Computing Absolute Binding Affinities by Streamlined Alchemical Free Energy Perturbation (SAFEP)

Ezry Santiago-McRae[1†], **Mina Ebrahimi**[2,3,4†], **Jesse W. Sandberg**[1], **Grace Brannigan**[1,5‡], **Jérôme Hénin**[3,4‡]

[1]Center for Computational and Integrative Biology, Rutgers University, Camden, New Jersey, 08102; [2]Department of Physics, Sharif University of Technology, Tehran, Iran; [3]Université Paris Cité, Laboratoire de Biochimie Théorique, CNRS UPR 9080, 75005, Paris, France; [4]Institut de Biologie Physico-Chimique – Fondation Edmond de Rothschild, PSL Research University, Paris, France; [5]Department of Physics, Rutgers University, Camden, New Jersey, 08102

This LiveCoMS document is maintained online on GitHub at https://github.com/jhenin/SAFEP_tutorial; to provide feedback, suggestions, or help improve it, please visit the GitHub repository and participate via the issue tracker.

This version dated July 27, 2023

Abstract Free Energy Perturbation (FEP) is a powerful but challenging computational technique for estimating differences in free energy between two or more states. This document is intended both as a tutorial and as an adaptable protocol for computing free energies of binding using free energy perturbations in NAMD. We present the Streamlined Alchemical Free Energy Perturbation (SAFEP) framework. SAFEP shifts the computational frame of reference from the ligand to the binding site itself. This both simplifies the thermodynamic cycle and makes the approach more broadly applicable to superficial sites and other less common geometries. As a practical example, we give instructions for calculating the absolute binding free energy of phenol to lysozyme. We assume familiarity with standard procedures for setting up, running, and analyzing molecular dynamics simulations using NAMD and VMD. While simulation times will vary, the human tasks should take no more than 3 to 4 hours for a reader without previous training in free energy calculations or experience with the VMD Colvars Dashboard. Sample data are provided for all key calculations both for comparison and readers' convenience.

***For correspondence:**
grace.brannigan@rutgers.edu (GB); jerome.henin@cnrs.fr (JH)

[†]These authors contributed equally to this work
[‡]These authors played an equal role in supervising this work

1 Introduction

In this tutorial we are principally concerned with computing the Absolute Binding Free Energy (ABFE) of a ligand to its receptor. While many methods of measuring free energies exist, alchemical Free Energy Perturbation (FEP) methods make use of the fact that, since the change in free energy is path independent, it can be calculated via an unphysical path. In the case of FEP, that unphysical path is defined by scaling the

non-bonded interactions of the ligand. [1–6] In essence, the user can make a bound ligand "disappear" from the binding site, make it re-appear in the bulk solution, and calculate the corresponding free energy difference.

While elegant and exact in principle, FEP calculations are often unwieldy in practice. One of the most stubborn challenges that most FEP implementations face is that the ligand must maintain the original bound configuration during decoupling, even as the very interactions that stabilize the bound configuration are weakened. Consequently, absolute binding schemes introduce restraints on the ligand to mimic the interactions that stabilize the bound ensemble. While there are many such schemes to accelerate convergence,[7–10] most do so in ways that require error-prone manual parameterization and many hours of the user's time. Many of these techniques have already been reviewed by Mey et al. [2] As discussed there, no restraint scheme is without cost; the restraints themselves will bias the estimated free energy and must be corrected in the coupled state, the decoupled (gas phase) state, or both. Conveniently, restraints which have simple geometries *and* do not affect the coupled state can be accounted for analytically. On the other hand, numerical approaches must be used to correct for restraints (such as an RMSD [10] or DBC collective variable [11]) that restrict the accessible configuration space to an irregular volume. However, if the restraint potential on the collective variable is chosen to have negligible impact on the coupled state (such as a flat well potential [12]), these calculations can be done cheaply in vacuum as illustrated in this tutorial. When more intrusive restraints (such as harmonic potentials [9, 10]) are applied in the coupled state, it is necessary to calculate numerical corrections in the coupled state regardless of the restraint geometry.

Streamlined Alchemical Free Energy Perturbation (SAFEP) is specifically designed to make FEP calculations faster and easier for the user without sacrificing accuracy of the final free energy estimate. SAFEP reduces conceptual and computational complexity by replacing conventional rotational and translational restraints for stabilizing the ligand in the binding site with a single Distance-to-Bound Configuration (DBC) coordinate. This makes setup easier because, instead of parameterizing six or more individual restraints, the user need only parameterize one. Furthermore, SAFEP can handle superficial binding sites in phase-separated bulk with atomistic resolution. [11] Statistically optimal FEP estimators require both decoupling and recoupling calculations; SAFEP uses Interleaved Double-Wide Sampling (IDWS) to extract both quantities from the same calculation, roughly halving the required simulation time. [13] (see the NAMD User Guide Free Energy Perturbation) SAFEP makes extensive use of the Colvars Dashboard in VMD, allowing the user to easily measure collective variables, impose biases, and generate restraint configuration files from one interface [14]. Finally, analysis tools and data visualizations are included in one Jupyter notebook allowing for comprehensive quality assurance along with the ΔG calculation.

Figure 1 depicts the thermodynamic paths at the heart of SAFEP. The desired quantity $\Delta G^{\circ}_{\text{bind}}$ (red, left column) is equal to the sum of the steps in the SAFEP method (black, right column), as shown in equation 1.

$$\Delta G^{\circ}_{\text{bind}} = -\Delta G^{*}_{\text{site}} + \Delta G_{\text{DBC}} - \Delta G^{\circ}_{\text{V}} + \Delta G^{*}_{\text{bulk}} \tag{1}$$

This equation forms the basis for the steps that follow in this tutorial.

1.1 Scope

The following steps will walk the user through the calculation of an Absolute Binding Free Energy (ABFE) using a computationally affordable example (phenol bound to a mutant lysozyme), but we have written these steps to be straightforward to generalize to other systems. These exact steps have been tested thoroughly for this particular system. More detailed discussions of each step can be found in Appendices A-E. To facilitate generalization of the method to other systems, we have provided additional troubleshooting advice in Appendix F, as well as example files for each step. These appendix entries are hyperlinked and referenced throughout the body of the tutorial.

1.2 Prerequisites

1.2.1 Background knowledge

We assume intermediate experience with running classical MD (any software) and beginner experience with NAMD 2.14 or later. If this is not the case, please see the NAMD Tutorial [15]. The latter portions that involve analysis are less important for this tutorial. Useful, but not required, material on alchemical free energy perturbations can be found in "*In-silico* alchemy: A tutorial for alchemical free-energy perturbation calculations with NAMD" [7]. Finally, basic knowledge of VMD and Python will be required for data analysis and visualization.

1.2.2 Software requirements

1. NAMD2 version 2.14 or later, or NAMD3 version 3.0b3 or later. The NAMD3 series enables GPU-accelerated alchemical simulations. These instructions contain the command "namd2": replace with "namd3" if applicable.
2. **VMD 1.9.4.a57** or later. Slightly older versions of VMD may be used, but will require manual update of the Colvars Dashboard. See the Colvars Dashboard README for more information on getting the latest version.

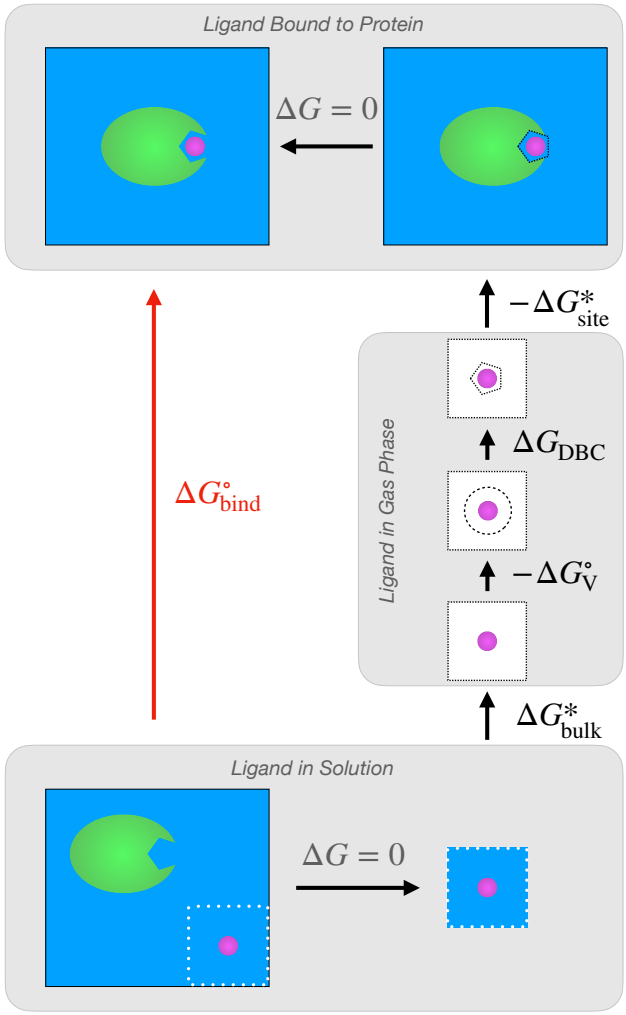


Figure 1. The SAFEP thermodynamic cycle. Computing the ABFE of a ligand bound to a protein ($\Delta G^\circ_{\text{bind}}$) is the ultimate goal. This is found by computing the free energies of several, smaller perturbations: 1) decoupling the unbound ligand from the condensed phase to the gas phase under no restraints (ΔG^*_{bulk}); 2) enforcing a restraint scheme ($-\Delta G^\circ_{\text{V}}$ and ΔG_{DBC}); and 3) coupling the ligand from the gas phase to its bound pose in the condensed phase ($-\Delta G^*_{\text{site}}$) under the Distance-to-Bound Configuration (DBC) restraint. The free energy contribution of the volumetric restraint ($\Delta G^\circ_{\text{V}}$) is calculated analytically, while the other three contributions are calculated via simulation.The free energy of the top horizontal leg vanishes in SAFEP due to design of the DBC restraint. See Appendix C for more details.

3. Python 3.9.12 or later
4. Jupyter
5. safep Python package and its dependencies (see Procedure below for installation instructions)

NAMD will be used to perform simulations. GPU acceleration of restrained free energy perturbations are available in NAMD3 (with `CUDASOAintegrate off`) [16, 17]. System setup, trajectory visualization, and restraint definition will be carried out in VMD [18]. Data analysis and visualization will be handled by a Jupyter notebook with the above dependencies.

High-performance computing resources are recommended, but not required. Sample outputs are provided for each step for users with limited compute resources or time.

The Colvars Dashboard is the recommended tool for writing and editing Colvars config files. Advanced users can also edit the text files directly. Example files are provided for each step.

1.3 Process Overview

Within the scope of free energy perturbations, absolute free energies of binding are typically calculated by the double-decoupling method (DDM) [1, 3, 4]. In this method, pair interactions (non-bonded terms) between the ligand and the rest of the simulation box are gradually scaled to zero (decoupled) from both a bound state and an unbound, solvated state.

In order to maintain the ligand in its bound state, most current approaches introduce a series of rotational and translational restraints on the ligand, each of which requires parameterization and an additional ΔG correction. In contrast, SAFEP uses just one restraint: a flat well on the "Distance-to-Bound Configuration" (DBC). This minimizes both the number of parameters to be optimized and reduces the number of simulations to be performed compared to layered restraints (See Figure 2 or Appendix C for details).

