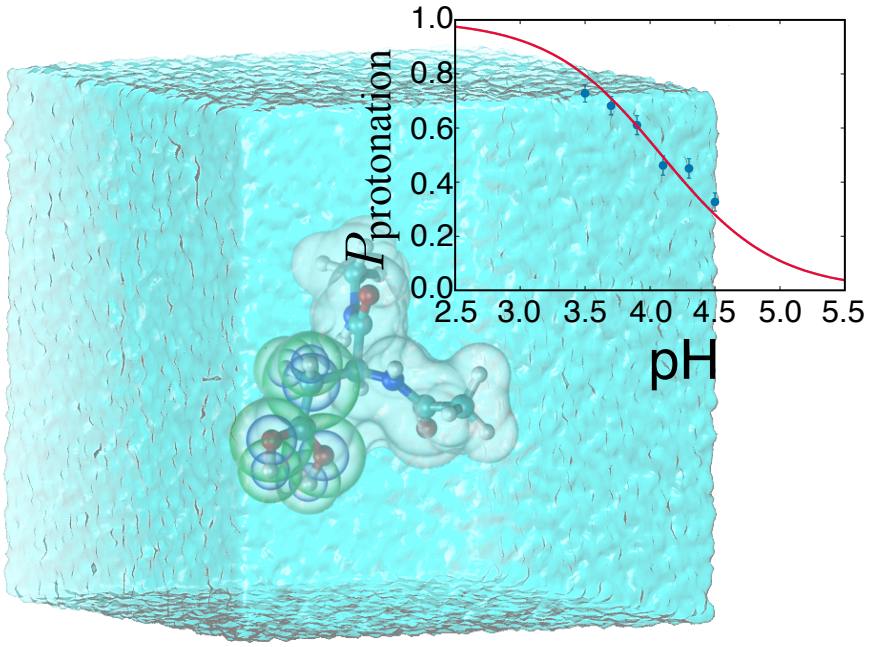
Leadership Computing Facility
Argonne National Laboratory

Department of Biochemistry and Molecular Biology
Gordon Center for Integrative Science
The University of Chicago

Centre National de la Recherche Scientifique
Laboratoire International Associé CNRS-UIUC
Université de Lorraine

University of Illinois at Urbana-Champaign
Beckman Institute for Advanced Science and Technology
Theoretical and Computational Biophysics Group

Simulating Biomolecules with Variable Protonation State: A Tutorial for Constant-pH Molecular Dynamics



Brian K. Radak
Benoît Roux
Christophe Chipot

September 4, 2018

Please visit www.ks.uiuc.edu/Training/Tutorials/ to get the latest version of this tutorial, to obtain more tutorials like this one, or to join the `tutorial-l@ks.uiuc.edu` mailing list for additional help.

Abstract

This tutorial is intended to demonstrate the application of molecular dynamics simulations at constant-pH. This method addresses the critical problem of what to do when modeling systems (especially biomolecular systems) that can attain multiple protonation states. While other approaches are possible, using a constant-pH technique naturally accounts for protonation state variation with little or no input beyond the physical model (*i.e.,* the force field). Once completed, several constant-pH simulations can be analyzed to yield titration curves in exact analogy to what is observed in introductory chemistry and biochemistry courses. Of course, the advantage of all-atom models is that these curves also correspond to a clear microscopic picture of the system under study and thus permit direct analysis of distinct protonation sites without the need to deconvolute experimental spectra. The methodology is used here on two examples – a simple peptide with predictable, but illustrative behavior and the "BBL" miniprotein, a bacterial homologue of the peripheral subunit-binding domain family. By analyzing the titration curves of both systems it is possible to compute estimated apparent pK_a values for a broad range of titratable amino acid residues and also analyze the correlations between residues.

Contents

1. Introduction

The intention of this tutorial is to acquaint the user with constant-pH molecular dynamics (MD), a simulation method designed to consider chemical systems in which the overall protonation state can change, perhaps in several ways [1]. This is to be contrasted with conventional force-field based MD simulations, which generally treat protonation states by assuming they are fixed. Consider, for example, a protein with two titratable residues which may both be either protonated or deprotonated (Figure 1); the system has four possible protonation states. In the conventional route, the user must enumerate these possibilities, construct distinct topologies, and then simulate the cases individually. The simulations for each state must then be connected by either asserting knowledge about the system (*e.g.,* by assuming that only certain states are of biological importance) or by performing additional simulations to probe transitions between states directly (*e.g.,* by performing free energy calculations). In a constant-pH MD simulation, knowledge of the transformations is not assumed and is instead actively explored by interconverting between the various protonation states. This is especially useful when the number of protonation states is extremely large and/or prior information on the importance of particular states is not available.

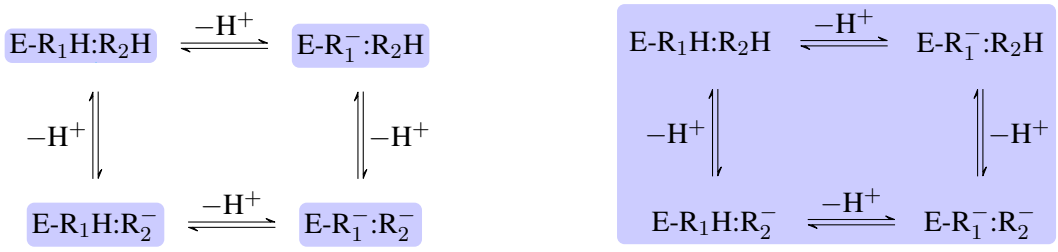


Figure 1: The core difference between conventional and constant-pH MD can be illustrated by a simple enzyme E with four protonation states describing the occupancy of two titratable residues, R_1 and R_2. A conventional MD simulation handles the states *separately* (left panel). The relative importance of the states must be known beforehand or computed by other means. Conversely, a constant-pH MD simulation handles the states *collectively* and actively simulates interconversion (right panel). Determining the relative importance of the states is a direct result of the simulation.

The history of constant-pH simulations is relatively short compared to conventional MD, nonetheless, several developments have taken place over the last few decades [1–10] and the approach is suitably mature as to be a mainstream, albeit state-of-the-art, technique. The developments leading to this tutorial follow from several recent advances in the area of hybrid nonequilibrium MD/Monte Carlo (neMD/MC) [1, 5, 10–13]. The neMD/MC approach is actually quite general but, as will be seen, is especially amenable to the simulation of variable protonation states in explicit solvent. Although the interested reader is invited to survey the literature on other routes to simulations at constant pH, the neMD/MC based approach is the only one currently implemented in NAMD and aspects of the discussion may or may not translate usefully to other methods.

Completion of this tutorial requires

- the various files contained in the archive `tutorial-constant-ph.tar.gz` located at `www.ks.uiuc.edu/Training/Tutorials/namd/tutorial-constant-ph`;
- NAMD 2.12 or later (`www.ks.uiuc.edu/Research/namd`);
- a working Python and NumPy distribution (for analysis only)
- PyNAMD (`https://github.com/radakb/pynamd`, for analysis only).

