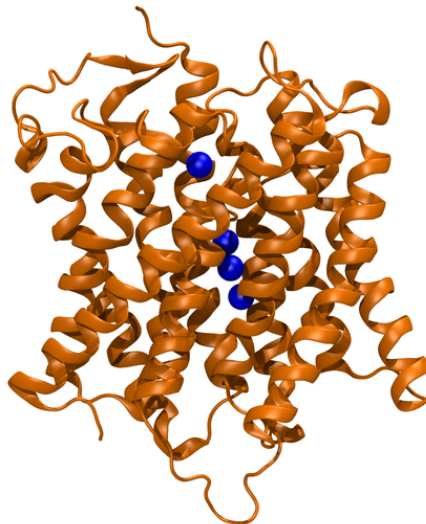School of Physics
Georgia Institute of Technology

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

# Methods for calculating Potentials of Mean Force

**Anthony Hazel**
**James C. Gumbart**

Sept. 2015

A current version of this tutorial is available at
http://www.ks.uiuc.edu/Training/Tutorials/
Join the `tutorial-l@ks.uiuc.edu` mailing list for additional help.

# Contents

# 1   Introduction

Living cells are defined in part by the possession of a membrane barrier surrounding their perimeter, thereby delineating the border between the inside and the outside. The existence of this barrier, as well as the proteins that mediate exchange of ions and other substrates across it, leads immediately to the question of how transport across the membrane, whether it be active or passive, is achieved and, in particular, how its done selectively. Although some common mechanisms exist, each protein has evolved specific features to optimally transport its particular substrate. The transport process is characterized typically by a substrate pathway along with associated conformational changes in the protein. These changes are often small or even non-existent for passive channels, but can be quite large for active transporters. The substrate pathway can also be characterized by the substrate's free energy along it, known as the potential of mean force (PMF), which dictates the speed of transport and the selectivity.

While X-ray crystallography and other experimental structural techniques can provide snapshots of membrane proteins, even in multiple states, they cannot display the dynamics connecting those states. Computational methods, namely molecular dynamics (MD) simulations, provide a means of animating the static structures, allowing for, e.g., the visualization of an entire permeation event of a substrate through a channel or transporter. However, the relevant time scales for many transport processes is beyond that afforded by MD, which is typically limited to a few microseconds currently. As such, methods to accelerate the process of interest in a simulation have been developed. One example is steered molecular dynamics (SMD) (1, 2), a method in which forces are applied to part of the system, e.g., the substrate, to drive it along a predefined direction. Commonly implemented using a constant force or a constant velocity, this method is useful for exploring possible permeation pathways through a membrane protein. However, because the forces imposed are usually orders of magnitude greater than would be experienced in a living system, interpretation of the results is often limited to a qualitative description of the possible behavior of the substrate and protein. For a quantitative picture of transport, one must turn to more advanced methods. For example, the results of multiple SMD simulations combined through application of Jarzynski's equality provides a means of recovering an equilibrium PMF from non-equilibrium events (3–7). Alternatively, adaptive biasing forces (ABF) can generate quasi-equilibrium trajectories from which the PMF can be deduced (8–11).

Numerous channels and transporters have been studied using MD-based methods. SMD has been used to address substrate binding and/or transport in the water channel aquaporin (12–15), the ADP/ATP carrier AAC (16), neurotransmitter transporters LeuT (17), the sugar transporter LacY (18, 19), and many others. In addition, SMD has been used to probe the conformational changes underlying function for many membrane proteins, such as the vitamin $B_{12}$ transporter BtuB (20) and the protein-conducting channel SecY (21–23). ABF has been used to find the free-energy profile for, e.g., glycerol in the channel GlpF (24), ADP in AAC (25), and ions in the nicotinic acetylcholine and glycine
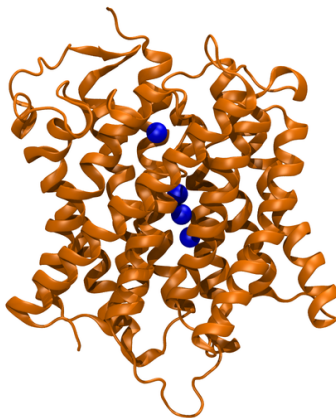
Figure 1: Crystal structure of AmtB. The top of the figure is the extracellular side of the channel and the bottom is the cytoplasmic side. The four crystallographically resolved $NH_3/NH_4^+$ molecules, denoted Am1 to Am4 from top to bottom, are also seen in blue.

receptors (26). Both methods, SMD and ABF, are clearly well established, but still require care to ensure their proper application.

In this tutorial, applications of SMD and ABF to the ammonium transporter AmtB will be explored (see Fig. 1). AmtB has been well studied by computational methods, including those used to calculate free energies (27–32). SMD will be used first to gain an approximate knowledge of the permeation pathway and the barriers along it. Then the PMF for ammonia in the central region of the channel will be calculated using ABF. The appropriate choice of parameters and potential difficulties will also be discussed.

The reader of this tutorial is assumed to be familiar with the use of NAMD to perform "standard" calculations, including energy minimization and MD simulations. General documentation, tutorials and templates of NAMD configuration files are available from the Documentation section of the NAMD web page.

**Many of the simulations in this tutorial can, out of necessity, take a significant amount of time to run on a single processor. If this time becomes prohibitive to completing the tutorial, example output is provided throughout to enable the reader to still carry out the desired analysis.**

The investigations of AmtB described below will utilize the 1.35-Å structure solved in 2004 (33). Prior to carrying out SMD and ABF simulations, a membrane-water system containing AmtB should be built and equilibrated.

## 1.1 Required programs

### 1 VMD:

Available at http://www.ks.uiuc.edu/Research/vmd/ (for all platforms)

### 2 NAMD (ver. 2.9 or higher compiled with the `replica` option):

Available at http://www.ks.uiuc.edu/Research/namd/ (for all platforms)

### 3 Plotting program:

The free program xmgrace (unix) is recommended. It can be downloaded from http://plasma-gate.weizmann.ac.il/Grace/. Other useful graphing programs are Mathematica, http://www.wolfram.com/, Matlab, http://www.mathworks.com/, and gnuplot, http://www.gnuplot.info/. For Windows, Microsoft Excel can also be used.

### 4 wham:

Available at http://membrane.urmc.rochester.edu/content/wham (for Linux and similar variants)

## 1.2 Getting started

1 A fully built and equilibrated system is required for the tutorial. The necessary files are provided in `channel-tutorial-files/common/`, or can be downloaded from http://www.ks.uiuc.edu/Training/Tutorials/science/channel/. Example input files for running the simulation as well as output are also provided.

2 To build the system from scratch using the tools within VMD, begin by downloading the structure of AmtB from the Protein Data Bank at http://www.pdb.org/ (PDB code: `1U7G`). Then follow the **Membrane Protein Tutorial** to prepare and equilibrate your system.