The thermodynamic cycle used for absolute binding free energies in SAFEP is shown in Figure 1 while the unknown values (black arrows) can be calculated by the simulations outlined in Figure 3. More precisely, the thermodynamic cycle (Fig 1) and the corresponding simulations (Fig 3) are broken into three main steps involving three simulation systems: 1) the ligand bound to the protein, 2) the ligand in the gas phase, and 3) the ligand in the bulk. The order of computations is unimportant so long as the end-points are defined consistently (e.g. the same temperature is used throughout and restraints are used consistently). For the sake of clarity, we have arranged the process linearly: Steps A and B are concerned with calculating ΔG^*_{site}; step C addresses the free energy of the DBC (ΔG_{DBC}); step D measures ΔG^*_{bulk}; and step E calculates an analytical correction ($\Delta G^\circ_{\text{V}}$) and combines all the preceding terms into the overall $\Delta G^\circ_{\text{bind}}$ using Equation 1.

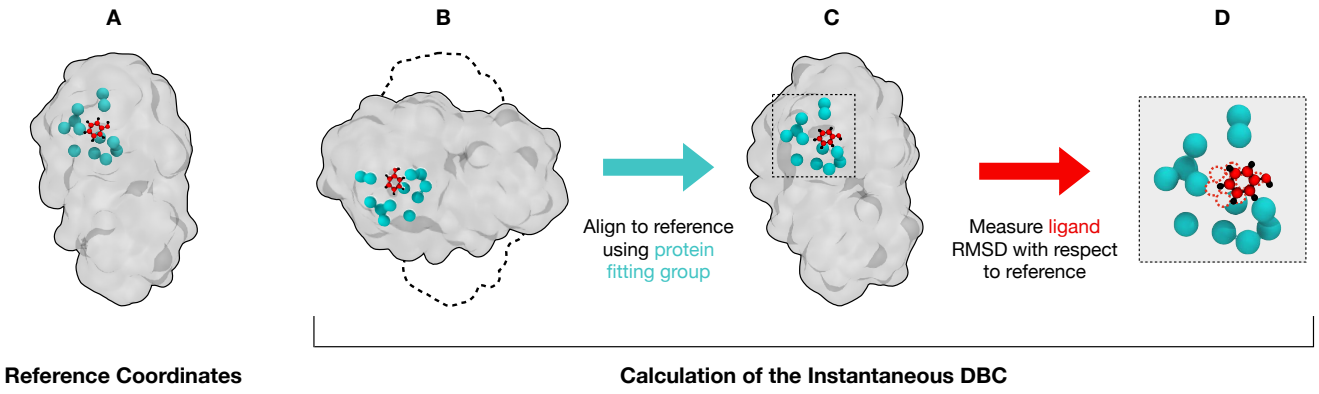


Figure 2. The Distance-to-Bound-Configuration (DBC) coordinate. The DBC coordinate is used as a bias to prevent ligand dissociation during uncoupling. The user specifies a subset of protein atoms as the fitting group (teal) and a subset of ligand atoms (red); also shown are the protein surface (gray) and remaining ligand atoms (black). A) User-specified reference coordinates for both protein and ligand. B) During simulation, both protein and ligand will drift from the reference coordinates (black dashed outline). C) In order to remove rotational and translational diffusion of the protein from calculation of the DBC, Colvars aligns the system to the reference coordinates using only the protein fitting group atoms. D) The DBC is the RMSD of the user-specified ligand atoms (solid) with respect to the reference coordinates (dashed).

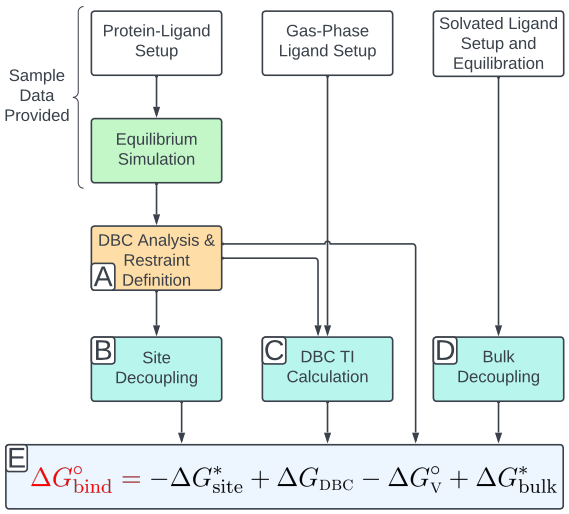


Figure 3. The overall SAFEP workflow. The dependencies in this flowchart can be used to decide in what order steps can be performed, and which simulations can be run simultaneously. From top to bottom and left to right: 1) the ligand must be setup (as for classical MD) in each of the three states (bound, solvated, gas phase) and minimally relaxed (white boxes); 2) a longer, unbiased simulation of the ligand-protein complex is necessary to sample the bound state (green box) which is used to determine the distribution of the DBC (orange box, Step A); 3) two FEP calculations and a thermodynamic integration (TI) calculation are carried out (blue boxes, Step B, Step C, and Step D); and 4) the resulting values are combined to get the standard free energy of binding (gray box; Step E).

2 Protocol

The following steps demonstrate the SAFEP protocol applied to a computationally affordable example: calculating the binding affinity of phenol to a lysozyme mutant. For more details on the rationale behind this choice, see Appendix A.

Procedure:

1. **Clone the SAFEP_tutorial repository to your local environment** Tip: use `--depth 1` to avoid downloading the entire commit history.
   ```
   git clone https://github.com/jhenin/SAFEP_tutorial.git --depth 1
   ```
2. **Navigate to the cloned repo**
   This is the starting path for all command line prompts in this tutorial.
   ```
   cd SAFEP_tutorial
   ```
3. **Install the SAFEP package by running:**
   ```
   pip install git+https://github.com/BranniganLab/safep.git
   ```
4. **Tips:**
   - All run.namd files contain a line that reads `set useSampleFiles 0`. To use the sample data provided, set the value to 1. Otherwise, NAMD will use your inputs (provided they are named exactly as described in this document).
   - If you run VMD and your simulations on different computers, then you will need to manually edit paths later when you are running simulations.
   - Some simulations will take several days on a single core. To use 4 cores in parallel we have included the `+p4` argument in the commands for the longer NAMD runs. This number may need to be optimized for your particular compute resources.
   - If you are using namd3 with CUDA (GPU) acceleration, you should consult the NAMD User Guide [13] for optimal settings. As of NAMD3b3, CUDAOSOAIntegrate should be left OFF, and "+idlepoll" should be included in the run command.
   - Common settings used by multiple simulations are in *common/common_config.namd*, which is sourced by the individual configuration files. This simplifies the individual configuration files and ensures consistency between calculations, which is a critical part of any free energy method.
   - **Old versions of Colvars Dashboard only** Colvars has a functionality called "auto-update selections." Please turn it off by deleting the comment line that begins with "auto-updating" in all Colvars config files. It is off by default in newer CV Dashboard versions.
5. **Move on to Step A**

Step A: Sample the bound state and define the binding restraint

Alchemical decoupling removes the interactions that stabilize the occupied ensemble. Consequently, during decoupling the ligand may spontaneously diffuse into the bulk. Therefore, we need to impose an external restraint to force the ligand to occupy the bound state throughout decoupling. With SAFEP we apply a single restraint on the Distance-to-Bound Configuration (DBC) collective variable as illustrated in Fig. 2. This restraint is straightforward to define and relatively insensitive to small differences in parameters [11]. Its sole correction factor is calculated via Thermodynamic Integration later in this tutorial. For more details about the DBC restraint see Appendix C and [11]. For a discussion of the merits in accounting for symmetry when computing the DBC of a small, symmetric ligand see Appendix C.2.2 and [19].

Required Input:

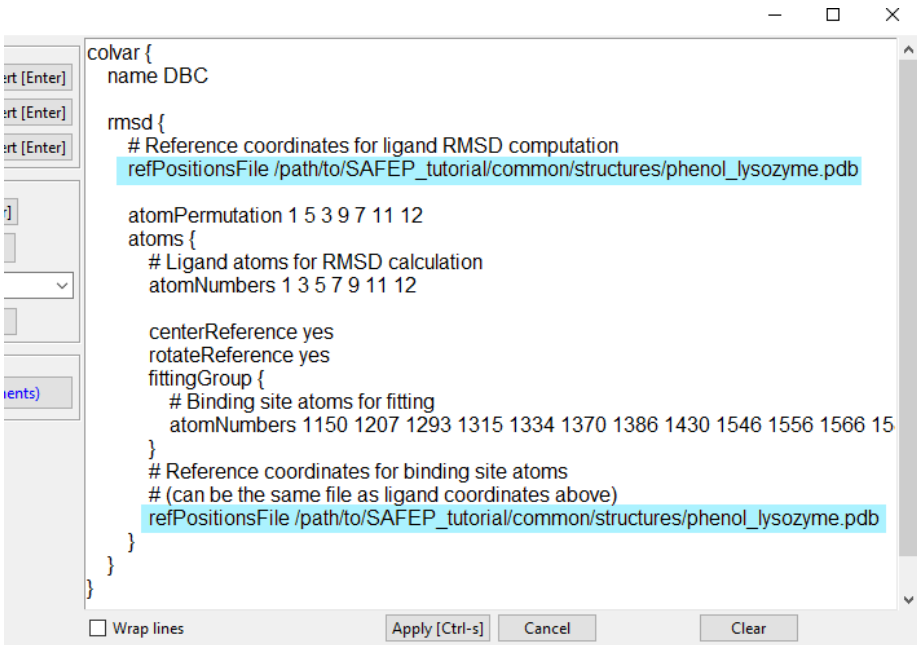
- Structure file: *common/structures/phenol_lysozyme.psf*
- Coordinate file: *common/structures/phenol_lysozyme.pdb*
- Equilibrium trajectory: *stepA_create_DBC/inputs/unbiased-sample.dcd*

Essential Output:

- DBC restraint parameters: *stepA_alchemy_site/outputs/DBC_restraint.colvars*

Procedure:

0. **Run standard MD of the occupied state.**
   This simulation should be long enough (∼50 ns) for the ligand to explore the configuration space of the bound state. See Appendix A for more details. **Note: For this tutorial, we have done this step for you and you may skip to the next step using the trajectory provided.**
1. **Define the Distance-to-Bound Configuration (DBC) Coordinate**
   a. **Open the Colvars Dashboard in VMD:**
      - Open VMD.
      - Load the psf, pdb, and dcd files listed above under "Required Input". You may choose to load your own dcd if you completed Step 0.
      - From VMD's main window options select: *Extensions→Analysis→Colvars Dashboard*
   b. **Create a DBC colvar:**
      - Click `New [Ctrl-n]` to start editing a new collective variable.
      - Delete all sample text shown in the editor on the right-hand side.
      - Open the *Templates→colvar templates* drop-down list and select `DBC (ligand RMSD)` to populate the editor with a template that now needs to be edited.
   c. **Define the atom selection for the ligand atoms: (Fig. 2, red)**
      - Delete `atomNumbers 1 2 3 4` from the `atoms` block and leave your cursor on the now-empty line.
      - Select the left panel text box *Editing helpers→Atoms from selection text* and enter `resname PHEN and noh`.
      - Press Enter or click `Insert [Enter]` to insert the new selection into the configuration text at your cursor.
   d. **Identify equivalent, symmetric atoms:**
      - In the rmsd block, add `atomPermutation 1 5 3 9 7 11 12` to the line above the atoms keyword. This indicates equivalence between ligand atoms listed in `atomNumbers`. That is, (5 and 3) and (9 and 7) are interchangeable. See the Colvars User Guide and Appendix C.2.2 for more details on symmetric DBC and `atomPermutation`.
   e. **Define the atom selection for the binding site atoms: (Fig. 2, teal)**
      - Delete `atomNumbers 6 7 8 9` within the `fittingGroup` block and leave your cursor on the now-empty line.
      - Select the left panel text box *Editing helpers→Atoms from selection text* and enter `alpha and same residue as within 6 of resname PHEN`.
      - Press Enter or click `Insert [Enter]` to insert the new selection into the configuration text at your cursor.
   f. **Set the reference positions for the RMSD and alignment calculations:**
      - In the initial `RMSD` block, before the `atoms` block, delete `refpositionsfile reference.pdb # PDB or XYZ file` (the first highlighted line in the figure below) and leave your cursor on that line.
      - In the left panel under *Editing helpers*, select the radio button ∘ `refPositionsFile` and click `Pick File`.
      - Select the *phenol_lysozyme.pdb* file you used as input for this section. This will insert a line in the dashboard text editor that indicates the file that will be used for the DBC reference coordinates.
      - Copy the line just inserted and replace the `refpositionsfile` line at the bottom of the `atoms` block (the second highlighted line in the figure below). This sets the same PDB file to be used for aligning to the protein frame-of-reference.
      - For NAMD builds older than October 31, 2022: change "centerToReference" and "rotateToReference" to "centerReference" and "rotateReference" respectively.
      - The colvar config editor should now look like the screenshot below with your file's path in place of the two highlighted lines.



g. **Save your edits:**
Click the `Apply [Ctrl-s]` button.

