engineering) and scholastic levels (undergraduate or graduate).

KEY WORDS molecular dynamics; phase transition; protein folding-unfolding transition; specific heat capacity; physical chemistry

I. INTRODUCTION

Students without a strong theoretical physics or chemistry background encounter the term "phase transition" in their first biophysics or physical chemistry course. A first-order phase transition is associated with the heat capacity, typically defined at a given temperature as a first partial derivative of the enthalpy (constant pressure heat capacity) or internal energy (constant volume heat capacity). Students can measure the heat capacity of various materials in the laboratory by using a calorimeter as part of a laboratory exercise during a full-fledged bachelor's-level course in physical chemistry or fundamental biophysics. In biology, such a material could be a folded protein, a single-component or 2-component lipid membrane, or a DNA double

"*" corresponding author

Received: 15 June 2023
Accepted: 22 May 2024
Published: 29 July 2024

helix. Performed carefully, such an experiment can measure the peakof the heat capacity of the material as it approaches a temperature-induced phase transition. Although this approach can reveal macroscopic properties of the phase transition, the underlying microscopic behavior of the material that led to the phase transition is often left completely unexplored. Students gain no insights into the soft forces that drive the assembly and evolution of biological macromolecules. Molecular dynamics (MD) simulations can be used to explore such microscopic behavior because the simulations enable calculation of the macroscopic properties (such as the heat capacity) while simultaneously providing atomic-resolution information on the molecular interactions in the biological material.

In this article, we describe a computer laboratory for a 2nd-year Physical Biochemistry course in which we use MD simulations to explore the folding-unfolding transition of a poly-alanine Ala_{30} molecule in vacuum, with the objective of establishing the link between the heat capacity of the phase transition (macroscopic thermodynamic property) and the molecular details accompanying the phase transition.

A key challenge in running such a laboratory is that bachelor's students with a background in biochemistry or molecular biology often have no prior knowledge of MD simulations or training in statistical mechanics. To circumvent these challenges, we use a graphical user interface and web platform: Scandinavian Online Kit for Nanoscale Modeling (VIKING) (1), which sets up an MD simulation and analyzes the data automatically. The students encounter the actual simulation as a black box but are made familiar with the basic principles of MD simulations both before and during the laboratory exercise. In this way, the students can use MD simulations as a tool without having to undergo a rigorous initiation into statistical mechanics or numerical integration.

The rest of the article is organized as follows. In the section “Background,” we describe the basic principles of MD simulations and calculation of the heat capacity. Under “Materials and Methods,” we describe the MD method that we use to calculate the heat capacity in a research setting. Under the section “Implementation as a Computer Laboratory,” we describe exactly how the method is implemented in a computer laboratory, which lasts ~3 h. We provide as many details as possible to enable easy incorporation into a 1st or 2nd year bachelor's-level course in biophysics or physical or biophysical chemistry.

II. BACKGROUND

The proposed simulations use the MD package NAMD (2) to model a transition from an α-helix to a random coil phase of a single poly-alanine Ala_{30} molecule. The laboratory design and simulations presented here are inspired by Solov'yov et al. (3).

The overall idea is to numerically solve the equation of motion for each atom in the poly-alanine Ala_{30} molecule, which in this case is the Langevin equation (4) given by

$$m_i \ddot{\vec{r_i}} = -\frac{\partial U(\vec{R})}{\partial \vec{r_i}} - \gamma m_i \dot{\vec{r_i}} + \sqrt{2 m_i \gamma k_B T}\,\vec{\eta}(t) \tag{1}$$

Here, m_i is the mass of an atom and $\vec{r_i}$ its position. The first term on the right corresponds to systematic forces arising from the interactions between particles of the solute. $U(\vec{R})$ is the potential energy of the system that depends on the position $\vec{R}$ of all atoms. The negative gradient of $U(\vec{R})$ with respect to $\vec{r_i}$ is the force that acts on particle i. The 2nd and 3rd term on the right are responsible for coupling the particle to a heat bath. Here, γ is the damping coefficient, k_B the Boltzmann constant, T the temperature of the system, and $\vec{\eta}$ a Gaussian distributed noise term that satisfies $\langle \vec{\eta}(t) \rangle = 0$ and $\langle \vec{\eta}(t)\,\vec{\eta}(t') \rangle = \delta(t - t')$. We use the semi-empirical CHARMM27 force field (5) to calculate the system potential energy $U(\vec{R})$. The CHARMM27 force field is well suited for MD simulations of large biomolecular systems such as proteins, peptides, lipids, and nucleotides.

The solution of the Langevin equation is carried out in small time steps. In these time steps, the various derivatives of the Langevin equation are assumed to be constant, enabling discretization of the equation. By repeatedly solving the discretized Langevin equation, the dynamical behavior of the poly-alanine Ala_{30} molecule is simulated. Effectively this is done by using the velocity Verlet algorithm, which is described in the appendix of Swope et al. (6). The system energy $E(t)$ can be accumulated as a function of simulation time.

The heat capacity C_V of the system can then be calculated from the time averaged values $\langle E \rangle$ and $\langle E^2 \rangle$ of the system energy by considering heat capacity as a derivative of the mean energy with respect to temperature T:

$$C_V(T) = \frac{\partial \langle E \rangle}{\partial T} = \frac{1}{k_B T^2} \frac{\partial}{\partial \beta} \left(\frac{1}{Z} \frac{\partial Z}{\partial \beta} \right) = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}. \quad (2)$$

Here, $\beta = (k_B T)^{-1}$, with k_B being the Boltzmann constant, and $Z = \sum_n \exp(-E_n \beta)$ is the partition function, which can be shown to be related to the mean value of the potential energy as $\langle E \rangle = -\frac{1}{Z}\frac{\partial Z}{\partial \beta}$ and $\langle E^2 \rangle = \frac{1}{Z}\frac{\partial^2 Z}{\partial \beta^2}$. The simulation should be repeated at different temperatures, to characterize the thermodynamic properties of a phase transition, by estimating $\langle E \rangle$ and $\langle E^2 \rangle$ for each simulation to calculate the heat capacities for the corresponding temperatures.

Unlike in statistical mechanics, the pseudo phase transition is macroscopically observed as a peak in the heat capacity when the heat capacity is plotted against temperature. This is in contrast to the statistical mechanical description of a phase transition of a system containing a very large number of degrees of freedom, in which the heat capacity diverges around a critical point (7). The finite peak of the heat capacity in the simulations manifests because of the finite size and sampling of the system. This is in general a limitation in the investigation of systems around a critical point using numerical methods that can be used only to investigate finite-sized systems. The peak of C_V is also rather wide because the transition from an α-helix to the random coil phase occurs gradually as temperature increases, meaning that during the phase transition, the poly-alanine Ala_{30} molecule retains some α-helix-like and random coil flavor. The teacher may choose to emphasize these differences depending on the educational background of the students.