# 2 Exploring the pathway with Steered MD

Before investing computer time in relatively lengthy free-energy calculations, SMD simulations will be used to gain a qualitative picture of the permeation pathway for ammonia in the AmtB channel. The crystallized ammonium ion on the extracellular side of the channel, mutated to a neutral ammonia molecule, will be pulled toward the cytoplasmic side at a constant velocity. You will then analyze the resulting output.

## 2.1 Setting up and running the SMD simulation

**1** The equilibrated system will be used as the starting point for the constant velocity SMD simulations. Set up a configuration file for a standard equilibrium simulation, which will be modified accordingly.

**2** To use the SMD feature in NAMD, the following parameters need to be set in the configuration file:

```
SMD on
SMDFile smd01.ref
SMDk 5
SMDVel .00001 ;# .00001 A/timestep = 10 A/ns (1 fs timesteps used)
SMDDir 0 0 -1
SMDOutputFreq 100
```

The parameter `SMDVel` defines a pulling speed of 10 Å/ns, which is fast enough to traverse the channel, defined by `SMDDir` to be in the $-z$ direction, in an ~5-ns simulation. The file, `smd01.ref`, is almost identical to the PDB of the entire system and controls to which atoms the SMD force will be applied. A non-zero value in the occupancy column of this file indicates forces will be applied, a "0" not. For pulling a single ammonia molecule, only the nitrogen of the ammonia will be set to 1. Finally, `SMDk` specifies the force constant (in kcal/molÅ$^2$) for the spring connecting the SMD atom to the imaginary atom moving at constant velocity.

**Setting the speed.** The pulling velocity should be as slow as possible in order to maximize sampling of the channel environment. Choose a velocity that allows you to complete the simulation in a reasonable amount of time given the computer power available.

> **Setting the force constant.** The force constant affects how well the constant velocity of the pulled atom is maintained. It should be high enough to ensure that the potential in the channel is accurately measured but not so high that the measurement is dominated by noise; this is also known as the "stiff-spring approximation" (6). The appropriate value will depend on the size of the barriers encountered, although a good rule-of-thumb is that the deviation due to thermal noise, i.e., $\Delta z = \sqrt{\frac{k_B T}{k}}$, is at least less than 0.5 Å. See Fig. 2 for a comparison of the resulting force and distance profiles using different force constants.

**3** To prevent the protein, or even the entire system, from drifting under the applied force, restraints must be applied to counterbalance it. However, it is important that these restraints do not limit any potential conformational changes of the protein during the simulation. The following parameters apply positional restraints to a limited number of atoms:

```
constraints on
consexp 2
consref rest6.ref
conskfile rest6.ref
conskcol O
selectConstraints on
selectConstrX off
selectConstrY off
selectConstrZ on
```

Of particular note is the restriction to apply restraints in only the $z$ direction, which corresponds to the SMD direction. Similar to the `SMDFile` above, the `conskfile` specifies which atoms to apply restraints to as well as the force constant for those harmonic restraints; in this case, the $C_\alpha$ atoms of residues 7, 97, 149, 225, 333, and 348, which are at the periphery of the extracellular side of AmtB, are chosen with a force constant of 5.0 kcal/mol·Å$^2$ The file in `consref` specifies the positions to which those atoms should be restrained, and many times is identical to the input PDB file.

> **Alternative restraints.** To restrain the center-of-mass of a large group of atoms in a single dimension ($x$, $y$, or $z$), SMD can also be used. Set the `SMDFile` to one containing the atoms to restrain, the `SMDDir` to 0 0 1, and `SMDVel` to 0. Because of the way SMD is formulated, non-axial directions cannot be used in this manner. Also, should SMD be used for restraints, an alternative method will be needed to apply the steering forces, such as `tclForces`.

**4** In some cases, the molecule being pulled may find it easier to move laterally outside of the channel before continuing translocation than to cross a large barrier. To limit such a possibility, restraints in the $xy$ plane can be applied. However, these restraints should only affect the molecule if it

tries to move outside the channel, and otherwise be zero. One way to implement such a restraint in NAMD is through `tclBCforces`. Additionally, the `colvars` (short for collective variables) module is useful for defining restraints, as well as for calculating free energies (see below).

In the NAMD configuration file, an external file containing the colvars setup is referenced:

```
colvars on
colvarsConfig restrain.in
```

In `restrain.in`, you will find the following:

```
colvarsTrajFrequency 1000
colvarsRestartFrequency 1000

colvar {
    name restrain01

    width 0.5

    lowerboundary 0.0
    upperboundary 8.0

    lowerwallconstant 100.0
    upperwallconstant 100.0

    distanceXY {
      main {
        atomnumbers { <list of ammonia atom(s)> }
      }
      ref {
        atomnumbers { <list of protein atoms> }
      }
      axis ( 0.0, 0.0, 1.0 )
    }
}
```

The `distanceXY` collective variable defines a restraint that, in this case, limits the motion of the ammonia molecule to a cylinder within 8 Å of the central axis of the protein.

> **Atom numbering.** In `colvars`, as in NAMD, atom numbering begins from 1, whereas in VMD, it begins from 0. An easy way to get the 1-based atom numbers in VMD is to use the keyword `serial` instead of `index`.

**5** Run the simulation. For a 5-ns simulation, it may take 1-3 days to complete, depending on available computational resources. Sample output is provided.

## 2.2   Analysis of the SMD simulation

**1** For a constant-velocity simulation, it is particularly useful to look at the force as a function of position in the channel. To extract this information from the log file, tools such as awk or grep (Linux) can be used. Alternatively, one can write a script in VMD to extract the required information and output it to a new file. For the SMD simulation run here, the content of the script reads as follows:

```
### Open the log file for reading and the output .dat file for writing
set file [open "AmtB-SMD01.log" r]
set output [open "FvsP.dat" w]

### Loop over all lines of the log file
while { [gets $file line] != -1 } {
### Determine if a line contains SMD output.  If so, write the
### current $z$ position followed by the force along $z$ scaled by
### by the direction of pulling (0,0,-1)

   if {[string range $line 0 3] == "SMD "} {
     puts $output "[expr [lindex $line 4]] [expr -1*[lindex $line 7]]"
   }
}
### Close the log file and the output file
close $file
close $output
```

Run the script in VMD and plot the resulting output in, e.g., Grace or Excel (see Fig. 2). Right now, the data appears very noisy. To clean it up, calculate a running average with a window size of 500. This can be done in Grace by going to the menu **Data → Transformations → Running Averages** or in Excel by typing in an adjacent column of the first row `AVERAGE($B1:$B500)`, and then using the "fill down" function (location depends on Excel version) to fill the remaining cells.

