UC Irvine
UC Irvine Previously Published Works

Title
Guidelines for the analysis of free energy calculations

Authors
Klimovich, Pavel V
Shirts, Michael R
Mobley, David L

Guidelines for the analysis of free energy calculations

Pavel V. Klimovich, Michael R. Shirts, and David L. Mobley

Abstract Free energy calculations based on molecular dynamics (MD) simulations show considerable promise for applications ranging from drug discovery to prediction of physical properties and structure-function studies. But these calculations are still difficult and tedious to analyze, and best practices for analysis are not well defined or propagated. Essentially, each group analyzing these calculations needs to decide how to conduct the analysis and, usually, develop its own analysis tools. Here, we review and recommend best practices for analysis yielding reliable free energies from molecular simulations. Additionally, we provide a Python tool, `alchemical-analysis.py`, freely available on GitHub at `https://github.com/choderalab/pymbar-examples,` that implements the analysis practices reviewed here for several reference simulation packages, which can be adapted to handle data from other packages. Both this review and the tool covers analysis of alchemical calculations generally, including free energy estimates via both thermodynamic integration and free energy perturbation-based estimators. Our Python tool also handles output from multiple types of free energy calculations, including expanded ensemble and Hamiltonian replica exchange, as well as standard fixed ensemble calculations. We also survey a range of statistical and graphical ways of assessing the quality of the data and free energy estimates, and provide prototypes of these in our tool. We hope these tools and discussion will serve as a foundation for more standardization of and agreement on best practices for analysis of free energy calculations.

Keywords hydration free energy · transfer free energy · free energy calculation · analysis tool · binding free energy · alchemical

P. Klimovich · D. Mobley
Department of Pharmaceutical Sciences
and Department of Chemistry
147 Bison Modular
University of California, Irvine
Irvine, CA 92697
Tel.: 949-824-6383
Fax: 949-824-2949
E-mail: dmobley@mobleylab.org

M. R. Shirts
Department of Chemical Engineering
University of Virginia
Charlottesville, VA 22094

D. Mobley
Department of Chemistry
University of New Orleans
2000 Lakeshore Drive
New Orleans, LA 70148

1 Introduction

1.1 Free energy calculations assist drug discovery

Complex chemical and biological systems pose a key challenge for modern molecular and computational science. We seek computational models which can provide quantitative predictions, not just qualitative insight. Researchers seek to answer questions such as "how much?", "how big?", "how tight?" and so on, and increasingly apply physically-detailed computation to help answer these questions. Models seek to mimic or simulate the processes in question, helping reveal and provide new understanding of mechanisms and phenomena which might be challenging or impossible to probe experimentally [20, 23, 38, 43].
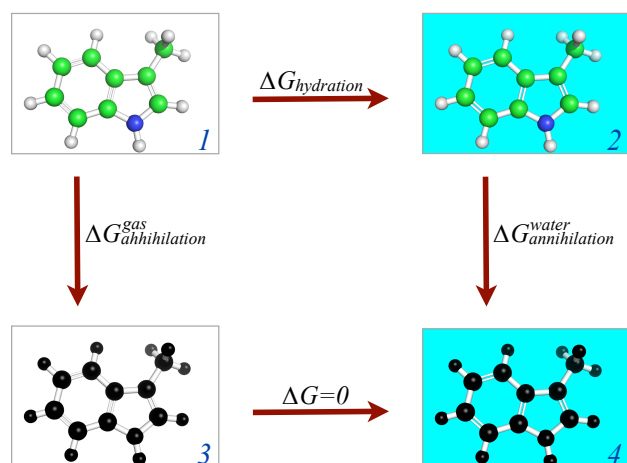
Free energy calculations [11, 7, 25, 9, 18] provide a good example of a computational technique which provides a quantitative answer to a specific question – in this case, "what is the free energy difference between the two thermodynamic end states of the system?" This question arises, for example, in drug discovery [16] where drugs need to be ranked

Fig. 1 The thermodynamic cycle for a standard hydration free energy calculation. Here, the blue background represents water and the clear background represents gas. The goal is to find the free energy difference $\Delta G_{hydration}$ between the two states: *end state 1* (upper left) representing the solute in the gas phase, and *end state 2* (upper right) depicting the solvated molecule. The $\Delta G_{hydration}$ is found as a sum of the free energy changes between the end states 1 and 2 and the intermediate alchemical states, *intermediate state 3* (lower left) and *intermediate state 4* (lower right), introduced along the alternative pathway $1 \rightarrow 3 \rightarrow 4 \rightarrow 2$, which may include additional nonphysical states interpolating between the legs $1 \rightarrow 3$ $3 \rightarrow 4$, and $4 \rightarrow 2$. The black-and-white appearance of the solute molecule, at bottom, indicates the solute is in a state where it has no non-bonded interactions with its environment, and possibly also no internal non-bonded interactions as well. These scenarios are referred to as *decoupling* and *annihilation*, respectively, as discussed in the text. In the case of decoupling, the transformation pathway reduces to the single leg $2 \rightarrow 4$, and the hydration free energy is the negative of the free energy for this leg. In this case of annihilation, which also modifies internal solute interactions, both legs are necessary.

by, among other criteria, their binding affinity [9] to a target protein. To use free energy calculations to answer this question, one must first build a model of the system, and then identify the end states between which the free energy difference is to be computed.