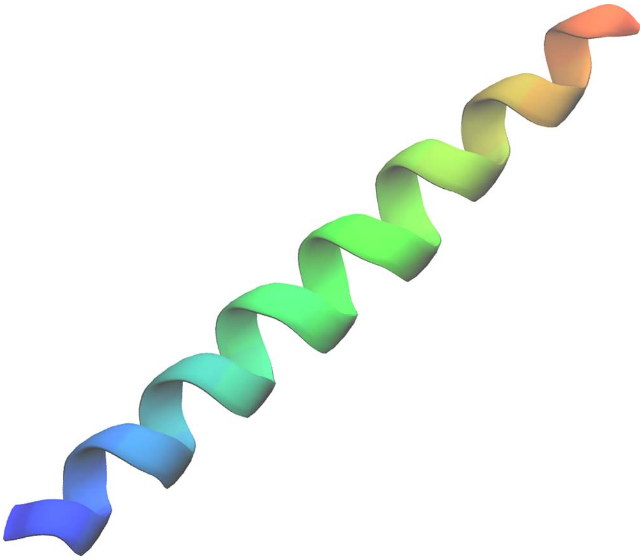


Fig 1. The poly-alanine Ala_{30} α-helix.

III. MATERIALS AND METHODS

To calculate the heat capacity of an alanine polypeptide, a number of steps must be carried out by using different computational tools.

First, the initial 3-dimensional coordinates of the poly-alanine Ala_{30} molecule must be defined. The equations of motion, Eq. (1), are second-order differential equations that require initial positions of all particles to be specified to permit a solution. The coordinates of a perfect α-helix poly-alanine Ala_{30} molecule were generated by using PyMOL (Fig 1) (8).

Second, the Langevin equations must be solved numerically by using NAMD. NAMD calculates the total energy of the system and several other properties and exports them to output files. To run NAMD, an input configuration file is generated. The configuration file points to the coordinates of the poly-alanine

Ala_{30} molecule and to the CHARMM27 force field and specifies various other simulation parameters such as the temperature, the number of integration time steps, the size of the time step, and the frequency (number of time steps) with which total system energy is written to an output file.

The MD simulation is performed in 3 stages: (i) 100 steps of energy minimization, (ii) 1 ns of an equilibration simulation with an integration time step of 2 fs, and (iii) a production simulation of 200 ns. Note that the simulations performed for this article were equilibrated in a 20-ns simulation. The total energy of the system is sampled every 2,000 time steps during the production simulation. Note further that these timescales are nonphysical and will be discussed further in the discussion. More frequent sampling of data can be time consuming, and we want to avoid sampling highly correlated data points, which would not significantly improve the estimate of the mean of the total system energy.

The simulations were conducted on central processing unit (CPU) nodes with Intel Xeon Platinum 8358 32-core CPUs. Multiple cores per simulation decreased the wall time required per simulation up to about 8 cores, after which the required wall time increased as more cores were used. Because of the relatively small system size, it was optimal to run 1 simulation per CPU core, allowing for many simulations to be run simultaneously on each CPU node. It took ~20 h to finish a simulation on one core, meaning that 32 simulations could be completed per CPU within a day.

It is advisable to benchmark the performance of the simulations on the available hardware, such that an optimal distribution of computer resources can be made during the actual class (see the next section).

Significant computer time is required for running 501 simulations: 1 for every temperature value between 500 and 1,000 K. It is advisable to let each student or group of students run simulations for a small, predefined set of temperatures such that all the students doing the exercise can collect their data in a shared spreadsheet and collectively make a high-resolution plot of the heat capacity as a function of temperature. The teacher can provide the other data points a priori in the Excel sheet.

IV. IMPLEMENTATION AS A COMPUTER LABORATORY

A. Background and prerequisites

At the University of Southern Denmark (SDU), the computer laboratory exercise described below is part of a 5-credit Physical Biochemistry course, intended for 2nd-year students of Biochemistry, Molecular Biology, and Biomedicine (https://bit.ly/3RI72L9). Half of the course focuses on conventional undergraduate physical chemistry, while the other half focuses on research-based physical chemistry methods particularly suited for Biochemistry and Molecular Biology majors, such as single-molecule diffusion, reaction kinetics, and binding equilibria. Before coming to the laboratory, students

1. are already familiar with the macroscopic concepts of heat capacity.
2. have been given a 30-minute introduction to the basic principles of MD simulations in a lecture and provided sufficient background material in the form of videos and slides about the general applications of MD simulations in Biology.
3. have read the relevant pages in the textbook that discuss the energy exchange process when heat is supplied to a macromolecule undergoing a phase transition, such as DNA melting or protein unfolding (e.g., references 9 and 10).
4. are not familiar with the Fluctuation Dissipation theorem, which leads to Eq. 2. Students are also not familiar with the concepts of Equilibrium Statistical Mechanics, other than the Kinetic Theory of Gases.
5. are provided the lab manual $\geq$1 week before the laboratory session.
6. are asked to install PyMOL before arriving to the laboratory. The PyMOL installation files

for all major operating systems can be found at https://PyMOL.org/2/, and PyMOL can be used for free for educational purposes.

NAMD and most other publicly available MD simulation software such as GROMACS (11) and LAMMPS (12) necessitate command-line-based interaction, familiarity with a Unix-based system, and the ability to deal with various types of input files in different formats. To avoid overwhelming the students, we control NAMD by using the VIKING web interface. VIKING asks the user for several essential inputs such as temperature, number of steps, and time step duration while also allowing for more advanced input depending on the user's confidence. The molecular visualization of the output can be performed in both VIKING and PyMOL.

VIKING also handles scheduling of MD simulations on the computer clusters linked to VIKING. Some details about the practical aspects of the interface between VIKING and a computer cluster are discussed later. VIKING can automatically calculate the mean of the total energy $\langle E \rangle$ and the mean of the total energy squared $\langle E^2 \rangle$. Using these numbers, students can promptly calculate the heat capacity by using Eq. (2). Students can also download trajectory files, which describe the dynamical behavior of the molecules, and visualize the trajectories by using PyMOL or Visual Molecular Dynamics (VMD) (13).

V. DETAILED WORKFLOW

In the following sections, we illustrate in detail the workflow that the students experience. The class, typically 25 students, is divided into groups of 2–3 students. Each group is assigned to perform simulations at one temperature value to reconstruct the temperature dependence of the heat capacity curve, ensuring that the three main regimes of the unfolding transition (folded state, unfolding transition, unfolded state) are captured. Individual groups upload their data on a shared spreadsheet. The compiled data in the document can then be used to make the final calculations and construct a report. Each group submits an individual report.