**2** Examine the simulation trajectory in VMD and try to correlate the force peaks in the plot with specific events. For example, the first two large peaks correspond to breaching the first two hydrophobic gates in AmtB, formed by Phe107 and Phe215. The three internal binding sites for $NH_3$ are also revealed by this plot. Compare them to the location of those found in the crystal structure. Finally, the other gate formed by Tyr32 and Phe31 can also be identified. If you have not completed your own simulation yet, an example trajectory with frames taken every 10 ps for the first SMD simulation is provided (`AmtB-SMD01-10ps.dcd`).
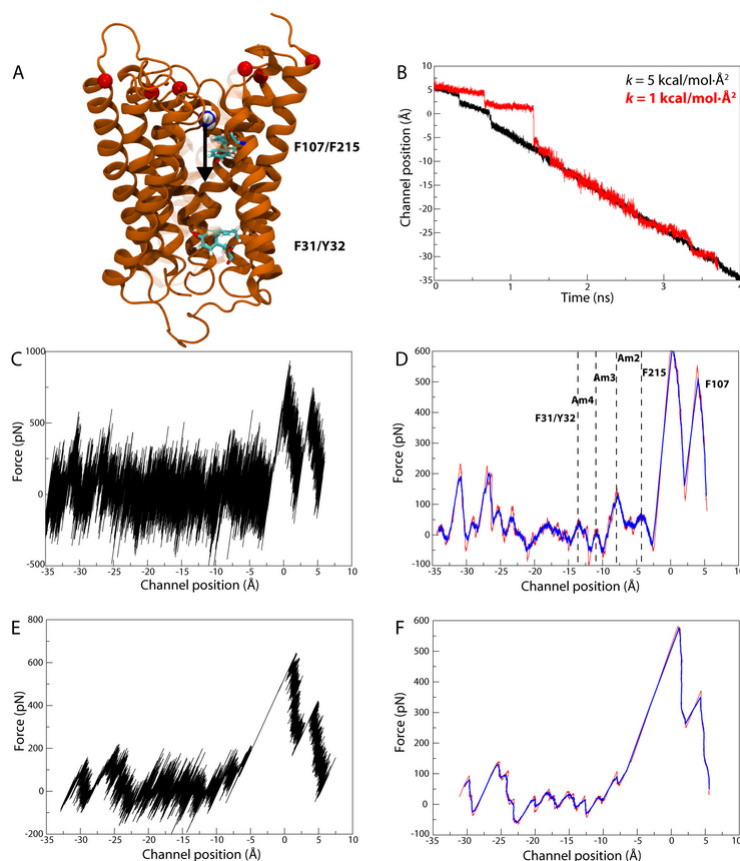
Figure 2: SMD as applied to $NH_3$ in AmtB. (A) Simulation setup. The $NH_3$ molecule begins above the first gate and is pulled in the $z$ direction toward the second gate and beyond. Red spheres near the top indicate residues restrained to negate any net motion arising from the SMD forces. (B) Position of $NH_3$ during the SMD simulations for two different values of the SMD spring constant $k$. Notice how a larger $k$ (black) enforces better the constant velocity of the pulled atom than the smaller $k$ (red). (C) Force vs. position for SMD with $k = 5$ kcal/mol·$Å^2$. (D) Running average of force vs. position with averages taken over 0.5-ns (light, red line) and 1-ns (heavy, blue line) windows. Force peaks are correlated with specific channel features. (E,F) Same as (C,D) but with $k = 1$ kcal/mol·$Å^2$.

# 3 Determining the PMF with Adaptive Biasing Forces

Now with a basic understanding of the $NH_3$ permeation pathway and the barriers along it, a more quantitative characterization can be carried out. We will use the adaptive biasing forces method in NAMD (10, 34) to determine the potential of mean force (PMF) along the channel axis.

## 3.1 Setting up and running the ABF simulation

**1** First, an appropriate reaction coordinate must be defined. While an obvious choice would be just to use the absolute position of $NH_3$ along the channel axis, this does not account for any fluctuations of the channel in $z$. A more apt choice is to relate the position of the molecule with that of the channel's center of mass; this also has the benefit of negating the need for additional restraints. Next, appropriate bounds for this coordinate must be defined. Using VMD, load the equilibrated PDB file for the AmtB system. Make two atom selections, one for $NH_3$ (i.e., "resname AMM1 and noh") and one for the protein $C_\alpha$ atoms (i.e., "protein and name CA"). You can get the distance along $z$ separating them with the `tcl` command

```
expr [lindex [measure center $AMM1sel] 2] - [lindex [measure center $protCAsel] 2]
```

which is, approximately, the beginning of your reaction coordinate. We will use the two hydrophobic gates to more explicitly define the range, i.e., residues 107 and 215 on one side and 31 and 32 on the other. Again, using the `tcl` command above, but with $upperCAsel and $lowerCAsel instead of $AMM1sel, find the relevant region of the channel.

**2** To enhance the efficiency of the ABF algorithm, the span of the reaction coordinate will be subdivided into equally spaced windows. The range found above is approximately [-13,5], while the $NH_3$ is initially positioned 9 Å above the center of the protein. Therefore, we will use four windows, each 5-Å wide, over the range [-13,7]. Create four directories, one for each window. Using the trajectory from the SMD simulation, find frames in which $NH_3$ is somewhere in each of the windows, i.e., between 2 and 7 Å above the protein's center of mass and so on. Write out a PDB for the entire system for each representative frame named, e.g., `win1start.pdb`. These will be the starting points for the four individual ABF simulations.

**3** The NAMD configuration file used for the SMD simulations can be used again here with a few minor edits. First, the keyword `coordinates` should reference the appropriate starting frame for a given window. Remove any reference to SMD and constraints, but leave that for colvars.

**4** As in the SMD section above, colvars will be used for the ABF calculations. The colvars input is similar, with a few important differences:

```
colvarsTrajFrequency 1000
colvarsRestartFrequency 1000

colvar {
   name Translocation

   width 0.1

   lowerboundary --z1--
   upperboundary --z2--

   lowerwallconstant 100.0
   upperwallconstant 100.0

   distanceZ {
     main {
       atomnumbers { <list of ammonia atom(s)> }
     }
     ref {
       atomnumbers { <list of protein atoms> }
     }
     axis ( 0.0, 0.0, 1.0 )
   }
}

abf {
   colvars Translocation
   fullSamples 1000
}
```

In this case, instead of `distanceXY` used to restrain the off-axis movement, the `distanceZ` variable is used. The width of 0.1 Å defines the bin size and is sufficiently small to generate a smooth PMF. The biasing forces and PMF calculation are invoked by the `abf` block. Create a colvars input file for each window, defining the lower and upper boundaries appropriately.

**5** Run the simulation for each window. To restart an ABF simulation, in the subsequent configuration files, add the line

```
colvarsInput <inputname>.restart.colvars.state
```

which references the output of the previous run.