### 1.2 Free energy calculations begin with a definition of the end states

The thermodynamic end states (for example, Fig. 1 states 1 and 2) are the key starting point in free energy calculations. In principle, free energy differences between the end states can be computed simply from simulations conducted in one or both states [11]. But in practice, this is typically not possible for biomolecular systems on reasonable timescales. To compute accurate free energy differences between states, their phase space integrals must have sufficient overlap which in practice is attainable only when both states are extremely similar. When this is not the case, it can be impossible to directly compute the free energy difference between end states

1 and 2. In such cases, we can instead compute free energy differences between a series of intermediate states which *do* have sufficient overlap, leading from state 1 to state 2. These intermediate states are typically artificial, unphysical states constructed to link the physical states of interest, and form part of a thermodynamic cycle (see Fig. 1) linking the two end states of interest. For most free energy calculations relevant to binding, solvation, and solubility, this alternate, unphysical pathway involves effectively deleting and/or inserting some atoms, while possibly also making parameter changes to those and other atoms.

1.3 The thermodynamic cycle depicts alternate paths between the end states

The thermodynamic cycle for standard hydration free energy calculations is comprised of the four legs joining the four states of interest (Fig. 1): (2) a molecule interacting with a box of water, (1) the same molecule present alone in the gas phase, (4) the molecule in water but not interacting with the surrounding water, and (3) the non-interacting molecule again alone in the gas phase. We thus compute the solvation free energy by modifying the solute molecule in each of its environments. To do so, we compute the free energy of turning off the solute's non-bonded interactions with its environment (called *decoupling*) or turning off both internal non-bonded interactions and interactions with the environment (called *annihilation*) (Fig. 1)[1]. Specifically, we compute the free energies associated with Figure 1, $1 \rightarrow 3$ and $2 \rightarrow 4$, turning off the molecule's interactions in gas phase and in water, respectively. Most commonly, both of these transformations are carried out by first scaling (usually linearly) the solute charges to zero and then turning off the solute's Lennard-Jones (LJ) interactions (usually via the "soft-core" scheme [2]). These transformations are done in a series of steps by introducing a parameter λ which modulates the potential energy of the system, so that as λ goes from 0 to 1 the potential energy transitions between that of the initial state and that of the target final state. Simulations are then run at a set of different λ values connecting the two states. In other words, each of the two transformation pathways is subdivided into a variety of individual steps, where

---

[1] Annihilation and decoupling can be thought to differ primarily in how they handle charge and Lennard-Jones parameters. Specifically, annihilation involves actually setting solute partial charges to zero, while decoupling involves turning off charge interactions with environment. Likewise, annihilation involves actually setting the Lennard-Jones parameters to zero, while decoupling involves turning off interactions with environment. Our current explanation is specific to the more general case, annihilation, but in case of decoupling no gas transformation $1 \rightarrow 3$ is needed and the overall transformation reduces to the single leg $2 \rightarrow 4$, i.e. the hydration free energy change is found as the negative of $\Delta G_{decoupling}^{water}$, with the possible exception of an anaytical standard state correction depending on the experimental reference state employed.
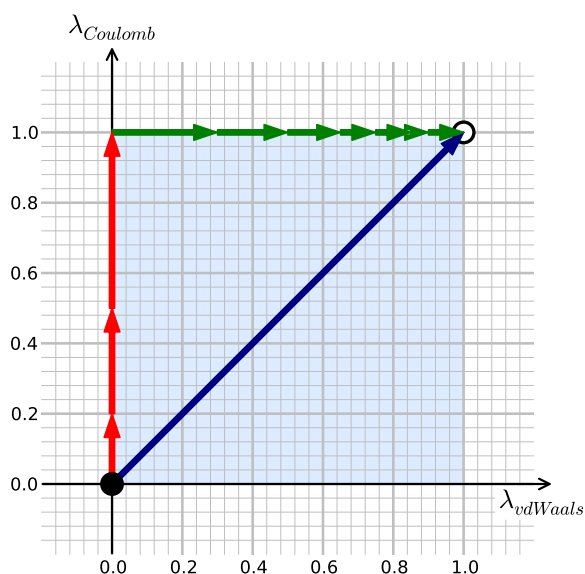
Fig. 2 The transformations $1 \rightarrow 3$ and $2 \rightarrow 4$ from Fig. 1 can be thought of as a pathway from (0,0) (filled circle) to (1,1) (hollow circle) in the lambda vector space which is shown as a pale blue square on the Cartesian plane formed by the axes $\lambda_{vdWaals}$ and $\lambda_{Coulomb}$, which control the solute van der Waals and Coulomb interactions, respectively. For the reasons discussed in the text, we start at point (0,0), which corresponds to the fully interacting molecule, and proceed through several alchemical intermediate states (with locations indicated by arrowheads) along the $\lambda_{Coulomb}$ axis (red arrows) until we reach point (0,1) which corresponds to the electrostatically non-interacting molecule. Then, we modify $\lambda_{vdWaals}$ (green arrows) until we reach our target state (1,1), corresponding to the non-interacting molecule. If instead transformation progress was controlled by a single λ value rather than a vector, the transformation path would have lain along the square diagonal (blue arrow).