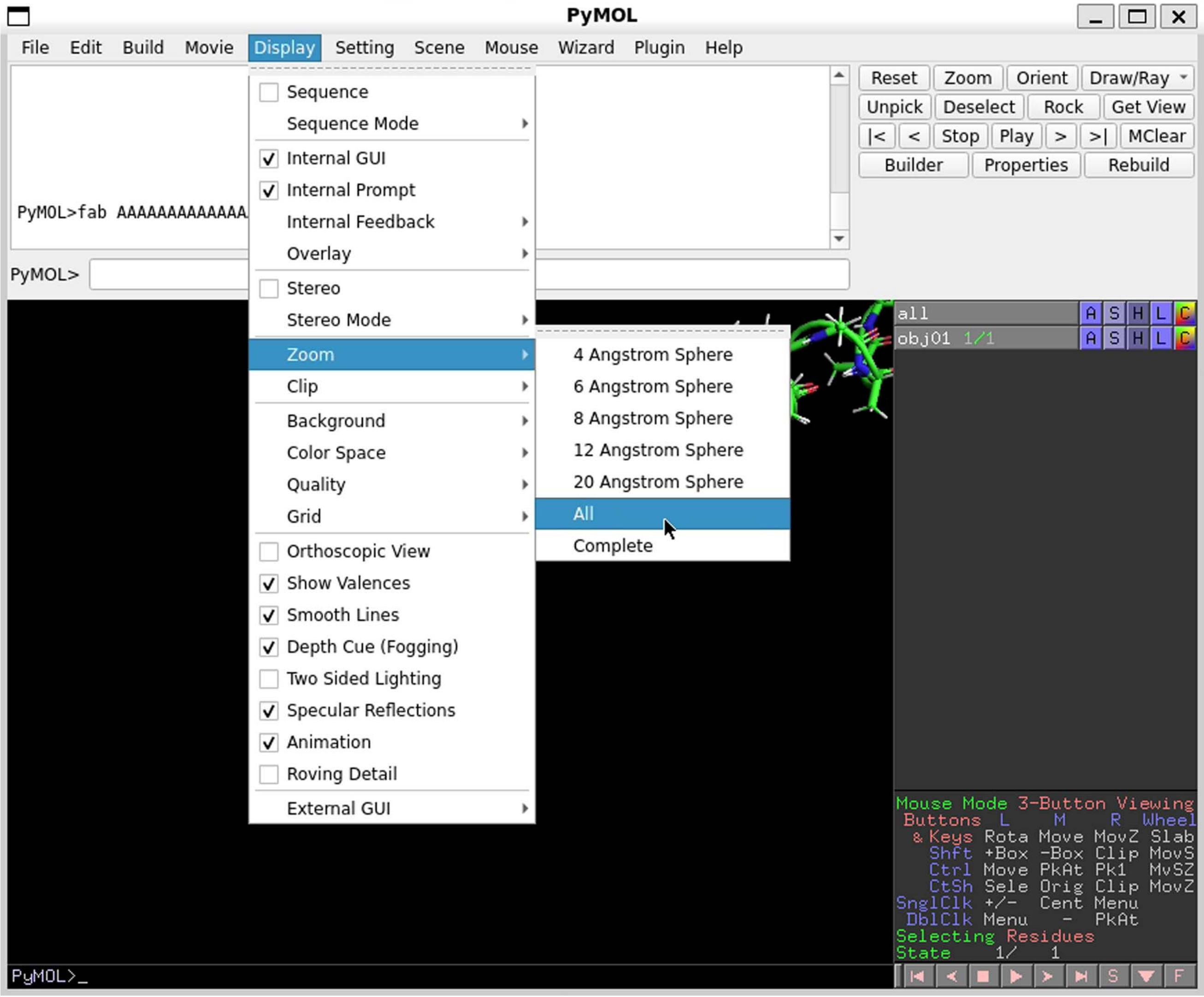
A. Generation of initial coordinates by using PyMOL

The coordinates of the poly-alanine Ala$_{30}$ molecule can be created by following these steps:

1. Open PyMOL.
2. On the menu bar at the top of PyMOL, select *Build* then *Residue* and finally *Helix*.
3. Now enter *fab AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, ss=1* in the PyMOL command input and hit enter, at which point a poly-alanine Ala$_{30}$ molecule should appear in the view.
4. To center the newly created structure in the view window, select *Display*, *Zoom*, and finally *All* from the top menu bar, as seen in Fig 2.
5. The representation of the structure can be changed by finding the *ala* or *obj01* molecule in the object overview on the right of PyMOL and clicking the *A* and selecting *preset* and finally *pretty* as seen in Fig 3.
6. The coordinates of the molecule can now be exported to the PDB format by selecting *File* and *Export Molecule* from the top menu. A new window should open. In this window, set the *Selection* drop-down to *ala* or *obj01* and click *Save*. Select a location to save the file, give it a name, and set the file type to *PDB*.
7. To check that the exported file represents the appropriate molecule, open it with any text editor and verify that it contains 300 atoms corresponding to 30 residues. Make sure not to edit the file, to avoid corrupting it.

B. Test simulation with VIKING

Now that the coordinates of the poly-alanine Ala$_{30}$ molecule are ready, the simulation can be started. The following steps illustrate how this can be accomplished by using VIKING. Note that refreshing the web page while setting up

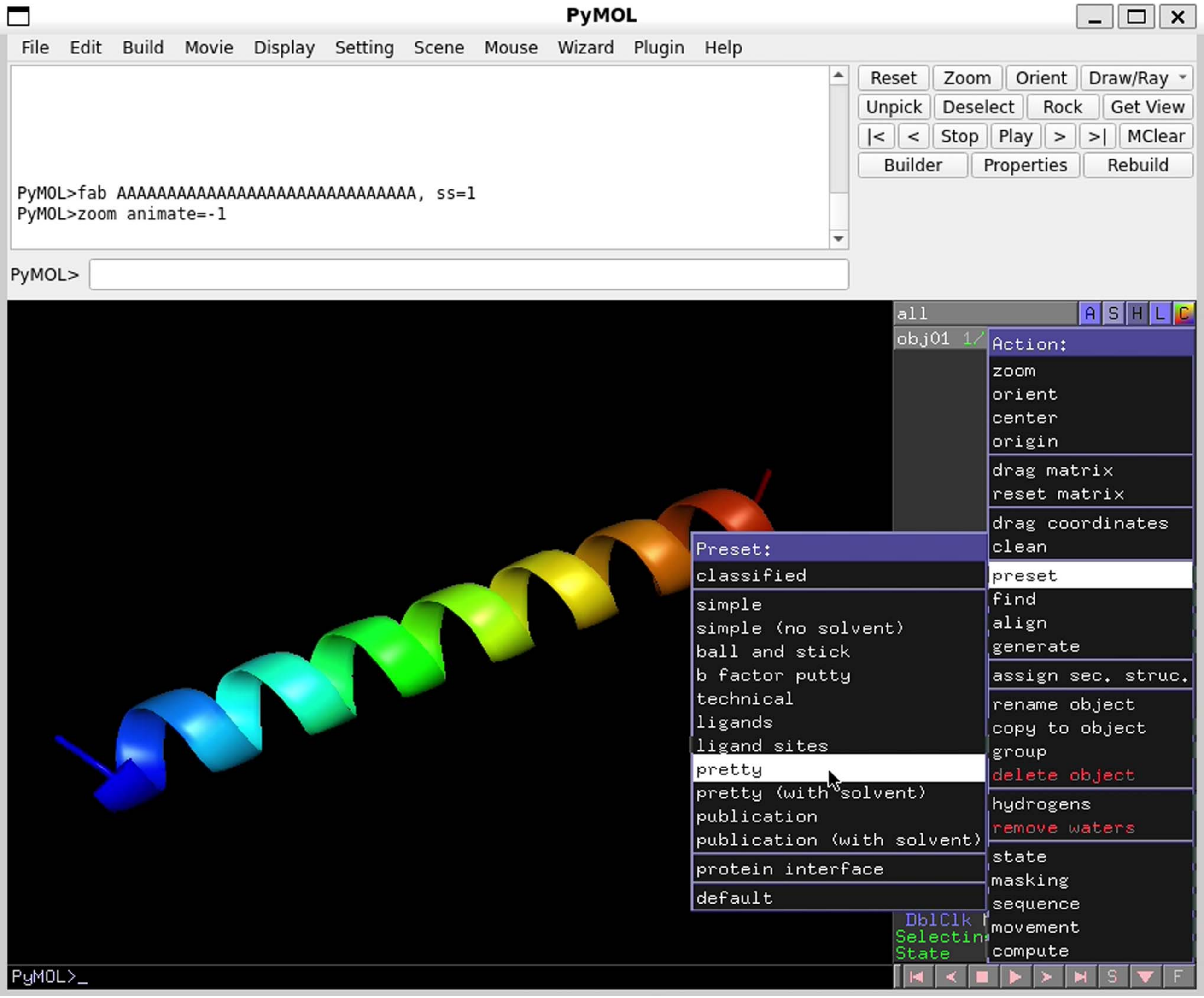


```
PyMOL>fab AAAAAAAAAAAA
```

Fig 2. Centering the structure in PyMOL.

the simulation may require the process to restart from the beginning.