2. **Impose a restraint based on the DBC coordinate**

a. **Determine the Upper Wall of the DBC restraint:**

- In the *Plots and real-time visualizations* panel of the dashboard, click `Histogram`. If you don't see such a button, you need to upgrade your VMD installation. See Software requirements for more details.
- From the histogram, estimate the the 95th percentile of the bound state's DBC coordinate. Use the cumulative distribution line graph as a guide. The value doesn't need to be precise. We selected 1.5 Angstroms. See Appendix C.2.2 for more details.
- Write this value down; you will need it in the next step.

b. **Impose a flat-bottom harmonic potential:**

- Open the *Biases* tab on the Colvars Dashboard and click `New bias [Ctrl-n]` to create a new biasing potential.
- Delete the default text.
- From the *bias templates:* drop-down menu select `harmonicWalls` and click `Insert [Enter]`.
- Modify the bias to match the following parameters:

```
colvars         DBC
upperWalls      [the DBC's 95th percentile just identified]
forceConstant   200
```

The force constant in this case is in units of kcal/(mol·Å^2). The strength of restraint should be neither so great that it causes instabilities nor so weak that it fails to cleanly separate the bound and unbound ensembles.

c. **Save your edits:**
Click the `Apply [Ctrl-s]` button.

3. **Save the Colvars configuration to a file**

a. **Click `Save` at the top of the dashboard**

b. **Save your file to** *stepA_create_DBC/outputs/DBC_restraint.colvars*
Note that if you choose to use a different file name or path you will need to update the files in the next step with the new name.

Step B: Decouple phenol from the protein via FEP

In this section we will calculate ΔG^*_{site} by decoupling the ligand from the protein binding site (and all other contents of the simulation box) using alchemical FEP. We will maintain the ligand in the bound configuration relative to the protein by restraining the DBC coordinate as defined in the previous subsection.

Required Input:

- Structure file: *common/structures/phenol_lysozyme.psf*
- Coordinate file: *common/structures/phenol_lysozyme.pdb*
- DBC restraint parameters: *stepA_create_DBC/outputs/DBC_restraint.colvars*
- NAMD configuration file: *stepB_alchemy_site/inputs/run.namd*

Essential Output:

- FEP configuration file: *stepB_alchemy_site/outputs/alchemy_site.pdb*
- FEP trajectory file: *stepB_alchemy_site/outputs/alchemy_site.dcd*
- FEP output file: *stepB_alchemy_site/outputs/alchemy_site.fepout*

Procedure:

1. **Specify which atoms will be decoupled using the pdb beta field**
   a. **Open VMD and load the psf and pdb files specified in "Required Input".**
   b. **Set and write beta values:**
      - Open the Tk Console from the Extensions menu.
      - Ensure that your Tk Console is in the correct directory:
        `cd stepB_alchemy_site/outputs`
      - Set the beta value of all atoms to 0:
        `[atomselect top all] set beta 0`
      - Set the beta values of the ligand atoms to -1 for decoupling:
        `[atomselect top "resname PHEN"] set beta -1`
      - Save as a pdb file:
        `[atomselect top all] writepdb alchemy_site.pdb`
2. **Perform the FEP simulation**
   We have provided a configuration file for this FEP run: *stepB_alchemy_site/inputs/run.namd*. See the in-line comments in that file and Appendix B for a detailed description of the settings relevant to running FEP in namd.
   a. **Run the decoupling FEP**:
      Enter the following in your terminal window:
      ```
      cd stepB_alchemy_site/inputs/
      namd2 +p4 run.namd > ../outputs/alchemy_site.log 2> ../outputs/alchemy_site.err
      ```
   b. **[Optional] Start Step C:**
      If you have access to more compute resources, you can continue on to Step C while this FEP calculation is running. **Don't forget to return to analyze these data once the simulation is complete.**
3. **Analyze the trajectory**
   a. **Visually inspect the trajectory in VMD:**
      - Open VMD.
      - Load the .psf (*common/structures/phenol_lysozyme.psf*) and .dcd file(s) from the outputs of stepB.
      - Ensure that the ligand remains in a bound-like configuration for the duration of the simulation.
   b. **Measure the restraint energy:**
      - Open the Colvars Dashboard.
      - Click `Load` and import your DBC restraint file (*DBC_restraint.colvars*).

- Open the *biases* tab, select the DBC restraint, and click `Energy Timeline`.
- The restraint energy should remain near zero for several nanoseconds, then increase and reach a maximum in the second half of the simulation (when the ligand is fully decoupled). If this is not the case, see Appendix F.

c. **Calculate ΔG^*_{site} in the Jupyter Notebook:**

- Navigate back to the tutorial root directory.
- Begin a Jupyter session and open the notebook titled *SAFEP_Tutorial_Notebook.ipynb*.
- Follow the in-notebook prompts to parse your new fepout file (*stepB_alchemy_site/output/AFEP2-02.fepout*). By default, we use the sample output. Be sure to update the paths as indicated in the notebook:

```
bound_fep_path=f'{root}/stepB_alchemy_site/sample_outputs/'
restraint_perturbation_path = f'{root}/stepC_restraint_perturbation/sample_outputs/'
bulk_fep_path=f'{root}/stepD_alchemy_bulk/sample_outputs/'
```

- Compare your outputs to the sample outputs found in Appendix B.3.
- Caution: the errors reported are based on the PyMBAR BAR estimator and are subject to the corresponding assumptions and caveats. [20]

Step C: Compute the DBC restraint free energy correction

We designed the DBC restraint so that it doesn't do any significant work in the fully coupled system. However, it does reduce the entropy of the fully decoupled ligand, which would otherwise be exploring an "empty" simulation box. We need to calculate the corresponding free energy cost so we can correct for it. In this section we will use thermodynamic integration (TI) to calculate ΔG_{DBC}; the free energy difference between a gas-phase ligand under DBC restraints *vs* a (spherical) volumetric restraint. For more details see Appendix D.

Required Input:

- Structure file: *common/structures/phenol_gas_phase.psf*
- Coordinate file: *common/structures/phenol_gas_phase.pdb*
- NAMD configuration file: *stepC_restraint_perturbation/inputs/run.namd*

Essential Output:

- Colvars configuration file: *stepC_restraint_perturbation/outputs/DBC_Restraint_RFEP.colvars*
- FEP trajectory file: *stepC_restraint_perturbation/outputs/RFEP.dcd*
- Colvars output file: *stepC_restraint_perturbation/outputs/RFEP.colvars.traj*

Procedure:

1. **Create coordinates upon which to base your restraints**

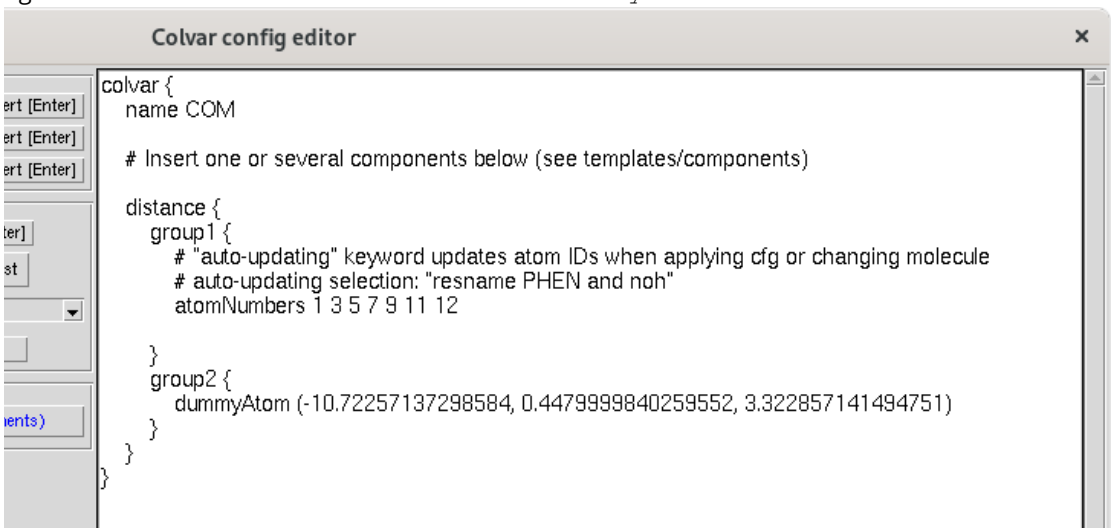
   a. **Open VMD and load the input files:**

   - Open VMD.
   - Open the Tk Console.
   - Open the Colvars Dashboard.
   - Load *phenol_gas_phase.psf* and *phenol_gas_phase.pdb*
   - **[Optional]** Extract the phenol from the phenol-lysozyme complex by running the following in the tkConsole. **Note: We have completed this step for you. The sample files can be found in *common/structures*.**

```
%cd common/structures
%mol load psf phenol_lysozyme.psf pdb phenol_lysozyme.pdb
%set ligand [atomselect top "resname PHEN"]
%cd ../../stepC_restraint_perturbation/outputs
%$ligand writepsf phenol_gas_phase.psf
%$ligand writepdb phenol_gas_phase.pdb
```

   b. **Define the gas-phase spherical coordinate:**

- In the Colvars Dashboard, click `New [Ctrl-n]`.
- In the second line of the editor, replace the default name `myColvar` with `COM`.
- Delete `atomNumbers 1 2` and leave your cursor on that line.
- Using the *Atoms from selection text:* tool in the left panel, enter `resname PHEN and noh` and click `Insert [Enter]`.
- Get the geometric center of the heavy atoms by the following in the Tk Console:
  `measure center [atomselect top "resname PHEN and noh"]`
- Set the atoms of group 2 to `dummyAtom (x0, y0, z0)` where x0, y0, and z0 are the coordinates of the geometric center of the ligand you just retrieved in the previous step. Your editor should look similar to the figure below. Note the inclusion of commas in the `dummyAtom` statement.



- Save and close the colvar editor by clicking `Apply [Ctrl-s]`.

c. **Define the gas-phase DBC coordinate:**

- Click `New [Ctrl-n]` again.
- In the second line of the editor, replace the default name `myColvar` with `DBC`.
- Delete the default distance component `distance{...}` and leave your cursor on that line.
- From the *component templates* dropdown menu select `rmsd` and click `Insert [Enter]`.
- As before, add `atomPermutation 1 5 3 9 7 11 12` to the rmsd block to define the ligand symmetry.
- Delete `atomNumbers 1 2 3` and leave your cursor on that line.
- In the field labeled *Atoms from selection text:* enter `resname PHEN and noh` and click `Insert [Enter]`.
- Add `rotateReference off` and `centerReference off` to the `atoms` block.
- Replace the default `refPositionsFile @` line using the `o refPositionsFile` radio button and the `Pick file` button to select *phenol_gas_phase.pdb*.
- Save and close the colvar editor by clicking `Apply [Ctrl-s]`.