> **Judging convergence.** It is difficult *a priori* to define the amount of sampling required to obtain a converged PMF. One way to assess convergence is to plot sequential PMFs and observe any changes as the simulation progresses. After two or more runs for a given window, plot the last two PMFs and compare. You can also plot the counts, which measure the number of times each bin has been sampled. When the PMF is converged, $NH_3$ should diffuse freely along the reaction coordinate, as the biasing forces cancel out any barriers and wells in the free energy that are encountered. Therefore, the counts should also increase uniformly for all bins after convergence. As another way to judge convergence, you can plot the forces stored in the `*.grad` files for all windows, checking for continuity across the window boundaries.

## 3.2  Analysis of the ABF simulations

Because some windows may require significant sampling to obtain a reasonably accurate PMF, you may also use the provided output for further analysis.

**1** To re-combine the output from the separate windows in a single PMF, we will again use ABF. Create a new directory called **Merge** and place in there the last **grad** and **count** files from each window. Also copy a NAMD configuration file and a colvars input file.

**2** In the configuration file, set it to `run` for 0 steps. Ensure that it still properly references PDB, PSF, and restart files. It is not important which input files you use for this simulation.

**3** You will need to modify the colvars input file in two ways. First, change the upper and lower boundaries so that they encompass the entire range, i.e., -13 and 7. Then, add the following line to the **abf** block:

```
inputPrefix <win1name> <win2name> <win3name> <win4name>
```

where `win_name` refers to the prefix for the output files of that window.

**4** Run the new simulation. NAMD will read in the gradients and counts from the provided output of the individual windows and generate a new set of outputs, including a new PMF, covering the entire reaction coordinate.

**5** Plot the final PMF. Through comparison to the simulation, try to correlate the various extrema to specific locations in the channel. The minima should correspond to the locations of the crystallized ammonia molecules in the PDB `1U7G`, as shown in Fig. 3.
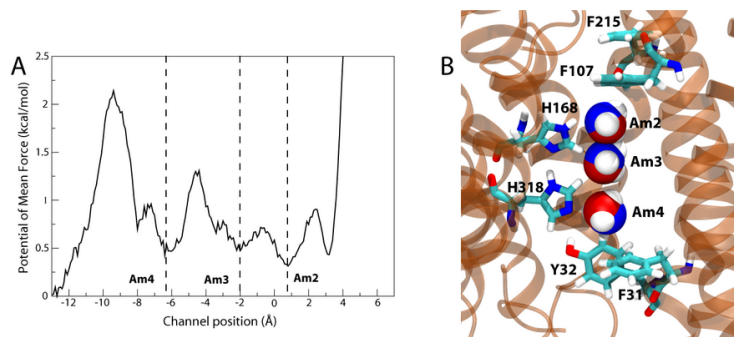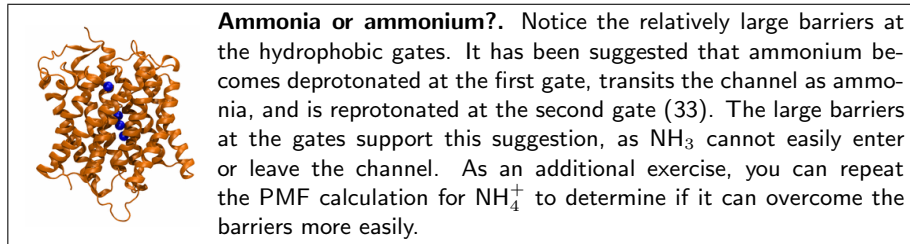
**Ammonia or ammonium?.** Notice the relatively large barriers at the hydrophobic gates. It has been suggested that ammonium becomes deprotonated at the first gate, transits the channel as ammonia, and is reprotonated at the second gate (33). The large barriers at the gates support this suggestion, as $NH_3$ cannot easily enter or leave the channel. As an additional exercise, you can repeat the PMF calculation for $NH_4^+$ to determine if it can overcome the barriers more easily.



Figure 3: ABF as applied to $NH_3$ in AmtB. (A) PMF as a function of $NH_3$ position relative to the protein's center. (B) Close-up on the interior of the channel. The locations of the three minima in the PMF are indicated by the blue and white $NH_3$ molecules, with the darker red spheres showing the crystallized positions. Notice the near perfect overlap. Key residues at the gates and in the channel are also highlighted.

# 4  Determining the PMF with Umbrella Sampling

Now that we've determined the PMF along the channel axis for $NH_3$ using ABF, we'd like to verify this result with another method. Here we will use umbrella sampling (US) to calculate the same PMF in the previous section. We will also make use of replica exchange molecular dynamics (REMD) to get better sampling along the reaction coordinate.

> **Windows or Windows<sup>TM</sup>?.** Due to the significant amount of command-line, Unix-based tools, it is recommended to only proceed with this section if using Linux or Mac.

> **US vs. REMD-US.** While this section is focused on running replica-exchange MD combined with umbrella sampling, one can also run normal umbrella sampling if sufficient computational resources for REMD-US aren't available. Some steps are essentially unchanged between the two methods (although beware following the instructions mechanically without any thinking!), others are labeled as **REMD-US only** or **US only**.

## 4.1  Umbrella sampling

Umbrella sampling consists of running separate "windows" of the reaction coordinate simultaneously. This parallelization can greatly increase the efficiency of the overall simulation, limited mostly by the number of processors available. First, the reaction coordinate space is divided into windows, and, for each independent simulation, the system is allowed to only sample within that window. By allowing neighboring windows to overlap, and ensuring there are enough windows to cover the entire reaction coordinate space, a PMF can be calculated by properly combining the data from each window, giving statistical weighting to the overlapping regions between windows. This is typically achieved by applying a harmonic restraint to the reaction coordinate:

$$U_i(\xi) = k_i(\xi - \xi_{0,i})^2, \tag{1}$$

with the equilibrium position, $\xi_{0,i}$, of the harmonic restraint defining the center of window $i$, and the force constant, $k_i$, defining the corresponding width. Generally, one decides *a priori* the centers and widths of all the windows and then chooses the corresponding equilibrium positions and force constants of the harmonic restraints.

> **Defining the windows.** Some care must be taken when defining the centers and widths of windows as high energy barriers can push windows away from their designated centers unless sufficiently high force constants are used. We will ignore this for now and simply define our windows to be equally spaced with identical force constants.