1.1. Theoretical Background

In formal terms, conventional MD samples from a canonical ensemble, whereas constant-pH MD samples from a semi-grand canonical ensemble. The new partition function,

$$\Xi(\mathrm{pH}) = \sum_{\boldsymbol{\lambda}\in\mathcal{S}} Q_{\boldsymbol{\lambda}} 10^{-n_{\boldsymbol{\lambda}}\mathrm{pH}}, \tag{1}$$

is essentially a weighted summation of canonical partition functions, $Q_{\boldsymbol{\lambda}}$, each of which are defined by an occupancy vector, $\boldsymbol{\lambda}$. The elements of $\boldsymbol{\lambda}$ are either one or zero depending on whether a given protonation site is or is not occupied, respectively. For a vector of length m, the set of all protonation states, $\mathcal{S}$, has at most 2^m members. In order to sample from the corresponding semi-grand canonical distribution function, a simulation must explore *both* the phase space defined by the canonical paritition functions and the state space defined by the different occupancy vectors. The fraction of simulation time spent in each state is dictated by the weights in the summation and these depend on the pH and the number of protons, $n_{\boldsymbol{\lambda}}$, in the system (*i.e.*, the sum of the elements in $\boldsymbol{\lambda}$).

1.1.1. Hybrid neMD/MC

The basic scheme in NAMD is to alternately sample the protonation state and then the configuration space within that state [1]. Protonation state sampling is accomplished by the aforementioned neMD/MC protocol, whereby a proton that is non-interacting (*i.e.,* its site is empty) has its interactions forcibly turned on (or vice-versa) in a time-dependent manner (*i.e.,* over the course of a short trajectory). The result of this trajectory then enters into an accept/reject criterion.

At this point there are two important conceptual changes with respect to conventional MD that are worth emphasizing. First, rather than being a continuous trajectory, a constant-pH simulation is a series of cycles composed of an MD and neMD/MC step (Figure 2). This means that the length of the simulation is no longer simply determined by the number of steps (`numsteps`) but rather the number of cycles. The

length of a cycle is in turn determined by two parts – the number of steps spent on equilibrium sampling and the number of steps for executing the switch. The second change is that the neMD/MC move, just as any MC procedure, can result in rejected states and configurations. This means that a switch can "fail" and cause the trajectory to be discarded (see the red "X" in Figure 2).

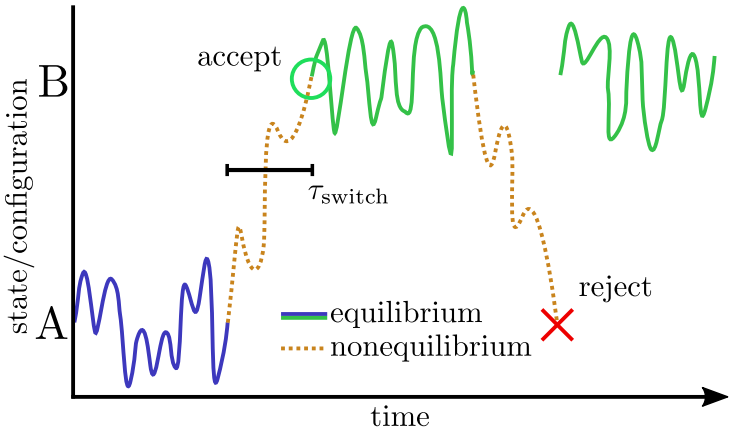


Figure 2: The basic constant-pH MD scheme in NAMD is to alternate equilibrium sampling in a fixed protonation state followed by a neMD/MC move to sample other protonation states. The latter move can be accepted or rejected. If accepted, the simulation continues in the new protonation state. If the move is rejected, sampling continues as if the move were never attempted at all.

The main challenge in running a successful constant-pH simulation with NAMD is to suitably adjust the relevant time parameters, especially the switch time, τ_{switch} (Figure 2). That is, a choice must be made with regards to how much time is spent sampling configurations in the usual way and how much time is spent sampling new protonation states (although this also samples configurations). If a switch is too short, then the move is likely to be too sudden and is unlikely to be accepted; effort will be wasted when the move is rejected. However, if a switch is too long, then an inordinate amount of effort will be spent sampling the state space and there will be fewer resources left for exploring the configuration space. Some basic qualities of the system that affect sampling have been determined using nonequilibrium linear response theory [14]. In short, there are intrinsic limits based on:

1. the extent that differing interactions between each state fluctuate (according to some variance, σ_0^2)
2. the "molecular" time scale, τ_{m} on which these fluctuations change.

These effects are roughly captured by the expression [14]:

$$\tau_{\text{opt}} \leq \frac{\sigma_0^2 \tau_{\text{m}}}{2.83475}, \tag{2}$$

where τ_{opt} is some optimal switching time, in the sense of maximizing the rate at which protonation states interconvert. Overall, switching times on the order of tens of picoseconds tend to be optimal in that they

balance the high cost of switching versus the high acceptance rate at longer switching times (in the infinite time limit the perturbation is adiabatic and exerts zero work). For titratable groups exposed primarily to aqueous solvent, a switch on the order of 10-20 ps appears to give near optimal results [1, 14]. An equivalent formulation of the above expression is that mean acceptance rates around 20-25% are likely near optimal.

1.1.2. The Two-step Inherent pK_a Algorithm

The astute reader may have noticed that the neMD/MC moves used to sample protonation states can be considerably expensive, especially when the rejection rate is high. This can be especially problematic for transitions to *rare* protonation states. For example, one might hardly expect aspartate ($pK_a \sim 4$) to be found in a deprotonated state at low pH (around 2, say). Attempting to sample a transition from a protonated state to a deprotonated state at such low pH, at least with high frequency, would be the height of foolishness; nearly all of the moves will be rejected and considerable resources will have been wasted.

A profitable solution to this problem is to pre-screen transitions based on the pH and a reasonable estimate for pK_a value of each residue (what will be referred to here as the "inherent" pK_a). This approach, first proposed by Chen and Roux [10] naturally shifts the majority of effort towards switching trajectories that are the most likely to be accepted. It also permits updates of the inherent pK_a values without disrupting the final statistics in any way. It can also increase the rate at which switches are accepted at the expense of decreasing the rate at which they are proposed. This last part can be subtle – the core of the algorithm is to examine the current protonation states of the system globally and then propose transitions using a pH dependent Metropolis criterion. This process is inexpensive and extremely efficient, especially when the estimates for the inherent pK_a values are accurate.

Of course, when inherent pK_a values are inaccurate, performance can also suffer. For example, there are well-documented cases of lysine residues achieving fairly low pK_a values near neutrality. However, a naive estimate for their inherent pK_a values might be closer to 10, which would result in a severe under sampling of such residues. Probably the most effective solution to this problem is to always sample a broad range of pH values, at least across a range of four pH units. When in doubt, the algorithm can also be effectively deactivated by setting the inherent pK_a of a residue equal to the pH at which it is being sampled.