2. **Define the restraints**

a. **Create the spherical restraint:**

- In the *biases* tab of the Colvars Dashboard, click `New bias (Ctrl-n)` and delete the default text.
- From the *bias templates* dropdown menu, select `harmonic walls` and press `Insert [Enter]`.
- Recall the upperWalls value you used for the DBC restraint in subsection 2.b from Step A. You will need this value in this and the next step.
- Modify the bias to match the following parameters (see Appendix D):

```
name            distance_restraint
colvars         COM
outputEnergy    on
upperWalls      [DBC upperWalls plus 1]
forceConstant   200
```

- Save and close the bias editor by clicking `Apply [Ctrl-s]`.

b. **Save the config file:**
   - Click the `Save` button on the Colvars Dashboard.
   - Save the file as *stepC_restraint_perturbation/outputs/DBC_restraint_RFEP.colvars*

c. **Create a DBC restraint that gradually releases:**
   - We will use the provided setTI Tcl procedure.
   - Open stepC_restraint_perturbation/inputs/run.namd in a text editor
   - Find the block labeled "COLVARS"
   - Edit the input variables to match the following

```
cvName                DBC
biasType              harmonicWalls
upperWalls            [DBC upperWalls as determined in step A]
targetForceConstant   200.0
forceConstant         0.0
targetForceExponent   6.0
targetEquilSteps      500
targetNumSteps        300000
nWindows              40
releaseFlag           True
```

3. **Run the TI simulation**

a. **Enter the following in your terminal:**

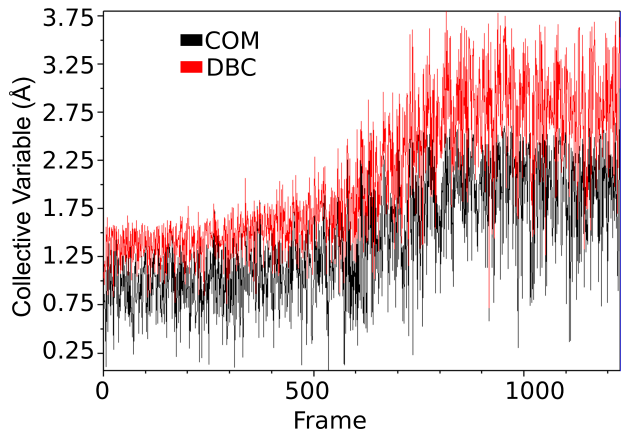
```
cd stepC_restraint_perturbation/inputs
namd2 +p1 run.namd > ../outputs/DBC_FreeEnergy.log 2> ../outputs/DBC_FreeEnergy.err
```

b. **[Optional] Start Step D:**
   If you have access to more compute resources, you can continue on to Step D while the TI calculation is running. **Don't forget to return to analyze these data once the simulation is complete.**

4. **Analyze the output** If any of these checks fails, check the Troubleshooting section of the Appendices (Appendix F).

a. **Visually inspect the trajectory in VMD:**
   - Open VMD.
   - Load the .psf, .pdb, and .dcd files associated with this tutorial step.
   - The ligand should initially fluctuate roughly in place at the start and gradually explore the COM restraint space as the DBC restraint is released.

b. **Check the collective variable trajectories:**
   - Open the Colvars Dashboard
   - Click `Load` and open the Colvars configuration file *DBC_restraint_RFEP.colvars*
   - Select both the COM and DBC restraints
   - Click `Timeline plot`
   - Both coordinates should start low and gradually increase. The COM restraint should plateau near the value of the restraint's `upperWalls` as shown in the figure below:



c. **Calculate ΔG_{DBC} in the Jupyter Notebook:**
   - Open the Jupyter Notebook as in subsection 3.c from step B. Update the paths in "User Settings" as needed. Remember, sample data will be used by default.
   - Run the first several cells at least until the first FEP analysis section.
   - Run the first cell in the section titled "Process the DBC TI calculation" and make sure the DBC and COM walls are correct.
   - Run all the remaining cells in that section. The output will include ΔG_{DBC} as well as an error estimate. Sample results and more explanation can be found in Appendix D.

Step D: Decouple phenol from bulk solvent

You have completed one alchemical FEP calculation already, but double-decoupling methods require two such calculations to close the thermodynamic cycle. We need to know the free energy of transferring the ligand from the binding site into vacuum, *and* from vacuum into the bulk. In this section we will calculate the latter term, $\Delta G^*_{\mathrm{bulk}}$, by decoupling the ligand from the bulk solution.

Required Input:

- Structure file: *common/structures/phenol_water.psf*
- Coordinate file: *common/structures/phenol_water.pdb*
- NAMD configuration file: *stepD_alchemy_bulk/inputs/run.namd*

Essential Output:

- FEP configuration file: *stepD_alchemy_bulk/outputs/alchemy_bulk.pdb*
- FEP trajectory file: *stepD_alchemy_bulk/outputs/alchemy_bulk.dcd*
- FEP output file: *stepD_alchemy_bulk/outputs/alchemy_bulk.fepout*

Procedure:

0. **Prepare the ligand as you would for traditional FEP.** Use a conservative box size; very small boxes are more prone to instabilities and self-interactions which will lead to artifacts in the final free energy estimate. **Note: this step has been completed for you.**
1. **Specify which atoms will be decoupled using the pdb beta field**
   a. **Open VMD and load the psf and pdb files specified in "Required Input".**
   b. **Set and write beta values:**
      - Open the Tk Console
      - Ensure that your Tk Console is in the correct directory:
        ```
        cd stepD_alchemy_bulk/outputs
        ```
      - Set the beta value of all atoms to 0:
        ```
        [atomselect top all] set beta 0
        ```

- Set the beta values of the ligand atoms to -1 for decoupling:
  `[atomselect top "resname PHEN"] set beta -1`
- Save as a pdb file:
  `[atomselect top all] writepdb alchemy_bulk.pdb`

2. **Run the ligand decoupling simulation in bulk solvent**
```
cd stepD_alchemy_bulk/inputs
namd2 +p4 run.namd > ../outputs/alchemy_bulk.log 2> ../outputs/alchemy_bulk.err
```
3. **Analyze the output**
   a. **Visually inspect the trajectory in VMD:**
      - Open VMD.
      - Load the .psf, .pdb, and .dcd files associated with your simulation.
      - The ligand should diffuse normally at the start of the simulation but behave more and more like a gas-phase molecule.
   b. **Calculate ΔG^*_{bulk} in the Jupyter Notebook**
      - Open the Jupyter Notebook as in subsection 3.c from Step B.
      - Confirm that `bulk_fep_path` points to your files
      - Parse the *.fepout* file by running all the cells in the Jupyter notebook section titled "Decoupling from Solvent".
   c. Caution: as noted above, the errors reported are based on the PyMBAR BAR estimator and are subject to the corresponding assumptions and caveats. [20]

Step E: Calculate corrections and combine quantities

We will now calculate $\Delta G^\circ_{\text{V}}$ analytically. With this final piece of information, we can calculate the dissociation constant and estimate a titration curve based on the probability of occupancy assuming a two-state system: $P_{\text{bind}} = \frac{[PHEN]}{K_{\text{d}}+[PHEN]}$ where K_{d} is the dissociation constant given by $K_{\text{d}} = e^{\beta \Delta G^\circ_{\text{bind}}}$. For additional information see Appendix E.

Required Input:

- Site FEP data: *stepB_alchemy_site/outputs/alchemy_site.fepout*
- Restraint perturbation data (RFEP/TI): *stepC_restraint_perturbation/outputs/RFEP.colvars.traj*
- Bulk FEP data: *stepD_alchemy_bulk/outputs/alchemy_bulk.fepout*

Essential Output:

- $\Delta G^\circ_{\text{bind}}$
- *titration_curve.pdf*

Procedure:

1. Complete any unfinished analyses in previous steps (i.e. steps B.3, C.4., and D.3). Some issues are only obvious when visualizing the trajectories. Don't skip those steps!
2. Open the Jupyter notebook and navigate to the section labeled "Volumetric Restraint Contribution".
3. Run the section to calculate the volumetric free energy contribution. See Appendix C.3 for a more detailed explanation. **Note:** At this point you will either need to have completed all simulations or use the sample data provided. To use the sample data, change the path variables (`bound_fep_path`, `restraint_perturbation_path`, and `bulk_fep_path`) to use the files in their respective *./sample_outputs* directories.
4. Calculate the overall $\Delta G^\circ_{\text{bind}}$ and compute a titration curve by running the cells in the section "Binding Free Energy".
5. Compare your final $\Delta G^\circ_{\text{bind}}$ to the literature value, reported as -5.44 kcal/mol [21].
6. Compare your titration curve to Figure 4 below.

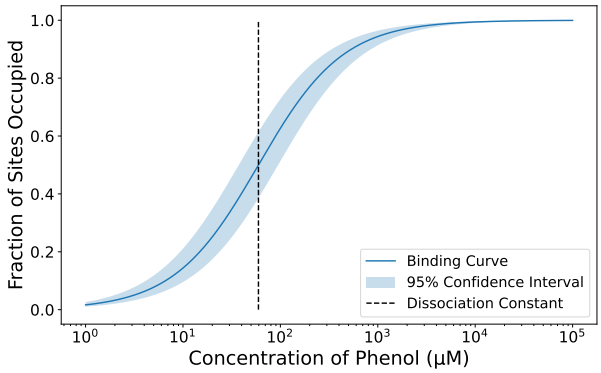


Figure 4. An example titration curve generated using Equation 23. The 95% confidence interval represents $\pm 1.96 * \text{SEM}$ of ΔG_{bind}.

3 What's next?

Congratulations! You have completed all the essential steps of the SAFEP protocol. You can now apply the protocol above to your own receptor-ligand binding problem, applying minor changes as needed. Of course, the outcome of your SAFEP calculation will be only as realistic as your model and force field. As you proceed, it is critical to visualize the simulations, track evolution of the DBC, and evaluate convergence of your own calculations. Specifically, for a more in-depth analysis of the alchemical calculations, check out the general-purpose notebooks from the SAFEP repository at https://github.com/BranniganLab/safep. Additional information and a (non-exhaustive) troubleshooting guide can be found in the appendices below. If you have trouble with this tutorial or the SAFEP package, please contact us either directly or by opening an issue on the corresponding GitHub page.

Table 1. Symbols used in this tutorial.

Symbol	Definition
d	Distance-to-Bound-Configuration (DBC)
$\Delta G^{\circ}_{\text{bind}}$	Standard free energy of binding
$\Delta G^{*}_{\text{bulk}}$	Free energy of decoupling the ligand from the bulk solution
ΔG_{DBC}	Free energy of imposing the DBC restraint on the ligand from a COM restraint
$\Delta G^{*}_{\text{site}}$	Free energy of decoupling the ligand from the binding site
$\Delta G^{\circ}_{\text{V}}$	Free energy of releasing the ligand to the standard state from the Center of Mass restraint
ΔG^{EXP}	Free energy estimated by exponential averaging
k	A force constant for a restraint
k_{λ}	A force constant that is a function of λ
k_0 and k_1	Force constants when $\lambda = 0$ and $\lambda = 1$, respectively
L and L'	Arbitrary ligand concentrations
m	Number of non-decoupled ligands in the bulk system
n	Number of (TI) lambda windows
p_{occ} and p_{unocc}	The probability of a site being occupied or unoccupied, respectively
r	Center Of Mass (COM) displacement
r_R	Upper wall of a COM restraint
U_{FW}	Energy function of a flat-well potential
V	Volume of the bulk system
V_R	Volume of a sphere with radius r_R
Z_{occ} and Z_{unocc}	Partition functions for the occupied state and unoccupied state, respectively
α	An exponential smoothing parameter
β	Thermodynamic beta with units of $\frac{kcal}{mol}$
δ_{λ}	A measure of forward-backward hysteresis; the difference between forward and backward ΔG^{EXP} for a given lambda window (Eq. 2)
Θ_i	DBC test function for a single ligand
Θ_{occ}	Occupancy test function for a single binding site
λ	A coupling parameter $\lambda \in \{0, 1\}$
ξ	An arbitrary collective variable
ξ_{max}	Upper wall of a flat-well restraint on ξ

Appendix A System Selection and Setup

Lysozyme L99A/M102H (PDBid 4I7L) was chosen for several reasons. Lysozyme L99A/M102H is a small protein that binds a small, rigid molecule with reasonably high affinity which has already been measured experimentally. These properties make it well-suited as a model for prototyping and validating free energy calculation methods generally.