## 4.2   Replica exchange molecular dynamics

In order to ensure more uniform sampling along the range of the reaction co-ordinate(s), different replicas of the same system can be exchanged in order to increase the number of different conformations each replica can occupy through-out the simulation. By changing parameters between different replicas, each replica may find conformations and states that could be inaccessible for another replica due to high energy barriers along the reaction coordinate. Exchanging replicas with one another allows each replica to move around energy barriers through other degrees of freedom (DOF) orthogonal to the reaction coordinate, thus allowing them to sample states which may otherwise be thermodynamically inaccessible. Not only does this ensure more uniform global sampling along both the reaction coordinate and orthogonal DOF, but it also increases the speed at which all states are sufficiently sampled, thus increasing the efficiency of the overall simulation and improving statistics.

First, different replicas are defined. Originally, different replicas were sim-ply simulated at different temperatures (35), and this is generally known as temperature exchange, or in some instances, parallel tempering. One would have a simulation at the desired temperature – say $300\,\mathrm{K}$ – with other replicas of that system at progressively higher temperatures – maybe at $350\,\mathrm{K}$, $400\,\mathrm{K}$, and $450\,\mathrm{K}$. One would allow neighboring windows to exchange with one another periodically, e.g. swapping the temperatures of the $300\,\mathrm{K}$ and $350\,\mathrm{K}$ replicas, then resuming the respective simulations at the new temperature. The proba-bility that two replicas, $i$ and $j$, will exchange with one another is given by the metropolis criterion:

$$p_{ij} = \min\left(1, \frac{\exp(-\frac{E_j}{kT_i} - \frac{E_i}{kT_j})}{\exp(-\frac{E_i}{kT_i} - \frac{E_j}{kT_j})}\right), \tag{2}$$

where the exponential in the numerator is the Boltzmann weighted energies after the replicas are exchanged, and the denominator the energies before the exchange (**pay attention to the indices!**). Therefore, if the combined energy is lower after the exchange, the replicas will always exchange. Otherwise, they will exchange with some Boltzmann weighted probability. In the end, only the data for the $300\,\mathrm{K}$ simulation would be used for further analysis.

For our simulations, we will use a slightly different version of the replica exchange described above. Instead of temperature exchange, we will use Hamil-tonian exchange (36). The Hamiltonian to be exchanged, in our case, will be the US harmonic potential given in Eq. 1. Each replica will be associated with a force constant, $k_i$, and equilibrium position, $\xi_{0,i}$. Using the same metropolis criterion (Eq. 2), but with the temperatures all equal ($T_1 = T_2 = \ldots$), we allow replicas to exchange between neighboring biasing potentials. Also, we will use the data from *all* replicas, instead of a single replica as is done in temperature exchange.

## 4.3   Setting up the US windows

**1** As stated previously, the reaction coordinate used for the US simulations will be the same reaction coordinate used for the ABF simulations, i.e. the $z$ distance along the channel axis. The bounds of the reaction coordinate will be kept the same as well, i.e. $z \in [-13, 5]$. Windows will be spaced 1 Å apart, centered at 5 Å, 4 Å,..., -12 Å, -13 Å. We can use the SMD trajectory previously calculated to generate the starting states (PDBs) for these windows by looking for states near the window centers. For convenience, the naming scheme employed will designate `window0.pdb` to be near 5 Å, `window1.pdb` near 4 Å, etc. There should be a total of 19 windows, i.e., windows 0 to 18. An example of a script to find these states from a trajectory and create the appropriate binary coordinate and extended system configuration files is given in `example-input/get-windows.tcl`.

> **Generating the starting states.** Note that one has to be very careful to ensure the starting states are well equilibrated. Using an SMD trajectory may be convenient, but runs the risk of an accumulation of perturbations to the system over the course of the pulling. Often, a better way is to place the solute in each window center, thus generating only a local perturbation.

**2** A colvar file similar to the one used in the ABF simulations will be used for the US simulations with one notable change at the bottom:

```
harmonic {
    name Z
    colvars Translocation
    centers 0.0
    forceConstant 0.025;# 2.5 * 0.1^2 = 0.025
}
```

The key difference here is instead of an `abf` block, there is a `harmonic` block, which defines the harmonic potential to be applied to the collective variable. For the `forceConstant` parameter, note that the force constant given to NAMD is the actual force constant scaled by the square of the `width` of the collective variable. For 1-Å windows, a force constant of 2.5 kcal/mol·Å$^2$ will give us good overlap between neighboring windows. Since the width of the `Translocation` colvar is 0.1 Å, the force constant given here will be $2.5 \times 0.1^2 = 0.025$. For a simple US simulation (without REMD), a colvar file would be written for each window, with the `centers` parameter set to the harmonic center for that window, then each window would be run independently. Post-processing of the data for all simulations would then produce the PMF. However, the replica exchange algorithm periodically updates the `centers` parameter to account for the exchange between windows, so for now this parameter can simply be set to 0.0. **For US-only simulations (without REMD), see below.**

## 4.4 Preparing the simulations

**1** Move to the **REMD** directory. Create two new directories, **input** and **output**. In the **output** directory, create directories **0** through **18**. To create the output directory, there is a convenient shell script distributed with NAMD in the `lib/replica/` directory called `make_output_dirs.sh`. To save time, the script is also included as part of the tutorial files. To use it, type

```
../common/make_output_dirs.sh output 19
```

Copy the minimization restart files (the `*.coor`, `*.vel`, and `*.xsc` files) into the **input** directory. Rename these files to adhere to the following naming convention: `AmtB-<window #>.restart.<extension>`. An easy way to do this would be with the simple `bash` script found in `example-input/rename_windows.bash`.

This will create links, so that the original files are unaltered. Also add the PSF and PDB files of the AmtB system, as well as the restraints file (`rest6.ref`), to the **input** directory. Example input files are provided for you if you do not have an SMD trajectory available.

**2** **REMD-US only:** In the **REMD** directory, there are several files. These include any CHARMM parameter files, as well as the colvars input file, `US-Z.in`. There are also four NAMD files (3 `.conf` files and 1 `.namd` file):

- The first file is called `AmtB_base.conf`. This contains most of the NAMD parameters that will be the same for every window, such as the structure, parameter files, periodic boundaries, etc.