1.2. Important Details of the Implementation

There are a number of specific choices made in the development of constant-pH MD in NAMD that could be done differently with little or no effect on the resulting simulations. Unfortunately, these de-

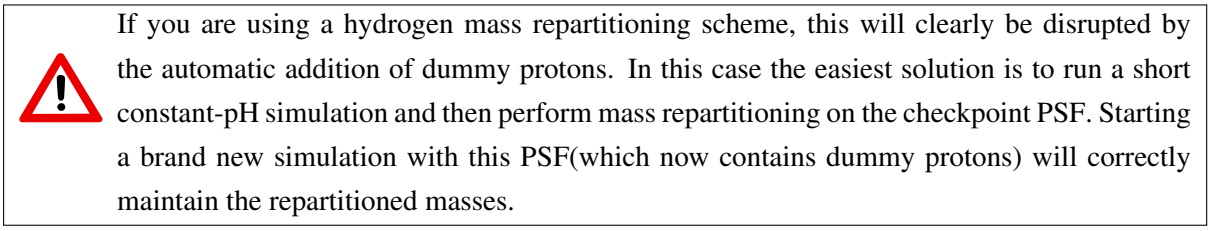
cisions largely have to do with file formats and visualization and thus are of direct concern to the user. Most importantly, these decisions result in additional `parameter` and `topology` files which provide additional information beyond that usually used to specify a force field. Much of this information can be derived in a concerted, albeit complicated, fashion. However, the details of this procedure are inessential at present – we will only describe the purpose of these additional files from the perspective of how they make constant-pH MD different from conventional MD. For the CHARMM36 protein force field [15–17], these additional files are:

- `top_cph36_prot.rtf` – additional topology information
- `par_cph36_prot.prm` – additional parameter information
- `conf_cph36_prot.json` – meta data for additional topology and parameter information

1.2.1. Dummy Protons

As per any (semi-)grand canonical ensemble, the number of atoms in a constant-pH ensemble fluctuates across a distribution of values. However, rather than removing protons from the system they instead persist as "dummy" protons without the usual force field interactions. This is useful for two reasons. First, when writing coordinates, *etc.* the number of fields can remain fixed and thus files can be recognized by visualization and analysis codes as if they were from conventional MD. Second, the dummy positions and velocities become relevant during switching trajectories because the corresponding atom in the auxilliary representation may itself be interacting (see Figure 3). By defining a minimal set of bonded interactions for dummy protons, two things can be assured. First, the dummy proton will have positions and velocities that are physically plausible as if the proton were interacting. Second, the underlying thermodynamics of the system are unaffected, that is, there would be no statistical difference in the system if the dummy were instead removed. However, as on might intuit, the dynamics of the system *are* perturbed, although this difference is likely on par with perturbations from normal Newtonian dynamics due to the use of Langevin dynamics, say.

Dummy protons are *automatically* added to a system when a constant-pH simulation is *initialized* – no user input is required. However, when a simulation is *restarted*, the dummy protons are assumed to exist already and therefore must be present in the PSF, PDB, and binary files. Appropriate files are written whenever a constant-pH simulation is checkpointed (including when it finishes normally).



If you are using a hydrogen mass repartitioning scheme, this will clearly be disrupted by the automatic addition of dummy protons. In this case the easiest solution is to run a short constant-pH simulation and then perform mass repartitioning on the checkpoint PSF. Starting a brand new simulation with this PSF(which now contains dummy protons) will correctly maintain the repartitioned masses.

1.2.2. *Auxilliary Coordinates*

In NAMD, the nonequilibrium "switching" is accomplished with the alchemy code, which follows the "dual-topology" paradigm [18, 19], even though the constant-pH trajectory itself is represented by a normal "single-topology." During the course of the simulation transformation between these two representations will occur on-the-fly by using an auxilliary set of coordinates (these will not appear in the final trajectory). The user does not strictly need to be aware of the actual details of such transformations (see Figure 3), but many simulation parameters affecting efficiency can best be understood by knowing that they occur.

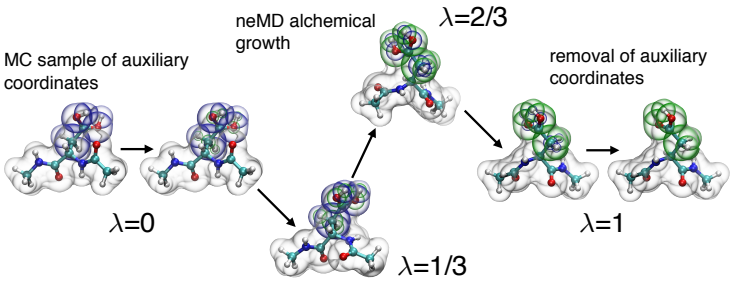
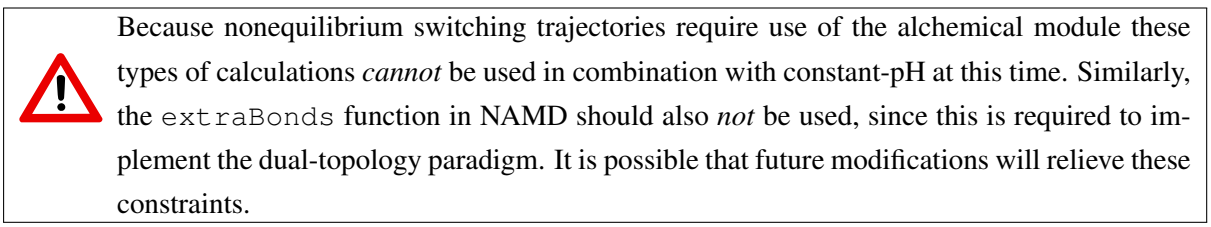


Figure 3: A switch move contains three main steps: 1) an exact MC sampling of auxilliary sidechain atoms, 2) neMD propagation of the coordinates and coupling constant λ as the original coordinates and state (blue spheres) decouple and the new coordinates and state (green spheres) become coupled, and 3) removal of the non-interacting atoms. If the neMD/MC move is rejected, then the simulation continues from the original coordinates.



Because nonequilibrium switching trajectories require use of the alchemical module these types of calculations *cannot* be used in combination with constant-pH at this time. Similarly, the `extraBonds` function in NAMD should also *not* be used, since this is required to implement the dual-topology paradigm. It is possible that future modifications will relieve these constraints.

1.2.3. *Restarting Simulations*

As mentioned in Section **1.2.1.**, restarting a constant-pH MD simulation requires some additional considerations compared to conventional MD. Normal NAMD behavior is to produce final coordinates, velocities, and extended system files using the `outputname` keyword as a base name. During constant-

pH MD this behavior is extended to also produce new PSF and PDB files with appropriate atom counts. *These files should be used when restarting, not the initial ones.* In addition, a constant-pH specific restart file with the extension `cphrst` is also produced and contains the relevant protonation state information (the final PSF information is accurate, but ignored during a restart). This nuance will be discussed in detail in the simulation examples and when discussing visualization.

1.2.4. Intermediate Files from `psfgen`

The implementation utilizes file-system based communication with `psfgen`, which requires the use of temporary intermediate files at various points during the simulation. Fortunately, the user does not need to know too much about these files other than where they are created. A default file name is used in the working directory in which NAMD is called; this is perfectly adequate for simple use cases without any adjustment. However, when running multiple simulations, especially at the same time, care must be taken to give these files unique names and/or locations so that file-system communication is not muddled between different runs. This would be very bad and will likely result in meaningless calculations. Lastly, these files should be written/read from the fastest available disk. On most supercomputing systems this is a scratch or tmp directory, rather than the user's home directory.