Because lysozyme is elongated, we save some computation time by using a narrower box. Rotations of the protein could cause self-interactions, which we avoid by imposing a soft harmonic restraint on its orientation, as defined in *common/protein_tilt.colvars*. The provided systems were prepared using CHARMM-GUI[22, 23] using a truncated lysozyme (PDBid 4I7L, residues 3 to 157) and solvated using default parameters (TIP3P water, 0.15 M NaCl). The production run uses largely default parameters and settings. The only notable exception is that `WrapAll` should be set to `off`. This is because wrapping across the PBC can cause unexpected results during analysis which can compromise the FEP and TI calculations.

Appendix B Running FEP in NAMD

Appendix B.1 Configuration Files

In addition to the configuration, forcefield, and structural files, running FEP in NAMD requires a particular pdb file (sometimes called a "fep file") that contains flags that indicate which atoms are being coupled or decoupled. This is usually indicated in the beta column as '-1' for decoupling or '1' for coupling. All other beta fields should be 0.

The configuration file also contains some additional options that are detailed in the NAMD user guide [13] and described briefly in the provided configuration files. While most of the settings should remain unchanged in a wide range of systems, there are a few exceptions.

`alchOutFreq` determines the number of steps between collecting FEP samples. It should be set to a multiple of `fullElectFreq` to ensure accurate energy estimates. Later versions of NAMD should resolve this issue automatically; see Bug advisory and Workaround. Additionally the sampling frequency should be between 50 and 200 steps; sampling too frequently will result in bloated data sets of highly autocorrelated samples while sampling infrequently will result in too few samples to get a well-converged estimate of the change in free energy.

`alchVdwShiftCoeff` controls the strength of the soft-core potential which is essential to prevent "end-point catastrophes" in which one or more Lennard-Jones potentials diverge to infinity near lambda=0 or lambda=1. See [24] and [19] for more details.

`alchEquilSteps` hard-codes the time between starting a new lambda value and beginning to sample the ensemble. Alchemlyb and PymBAR provide functions that will downsample the data set using automated equilibration and autocorrelation detection schemes. We have found that auto-

mated equilibrium detection performs about as well as manually setting `alchEquilSteps` and autocorrelation is the bigger problem when trying to assess convergence. See [20] for a more detailed discussion of equilibrium detection, autocorrelation, and their effects on free energy estimation. See Appendix B.3 or the provided Jupyter notebook for more information on how these are applied to analysis.

`deltaLambda` is passed as a parameter to the runFEP function and determines the width of the lambda windows. Narrower windows will converge faster but will increase the total number of windows required to span $\lambda = 0$ to $\lambda = 1$. As a result, we need to empirically optimize the number and length of windows. See Appendix B.3 and Appendix F for more details on assessing and optimizing these parameters. The number and length of windows used here ($\sim$ 40 ns total simulation time) are a good starting point, but very flexible ligands or binding sites may require much more sampling.

`IDWS` (interleaved double-wide sampling) tells NAMD to alternate between sampling the forward and reverse lambda directions (via the runFEP function, which is passed the `alchLambdaIDWS` parameter). This should be set to "true" thus removing the need for independent forward and backward runs.

Appendix B.2 Parsing and Data Analysis

In this tutorial we have recommended using a Jupyter notebook for analysis.

After initial reading and parsing, each window's equilibration period will be automatically detected and the samples decorrelated as implemented by PyMBAR. [20] The free energies will be calculated and you will see the estimated ΔG^* with Alchemlyb-estimated statistical error. We have used conservative FEP settings in this tutorial which (though not the most efficient) should result in good convergence for this system. As noted in the previous section, more complicated systems (with more internal degrees of freedom) may require much longer sampling and narrower lambda windows. In such systems, errors as high as 1 kcal/mol are not uncommon. Error estimates larger than 1 kcal/mol often indicate poor convergence, which can also be diagnosed by analyzing hysteresis. See Appendix F for more information on how to identify and resolve the underlying causes.

Appendix B.3 Interpreting the Figures

In this section we describe the contents and meaning of each of the figures generated by the provided Jupyter notebook. See Appendix F for strategies to address discrepancies between your own results and those described here. An example of a well-converged calculation is shown in Figure 5.

Cumulative and per-window ΔG curves (first and second panels of Figure 5) should be fairly smooth, and a magnitude of ΔG greater than a few kcal/mol per window is often associated with poor convergence. Sharp cusps and large jumps often indicate either insufficient samples or lambda windows that are too wide.

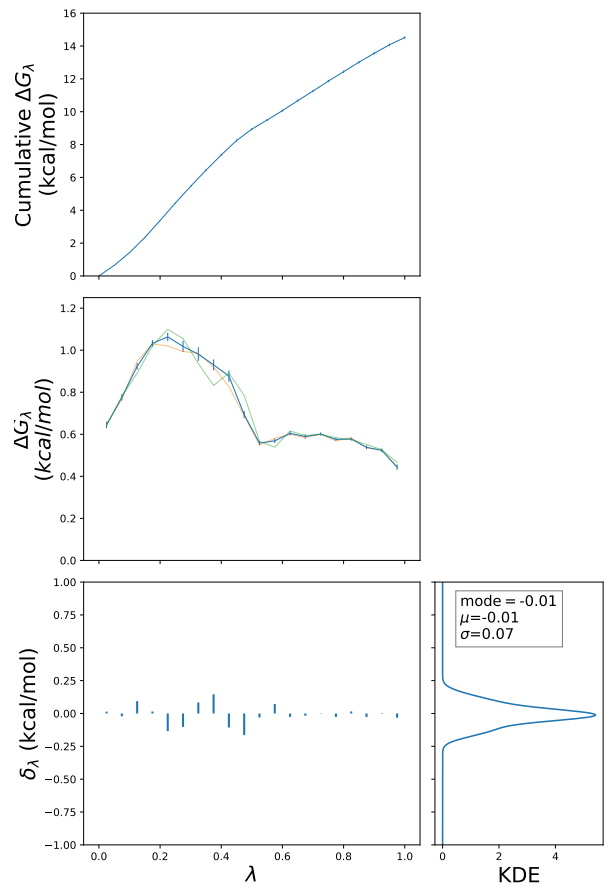


Figure 5. Example results from protein-phenol decoupling calculation. The first panel shows the cumulative change in free energy with accumulated error. The second panel shows per-window difference in free energy (ΔG_λ) calculated by the BAR estimator (blue), and exponential estimators for forward (orange) and backward (green) samples. The third and fourth panels show the free energy hysteresis (δ_λ) and its probability density function estimated by kernel density estimation (KDE), respectively. Error bars indicate standard error of the mean; cumulative (first panel) or per-window (second and third panels).

The δ_λ plots (third and fourth panels of Figure 5) are used to diagnose hysteresis with respect to lambda:

$$\delta_{\lambda,i} \equiv \Delta G^{\mathrm{EXP}}_{i+1,i} - \Delta G^{\mathrm{EXP}}_{i,i+1} \tag{2}$$

No value should be more than about 1 kcal/mol with a mean (μ) and mode close to zero. Similarly, the standard deviation (σ) should be less than $\approx$ 0.5. Failure to meet any of these criteria indicates that one or more of the lambda windows has not, in fact, reached equilibrium or converged.

Finally, the convergence plot should display two curves that meet quickly (before 0.5), and both curves should level

out well before 1 like the example shown in Figure 6. If they are still changing at 1 or have not gotten to within 0.5 kcal/mol by λ = 0.5, the system is unlikely to be converged at one or more lambda values and the final ΔG estimate is likely to be inaccurate.

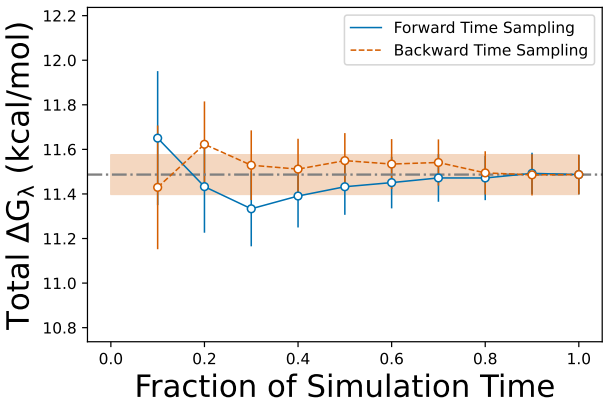


Figure 6. Example convergence data. We believe this calculation is well-converged due to the overlap near the half-way point and the leveling out of both curves well before the end. Error bars indicate standard error of the mean.

Appendix C Restraints

In the simulation in which the ligand is decoupled from the site, restraints that keep the ligand from diffusing away must be applied. This serves two purposes [5]: first, it prevents spontaneous unbinding of the ligand over the course of the simulation; second, it accelerates convergence of the computation by limiting the space to be sampled. Thus binding restraints are essential both for estimating a well-defined free energy of binding, and for minimizing the statistical noise on that estimate. This is often achieved by layering several rotation and translational restraints on the ligand. [1, 3, 4, 6, 8, 9]. The main draw-back of such approaches is that each restraint must be designed and parameterized which adds to 1) the time and effort required to setup the simulations and 2) the complexity of the simulations themselves. As a result, troubleshooting and interpretation are more difficult and time-consuming. SAFEP, in contrast, uses just one restraint, the distance-to-bound-configuration (DBC), which is both robust and requires minimal parameterization [11].

Appendix C.1 Flat-well Restraints

An ideal restraint for decoupling simulations would precisely encapsulate the bound ensemble without modifying it. That is, it would be of the form:

$$U(\xi) = \begin{cases} 0 & , \xi \text{ in the bound state} \\ \infty & , \text{otherwise} \end{cases} \tag{3}$$

This infinitely sharp potential, however, would create numerical instability in a molecular dynamics simulation. We, therefore, impose continuous flat-well restraints which result in finite restorative forces when the system approaches the boundary of the bound state without modifying the bound ensemble itself. Such restraints approximate square wells with the form:

$$U_{\mathrm{FW}}(\xi) = \begin{cases} \frac{1}{2}k\,(\xi - \xi_{\max})^2 & , \xi > \xi_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Appendix C.2 The Distance-to-Bound Configuration (DBC) Coordinate

Appendix C.2.1 Definition of DBC

The DBC is the root-mean-square deviation (RMSD) of a subset of ligand atom coordinates from a typical bound pose *in the frame of reference of the binding site*. This single, scalar coordinate is designed to optimize the convergence of alchemical decoupling from the site without biasing the initial, coupled state. It captures any relative motion of the ligand with respect to the binding site as well as conformational changes of the selected ligand atoms. In general, all heavy atoms of the ligand can be included in the DBC definition, but larger, more flexible ligands may be better restrained using a smaller subset of atoms.

To impose a DBC restraint, we apply a flat-well potential defined by Equation 4 to a DBC coordinate. See Ref. 11 for details.