1. Open a Web browser and navigate to https://viking-suite.com/, read the privacy policy, and accept the cookies. Enter the username and password supplied in the class in the sign-in form at the top of the web page.
2. Go to *Projects* and select the project with the temperature assigned to your group.
3. Create a new task by clicking the *New task* button above the project overview. Select *Molecular Dynamics* and then *Equilibrium MD*. Name the tasks (e.g., *MD simulation, Ala30, 450K*). Go to the next step by clicking the next button on the bottom right.
4. Click the *new structure* button above the project overview, click the *Browse* button and select the PDB file exported from PyMOL, and finally click *Load*. The structure can be previewed by hovering the cursor over it in the project overview and clicking the three horizontal bars on the right. Go to the next step.
5. Deselect the *Water* and *Proteins* potentials at the top by clicking on them, and then select the *CHARMM27* potential by clicking on it under the *custom potentials in projects* at the bottom. Go to the next step.
6. Choosing a solvent: Select - *None* - because the poly-alanine Ala_{30} molecule does not

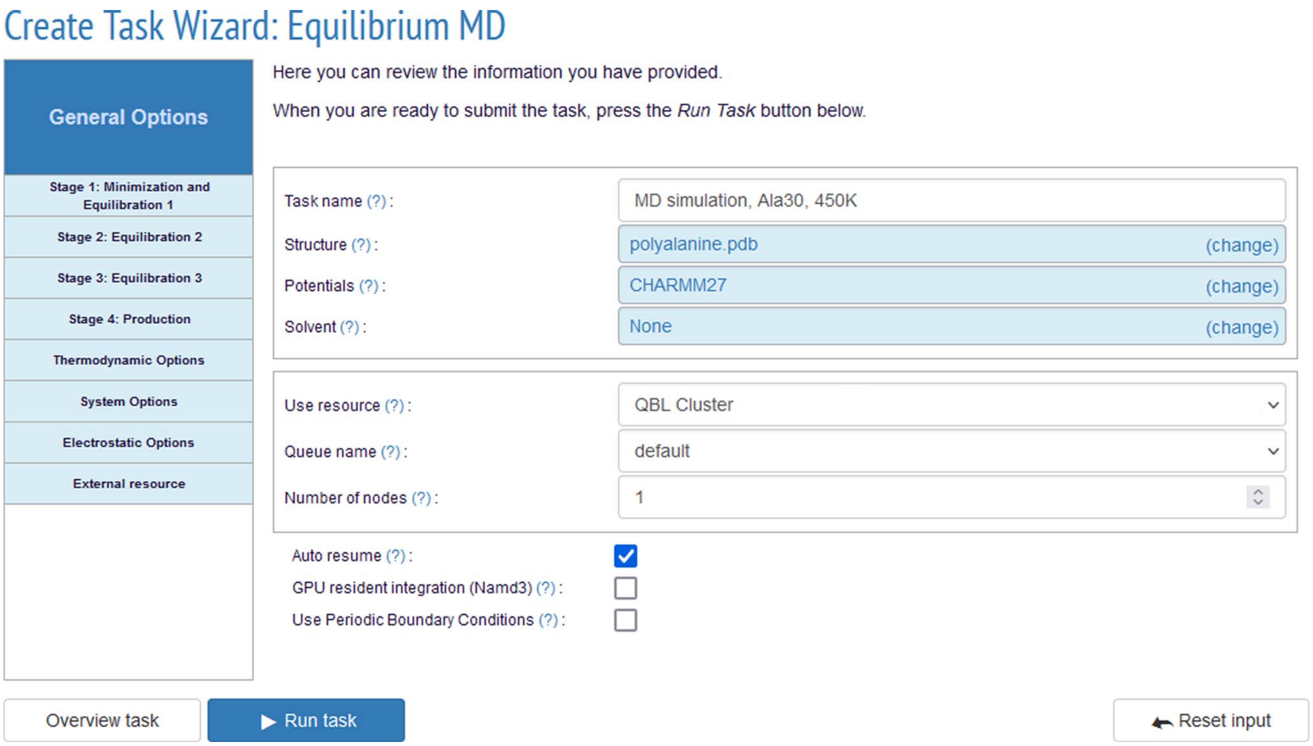


```
PyMOL>fab AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, ss=1
PyMOL>zoom animate=-1
```

Fig 3. Changing the visual representation of structures in PyMOL.

need to be immersed in a solvent. Go to the next step.

7. Set the simulation time to *2 ns* for an in-class demonstration to ensure that the simulation is set up correctly. Before launching the test simulation, the teacher should check the simulation setup. Once this is done, go to the next step.
8. Set the temperature assigned to your group. Fast forward to the final step by pressing the double arrows to the left.
9. The screen that emerges provides an overview of all the simulation parameters; at the bottom, uncheck the *Use Periodic Boundary Conditions* check box (Fig 4).
10. For this simulation, only two equilibration and production run stages are required. To turn off stages 1 and 2, go to their pages by clicking them at the left panel and unchecking *Do stage 1* and *Do stage 2* at the top of each page.
11. To speed-up the simulation conclusion for the in-class experiment, the final equilibration stage is also disabled. Navigate to the configuration page of stage 3 and uncheck *Do Stage 3* at the top of the page. For the longer simulation, stage 3 should be executed. The structure generated by using PyMOL must be minimized and equilibrated for the final simulation per standard



Create Task Wizard: Equilibrium MD

Here you can review the information you have provided.

When you are ready to submit the task, press the *Run Task* button below.

Task name (?) : MD simulation, Ala30, 450K

Structure (?) : polyalanine.pdb (change)

Potentials (?) : CHARMM27 (change)

Solvent (?) : None (change)

Use resource (?) : QBL Cluster

Queue name (?) : default

Number of nodes (?) : 1

Auto resume (?) : ☑

GPU resident integration (Namd3) (?) : □

Use Periodic Boundary Conditions (?) : □

Fig 4. Overview of the task parameters for the simulation in VIKING.

MD protocols. For this, set *Minimization Steps* to *100*, the *Simulation Length* to *1 ns*, *Restrain Hydrogen Bonds* to *all*. Here, 1 ns is the equilibration period (Fig 5).

12. Go to the overview of Stage 4 and check that the *Simulation Length* and *temperature* are set to the correct values.
13. Go to *Electrostatic options* and set *Cutoff Distance*, *Switching Distance*, and *Pairlist Distance* to *1000 Å*. Finally, un-check the *Use PME* option.
14. Finally, hit *Run task*, and the task will be created. The test simulation of 2 ns will be launched immediately if a reservation on the cluster computer has been made in advance.