1.3. Constant-pH Output

In addition to minor changes to the usual NAMD output, constant-pH MD also produces a new output file to `[ouputname].cphlog`. This constant pH specific output tracks the occupancy vector $\boldsymbol{\lambda}$ at each cycle. The user need not, in general, actually look at these files, but the format is quite simple:

```
#pH <pH value>
#<segid:resid:resname> <segid:resid:resname> ...
<cycle> <lambda_1> <lambda_2> ...
```

The first line simply reports the pH at which the data were generated. The second line lists the identities of the titratable residues tracked during the simulation – these are listed in `psfgen` format, using the `segid` and `resid` as identifiers as well as the residue name. Since different residue types have different numbers of sites, the residue name is essential for identifying which sites belong to which residue. After the header information, the data is a simple time series with the cycle number in the leftmost column and then each element of $\boldsymbol{\lambda}$.

1.3.1. Computing and Analyzing Titration Curves

The output from the `cphlog` can be used to compute titration curves. The principle objective in doing this is usually to extract some kind of $\mathrm{p}K_\mathrm{a}$ value. Although all of the protonation sites in the system are free to titrate, a $\mathrm{p}K_\mathrm{a}$ generally only corresponds to a reaction in which one proton moves on and off of a single site. Returning to the semi-grand canonical partition function in Eq. (1), for any given site half of the terms in the summation include the site being occupied (Ξ_1) and half include the site as unoccupied (Ξ_0). By taking out a factor of $10^{-\mathrm{pH}}$, this can be re-written as

$$\Xi(\mathrm{pH}) = \Xi_0(\mathrm{pH}) + \Xi_1(\mathrm{pH})10^{-\mathrm{pH}}. \tag{3}$$

Now, suppose that the protonation site in question corresponds to the sth element in $\boldsymbol{\lambda}$, λ_s. Since λ_s can only take the values zero and one, taking the ensemble average amounts to "masking out" the terms where it is zero and only keeping those where it is one. In other words:

$$\langle \lambda_s \rangle_\mathrm{pH} = \frac{\Xi_1(\mathrm{pH})10^{-\mathrm{pH}}}{\Xi_0(\mathrm{pH}) + \Xi_1(\mathrm{pH})10^{-\mathrm{pH}}} = \frac{1}{1 + \frac{\Xi_0(\mathrm{pH})}{\Xi_1(\mathrm{pH})}10^{\mathrm{pH}}} \tag{4}$$

This can be recognized as a statistical mechanical form of the well-known Henderson-Hasselbalch equation with the equality

$$\mathrm{p}K_\mathrm{a}(\mathrm{pH}) = -\log \frac{\Xi_0(\mathrm{pH})}{\Xi_1(\mathrm{pH})}. \tag{5}$$

Note that we have, however, now added a pH dependence into the $\mathrm{p}K_\mathrm{a}$, which arises from the fact that additional sites beyond s are *also* coupled to the same pH bath. This dependence vanishes when there is only one kind of site (or chemical species), but this is rarely the case for biopolymers. An extremely useful and common approximation is to expand $\mathrm{p}K_\mathrm{a}(\mathrm{pH})$ as a power series around some fixed "apparent" value, $\mathrm{p}K_\mathrm{a}^{(\mathrm{a})}$:

$$\mathrm{p}K_\mathrm{a}(\mathrm{pH}) \approx \mathrm{p}K_\mathrm{a}^{(\mathrm{a})} + (1-n)\left(\mathrm{pH} - \mathrm{p}K_\mathrm{a}^{(\mathrm{a})}\right). \tag{6}$$

This approximation introduces the Hill coefficient n and corresponds to the equation

$$\langle \lambda_s \rangle_\mathrm{pH} \approx \frac{1}{1 + 10^{n(\mathrm{pH}-\mathrm{p}K_\mathrm{a})}}, \tag{7}$$

which, in most cases, provides an excellent fit to the data and provides a quantification of the correlation between a given site and the other sites in the system. For $n = 1$, the correlations vanish, while values less than and greater than one indicate positive and negative correlation with other sites being occupied. That is, if $n > 1$ for a given site, then that site is more likely to be occupied when other sites in the system are *unoccupied* and vice-versa.

In general, the pattern will be to compute $\langle \lambda_s \rangle_{\mathrm{pH}}$ for the residue(s) of interest at multiple pH values and then perform non-linear regression according to Eq. (7). Better statistics can be acquired by using a multistate reweighting procedure, such as the weighted histogram analysis method or its variants [20–23], but the theory behind this is beyond the scope of this tutorial. It is also worth noting that the above analysis becomes slightly more complicated when analyzing specific sites in residues with multiple, non-equivalent sites. Again, this is beyond the scope of the present tutorial and is generally highly specific to the application at hand.

2. Example 1: Single Peptide Residue

The first example is a single peptide in explicit solvent. This is arguably less interesting than larger applications, but can be managed on a small amount of resources and captures the essential components of titration curve analysis. It also provides an excellent opportunity to become acquainted with the basic NAMD keywords associated with constant-pH MD. The necessary files for what follows can be found in the `asp` directory inside the `tutorial-constant-ph.tar.gz` archive.

2.1. Simulation Setup

Simulation setup follows exactly as it does for conventional MD simulations. The inputs in `asp/topology` were constructed using `psfgen` with solvation from VMD. The system is a terminally blocked aspartate "dipeptide" inside a ~ 40 Å cubic box of modified TIP3P water molecules and has already been equilibrated with conventional MD at constant temperature and pressure. Although not essential, the system has been prepared in the most common deprotonated state, with no protons present on the carboxylate moiety. As will be seen, all modifications for constant-pH MD (*e.g.,* the addition of dummy atoms) will be done automatically by NAMD. Lastly, because this system is being prepared with the CHARMM36 force field the additional `topology` and `parameters` files located in the `toppar` directory will be needed, *but only during the simulation, they have no impact on the system construction*.

The additional `topology` and `parameter` files are force field dependent. Just as `parameter` files must be properly matched with a PSF, so too must these files be matched with a system after it has been constructed. An improper match may cause simulations to crash or give incorrect results.

2.2. Running the Simulations

Once the system has been prepared and simulation parameters chosen as usual, the user must then specify constant-pH specific keywords and commands. Principle among these is the command to source the necessary TCL files. For convenience, version 1.0 (to be released with NAMD 2.13) is provided in the tutorial files in the directory `namdcph` and the correct relative path is built in to the example configuration files inside `asp/example`. On other systems this will require the user to know where the correct files are located – this is an unavoidable consequence of the way TCL is currently distributed with NAMD. Looking inside `asp/example/start.namd` you will see a standard list of NAMD keywords using the pre-equilibrated system from the previous section. The latter half contains the constant-pH keywords. Once this is done, the necessary `topology` and `parameter` files are loaded using the standard NAMD and `psfgen` keywords. The first new keyword is `cphConfigFile`, which is non-optional and loads additional meta information regarding the new `topology` files.

In the next section, a useful trick is employed that permits re-using configuration files for multiple runs at different pH values. A TCL variable is used to store the desired pH and then as a label for one of many output directories into which different simulations can be sorted. The only difference between these simulations will be the value specified by the `pH` keyword, which is also non-optional. Note that the normal output uses the same file names, but a different directory. Lastly, it should be noted that the new constant-pH specific temporary files are also directed to a new directory by use of the `cphMDBasename` and `cphSwitchBasename` keywords.