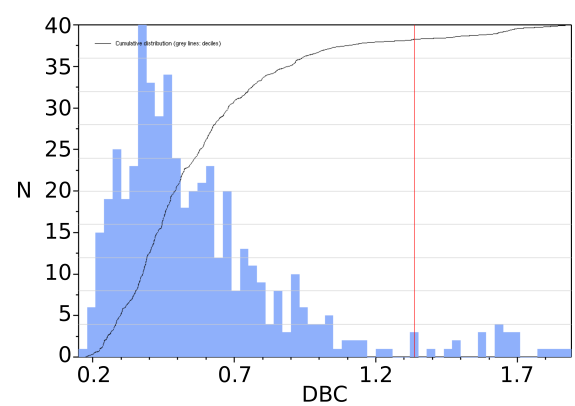


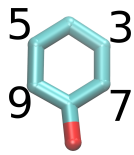
Figure 7. Screenshot from the Colvars Dashboard showing an example asymmetric DBC distribution from an unbiased simulation. If the phenol had flipped about the symmetric axis, there would be a second peak about 1.8 Å

Appendix C.2.2 Symmetric DBC

One peculiarity of the model system used here is that the ligand, phenol, is symmetric. Although this isn't strictly problematic, it does require a little extra accounting and care. This

is especially true for ligands with higher symmetry numbers which would require much more complicated analytical corrections. The Colvars keyword `atomPermutation` can be used to define such symmetries:

1. The easiest way to identify equivalent atoms is to label them.
2. Use the *Graphics→Representations* interface to hide all atoms except the ligand.
3. Use the labeling tool to label the four symmetric carbons as shown:



4. Open *Graphics→Labels...* .
5. Select all four labels from the list by clicking and dragging
6. Open the tab titled *Properties* and change the format string to `%1i`
7. You may wish to adjust other settings in this menu to make the labels more visible
8. Your view should now look something like the image above with the serial number of each atom indicated.
9. In the Colvars config editor window, place your cursor on the line before `atoms {` and add `atomPermutation {11 7 3 1 5 9 12}`
10. Your console should now look like this:

```
colvar {
  name DBC_sym
  rmsd {
    atomPermutation {11 7 3 1 5 9 12}
    atoms {
      atomNumbers {11 9 5 1 3 7 12}
. . .
```

These changes make atom 7 equivalent to atom 9 and atom 3 equivalent to atom 5 for purposes of RMSD calculation.

Appendix C.3 Isotropic center-of-mass restraint

The center-of-mass (COM) restraint is used as the container into which the ligand is released during RFEP (Step C). It is created by using the flat-well restraint in Equation 4, where ξ is the displacement of the ligand's center of mass. The free energy cost of imposing the COM restraint can be calculated analytically because we treat the ligand as a point particle in a well-defined volume (i.e. as an ideal gas). The free energy difference between the simulated volume and an arbitrary concentration L is:

$$\Delta G_{\mathrm{V}}(L) = -\frac{1}{\beta}\ln[LV_{\mathrm{R}}], \tag{5}$$

where $V_{\mathrm{R}} = \frac{4}{3}\pi r_{\mathrm{R}}^3$ is the volume of a sphere of radius r_R (the upper boundary of the COM restraint). [11] Recall that the width of the restraint is slightly (1 Å) larger than the width of the DBC restraint to avoid any edge cases in which the DBC may be larger than the COM displacement.

Appendix D Restraint free energy calculation

Appendix D.1 Restraint perturbation simulation

Although the DBC restraint, by design, does not affect the coupled state, it does modify the decoupled state, and this contribution must be accounted for in the overall free energy estimation. To that end, we use the Colvars Module to run a simulation in which the DBC restraint is removed progressively, and compute the free energy for that process. To make this computation more efficient, the ligand is not released into the whole simulation box, but it is kept confined in a spherical volume V_{R}. Be advised, MD simulation algorithms can prevent center-of-mass diffusion for the whole system (in NAMD, `zeroMomentum`). In RFEP, the ligand must be allowed to diffuse, so this option must be disabled.

Appendix D.2 Thermodynamic Integration and Analysis

As in FEP, restraint free energy perturbation (RFEP) scales certain energy terms and the associated forces using a perturbation parameter, $\lambda \in \{0, 1\}$. FEP estimates finite free energy differences between λ values while TI estimates the free energy derivative. As derived and discussed in most statistical mechanics and molecular dynamics textbooks (e.g. [25, 26]), the Helmholtz free energy F approximates the Gibbs free energy G when $P\Delta V$ can be neglected. Furthermore, the $P\Delta V$ terms tend to cancel out when subtracting the two alchemical free energy differences in our thermodynamic cycle, making the following approximation quite accurate:

$$\Delta G \approx \Delta F = \int_0^1 \left\langle \left.\frac{\partial U}{\partial \lambda}\right|_{\lambda} \right\rangle d\lambda \tag{6}$$

$$\approx \sum_{i=1}^{n} \left\langle \left.\frac{\partial U}{\partial \lambda}\right|_{\lambda_i} \right\rangle \times \Delta\lambda_i, \tag{7}$$

where $\Delta\lambda_i$ is the difference between adjacent lambda values.

The core of this expression, $\left.\frac{\partial U}{\partial \lambda}\right|_{\lambda_i}$, can be calculated analytically for a given value of ξ by applying the definition

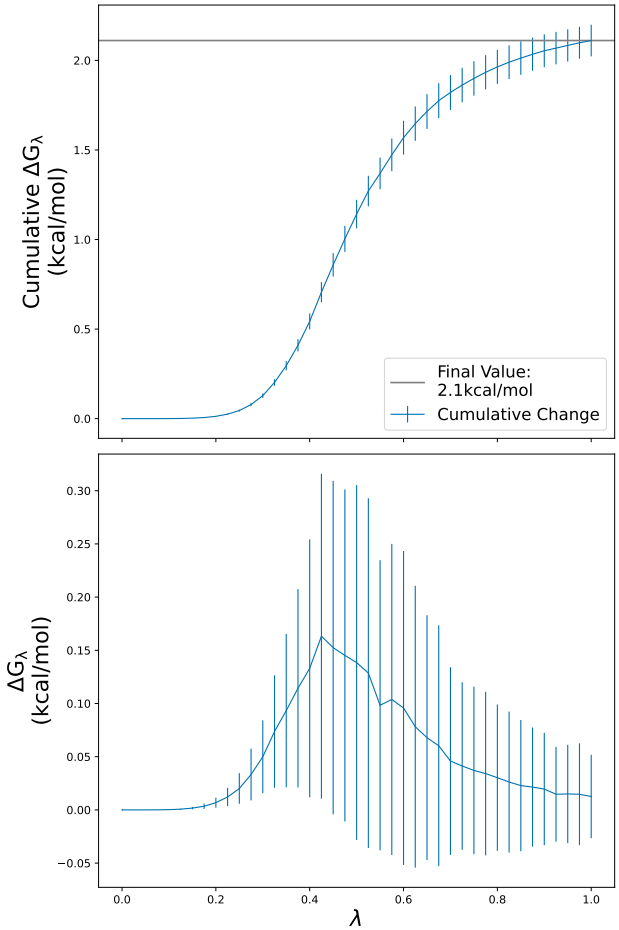


Figure 8. Restraint free energy (ΔG_λ), top) and its derivative with respect to the coupling parameter ($\partial G/\partial\lambda$, bottom), as a function of λ. Error bars indicate standard deviation of the mean.

$k_\lambda \equiv \lambda^\alpha (k_1 - k_0)$ to equation 4 and taking the partial derivative with respect to λ:

$$\left.\frac{\partial U}{\partial \lambda}\right|_{\lambda_i} = \begin{cases} \frac{1}{2}\alpha\lambda_i^{\alpha-1}(k_1 - k_0)(\xi - \xi_{\max})^2 & , \xi > \xi_{\max} \\ 0 & , \text{otherwise} \end{cases} \tag{8}$$

where k_0 is the force constant (`forceConstant`) when $\lambda = 0$, k_1 is the force constant when $\lambda = 1$ (`targetForceConstant`), and α (`targetForceExponent`) is a tuning parameter that improves convergence of TI by making the energy a smoother function of λ near $\lambda = 0$.

In Step C, ξ is replaced by d, the DBC, and $\xi_{\max}$ is replaced by the upper wall of the DBC restraint, $d_{\max}$. This is applied to the colvars trajectory data in the Jupyter notebook section associated with Step C.

Finally, the error is estimated as follows (Figure 8).

$$\sigma^2_{\Delta F} = \sum_{i=1}^{n} \sigma_i^2 \tag{9}$$

The error on each local average $\left\langle \left.\frac{\partial U}{\partial \lambda}\right|_{\lambda_i} \right\rangle$ is estimated as its standard deviation, thus avoiding any assumption on the number of independent samples, leading to:

$$\sigma_i^2 = \left[\left\langle \left.\frac{\partial U}{\partial \lambda}\right|^2_{\lambda_i} \right\rangle - \left\langle \left.\frac{\partial U}{\partial \lambda}\right|_{\lambda_i} \right\rangle^2 \right] \times \Delta\lambda_i \tag{10}$$

A finer estimate of the error can be obtained by running replicas of the TI calculation and computing their dispersion. Technical details of our implementation can be seen in *common/TI.tcl* which is just syntactic sugar for the Colvars implementation (See the Colvars user guide).

Appendix E Concentration Dependence and Non-Ideality

While in this tutorial we have only used a single, infinitely dilute, concentration to calculate ΔG_{bind}, SAFEP can also be used to predict concentration dependence in non-ideal and non-dilute solutions. Here we consider the underlying theory for interpreting such a calculation, and provide general suggestions for implementation.

We consider a "unitary" (single protein, single site[11]) system with m ligands in a volume V. The ligand concentration in this system is $L = m/V$. A DBC coordinate d_i can be defined for each of the m ligands, indexed by i.

The threshold on the DBC coordinate meaningfully divides the ensemble into two possible macrostates: occupied (one ligand occupies the site and $m-1$ ligands are in solution) and unoccupied (no ligands occupy the site and m ligands are in solution.) We formalize this here through the DBC "test function," which for an individual ligand i is a Heaviside step function of the form

$$\Theta_i = \begin{cases} 1 & , d_i < d_{\max} \\ 0 & , \text{otherwise} \end{cases} \tag{11}$$

The instantaneous site occupancy Θ_{occ} is determined by whether any of the m ligands occupy the site, given by the sum of all the individual test functions:

$$\Theta_{\text{occ}} = \sum_{i=1}^{m} \Theta_i. \tag{12}$$

Since here we consider the case in which the site can bind at most one ligand, Θ_{occ} is either 0 or 1.

The partition functions for the occupied and unoccupied states are thus Z_{occ} and Z_{unocc} respectively, where

$$Z_{\text{occ}} = \int \Theta_{\text{occ}} e^{-\beta U} \mathrm{d}^N\vec{r} \tag{13}$$

$$Z_{\text{unocc}} = \int [1 - \Theta_{\text{occ}}]\, e^{-\beta U} \mathrm{d}^N\vec{r}, \tag{14}$$

and the potential energy U is a function of the positions $\vec{r}$ of all N particles in the system, while Θ_{occ} is a function of the DBC coordinates d (and thus the positions of ligand and site atoms only).

The occupancy probability $P_{\text{occ}}(L)$ is thus

$$P_{\text{occ}}(L) = \frac{Z_{\text{occ}}}{Z_{\text{occ}} + Z_{\text{unocc}}} = \frac{\int \Theta_{\text{occ}} e^{-\beta U} \mathrm{d}^N \vec{r}}{\int e^{-\beta U} \mathrm{d}^N \vec{r}} = \langle \Theta_{\text{occ}} \rangle \tag{15}$$

which, as expected, yields the average occupancy $\langle \Theta_{\text{occ}} \rangle$.

Each macrostate has an associated free energy:

$$\beta G_{\text{occ}} = -\ln Z_{\text{occ}} \tag{16}$$

$$\beta G_{\text{unocc}} = -\ln Z_{\text{unocc}}, \tag{17}$$

where G_{occ} and G_{unocc} are the free energies of the occupied and unoccupied macrostates respectively, so

$$P_{\text{occ}}(L) = \frac{e^{-\beta G_{\text{occ}}}}{e^{-\beta G_{\text{occ}}} + e^{-\beta G_{\text{unocc}}}}. \tag{18}$$

We turn now to connecting these quantities to a SAFEP calculation. In step D of the protocol, we decoupled one ligand from a bulk that contained $m = 0$ fully coupled ligands, for an infinitely dilute concentration of $L = 0/V$. We then extrapolated to the standard concentration using an ideal gas correction that assumes ideality.