- Next is `AmtB-remd.conf`, which contains the following parameters:
  ```
  set num_replicas 19
  set temperature 310
  set steps_per_run 1000
  set num_runs 500
  # num_runs should be divisible by runs_per_frame * frames_per_restart
  set runs_per_frame 10
  set frames_per_restart 5
  set namd_config_file "AmtB_base.conf"
  set output_root "output/%s/AmtB-A" ; # directories must exist
  set pdbRoot "input/start.%d.pdb"
  ```

  This file sets the number of windows/replicas (`num_replicas`), the number of steps between exchange attempts (`steps_per_run`), as well as the total number of attempts (`num_runs`). There are also two procedures which tell NAMD the specifics of how to perform the replica exchange. The first procedure, `replica_bias`, returns the `centers` corresponding to each window:

```
proc replica_bias { i } {
    set c [expr 5.0 - $i]
    return [list Z "centers $c"]
}
```

The second procedure, `replica_neighbors`, tells NAMD which replicas can exchange with each other. For our purposes, we allow only closest neighbors to switch:

```
proc replica_neighbors { i } {
    set n 19
    if { $i % 2 } { set s -1 } { set s 1 }
    set result {}
    foreach dx { $s 0 -$s } {
        set j [expr $i + $dx]
        if { $j < 0 || $j >= $n } {
            lappend result $i ;# swap with self
        } {
            lappend result $j ;# swap with neighbor
        }
    }
    return $result
}
```

For example, replica 3 can exchange with replicas 2 and 4, whereas replica 0 can only exchange with replica 1 (and replica 18 can only exchange with replica 17). Since each replica has at most two neighbors (for one reaction coordinate) and can only exchange once per attempt, the REMD algorithm alternates between which neighbor it can exchange with. For example, the first exchange attempts might be between 0 and 1, 2 and 3, 4 and 5, etc., whereas the second exchange attempts will be between 1 and 2, 3 and 4, 5 and 6, etc., so that the number of exchange attempts for each exchange pair is half the total number of exchange attempts.

- The last configuration file is `job0.conf`. This configuration file is the one that is fed into NAMD to start the simulation.

- There is also a file called `umbrella-mod_AmtB.namd`. This file uses the procedures and parameters defined in `AmtB-remd.conf` to create and alter the individual configuration and colvars input files (derived from `AmtB_base.conf` and `US-Z.in`, respectively) that each replica will use to run its own simulation. Based on our `replica_bias` and `replica_neighbors` procedures, `umbrella-mod_AmtB.namd` will adjust the `centers` parameter of `US-Z.in` for each replica. It also calculates the metropolis criterion to determine if an exchange is accepted or rejected. The only notable changes from the original one distributed with NAMD in `lib/replica/umbrella.namd` and

the one here are the locations of `coordinates` and `bincoordinates`, `binvelocities`, and `extendedSystem`.

**3 US only:** For those not running REMD simulations, there is a directory named `USonly/`. Just as above, one creates an `output/` directory with 19 sub-directories in it. In each, a separate configuration file and colvars input file with the appropriate harmonic restraint for that window must be written. To simplify the process, a script, `setup-windows.tcl` is provided; this script copies the configuration and colvars input files, `win-base.conf` and `US-base.in` respectively, to each directory, adding the appropriate window-specific lines. After reading the script to ensure you understand how its parameters are set, run it simply by typing

```
tclsh setup-windows.tcl
```

## 4.5   Running the simulations

**1 REMD-US only:** REMD runs all individuals simulations simultaneously, in parallel. Thus, running the REMD simulation requires a number of processors equal to a multiple of the total number of replicas. Thus, significant computational resources are often required, unless the number of replicas is sufficiently small. Running REMD through NAMD also requires a version of NAMD compiled with the `replicas` option. If these criteria are met, running the REMD simulation is essentially the same as running a normal NAMD simulation, except with the additional `replicas` option:

```
namd2 +replicas <number of replicas> +p<N*number of replicas> \
  job0.conf +stdout output/%d/$NAME.%d.log
```

Additional simulations can be run using the `job1.conf` file. As `job0.conf` is running, it outputs restart `.tcl` files in the **output** directory. These restart files can then be called by subsequent jobs, i.e. `job1.conf` calls `output/AmtB-A.job0.restart500.tcl`.

**2 US only:** For US-only simulations, one needs to run each window individually. Although it is likely you will run them on a cluster with your own specific job-submission scripts, the simplest way to run a window is to navigate to its directory and type

```
namd2 +p<number of cores> win0A.conf > win0A.log
```

---

💡 **Directories.** It is important to note that the output directories (`output/0` to `output/18`) must already exist prior to running NAMD. An error will result otherwise.

## 4.6   Analyzing the simulations

If computational resources are limited, example output is provided for you in order to perform the analysis.

**1 REMD-only:** Since replicas have exchanged with one another, they may not be in the window in which they originally started. Because of this, we must "sort" the replicas back into their proper windows. This is done using the `sortreplicas` command, which is a binary provided with NAMD in the same directory. `sortreplicas` requires three arguments: (1) the output job name, (2) the total number of replicas, and (3) the number of runs per dcd frame (`runs_per_frame` parameter in `AmtB-remd.conf`). For our system, this is done by typing the following in the `output` directory:

  `sortreplicas %d/AmtB-A.job0 19 10`

This will output `*.sort.*` files into the numbered **output** directories. It is these sorted files which will be used to generate the PMF.

**2** We will use the Weighted Histogram Analysis Method (WHAM) (37) to construct the PMF from the sorted trajectory files. A script, `write-input.tcl`, as well as the `wham` program, is provided for you in the **example-output** directory. Copy `write-input.tcl` into the **output** directory. To run this script, simply type in the terminal

  `tclsh write-input.tcl <nanoseconds>`

where <nanoseconds> is the number of nanoseconds per window used to calculate the PMF. This will produce an input file, `INPUT_1-<nanoseconds>` (where "1" is the equilibration time that is ignored - see the Info box below), which tells WHAM which trajectory files to use as well as the corresponding harmonic force constant and equilibrium position for that trajectory file. It also tells WHAM the bounds in which to calculate the PMF, as well as the resolution of the PMF and the simulation temperature. The first line of the INPUT file is the necessary terminal command for running WHAM with that INPUT file. For example, to calculate the PMF for 5 ns/window you would type the following:

  `wham -13 5 180 0.0001 310.0 0 INPUT_1-5ns AmtB_REMD_US_1-5ns.pmf`

> 💡 **Force constants.**   WHAM requires the actual, unscaled force constant, not the rescaled force constant given in the colvars file.   Look for the following line in the NAMD log file to confirm: `The force constant for colvar "Translocation" will be rescaled to 2.5 according to the specified width.`

The calculated PMF is outputted to the file `AmtB_REMD_US_1-5ns.pmf`. Note that although the PMF may look slightly different, we can still identify minima associated with the three ammonia sites (Am2, Am3 and Am4) previously identified. An example PMF is plotted in Figure 4 along with the PMF calculated from ABF. The disagreement comes from insufficient sampling (only a few ns
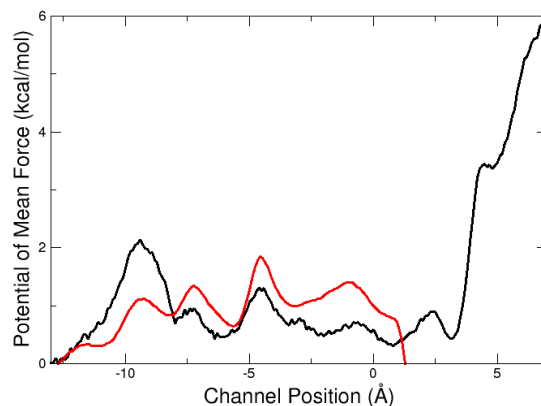
Figure 4: PMF calculated from ABF (black line) and REMD-US (red line). REMD-US PMF adjusted so that the 0 point matches the ABF PMF. Each REMD-US window was simulated for 5 ns, with the PMF calculated using the last 4 ns.

for each case). Although not a guarantee of convergence, one should not stop running the US simulations as long as the PMF is still changing.