Next, we modify new settings that control the nature of the neMD/MC moves. This is quite simple in the present case because the system only contains one type of residue and thus one type of move. In general, the most important setting is the length of the switching trajectories that will be attempted. This is specified with the `cphNumstepsPerSwitch` keyword. This keyword can have more nuanced use, but in the present case we only need to specify a single length. The general recommendation is switches on the order of 10-20 ps, unless poor performance is encountered.

Finally, the actual simulation command, `cphRun` is called with two arguments. The first is the number of steps of conventional MD to be run in between switches and the second is the number of cycles (including both conventional MD and neMD/MC) to run in total. Unfortunately, there are no comprehensive theoretical arguments for how the simulation time should be partitioned between both kinds of propagation. However, experience shows that the attempt frequency should be rather high, especially for systems with many titratable residues, and values on the order of 0.1–1 ps are not unreasonable. Also note that, since many proposals are rejected immediately due to the inherent pK_a, a small number of steps between cycles does not necessarily imply a large number of switches being performed.

2.3. Computing Titration Curves

In this section, we will discuss the basic principles behind computing titration curves from constant-pH MD simulations. At this point the user should either: 1) generate data at multiple pH values (some recommended values are given as directory names inside `asp/example`) or 2) use the data generated previously and stored in `asp/archive`. Both directories have parallel structure. Note also that, to save disk space, only the `cphlog` files are given for the archived trajectories. For most simulations, these are the only files needed to compute titration curves.

As discussed in the introduction, all constant-pH MD simulations produce output to `[outputname].cphlog` which stores the occupancy vector describing all protonation states; this file is updated after each simulation cycle. The aspartate residue in this example has two sites and so, in addition to the time stamp, there are two columns of data in the `cphlog` file. Since each column represents an element in the occupation vector $\boldsymbol{\lambda}$ we shall refer to column one as λ_1 and column two as λ_2. Simply taking the mean of any λ_s will give the fraction of time that site s = 1 or 2 was protonated at the given pH. However, this is not the quantity of interest in all cases. In the present case, we are most interested in the fraction of time that the residue as a whole is protonated; this requires defining a more complicated indicator function, which shall be denoted χ. Since the two sites are equivalent, this can be conceived of in two different ways: 1) the sites should be *added* together, $\chi = \lambda_1 + \lambda_2$, or 2) we should compute the joint probability, $\chi = \lambda_1(1-\lambda_2) + (1-\lambda_1)\lambda_2$. Because λ_1 and λ_2 are never both equal to one (we do not consider the possibility of a doubly protonated carboxylate), these two formulations are equivalent, but the latter is more general.

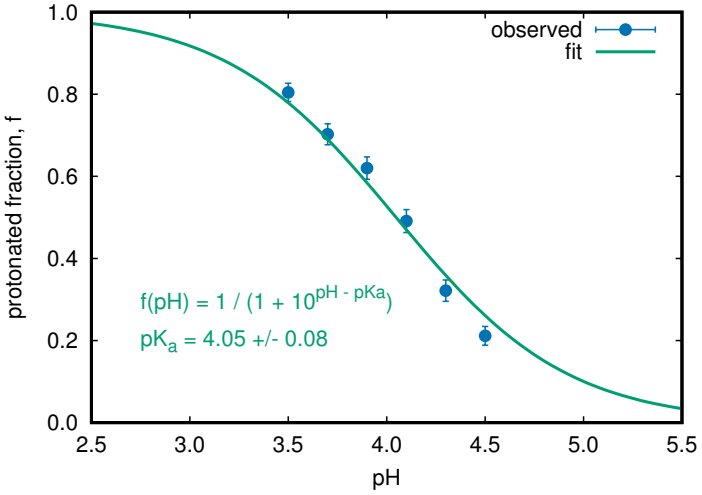


Figure 4: Collecting the protonated fraction observed at each pH value constitutes a titration curve (shown here for aspartate dipeptide). A pK_a value can be extracted by fitting to the Henderson-Hasselbalch equation. Error bars represent estimated 95% confidence intervals.

A simple Python script, `analyze_asp.py` is included for the purposes of this tutorial. The interested reader is encouraged to examine this script in order to better understand how the calculation is performed. The script takes as arguments any number of `cphlog` files collected at any number of pH values; wild card selections are permissible. In order to analyze all of the data in archive one must simply type:

```
$ ./analyze_asp.py archive/*/*.cphlog
```

The output is a simple whitespace delimited file and is plotted in Figure 4 (blue data points). It is also possible to estimate confidence intervals by noting that, because χ can only be zero or one, the standard error of $\langle\chi\rangle$ is simply $\epsilon_\chi = \sqrt{\langle\chi\rangle(1-\langle\chi\rangle)/N}$, where N is the number of samples.

Finally, we can extract a pK_a value from the graph by performing a non-linear regression to the Henderson-Hasselbalch equation. This is identical to Eq. (7) when the Hill coefficient is exactly one. It is used here because there is only one residue in the system and so, aside from concentration effects (infinite dilution is assumed here), there are no other sites for the residue to be correlated with. This fit is also shown in Figure 4 (green line), as well as the resulting parameter values and error bars. The results are in excellent agreement with the force field benchmark (pK_a = 4.0).

3. Example 2: The BBL Miniprotein

Here a small protein, BBL, is now shown as a realistic application of constant-pH MD. Although BBL is relatively well-studied and has a rather stable structure, the resources needed to obtain adequate statistics are substantial, requiring at least a small, multi-node cluster. Once again, archived data is available for analysis so that the user can complete this section without too much expense. Most importantly, the principles used to submit simulations at multiple pH values can be extended directly to other computing systems and the somewhat advanced setup for designating inherent pK_a values is also quite general for similar systems with multiple titratable residues.

3.1. Simulation Setup

Here we use the dominant NMR structure stored under the PDB code 1W4H [24]. The system was solvated in a truncated octahedron (57 Å) with 200 mM NaCl as determined by the CHARMM-GUI [25] Quick MD Simulator (at present, this is the only way to extract proper extended system information for non-orthorhombic cells). Files containing a pre-equilibrated snapshot from a very short constant-pH MD simulation of the resulting system are available in `bbl/topology`. For speed, we have also produced a PSF for the system with a simple hydrogen mass repartitioning scheme on the protein [26]; this extra PSF is named `bbl.hmr.psf` and includes repartitioning of mass on to the dummy atoms.

Mass repartitioning should improve accuracy when using r-RESPA multiple time-stepping [27], but is otherwise unnecessary.

The CHARMM-GUI produces a lot of output, only some of which is properly compatible with NAMD. This occasionally requires manual modification of PSF and PDB files. It is not guaranteed that the files provided with this tutorial can be exactly reproduced with CHARMM-GUI.