For a ligand at finite concentration in a non-ideal bulk, it is neither useful or necessary to standardize the free energy. Instead, we would carry out Step D at the finite ligand concentrations of interest ($L = m/V > 0$), and adjust Step E to calculate the unstandardized free energy $\Delta G_{\text{bind}}(L)$ as follows:

$$\Delta G_{\text{bind}}(L) = -\Delta G^*_{\text{site}} + \Delta G_{\text{DBC}} - \Delta G_{\text{V}}(L) + \Delta G^*_{\text{bulk}}(L), \tag{19}$$

where the volume per molecule in the bulk is V/m and thus

$$\Delta G_{\text{V}} = -\frac{1}{\beta} \ln \frac{m V_{\text{R}}}{V}. \tag{20}$$

Since $\Delta G_{\text{bind}}(L) = G_{\text{occ}}(L) - G_{\text{unocc}}(L)$

Equation 18 can be rewritten in terms of $\Delta G_{\text{bind}}(L)$:

$$P_{\text{occ}}(L) = \frac{1}{1 + e^{\beta \Delta G_{\text{bind}}(L)}} \tag{21}$$

Even for a non-ideal bulk, we may assume the excess chemical potential is unchanging for small changes in concentration. Thus we would perform ligand decoupling (step D) at finite concentration L, and use the ideal gas correction to predict occupancy for nearby concentrations L', as long as $|L - L'|$ is small.

$$\Delta G_{\text{bind}}(L') = \Delta G_{\text{bind}}(L) - \frac{1}{\beta} \ln \frac{L'}{L} \tag{22}$$

Substitution of Equation 22 in Equation 21 yields the occupancy probability for concentration L':

$$P_{\text{occ}}(L') \sim \frac{L'}{L' + L e^{\beta \Delta G_{\text{bind}}(L)}} \tag{23}$$

Incidentally, for dilute L, $L e^{\beta \Delta G_{\text{bind}}(L)} = L^\circ e^{\beta \Delta G^\circ_{\text{bind}}} = K_d$, and Equation 23 reduces to a form familiar to biochemists $P_{\text{occ}}(L') = \frac{L'}{L' + K_d}$. In general, Equation 22 only holds if the change in excess chemical potential is negligible between L' and the simulation concentration L. This assumption can be tested by running bulk decoupling (Step D) at both L' and L and checking that the resulting change in ΔG_{bulk} is much smaller than the overall error. If we wish to calculate P_{occ} over a wider concentration range in which this assumption does not hold, we would need to explicitly calculate $\Delta G^*_{\text{bulk}}(L)$ for multiple simulation concentrations L and extrapolate to the intermediate concentrations, as in Ref. [11].

Appendix F Troubleshooting

We have written this tutorial to be as robust as possible but also generalizable to other systems. In the process of applying these steps to your own system of interest, however, additional challenges may arise. When calculations fail to converge or appear to converge to unreasonable values, it can be difficult to discern what has gone wrong without simply starting over. We provide here some of the most common issues and their respective fingerprints as cautionary tales and troubleshooting tools. If you encounter a problem with running the tutorial as written and do not see your issue listed below, please contact us.

Appendix F.1 Problems with Running; NAMD crashes

Alchemical FEP will make any instabilities in a system more apparent as well as introduce a few more possible sources of instability. The most common problems are RATTLE errors and box size instability

Appendix F.1.1 RATTLE Errors

```
ERROR: Constraint failure in RATTLE
algorithm for atom 593!
```

Causes:

If the usual culprits (poor equilibration, long time steps, and over-aggressive RESPA settings) have been ruled out, the most likely causes of RATTLE errors during FEP are 1) too-wide lambda windows and 2) a too-low soft-core potential exponent.

Solutions:

Lambda windows can easily be narrowed by reducing `dLambda`. Values between 0.05 and 0.005 give a good balance between efficiency and accuracy. Note, decreasing `dLambda` will result in more windows which will require more CPU time overall.

Appendix F.1.2 Box Size Instability

```
FATAL ERROR: Periodic cell has become
too small for original patch grid!

Possible solutions are to restart
from a recent checkpoint, increase
margin, or disable useFlexibleCell
for liquid simulation.
```

Causes:

While classical MD is "tolerant" to small periodic boxes and aggressive barostats, combining these with FEP is particularly unstable.

Solutions:

First, the periodic box should be at least twice the solute size or twice the cutoff distance (whichever is longer) in order to avoid violating the minimum image convention which can cause instabilities especially with charged solutes and during FEP decoupling. Second, at least with the Langevin barostat, slowing the piston dynamics can improve system stability at the cost of slowing box-size relaxation. We use `LangevinPistonPeriod` between 75 and 200, and `LangevinPistonDecay` between 50 and 100. `LangevinPistonDecay` should always be about half `LangevinPistonPeriod` . See The NAMD UG for more details. [13]

Appendix F.2 Problems with Results; Poor Convergence

Convergence within each step is a prerequisite to a reliable final estimate of $\Delta G^{\circ}_{\text{bind}}$. Large errors and internal inconsistencies often indicate poor equilibration or under-sampling of one or more ensembles. Each leg of a SAFEP calculation has unique challenges and edge-cases which we address below. In general, convergence may be improved by increasing the simulation time for each lambda value.

Appendix F.2.1 Local and Misleading Convergence

In the case of very slow fluctuations or in the presence of metastable states, a FEP calculation may be converge locally and give a biased outcome. The best way to detect this is to run multiple replicas that are as weakly correlated as possible (i.e. true replicas are better than pseudo-replicas with identical initial conditions, and pseudo-replicas are better than no replicas).

In this tutorial, we include analysis of the protein-ligand bound state ensemble because it directly affects the definition of the DBC. Simulation of the apo protein (without ligand in the binding pocket), however, can provide useful information about the decoupled end-point. In the case of lysozyme, for example, the binding pocket is frequently occupied by one or two water molecules. If the lysozyme binding pocket does not recover hydration once the ligand is fully decoupled during FEP, the calculation overestimates the strength of binding by up to 0.5 kcal/mol. This is a small error compared to the overall precision of the technique, but users should be aware that assessing endpoint hydration is particularly important for larger or more hydrated binding pockets.

Appendix F.2.2 Step A: Define the DBC

Symptom: Multimodal DBC Distribution (e.g. Figure 9)

Causes:

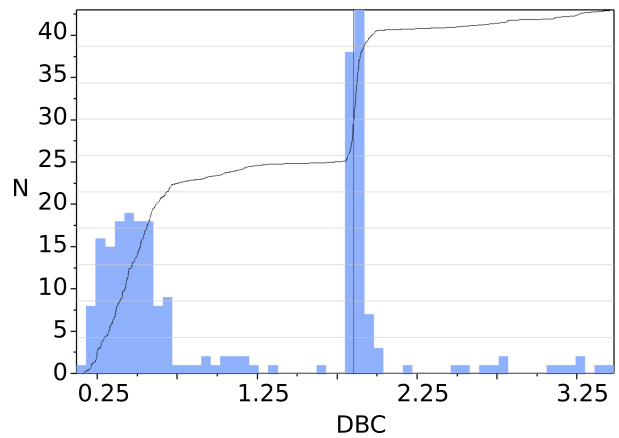


Figure 9. Screenshot from the Colvars Dashboard showing a bimodal distribution that resulted from using an asymmetric DBC for phenol. The second peak corresponds to a 180 degree rotation about the C-O axis.

There are three main causes of multimodal DBC distributions: 1) ligand unbinding, 2) multiple binding modes, and 3) multiple, nearby binding sites.

Solutions:

Use the Colvars Dashboard histogram tool to probe the conformations associated with each mode and decide which modes correspond to bound and unbound states.

Bound and Unbound modes	If all but one mode may be described as unbound, place the DBC restraint upper wall between the bound mode and the unbound mode with the lowest DBC value. Proceed with FEP.
Multiple, indistinguishable bound modes	Such modes are a result of symmetric ligands and are best addressed using a symmetric DBC. See Appendix C.2.2 for more details.
Multiple, distinguishable bound modes	If one or more bound mode(s) is meaningfully distinct from some other mode(s), select a representative frame for each class. These frames become reference poses for each binding "site" from which you must calculate ΔG_{site}, ΔG_{DBC}, and ΔG_{V} separately. Free energy contributions from each mode can be combined by exponential summing as shown in equation 12 of [27].

Appendix F.2.3 Step B or D: FEP Calculations

Symptom: DBC restraint energy stays close to 0

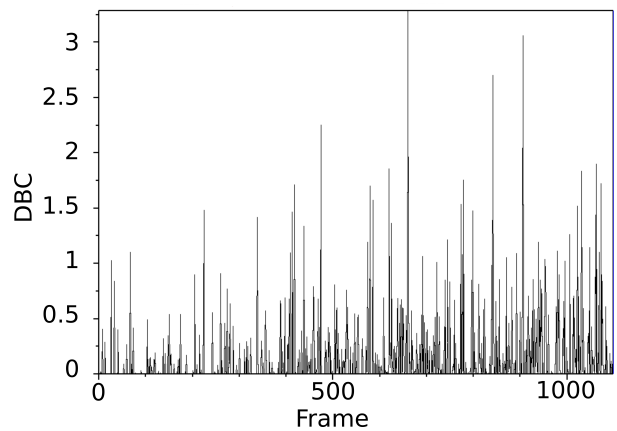
Causes:

The DBC restraint may be too wide (`upperWalls`) or too soft (`forceConstant`). The lambda windows may be too short to properly sample the decoupled ensemble.

Solutions:

First, watch the trajectory for any abnormal behavior; wide DBCs will often be obvious from the last several nanoseconds of a FEP run because the ligand will explore a much larger conformation space than expected. See the DBC debugging checklist below. If the system passes all checks, try running λ = 1 for longer to make sure the DBC restraint is functioning.

Symptom: DBC energy is consistently greater than 0



Causes:

This issue is most often due to too-narrow DBC restraints (`upperWalls`) or other mistakes in the colvar definition.

Solutions:

This problem is harder to diagnose from the trajectory alone unless there are obviously over-restrained degrees of freedom in the ligand. Consult the DBC debugging checklist below.

Symptom: Ligand Unbinding during FEP

Cause:

The most likely cause of unbinding during FEP is a DBC restraint that is too wide (`upperWalls`), too soft (`forceConstant`), inactive, or improperly defined.

Solutions:

Consult the checklist below:

DBC Debugging Checklist:

- ☐ Only the DBC restraint should be active during FEP (Step B)
- ☐ DBC restraint upper walls have the intended value. (2.b)
- ☐ DBC restraint force constant is appropriate (100 or 200). (2.b)
- ☐ NO lower walls (2.b)
- ☐ If the Colvars configuration file contains a "width" keyword, it should be 1. See [28] and the Colvars user guide for more details. (2.b)

Symptom: Very Large Hysteresis near λ = 0.5

Hysteresis (δ_λ) larger than 1 kcal/mol for any lambda window suggests poor convergence.

Causes:

Large hysteresis values are most often caused by: 1) insufficient equilibration, 2) short windows (less than a few hundred ps), or 3) wide windows (large `dLambda`).

Solutions:

If the system is well-relaxed and equilibrated by the usual metrics (box size, pressure, temperature, etc.), then it is

most likely that either the lambda windows are too short or too wide. Try increasing the sampling time or increasing the total number of windows. We have had good results with 120 windows of 3 ns each, but longer may be necessary for particularly unwieldy systems.

Symptom: Very Large Hysteresis near $\lambda = 0$ or $\lambda = 1$
Causes:
Large hysteresis near the end-points of the FEP calculation are most commonly caused by so-called "end-point catastrophes." See The NAMD UG for more details. [13]
Solutions:
Ensure `alchVdwShiftCoeff` is between 5 and 8. If this is already the case, and no other part of the calculation is problematic, try doubling the number of windows between the window with large hysteresis and the nearest end-point.