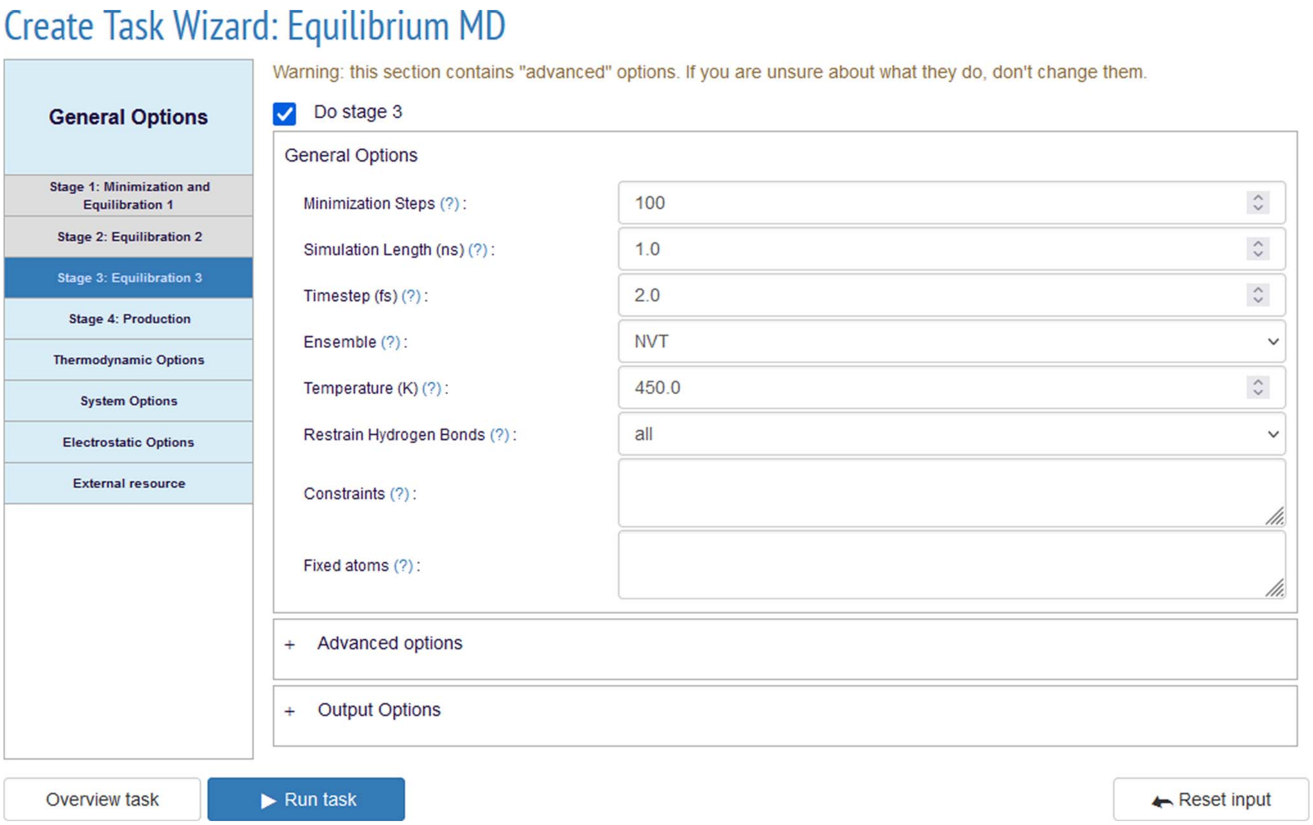
C. Verlet algorithm in class

The test simulation typically lasts ~20 min. During this period, students are asked to derive the Verlet algorithm from a Taylor series expansion (see the Supplemental Information for a copy of the instructions handed out to the students). An alternative is to arrange a visit for the students to the local super-computing center, which may be possible in some teaching environments.

D. Obtaining data from VIKING

When the in-class simulation is complete, the required data to construct the heat capacity can be extracted from the following steps:

1. Navigate to the VIKING projects and find your group's project, hover over the simulation task, and click the magnifying glass on the right. This should open up an overview page for the task, from which the task's status should be apparent. The data are ready when the *Status* says *done*.
2. Navigate to the *Results* tab on the top. Scroll down and click *Total energy*. This click will reveal the mean total system energy $\langle E \rangle$ and mean total system energy squared $\langle E^2 \rangle$. The heat capacity can now be calculated by using Eq. (2). Note that the numbers obtained during the simulation stage 4 should be used.
3. A trajectory file describing the dynamical behavior of the molecule during the simulation



Create Task Wizard: Equilibrium MD

Warning: this section contains "advanced" options. If you are unsure about what they do, don't change them.

☑ Do stage 3

General Options

Minimization Steps (?) : 100

Simulation Length (ns) (?) : 1.0

Timestep (fs) (?) : 2.0

Ensemble (?) : NVT

Temperature (K) (?) : 450.0

Restrain Hydrogen Bonds (?) : all

Constraints (?) :

Fixed atoms (?) :

Fig 5. Parameters for the equilibration simulation stage.

can also be downloaded. This is done by navigating to the *Files* tab at the top of the window. First, download the *psfgen.psf* file by hovering over it and clicking the arrow pointing down on the right.

4. Now click the *stage4* folder and then the *output* folder within it. Hover over the *stage4_1.dcd* file and click the arrow pointing down on the right to download it.

E. Visualizing data in PyMOL

The *psfgen.psf* and *stage4_1.dcd* files downloaded from VIKING permit visualizing the dynamical development of the poly-alanine Ala_{30} molecule in PyMOL.

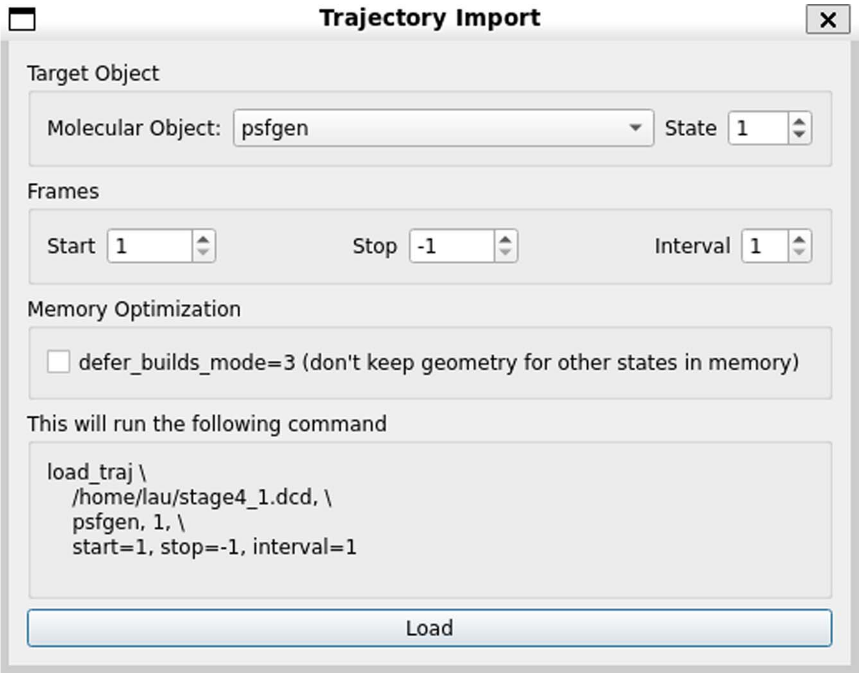
1. First open PyMOL and select *File* and *Open* . . . . Navigate to the *psfgen.psf* file downloaded from VIKING, select it, and press *Open*.
2. Select *File* and *Open* . . . again and navigate to the *stage4_1.dcd* file and select *Open*, which opens a new window shown in Fig 6. Click *load*.
3. Center the structure and set the representation to *pretty* as done previously in Section V.A. The dynamic evolution of the poly-alanine Ala_{30} molecule can now be seen by pressing the *play* button at the bottom of the object overview at the right (Fig 7).