3.1.1. *Inherent pK_a Estimates and Initializing States*

There are additional considerations in constant-pH MD that become increasingly important when a large number of titratable residues are present (11 in this case). First, it may be desireable to refine initial estimates for the inherent pK_a values in the system. By default, these are assigned based on the force field reference compounds. In the present parameterization, the reference compounds are single sidechain dipeptides, as in the previous example. Such estimates may be quite accurate if the residues are in a mostly aqueous environment and this is frequently the case for residues in globular proteins, with rare and notable exceptions. Often the best course of action is to use an experimental estimate or else a structure based estimate such as those from PROPKA [28]. Any number of residues can be re-assigned with the `cphSetResiduepKai` command. For example, E161 has an experimentally measured pK_a of 3.7, which is appreciably lower than the reference value of 4.4. This can be noted by including

```
cphSetResiduepKai PROA:161:GLU 3.7
```

in the NAMD configuration file, where `PROA` is the segid from the PDB file (in principle, this can be changed by the user). The command permits multiple arguments and/or calls in the same file, so that any or all of the residues in the system can be specifed.

It may also be desireable to specify directly the starting protonation state of all or some residues. If a starting state is not specified (and the simulation is not being restarted), then each residue is assigned a state according to the pH and the inherent pK_a. *No structural information is used in this assignment.* Note that protonation states are assigned by an ASCII code that differs by the residue type. Continuing with the E161 example, glutamate has three codes: “D” – deprotonated, “1” – protonated at OE1, and “2” – protonated at OE2. In order to initialize E161 in the deprotonated state, one would use

```
cphSetResidueState PROA:161:GLU D
```

Again, multiple arguments and/or calls are permitted. Be aware that the initially assigned protonation states may be unstable within the backbone and sidechain structure. It is thus a good idea to minimize the system with the new protonation states. Because of the extra PSF modifications, the constant-pH

MD code has its own minimization keyword, `cphNumMinSteps`. Once minimization is completed, velocities are automatically reinitialized according to the initial or thermostat temperature.

3.1.2. *Focused Sampling and Constructing neMD/MC Move Sets*

An additional functionality which will be described, but not used, here is the ability to focus sampling on certain residues and/or neMD/MC moves. By construction, the inherent pK_a algorithm naturally avoids proposing low probability *protonation states*, but it nonetheless permits all *residues* to titrate on equal footing. This may be undesireable in a system with a large number of residues that are known to titrate at the given pH, but are uninteresting to the physical problem at hand (perhaps we are primarily interested in a handfull of residues in an enzyme's active sight, say).

As an elaboration to the inherent pK_a algorithm, each neMD/MC move is begun by selecting a set including one or more residues. By default, the only residue sets that can be selected consist of each residue individually – it is up to the user to specify pairs (or triplets, *etc.*) that can titrate together. Each residue set is specified a fixed probability weight which, after normalization, constitutes a probability mass function that can be sampled directly. Again, by default, each residue is in its own set and specified a weight of 1.0 such that all residues can be selected equally. Setting a residue's weight to 10.0, say, would mean that it is 10 times more likely to select that residue as a candidate for titration than all of the other residues (individually, not in aggregate). Once more using E161 as an example, the residue weight can be set to 10.0 with the following keywords:

```
cphProposalWeight PROA:161:GLU 10.0
```

However, once it has been selected, this "special" residue is still subject to the inherent pK_a algorithm and will have a diminished chance of moving from a high probability protonation state to a low probability state if the pH and inherent pK_a differ significantly. Nonetheless, focusing proposals on only a few residues will increase the rate of sampling without perturbing the statistics in any systematic way. That being said, one should be cautious about over-emphasizing certain residues as there is a high risk that the simulation will: 1) undersample and produce very poor statistics for "de-emphasized" residues and 2) miss sampling correlations between residues.

The keywords in this section can be modified in between restarts of a constant-pH simulation. Most importantly, this information is stored in the `cphrst` file after a run is complete and is loaded in the next segment, even if the keywords are omitted in subsequent restarts. This dramatically simplifies the restart procedure once a set of parameters has been suitably chosen. Future versions of the program may even use the gathered statistics in order to *automatically* adjust parameters in between restarts without requiring input from the user.

3.2. Running and Analyzing the Simulations

The basic concerns here are identical to those for the simple peptide in the previous example. We shall instead focus on the additional complications of handling a larger system with more substantial resource requirements and the problem of analyzing multiple titratable residues. An example input file can be found in `bbl/example/run_mca.namd` (discussed in detail below). The main new settings involve using the `cphSetResiduepKai` keyword, as discussed in the previous section. In particular, all titratable aspartate, glutamate, and histidine are assigned inherent pK_a values equal to their experimentally determined values [29, 30].

3.2.1. Using Multiple Copies to Simulate Multiple pH Values in Parallel

Since the recommended usage of constant-pH MD is to simulate the same system at multiple pH values, it may also be recommendable to perform these simulations in parallel. The multiple copy algorithm (MCA) functionality in NAMD serve this purpose excellently. A simple framework for doing this is to define a TCL list of pH values to be simulated, where the number of compute resources are equally divisible by the length of the list (*e.g.,* 16 compute nodes and eight pH values). The `myReplica` command can then be used to extract an element from this list. Most importantly, because the MCA functionality uses a single input file, but multiple output streams, several output related settings need to be "tagged" with either the replica ID or pH value. In addition to the standard `outputname` and `stdout` keywords, constant-pH MD also utilizes temporary intermediate files, which need to be separately sorted per simulation and avoid name collision, preferably by using different working directories. Both file names can be changed with the `cphMDBasename` and `cphSwitchBasename` keywords. Examples of how to do this can be found amongst the tutorial files in `bbl/example/run_mca.namd`. The archival data was produced on nine 64-processor nodes (576 cores total) using an MPI enabled version of NAMD with a command simuilar to the following:

```
$ mpirun -np 576 namd2 +replicas 9 run_mca.namd &> cph.log
```

3.2.2. Titration Curves and the Hill Equation

The principles for computing titration curves are much the same when considering multiple residues. The main difference is that the proliferation of sites requires more elaborate bookkeeping. To this end, a Python-based utility, `cphanalyze` has been made which natively handles the NAMD `cphlog` format. Furthermore, the program also implements multistate reweighting techniques and automated non-linear regression in order to integrate statistics gathered at multiple pH values. The install procedure requires the user to first download and then install the code. The following commands will accomplish this:

```
$ git clone https://github.com/radakb/pynamd
$ cd pynamd
$ python setup.py install
```

The executable is then available under `pynamd/scripts/cphanalyze`. If difficulties are encountered, additional instructions are available on the `PyNAMD` github page.

`cphanalyze` is an *experimental* code at this time. Be very sure that you understand what the output is telling you before drawing conclusions. The most useful, if least accurate, option is the `--naive` flag, which implements the simple counting procedure demonstrated in the `analyze_asp.py` script from the previous example.