> **Equilibration time.** The first few nanoseconds in each window are often treated as an equilibrium phase, and as such are ignored for post-processing. This is done primarily to remove any non-equilibrium effects that may "contaminate" the PMF. For smaller systems such as this, throwing out the first nanosecond of data is generally sufficient, which was already done for you in the example output. As an additional exercise, recalculate the PMF **without** ignoring the first nanosecond to see the how the PMF changes. Plot the PMF with the first nanosecond included and compare it to the PMF with it ignored.

## 4.7   Sampling of reaction coordinate space

An important aspect of any free-energy calculation method is the exploration of the complete reaction space. Uniform sampling at every point along your reaction coordinate not only improves the accuracy of the produced PMF, but also leads to quicker convergence. Hamiltonian replica exchange should improve the uniformity of the sampling along the reaction coordinate. In order to take full advantage of this there needs to be a good exchange between windows. In this section we will examine the sampling along the channel as well as the exchange rates between neighboring windows.
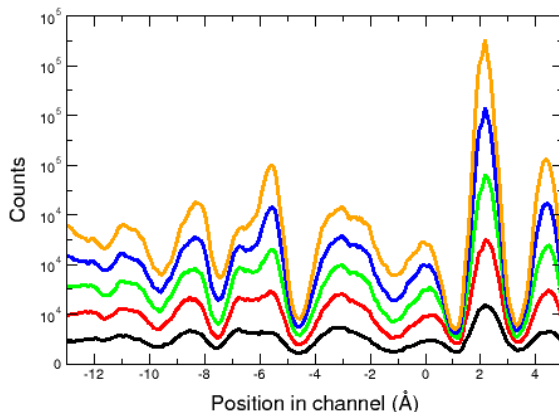
Figure 5: Histograms calculated from the trajectory data for the REMD-US simulations for 1 ns/window (black), 2 ns/window (red), 3 ns/window (green), 4 ns/window (blue), and 5 ns/window (orange).

1. First, using the `traj` files, plot histograms of the sampling along the channel distance reaction coordinate for every nanosecond and examine how the sampling is accumulating over time. The example output provided was used to create the histogram seen in Figure 5.

2. Next, we want to look at the exchange rates between windows. Ideally, neighboring windows should exchange 30-40% of the time. NAMD outputs the accepted exchanges into the main `log` file, as well as the exchange ratios for each neighbor pair. These can be found by searching the `log` file for `EXCHANGE_ACCEPT` and `EXCHANGE_RATIO`, respectively. Plot the exchange ratios for each pair as well as the average exchange rate for all pairs. The exchange rates for the example output is plotted in Figure 6.

3. Not only do we want good sampling, we also want to ensure reversibility. What this means in the terms of replica exchange is that a single replica has traversed the entire reaction coordinate space and returned to its original window. The unsorted `history` files contain the window number for each replica at each exchange attempt. Check these files, or the unsorted `traj` files, to see how much of the reaction coordinate space was explored by each replica. Figure 7 shows the motion of all 19 replicas for the example output provided.

Does the sampling accumulate uniformly? Are there regions where sampling is very low? If so, how do these low sampling regions manifest in the calculated PMF? In Figure 4, we can see a large deviation in the REMD-US PMF from
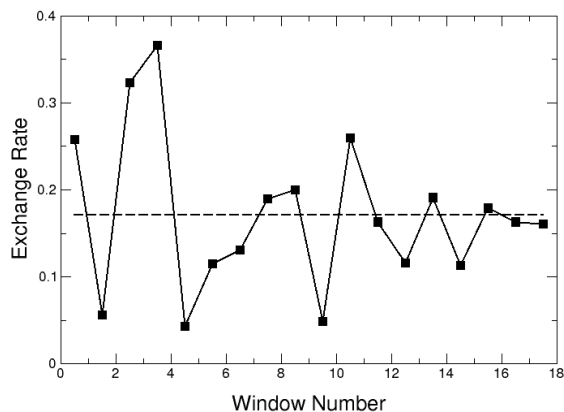
Figure 6: Exchange rates for neighboring windows after 5 ns/window, plotted as the midpoint between the windows (squares, solid line), as well as the average exchange rate (dashed line).
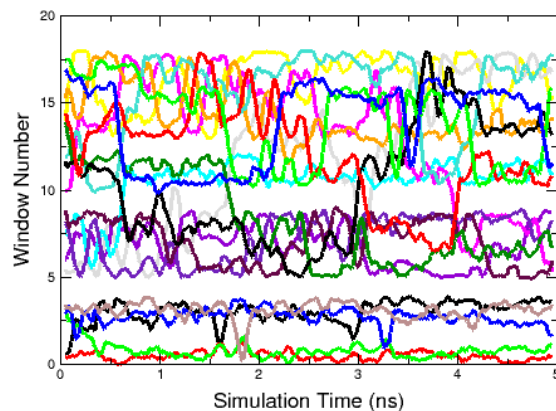


Figure 7: Average window number of each replica over time.

the ABF PMF around 0-2 Å that corresponds to a region of very low sampling. Similarly, we can see that the exchange rate between windows 4 and 5 (corresponding to window centers of 1 Å and 0 Å, respectively) is also very low (∼5%). There is also a distinct gap between windows 4 and 5 in Figure 7, suggesting little to no movement between these windows. Therefore, reversibility is not

guaranteed.

Although replica exchange has yielded better sampling, given the majority of the replica movement seen in Figure 7, there are still regions where sampling is insufficient, and further simulation time seems unlikely to remedy this problem, as the exchange rates plotted in Figure 6 appear to be converged after 5 ns/window. In such cases, additional replicas are often required. Try adding intermediate windows between existing windows in which there is insufficient sampling or low exchange, such as between windows 4 and 5, and observe how the PMF changes.

# Acknowledgements

# References

[1] Izrailev, S., S. Stepaniants, M. Balsera, Y. Oono, and K. Schulten. 1997. Molecular dynamics study of unbinding of the avidin-biotin complex. *Biophys. J.* 72:1568–1581.

[2] Sotomayor, M. and K. Schulten. 2007. Single-molecule experiments in vitro and in silico. *Science.* 316:1144–1148.

[3] Jarzynski, C. 1997. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.* 78:2690–2693.