Symptom: Hysteresis Oscillates or is Otherwise Correlated with λ.
As noted above, δ_λ should be independent of lambda with a mean of 0.
Causes:

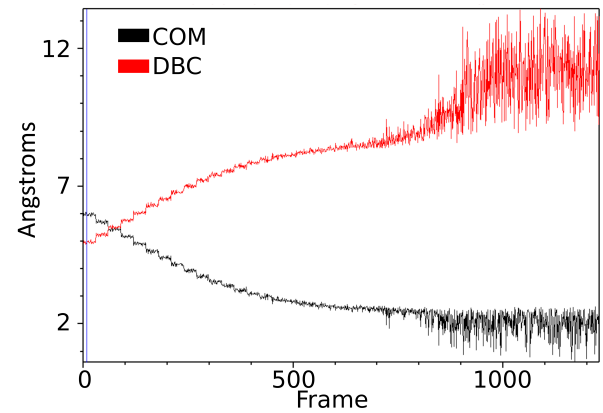
1. Some versions of NAMD have a bug that allows FEP data to be written on a step without energy calculations. This results in the use of stale energies (from a previous step) and inaccurate estimates for differences in energy.
2. The lambda windows may be too short to reach equilibrium

Solutions:

1. Manually ensure that `alchOutFreq` is a multiple of both `fullElectFrequency` and `nonbondedFreq`. See Bug advisory and Workaround for more details.
2. Extend the length of each window or decrease `dLambda`.

Appendix F.2.4 Step C: TI Calculation

The DBC restraint should do the most work early in the TI calculation then, as the force constant is scaled out, the COM restraint should take over and keep the center of mass in a well-defined volume. TI convergence issues are most easily diagnosed by watching the MD trajectory and examining the colvar trajectories.



Symptom: The collective variable trajectory is abnormal.
Causes:
Strange behavior is often caused by a mismatch between the coordinates used to determine the center of the COM restraint and the reference coordinates used for the DBC restraint as is the case in the figure above.
Solutions:
Update the reference coordinates to match those used for determining the center of the COM restraint. That is, revisit Step C1.b paying special attention to the coordinates used. Make sure you save your `refPositionsFile` paths are correct.

Symptom: COM restraint energies during TI are very high or very low and don't change much.
Causes:
The harmonic walls `forceConstant` is likely too stiff (high) or too soft (low).
Solutions:
Adjust the force constants to be between 100 (minimum) and 200.

Symptom: The ligand doesn't move during TI (or moves very little).
Causes:

1. The COM restraint is too narrow (`upperWalls` is too small).
2. The run is too short to visit all lambda values.
3. The DBC restraint isn't being released.

Solutions:

1. Double check the choice of wall position and ensure that it is correct in all Colvars configuration files.
2. Make sure your run is long enough; if you increase the `targetNumSteps` parameter, you must also increase the run time proportionally.
3. If you are using our *TI.tcl* script, double-check all settings and parameters. If you have written your own TI

Colvars config file, check all settings and compare them to the settings reported in the NAMD log file.

Symptom: The TI calculation doesn't converge *and* the ligand moves very far away from its initial position.
Causes:
The COM restraint is probably too wide (or non-existent).
Solutions:
Double check the existence and parameters of the COM restraint.

4 Author Contributions

ESM, ME and JWS ran, tested and refined the protocol. ESM wrote the notebook and analysis scripts. GB and JH designed and supervised the work. All authors wrote the document.

5 Other Contributions

The authors are grateful to Ms. Noureen Abdelrahman, Ms. Mariadelia Argüello-Acuña, Mr. Jahmal Ennis, Mr. Connor Pitman, and Mr. Jules Marien for providing feedback and initial testing of this tutorial. The authors acknowledge the Office of Advanced Research Computing (OARC) at Rutgers, The State University of New Jersey for providing access to the Amarel cluster and associated research computing resources.

6 Potentially Conflicting Interests

The authors declare no potential conflict of interests.

7 Funding Information

We acknowledge financial support from the National Science Foundation DGE 2152059 (to ESM, JWS, and GB), the Ministry of Science, Research, and Technology of the Islamic Republic of Iran for a Ph.D. candidate research grant to ME, and the French Agence Nationale de la Recherche (ANR) for grant LABEX DYNAMO (ANR-11-LABX-0011-01, to JH).

Author Information

ORCID:
Ezry Santiago-McRae: 0000-0002-0930-8277
Mina Ebrahimi: 0000-0001-6204-5886
Jesse W. Sandberg: 0000-0001-7466-8466
Grace Brannigan: 0000-0001-8949-2694
Jérôme Hénin: 0000-0003-2540-4098

References

[1] **Gilson MK**, Given JA, Bush BL, McCammon JA. The statistical-thermodynamic basis for computation of binding affinities: a critical review. Biophysical journal. 1997; 72(3):1047–1069.

[2] **Mey ASJS**, Allen BK, Bruce McDonald HE, Chodera JD, Hahn DF, Kuhn M, Michel J, Mobley DL, Naden LN, Prasad S, Rizzi A, Scheen J, Shirts MR, Tresadern G, Xu H. Best Practices for Alchemical Free Energy Calculations [Article v1.0]. Living Journal of Computational Molecular Science. 2020; 2(1):18378. https://doi.org/10.33011/livecoms.2.1.18378.

[3] **Hamelberg D**, McCammon JA. Standard free energy of releasing a localized water molecule from the binding pockets of proteins: double-decoupling method. Journal of the American Chemical Society. 2004; 126(24):7683–7689.

[4] **Woo HJ**, Roux B. Calculation of absolute protein–ligand binding free energy from computer simulations. Proceedings of the National Academy of Sciences. 2005; 102(19):6825–6830.

[5] **Hermans J**, Shankar S. The Free Energy of Xenon Binding to Myoglobin from Molecular Dynamics Simulation. Israel Journal of Chemistry. 1986; 27(2):225–227. https://doi.org/https://doi.org/10.1002/ijch.198600032.

[6] **Deng Y**, Roux B. Calculation of standard binding free energies: Aromatic molecules in the T4 lysozyme L99A mutant. Journal of Chemical Theory and Computation. 2006; 2(5):1255–1273.

[7] **Hénin J**, Gumbart J, Chipot C. In silico alchemy: A tutorial for alchemical free-energy perturbation calculations with NAMD. see www ks uiuc edu/Training/Tutorials/namd/FEP/tutorial-FEP pdf for the NAMD software. 2017; .

[8] **Hermans J**, Wang LU. Inclusion of loss of translational and rotational freedom in theoretical estimates of free energies of binding. Application to a complex of benzene and mutant T4 lysozyme. Journal of the American Chemical Society. 1997; 119(11):2707–2714.

[9] **Boresch S**, Tettinger F, Leitgeb M, Karplus M. Absolute binding free energies: a quantitative approach for their calculation. The Journal of Physical Chemistry B. 2003; 107(35):9535–9551.

[10] **Fu H**, Cai W, Hénin J, Roux B, Chipot C. New coarse variables for the accurate determination of standard binding free energies. Journal of Chemical Theory and Computation. 2017; 13(11):5173–5178.

[11] **Salari R**, Joseph T, Lohia R, Hénin J, Brannigan G. A streamlined, general approach for computing ligand binding free energies and its application to GPCR-bound cholesterol. Journal of chemical theory and computation. 2018; 14(12):6560–6573.

[12] **Helms V**, Wade RC. Thermodynamics of water mediating protein-ligand interactions in cytochrome P450cam: a molecular dynamics study. Biophysical Journal. 1995; 69(3):810–824. https://doi.org/https://doi.org/10.1016/S0006-3495(95)79955-6.

[13] **Bernardi R**, Bhandarkar M, Bhatele A, Bohm E, Brunner R, Buch R, Buelens F, Chen H, Chipot C, Dalke A, Dixit S, Fiorin G, Freddolino P, Fu H, Grayson P, Gullingsrud J, Gursoy A, Hardy D, Harrison C, Hénin J, et al., NAMD User's Guide version 2.14. Theoretical biophysics group, University of Illinois and Beckman Institute; 2020. https://www.ks.uiuc.edu/Research/namd/2.14/ug/, accessed Aug 8, 2022s.

[14] **Hénin J**, Lopes LJS, Fiorin G. Human Learning for Molecular Simulations: The Collective Variables Dashboard in VMD. Journal of Chemical Theory and Computation. 2022; 18(3):1945–1956. https://doi.org/10.1021/acs.jctc.1c01081, pMID: 35143194.

[15] **Phillips J**, Isgro T, Sotomayor M, Villa E, NAMD TUTORIAL. NIH Center for Macromolecular Modeling and Bioinformatics, Beckman Institute; 2017. http://www.ks.uiuc.edu/Training/TutorialsOverview/namd/namd-tutorial-unix.pdf.

[16] **Chen H**, Maia JDC, Radak BK, Hardy DJ, Cai W, Chipot C, Tajkhorshid E. Boosting Free-Energy Perturbation Calculations with GPU-Accelerated NAMD. Journal of Chemical Information and Modeling. 2020; https://doi.org/10.1021/acs.jcim.0c00745.

[17] **Phillips J**, Hardy D, Maia J, Stone J, Ribeiro J, Bernardi R, Buch R, Fiorin G, Hénin J, Jiang W, McGreevy R, Melo MCdR, Radak B, Skeel R, Singharoy A, Wang Y, Roux B, Aksimentiev A, Luthey-Schulten Z, Kale L, et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. The Journal of Chemical Physics. 2020; 153:044130.

[18] **Humphrey W**, Dalke A, Schulten K. VMD: visual molecular dynamics. Journal of molecular graphics. 1996; 14(1):33–38.

[19] **Ebrahimi M**, Hénin J. Symmetry-Adapted Restraints for Binding Free Energy Calculations. Journal of Chemical Theory and Computation. 2022; 18(4):2494–2502. https://doi.org/10.1021/acs.jctc.1c01235.

[20] **Shirts MR**, Chodera JD. Statistically optimal analysis of samples from multiple equilibrium states. The Journal of chemical physics. 2008; 129(12):124105. https://doi.org/10.1063/1.2978177.

[21] **Merski M**, Shoichet BK. The impact of introducing a histidine into an apolar cavity site on docking and ligand recognition. Journal of medicinal chemistry. 2013; 56(7):2874–2884.

[22] **Jo S**, Kim T, Iyer VG, Im W. CHARMM-GUI: A web-based graphical user interface for CHARMM. Journal of Computational Chemistry. 2008; 29(11):1859–1865. https://doi.org/10.1002/jcc.20945.

[23] **Lee J**, Cheng X, Swails JM, Yeom MS, Eastman PK, Lemkul JA, Wei S, Buckner J, Jeong JC, Qi Y, Jo S, Pande VS, Case DA, Brooks CL, MacKerell AD, Klauda JB, Im W. CHARMM-GUI Input Generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM Simulations Using the CHARMM36 Additive Force Field. Journal of Chemical Theory and Computation. 2016; 12(1):405–413. https://doi.org/10.1021/acs.jctc.5b00935.

[24] **Zacharias M**, Straatsma TP, McCammon JA. Separation-shifted scaling, a new scaling method for Lennard-Jones interactions in thermodynamic integration. The Journal of chemical physics. 1994; 100(12):9025–9031.

[25] **Tuckerman M**. Statistical mechanics: Theory and molecular simulation. Oxford Graduate Texts, London, England: Oxford University Press; 2010.

[26] **Frenkel D**, Smit B. Understanding molecular simulation: from algorithms to applications, vol. 1. Elsevier; 2001.

[27] **Mobley DL**, Chodera JD, Dill KA. On the use of orientational restraints and symmetry corrections in alchemical free energy calculations. The Journal of chemical physics. 2006; 125(8):84902.

[28] **Fiorin G**, Klein ML, Hénin J. Using collective variables to drive molecular dynamics simulations. Molecular Physics. 2013; 111(22-23):3345–3362.