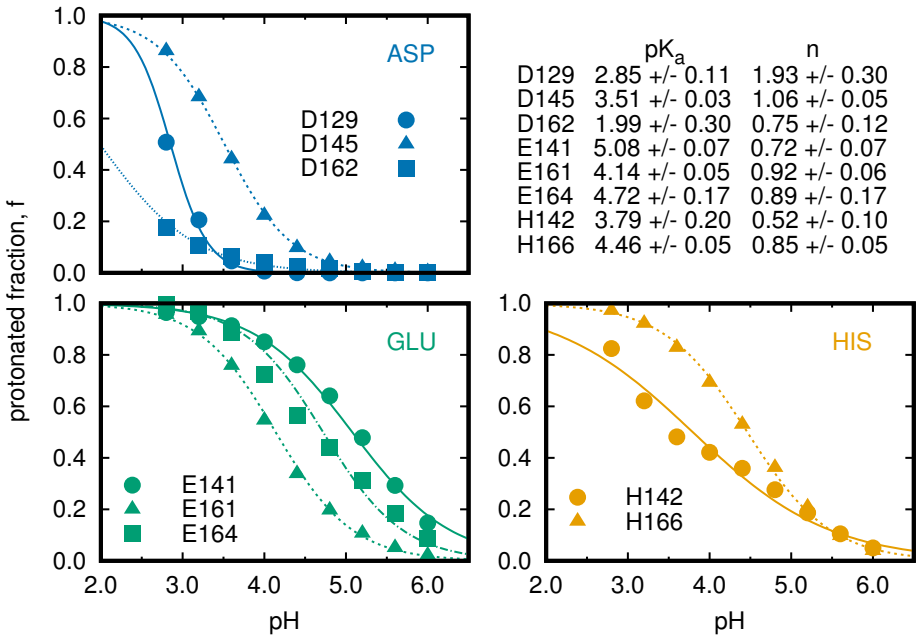


Figure 5: Titration curves can be computed for BBL using the `cphanalyze` program and a specialized reweighting procedure [1]. All titratable residues that actively changed in the selected pH range have been sorted by type and their macroscopic titration curves were fit to the Hill equation. Error bars represent estimated 95% confidence intervals.

Once again, archival data from several simulations of BBL are available in the tutorial files under `bbl/archive`. The `cphanalyze` utility can automatically sort multiple `cphlog` files. It also requires, as input, the same JSON configuration file needed to run constant-pH MD. From the `bbl/archive` directory, all of the data can be simply and cleanly analyzed by typing:

```
$ cphanalyze */*.cphlog --json ../../toppar/conf_cph36_prot.json
```

This immediately computes titration curves for every residue in the system, as well as error estimates. There is a also a crude filter syntax available for including/excluding residues by type (*e.g.*, ASP or GLU)

as well as by their segment and residue IDs (*e.g.,* PROA:161). Occasionally, some residues do not titrate during the course of the simulation. This generally indicates that a pK_a is outside of the pH range that was sampled, but it can also be indicative of poor statistics.

Titration curves for the actively titrating residues in BBL are shown in figure 5. Non-linear fits to Eq. (7) were carried out using `cphanalyze` (built using `NumPy/SciPy`), but any graphing program is appropriate (`gnuplot` is fairly good). Notably, the Hill coefficients all deviate from 1.0 by at least some degree; this is to be contrasted with the curve for a single peptide in solution.

3.2.3. Cautions Regarding Analysis and Convergence

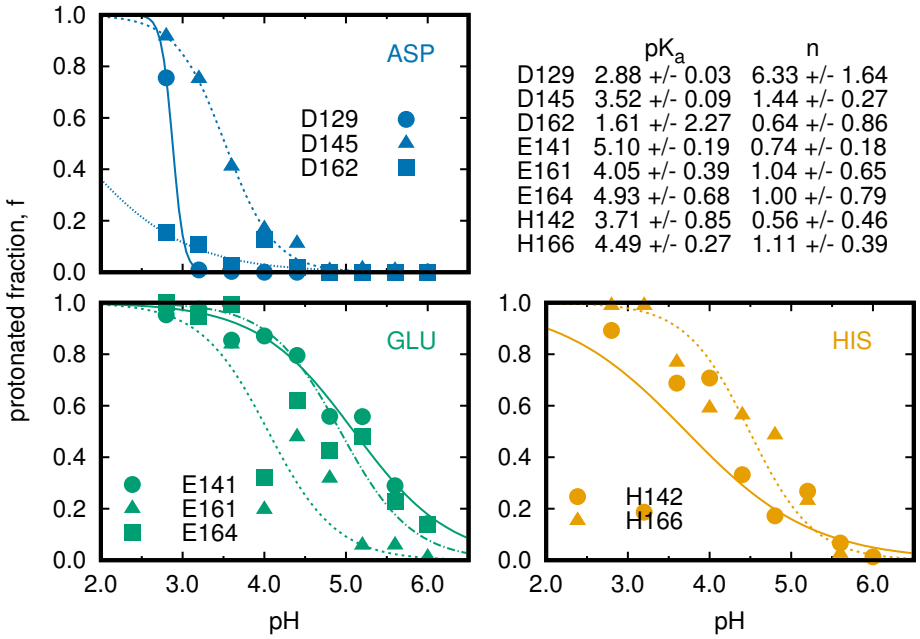


Figure 6: Titration curves can be computed for BBL using the `cphanalyze` program and a simple counting procedure. Note that many irregularities in sampling at different pH values are somewhat hidden by the reweighting algorithm used in figure 5. All titratable residues that actively changed in the selected pH range have been sorted by type and their macroscopic titration curves were fit to the Hill equation. Error bars represent estimated 95% confidence intervals.

In this section we will highlight some of the pitfalls and difficulties of constant-pH MD. Most specifically, the default reweighting approach implemented in `cphanalyze`, while very useful, can also very easily mask pathologies in the data. In the present case, re-computing the titration curves from figure 5 using a naive counting algorithm (as in Example 1), leads to the rather different curves shown in figure 6. Clearly many of the unweighted data points are very far from the fit curves, especially E161 and E164 at pH 4.0 (lower left) and H142 at pH 3.2 (lower right). Based on the consensus trend from the other simulations,

these points are very likely unconverged outliers and may indicate that the protein has a significantly different structure in these trajectories. For example, H142 coordinates with a protonated lysine residue, but only when neutral (data not shown). It is satisfying to see that, despite these infelicities, the pK_a estimates from both sets of curves vary by less than 0.5 units. Nonetheless, techniques such as pH-replica exchange may be effective in harmonizing convergence across the set of simulations [7, 8].

4. Concluding Remarks & Further Investigation

Thus ends this tutorial for constant-pH MD. It should be clear that this represents a powerful approach for addressing systems with variable protonation states and that both the theoretical underpinnings and implementational details are now mature enough for routine application. Nonetheless, as with many aspects of molecular simulations, challenges remain, especially with regards to accurate and reliable sampling, and ongoing developments aim to address these issues. The fact that this approach is now implemented in NAMD at a low level means that the intrinsic performance and portability, as well as the plethora of complementary enhanced sampling methods, can be applied towards these efforts.

A second, and perhaps equally important, advance shown here is the machinery for computing titration curves from the simulation output [1]. Although a great deal of insight can be gained from analyzing the behavior of individual residues or sites, this was not discussed in depth here. This is largely because these insights speak to the intrinsic complexity of biochemistry and not the method itself. The main message is that constant-pH MD simulations rigorously model the interactions between titratable residues using a physics-based potential – there is no recourse to *ad hoc* or empirical descriptions. Nonetheless, well-established approximations such as the Hill equation can be of great use, especially given their widespread use and ease of application. More complex calculations are of course possible, for example directly computing correlations betweens residues. However such approaches are not yet completely formalized and require specific insight from the user.