F. Uploading data to the shared spreadsheet

The heat capacity values obtained from the very short in-class simulation of 2 ns are rather inaccurate, as seen in Fig 8a. However, the students are still instructed to upload their data in the shared document to ensure that a preliminary, albeit inaccurate curve is obtained from the test simulations.

G. Launching the final production run on VIKING

Once the students and the teacher are confident that the in-class simulation has completed satisfactorily, the students are asked to repeat Steps A–F



Target Object

Molecular Object: psfgen State 1

Frames

Start 1 Stop -1 Interval 1

Memory Optimization

☐ defer_builds_mode=3 (don't keep geometry for other states in memory)

This will run the following command

```
load_traj \
   /home/lau/stage4_1.dcd, \
   psfgen, 1, \
   start=1, stop=-1, interval=1
```

Fig 6. Loading an MD trajectory file into PyMOL.

for a production run with a simulation length of 200 ns including the Stage 3 equilibration.

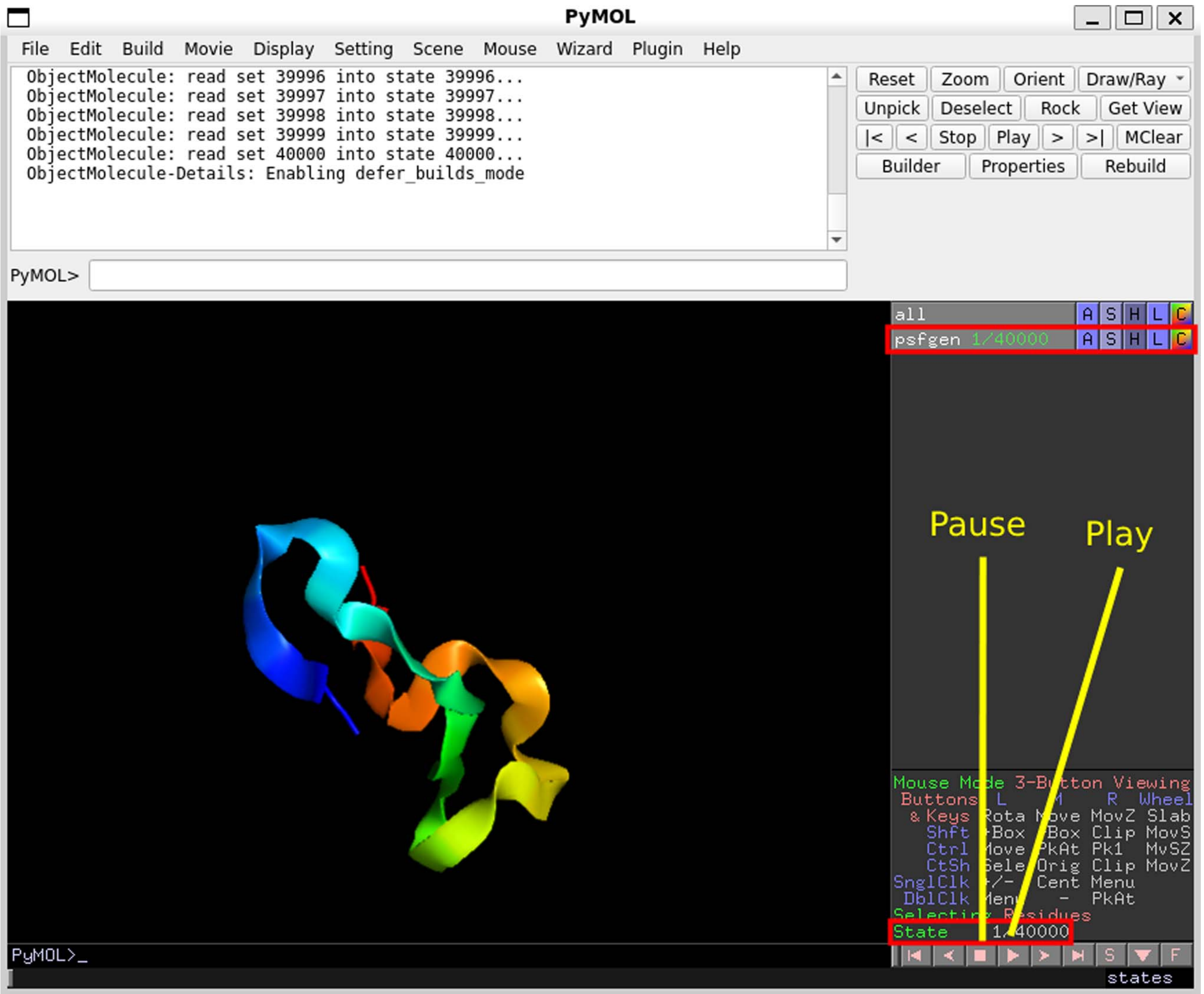
VI. RESULTS AND DISCUSSION

Student groups simulate the peptide at 8–12 different temperatures, depending on the number of students. In this article, we have performed the simulation 501 times for every integer step from 500 K to 1,000 K. Fig 8b shows the heat capacity calculated for each temperature. Students are encouraged to superimpose their data onto a similar plot after their analysis is complete.

The temperature dependence of the heat capacity in Fig 8b shows the expected behavior, with a peak starting ~640 K as hydrogen bonds begin to break and the poly-alanine Ala_{30} molecule begins to unfold. The heat capacity has a maximum at 740 K, and the peptide reaches the fully unfolded state at 840 K. Illustrations of the poly-alanine Ala_{30} molecule taken from the simulation at 580 K and 990 K (Fig 9) demonstrate that the peptide is unfolded at the higher temperature. Refer to the Supplementary Material for an animation showing the correlation between the temperature dependence of the heat capacity and the conformation of the peptide.

The heat capacity calculated from the simulations is in good agreement with earlier studies (3). Still, the rather large fluctuations at low temperatures indicate that more extended simulations may be necessary for a better sampling. From a pedagogical perspective, the shortcoming of limited sampling times should certainly be made clear to the students. However, using additional computer time to decrease the magnitude of the fluctuation in the heat capacity provides no further physical insight to the students.