[4] Jarzynski, C. 1997. Equilibrium free-energy differences from nonequilibrium measurements: A master equation approach. *Phys. Rev. E.* 56:5018–5035.

[5] Hummer, G. and A. Szabo. 2001. Free energy reconstruction from nonequilibrium single-molecule pulling experiments. *Proc. Natl. Acad. Sci. USA.* 98:3658–3661.

[6] Park, S., F. Khalili-Araghi, E. Tajkhorshid, and K. Schulten. 2003. Free energy calculation from steered molecular dynamics simulations using Jarzynski's equality. *J. Chem. Phys.* 119:3559–3566.

[7] Park, S. and K. Schulten. 2004. Calculating potentials of mean force from steered molecular dynamics simulations. *J. Chem. Phys.* 120:5946–5961.

[8] Darve, E. and A. Pohorille. 2001. Calculating free energies using average force. *J. Chem. Phys.* 115:9169–9183.

[9] Hénin, J. and C. Chipot. 2004. Overcoming free energy barriers using unconstrained molecular dynamics simulations. *J. Chem. Phys.* 121:2904–2914.

[10] Darve, E., D. Rodríguez-Gómez, and A. Pohorille. 2008. Adaptive biasing force method for scalar and vector free energy calculations. *J. Chem. Phys.* 128:144120.

[11] Hénin, J., G. Forin, C. Chipot, and M. L. Klein. 2010. Exploring multidimensional free energy landscapes using time-dependent biases on collective variables. *J. Chem. Theor. Comp.* 6:35–47.

[12] Jensen, M. Ø., S. Park, E. Tajkhorshid, and K. Schulten. 2002. Energetics of glycerol conduction through aquaglyceroporin GlpF. *Proc. Natl. Acad. Sci. USA*. 99:6731–6736.

[13] Ilan, B., E. Tajkhorshid, K. Schulten, and G. A. Voth. 2004. The mechanism of proton exclusion in aquaporin channels. *Proteins: Struct., Func., Bioinf.* 55:223–228.

[14] Wang, Y., K. Schulten, and E. Tajkhorshid. 2005. What makes an aquaporin a glycerol channel: A comparative study of AqpZ and GlpF. *Structure*. 13:1107–1118.

[15] Chen, H., B. Ilan, Y. Wu, F. Zhu, K. Schulten, and G. A. Voth. 2007. Charge delocalization in proton channels. I. The aquaporin channels and proton blockage. *Biophys. J.* 92:46–60.

[16] Wang, Y. and E. Tajkhorshid. 2008. Electrostatic funneling of substrate in mitochondrial inner membrane carriers. *Proc. Natl. Acad. Sci. USA*. 105:9598–9603.

[17] Celik, L., B. Schiott, and E. Tajkhorshid. 2008. Substrate binding and formation of an occluded state in the leucine transporter. *Biophys. J.* 94:1600–1612.

[18] Yin, Y., M. Ø. Jensen, E. Tajkhorshid, and K. Schulten. 2006. Sugar binding and protein conformational changes in lactose permease. *Biophys. J.* 91:3972–3985.

[19] Jensen, M. Ø., Y. Yin, E. Tajkhorshid, and K. Schulten. 2007. Sugar transport across lactose permease probed by steered molecular dynamics. *Biophys. J.* 93:92–102.

[20] Gumbart, J., M. C. Wiener, and E. Tajkhorshid. 2007. Mechanics of force propagation in TonB-dependent outer membrane transport. *Biophys. J.* 93:496–504.

[21] Gumbart, J. and K. Schulten. 2006. Molecular dynamics studies of the archaeal translocon. *Biophys. J.* 90:2356–2367.

[22] Gumbart, J. and K. Schulten. 2007. Structural determinants of lateral gate opening in the protein translocon. *Biochemistry*. 46:11147–11157.

[23] Gumbart, J. and K. Schulten. 2008. The roles of pore ring and plug in the SecY protein-conducting channel. *J. Gen. Physiol.* 132:709–719.

[24] Henin, J., E. Tajkhorshid, K. Schulten, and C. Chipot. 2008. Diffusion of glycerol through *Escherichia coli* aquaglyceroporin GlpF. *Biophys. J.* 94:832–839.

[25] Dehez, F., E. Pebay-Peyroula, and C. Chipot. 2008. Binding of ADP in the mitochondrial ADP/ATP carrier is driven by an electrostatic funnel. *J. Am. Chem. Soc.* 130:12725–12733.

[26] Ivanov, I., X. Cheng, S. M. Sine, and J. A. McCammon. 2007. Barriers to ion translocation in cationic and anionic receptors from the cys-loop family. *J. Am. Chem. Soc.* 129:8217–8224.

[27] Lin, Y., Z. Cao, and Y. Mo. 2006. Molecular dynamics simulations on the *Escherichia coli* ammonia channel protein AmtB: Mechanism of ammonia/ammonium transport. *J. Am. Chem. Soc.* 128:10876–10884.

[28] Luzhkov, V. B., M. Almlöf, M. Nervall, and J. Åqvist. 2006. Computational study of the binding affinity and selectivity of the bacterial ammonium transporter AmtB. *Biochemistry.* 45:10807–10814.

[29] Nygaard, T. P., C. Rovira, G. H. Peters, and M. Ø. Jensen. 2006. Ammonium recruitment and ammonia transport by *E. coli* ammonia channel AmtB. *Biophys. J.* 91:4401–4412.

[30] Lamoureux, G., M. L. Klein, and S. Bernèche. 2007. A stable water chain in the hydrophobic pore of the AmtB ammonium transporter. *Biophys. J.* 92:L82–L84.

[31] Bostick, D. L. and C. L. B. III. 2007. Deprotonation by dehydration: The origin of ammonium sensing in the AmtB channel. *PLoS Comput. Biol.* 3:e22.

[32] Yang, H., Y. Xu, W. Zhu, K. Chen, and H. Jiang. 2007. Detailed mechanism for AmtB conducting $NH_4^+/NH_3$: Molecular dynamics simulations. *Biophys. J.* 92:877–885.

[33] Khademi, S., J. O. III, J. Remis, Y. Robles-Colmenares, L. J. W. Miericke, and R. M. Stroud. 2004. Mechanism of ammonia transport by Amt/MEP/Rh: Structure of AmtB at 1.35 Å. *Science*:1587–1594.

[34] Phillips, J. C., R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten. 2005. Scalable molecular dynamics with NAMD. *J. Comp. Chem.* 26:1781–1802.

[35] Sugita, Y. and Y. Okamoto. 1999. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* 314:141–151.

[36] Sugita, Y., A. Kitao, and Y. Okamoto. 2000. Multidimensional replica-exchange method for free-energy calculations. *J. Chem. Phys.* 113:6042–6051.

[37] Kumar, S., J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman. 1992. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *J. Comp. Chem.* 13:1011–1021.