An aspect that was not discussed in this tutorial at all is the question of the force field. Although CHARMM36 is proven to be quite reliable, comparison studies with other force fields (existing or still under development) are always welcome. The effort to adapt additional force fields for use with constant-pH MD is straightforward, but substantial. The initial barrier is simply having the topology and parameters in the suitable RTF and PRM file formats. Interested users are invited to work with the developers in expanding this coverage.

References

[1] Brian K. Radak, Christophe Chipot, Donghyuk Suh, Sunhwan Jo, Wei Jiang, James C. Phillips, Klaus Schulten, and Benoît Roux. Constant-pH molecular dynamics simulations for large biomolecular systems. *J. Chem. Theory Comput.*, 13:5933–5944, 2017.

[2] António M. Baptista, Vitor H. Teixeira, and Cláudio M. Soares. Constant-pH molecular dynamics using stochastic titration. *J. Chem. Phys.*, 117:4184–4200, 2002.

[3] Michael S. Lee, Freddie R. Salsbury, Jr., and Charles L. Brooks III. Constant-pH molecular dynamics using continuous titration coordinates. *Proteins*, 56:738–752, 2004.

[4] John Mongan, David A. Case, and J. Andrew McCammon. Constant pH molecular dynamics in Generalized Born implicit solvent. *J. Comput. Chem.*, 25:2038–2048, 2004.

[5] Harry A. Stern. Molecular simulation with variable protonation states at constant pH. *J. Chem. Phys.*, 126:164112, 2007.

[6] Serena Donnini, Florian Tegeler, Gerrit Groenhof, and Helmut Grubmüller. Constant pH molecular dynamics in explicit solvent with λ-dynamics. *J. Chem. Theory Comput.*, 7:1962–1978, 2011.

[7] Jason A. Wallace and Jana K. Shen. Continuous constant pH molecular dynamics in explicit solvent with pH-based replica exchange. *J. Chem. Theory Comput.*, 7:2617–2629, 2011.

[8] Jason M. Swails, Darrin M. York, and Adrian E. Roitberg. Constant pH replica exchange molecular dynamics in explicit solvent using discrete protonation states: Implementation, testing, and validation. *J. Chem. Theory Comput.*, 10:1341–1352, 2014.

[9] Juyong Lee, Benjamin T. Miller, Ana Damjanović, and Bernard R. Brooks. Constant pH molecular dynamics in explicit solvent with enveloping distribution sampling and Hamiltonian exchange. *J. Chem. Theory Comput.*, 10:2738–2750, 2014.

[10] Yunjie Chen and Benoît Roux. Constant–pH hybrid nonequilibrium molecular dynamics–Monte Carlo simulation method. *J. Chem. Theory Comput.*, 11:3919–3931, 2015.

[11] Jerome P. Nilmeier, Gavin E. Crooks, David D. L. Minh, and John D. Chodera. Nonequilibrium candidate Monte Carlo is an efficient tool for equilibrium simulation. *Proc. Natl. Acad. Sci. USA*, 108:E1009–E1018, 2011.

[12] Yunjie Chen and Benoît Roux. Efficient hybrid non-equilibrium molecular dynamics - Monte Carlo simulations with symmetric momentum reversal. *J. Chem. Phys.*, 141:114107, 2014.

[13] Yunjie Chen and Benoît Roux. Generalized Metropolis acceptance criterion for hybrid non-equilibrium molecular dynamics–Monte Carlo simulations. *J. Chem. Phys.*, 142:024101, 2015.

[14] Brian K. Radak and Benoît Roux. Efficiency in nonequilibrium molecular dynamics Monte Carlo simulations. *J. Chem. Phys.*, 145:134109, 2016.

[15] A. D. MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. B*, 102:3586–3616, 1998.

[16] Jeffery B. Klauda, Richard M. Venable, J. Alfredo Freites, Joseph W. O'Connor, Douglas J. Tobias, Carlos Mondragon-Ramirez, Igor Vorobyov, Alexander D. MacKerell, Jr., and Richard W. Pastor. Update of the CHARMM all-atom additive force field for lipids: Validation on six lipid types. *J. Phys. Chem. B*, 114:7830–7843, 2010.

[17] Robert B. Best, Xiao Zhu, Jihyun Shim, Pedro E. M. Lopes, Jeetain Mittal, Michael Feig, and Alexander D. MacKerell, Jr. Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain χ_1 and χ_2 dihedral angles. *J. Chem. Theory Comput.*, 8:3257–3273, 2012.

[18] Jiali Gao, Krzysztof Kuczera, Bruce Tidor, and Martin Karplus. Hidden thermodynamics of mutant proteins: A molecular dynamics analysis. *Science*, 244:1069–1072, 1989.

[19] Paul H. Axelsen and Daohui Li. Improved convergence in dual-topology free energy calculations through use of harmonic restraints. *J. Comput. Chem.*, 19:1278–1283, 1998.

[20] Shankar Kumar, Djamal Bouzida, Robert H. Swendsen, Peter A. Kollman, and John M. Rosenberg. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *J. Comput. Chem.*, 13:1011–1021, 1992.

[21] Marc Souaille and Benoît Roux. Extension to the weighted histogram analysis method: Combining umbrella sampling with free energy calculations. *Comput. Phys. Commun.*, 135:40–57, 2001.

[22] Michael R. Shirts and John D. Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.*, 129:124105, 2008.

[23] Zhiqiang Tan, Emilio Gallicchio, Mauro Lapelosa, and Ronald M. Levy. Theory of binless multi-state free energy estimation with applications to protein-ligand binding. *J. Chem. Phys.*, 136:144102, 2012.

[24] Neil Ferguson, Timothy D. Sharpe, Pamela J. Schartau, Mark D. Allen, Christopher M. Johnson, Trevor J. Rutherford, and Alan R. Fersht. Ultra-fast barrier-limited folding in the peripheral subunit-binding domain family. *J. Mol. Biol.*, 353:427–446, 2005.

[25] Sunhwan Jo, Taehoon Kim, Vidyashankara G. Iyer, and Wonpil Im. CHARMM-GUI: A web-based graphical user interface for CHARMM. *J. Comput. Chem.*, 29:1859–1865, 2008.

[26] Chad W. Hopkins, Scott Le Grand, Ross C. Walker, and Adrian E. Roitberg. Long-time-step molecular dynamics through hydrogen mass repartitioning. *J. Chem. Theory Comput.*, 11:1864–1874, 2015.

[27] M. E. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Phys. Chem. B*, 97:1990–2001, 1992.

[28] Hui Li, Andrew D. Robertson, and Jan H. Jensen. Very fast empirical prediction and rationalization of protein pK_a values. *Proteins*, 61:704–721, 2005.

[29] Eyal Arbely, Trevor J. Rutherford, Timothy D. Sharpe, Neil Ferguson, and Alan R. Fersht. Downhill versus barrier-limited folding of BBL 1: Energetic and structural perturbation effects upon protonation of a histidine of unusually low pK_a. *J. Mol. Biol.*, 387:986–992, 2009.

[30] Eyal Arbely, Trevor J. Rutherford, Hannes Neuweiler, Timothy D. Sharpe, Neil Ferguson, and Alan R. Fersht. Carboxyl pK_a values and acid denaturation of BBL. *J. Mol. Biol.*, 403:313–327, 2010.