The simulation of the peptide is performed without a solvent to minimize computational costs associated with running a laboratory session. The results would markedly differ and be more realistic if the simulations were performed in a solvent such as water, because particles will interact differently in an explicit solvent. For example, the hydrophobic effect and electrostatic screening due to polarization of water are not captured in the current setup. These effects are further exaggerated by the lack of calibration of the coupling to the heat bath, which results in a mismatch of the timescale of the unfolding transition and temperatures at which the transition happens. Usually folded proteins unfold in water upon heating beyond 340 K. For


PyMOL
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
Reset Zoom Orient Draw/Ray
Unpick Deselect Rock Get View
|< < Stop Play > >| MClear
Builder Properties Rebuild
PyMOL>
all A S H L C
psfgen 1/40000 A S H L C
Pause
Play
Mouse Mode 3-Button Viewing
State 1/40000
S F
states

```
ObjectMolecule: read set 39996 into state 39996...
ObjectMolecule: read set 39997 into state 39997...
ObjectMolecule: read set 39998 into state 39998...
ObjectMolecule: read set 39999 into state 39999...
ObjectMolecule: read set 40000 into state 40000...
ObjectMolecule-Details: Enabling defer_builds_mode
```

Fig 7. Playing a loaded trajectory in PyMOL.

a discussion of the folding timescale of proteins, see, for example, Kubelka et al. (14). However, the concepts and the physics of calculating a heat capacity of a phase transition is in no way undermined by using no solvent (9).

There are various theoretical approaches for modelling the helix to coil phase transition (15, 16). Although such statistical mechanical models capture the essence of the transition, the laboratory is not targeted to students with an advanced working knowledge of statistical mechanics. In addition, the purely theoretical approach will not serve as an introduction to biological phase transitions or to the coupling between microscopic and macroscopic properties. The proposed MD simulations demonstrate the coupling as well as introduce the use of computer simulations as a research tool. The coupling is achieved because the students have the ability to peer into the microscopic behavior of the Ala$_{30}$ molecule at any point in time and link it directly to a macroscopic property (i.e., the heat capacity).

A. Practical aspects on implementation in VIKING

Performing a task through the online platform VIKING requires access to a computing resource with available computation time. VIKING is an easy-to-use environment requiring some features to be automated before usage. Because every computing cluster has slightly different

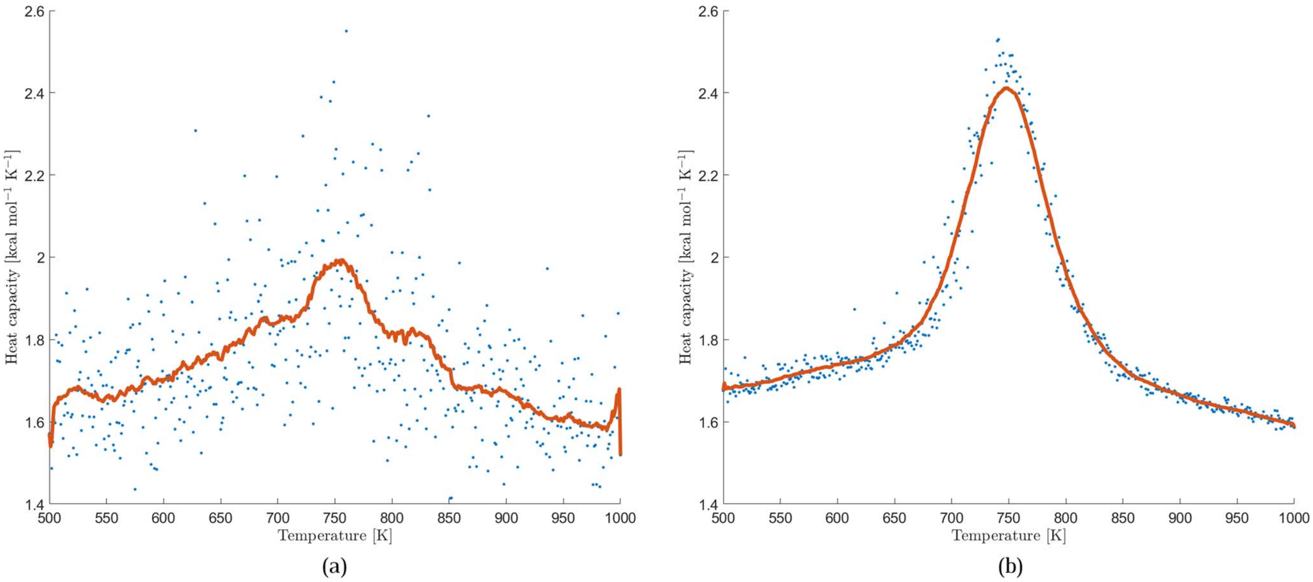


Fig 8. Temperature evolution of the heat capacity of Ala$_{30}$. The dots correspond to the values calculated from each individual simulation. The solid line shows a running average of the heat capacity. In (a), 2-ns simulations (500 snapshots) were used, corresponding to the in-class test simulation. In (b), the entire 200-ns production simulations were used.

settings, modules, and security guidelines, linking the available cluster to the VIKING framework requires contact with the VIKING support and development team, which is undertaken by the research group of Prof. Dr. Ilia A. Solov'yov at the Carl von Ossietzky University in Oldenburg. Because the procedure requires initial access to the computing resource, setting up the connection is a joint effort between the development team, the user, and respective IT services. Furthermore, a user identifier can be added upon request to hide the availability of the new resource from arbitrary VIKING users not involved in the class.

Once the computing cluster is added to the VIKING interface, the available software on the cluster is checked for all tasks to allow a seamless integration of all job types.

In the end, VIKING will access the cluster as a representative of the user, queueing the task and monitoring the task's progress to then retrieve the files by providing the download link in the browser. This process allows a user to run, for instance, the MD simulation without any knowledge of establishing a secure shell connection and task queueing systems used by the computing resource.

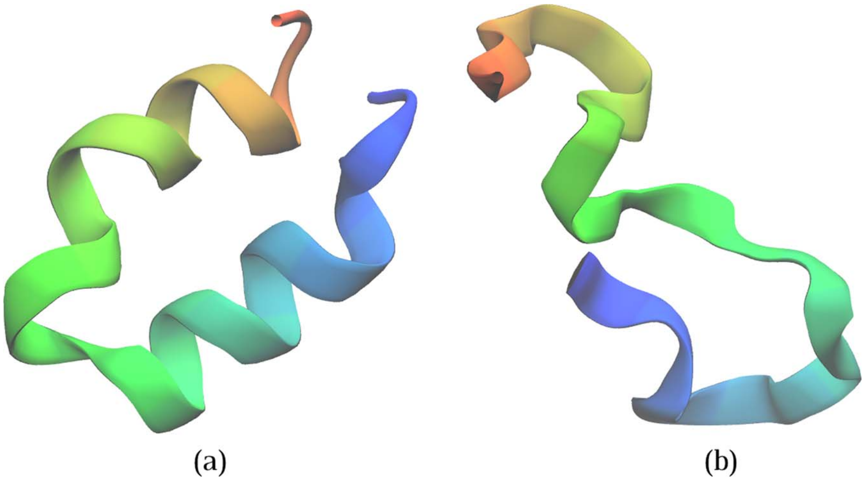


Fig 9. Final simulation snapshot of Ala$_{30}$ taken from the simulation at (a) 580 K and (b) 990 K.

VIKING can be linked to any computing cluster that does not require the end-user to be logged in to a VPN to access the resource. Computing resources using a double-authentication security system may also be added. The authentication token will have to be entered through VIKING every time that data are meant to be retrieved from the resource.

To get access to the VIKING platform, the following steps should be taken:

1. Obtain access to a computing resource satisfying the above restrictions.
2. Contact the VIKING support team through the online form found at https://perpevit.com/contact-us. Include a basic description of the computing resource and your intent to use VIKING for carrying out this computer laboratory exercise. You may contact the VIKING team before getting access to a computer resource, if you are interested in a tour of the system.
3. Upon receiving your inquiry, the VIKING support team will reach out to you such that authentication details for the computing resource and the VIKING platform can be exchanged. Here, the queueing manager and the availability of installed software are discussed. On some computing resources, it might be required to install, for example, NAMD, directly.
4. The computer resource will be added to the VIKING platform, and the VIKING support team will make sure that all programs can be accessed as needed. After this step, the VIKING platform will be able to verify the programs automatically.
5. The VIKING platform is now ready to run simulations on the computing resource, and the computer laboratory exercise can be carried out as described in Section V.

The completion time of setting up the computing resource can vary on the basis of the computing resource and the software needs. In the simplest case, in which, for instance, all programs can be loaded as modules, setting up the VIKING connection can typically be completed within a month.

B. Running simulations without VIKING

It is also possible to run the computer laboratory exercise on student laptops without using the VIKING platform. However, this requires the students to be somewhat familiar with the command-line interface (terminal) on their operating system of choice. The implementation of the simulation into a computer laboratory exercise will be considerably different from what was described in Section V. Running the simulations without VIKING will make the students familiar with the workings of the simulation software and their own computer. We have made the files required for running the simulations available at https://bit.ly/3Pv4OvW. On this page, a detailed step-by-step guide for running the simulations on a personal macOS, Windows, or Linux computer can be found. We have also added a tool on this page that can be used to calculate the mean energy and mean energy squared from the NAMD output file.

VII. CONCLUSION

The simulation laboratory presented here predicts the heat capacity of the folding-unfolding transition of a single poly-alanine Ala_{30} molecule. The computer laboratory exercise gives the students molecular insights into a biophysical system and the molecular and macroscopic definitions of the heat capacity of a first-order phase transition. The simulation laboratory cannot replace a conventional laboratory where students can, for example, investigate the phase transition of a lipid bilayer by using a differential scanning calorimeter. However, we are confident that the computational calculation provides molecular and statistical insights into a simple concept from physical chemistry, which is impossible to obtain from a macroscopic experimental laboratory alone.

As extensions of the laboratory, students can investigate the temperature and width of the phase transition as a function of peptide length.

Combined with a graphical user interface of VIKING, the simulation also serves as a good introduction for students to working with MD simulations that plays an important role in modern biophysical education and research.

AUTHOR CONTRIBUTIONS

HK and IAS conceptualized the laboratory exercise. HK implemented the laboratory in the course BMB540: Physical Biochemistry at SDU. DB and VD wrote the lab manuals and developed the laboratory with HK. LBG wrote the first draft of the manuscript and ran all the calculations. FS and IAS implemented the simulations in VIKING. All authors contributed to the final draft.

ACKNOWLEDGMENTS

HK is funded by a Lundbeck Ascending Investigator grant #R344-2020-1023. Part of the numerical results presented in this work were performed on the Novo Nordisk Foundation-funded ROBUST Resource for Biomolecular Simulations #NNF18OC0032608.

Part of the computation done for this project was performed on the UCloud interactive HPC system, which is managed by the eScience Center at the University of Southern Denmark with support from the Danish e-Infrastructure Cooperation.

The authors thank the Volkswagen Foundation (Lichtenberg Professorship to IAS), the Deutsche Forschungsgemeinschaft (DFG; GRK1885: Molecular Basis of Sensory Biology, SFB 1372: Magnetoreception and Navigation in Vertebrates, and TRR386/1 – 2023 HYP*MOL, no. 514664767), the Ministry for Science and Culture of Lower Saxony (Simulations Meet Experiments on the Nanoscale: Opening up the Quantum World to Artificial Intelligence (SMART) and Dynamik auf der Nanoskala: Von kohärenten Elementarprozessen zur Funktionalität (DyNano)). Computational resources for the simulations were provided by the CARL Cluster at the Carl-von-Ossietzky University, Oldenburg, supported by the DFG and the Ministry for Science and Culture of Lower Saxony. The authors gratefully acknowledge the computing time granted by the Resource Allocation Board and provided on the supercomputer Lise and Emmy at NHR@ZIB and NHR@Göttingen as part of the NHR infrastructure. The calculations for this research were conducted with computing resources under the project nip00058.

The authors declare no competing interests.

REFERENCES

1. Korol, V., P. Husen, E. Sjulstok, C. Nielsen, I. Friis, A. Frederiksen, A. B. Salo, and I. A. Solov'yov, 2020. Introducing VIKING: a novel online platform for multiscale modeling. *ACS Omega* 5:1254–1260.
2. Phillips, J. C., D. J. Hardy, J. D. C. Maia, J. E. Stone, J. V. Ribeiro, R. C. Bernardi, R. Buch, G. Fiorin, J. Hénin, W. Jiang, R. McGreevy, M. C. R. Melo, B. K. Radak, R. D. Skeel, A. Singharoy, Y. Wang, B. Roux, A. Aksimentiev, Z. Luthey-Schulten, L. V. Kalé, K. Schulten, C. Chipot, and E. Tajkhorshid, 2020. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J Chem Phys* 153:044130.
3. Solov'yov, I. A., A. V. Yakubovich, A. V. Solov'yov, and W. Greiner, 2008. α-helix ↔random coil phase transition: analysis of ab initio theory predictions. *Eur Phys J D* 46:227–240.
4. Livi, R., and P. Politi, 2017. Nonequilibrium Statistical Physics. Cambridge University Press, Cambridge, UK.
5. Brooks, B. R., C. L. Brooks III, 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, and M. Karplus, 2009. CHARMM: the biomolecular simulation program. *J Comput Chem* 30:1545–1614.
6. Swope, W. C., H. C. Andersen, P. H. Berens, and K. R. Wilson, 1982. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters. *J Chem Phys* 76:637–649.
7. Gasic, A. G., M. M. Boob, M. B. Prigozhin, D. Homouz, C. M. Daugherty, M. Gruebele, and M. S. Cheung, 2019. Critical phenomena in the temperature-pressure-crowding phase diagram of a protein. *Phys Rev X* 9:041035.
8. Schrödinger, LLC, 2015. The PyMOL Molecular Graphics System, Version 1.8.
9. Yakubovich, A., 2011. Theory of phase transitions in polypeptides and proteins. Springer Theses. Springer Berlin Heidelberg.
10. Kuriyan, J., B. Konforti, and D. Wemmer, 2012. The Molecules of Life: Physical and Chemical Principles. W. W. Norton, New York.
11. Abraham, M. J., T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, 2015. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1–2:19–25.
12. Thompson, A. P., H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, 2022. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput Phys Commun* 271:108171.
13. Humphrey, W., A. Dalke, and K. Schulten, 1996. VMD – visual molecular dynamics. *J Mol Graph* 14:33–38.
14. Kubelka, J., J. Hofrichter, and W. A. Eaton, 2004. The protein folding "speed limit." *Curr Opin Struct Biol* 14:76–88.
15. Vitalis, A., and A. Caflisch, 2012. 50 Years of Lifson-Roig Models: Application to molecular simulation data. *J Chem Theory Comput* 8:363–373.
16. Yakubovich, A. V., I. A. Solov'yov, A. V. Solov'yov, and W. Greiner, 2008. Ab initio theory of helix ↔coil phase transition. *Eur Phys J D* 46:215–225.