each step involves a transition between two λ values. The number and spacing of λ values is chosen to assure adequate overlap between the conformational spaces of the two states being considered. These intermediate states, and their corresponding λ values, are normally said to be *alchemical*, since they correspond to unphysical states, often involving a change in the chemical identity of the species considered.

Following alchemical transformation of the molecule in gas and solution, it remains to connect the two end states ((3) and (4) in Fig. 1). However, the free energy of the non-interacting molecule does not depend on the nature of its environment, and so the transfer free energy of the non-interacting molecule between environments is 0. Thus, there is no associated free energy change going from states (3) to (4) in Fig. 1.

1.4 The interactions can be decoupled

Changes in electrostatic interactions are often separated from changes in LJ interactions to avoid inaccuracy in the free energy estimate and sampling challenges. Specifically, if electrostatic interactions are retained while an atom's LJ interactions are being removed, the associated charge becomes more and more exposed, and can create huge electrostatic forces leading to large and expensive-to-converge free energy differences. In extreme cases this can result in the lack of separation between positive and negative charges, which is especially problematic, potentially leading to numerical instabilities and simulation crashes [3, 31]. An additional benefit of separating these transformations is it provides a mechanism to maintain optimally efficient linear scaling of the charge interactions [27] with λ while using alternative scaling schemes for LJ interactions. Specifically, to avoid situations when the derivative $\partial U / \partial \lambda$ (needed for the TI analysis) would have been discontinuous, the potential for LJ transformations is typically treated via the *soft-core* potential [2, 42, 36, 39]. Although we focus here on the decoupled scheme, our general analysis would apply to the combined case (the one that would require no electrostatic decoupling [5, 12]) as well.

1.5 The intermediate lambda states can be controlled by lambda vectors

The MD packages like GROMACS [33] and DESMOND [37] can handle free energy calculations via multiple λ values controlling progress of different interaction types so that, for example, Coulomb, LJ, and restraining transformations can be controlled separately. Each step along the transformation path is associated with a unique set of λ values that is often referred to as the λ vector. For the thermodynamic cycle in Fig. 1 the λ vector has two components that control the Coulomb and LJ interactions. Each of the transformation paths $1 \rightarrow 3$ and $2 \rightarrow 4$ can then be presented as a train of intermediate coupled states $(\lambda_{coul}, \lambda_{LJ})$, the initial and final states being (0, 0) and (1,1) as depicted in Fig 2. If λ controls the strength of Coulomb and LJ interactions with the solute's environment, then as λ progresses, solute-solvent interactions gradually decrease until, at the end state, the system consists of pure solvent, overlaid by a parallel system consisting of the non-interacting, isolated solute with full internal interactions. In calculations where internal solute non-bonded interactions are removed as well, the solute end state is slightly different, consisting of an assembly of atoms which interact only via their bonded interactions.

Once λ states are selected, equilibrium simulations are carried out, storing the necessary information for analysis (typically $\partial U / \partial \lambda$, the derivative of the potential with respect to λ, and $\Delta U_{i,j}$, the potential energy differences between

states at the different λ values evaluated from individual trajectories).

1.6 The automated analysis should be an essential part of the free energy calculations

Free energy calculations have typically been an experts-only endeavor, and one reason for this is that both their setup and analysis require, as a rule, substantial manual intervention. Analysis often involves in-house scripts and as more researchers get involved with the free energy calculations, standard analysis tools become increasingly important both to help ensure best practices are followed, and to avoid duplication of effort.

Our focus here is on the analysis of free energy calculations, which typically consists of a series of sequential steps. These free energy calculations themselves can be conducted with a variety of different sampling techniques, and our focus here is primarily on the analysis stage, regardless of sampling technique.

Here we present what we believe are current best practices for analysis of alchemical free energy calculations. Conceptually, we break analysis into four main stages:

1. subsampling the data to retain uncorrelated samples
2. calculating free energy differences along with the corresponding statistical errors via a variety of TI- and FEP-based methods
3. producing textual and graphical outputs of the computed data
4. inspecting for
   – for convergence and identifying the equilibrated portion of the simulation
   – good phase space overlap for all pairs of adjacent lambda states

2 Analysis concepts, theory, and free energy estimation

We focus our attention on the analysis of a model free energy calculation—in this case, we choose a hydration free energy calculation of 3-methylindole as an illustrative example. Subsequent discussion will assume the reader already has run a free energy calculation and wishes to analyze the resulting data. In our case, we have run this free energy calculation in GROMACS, and we provide a Python tool which implements the procedures described here for GROMACS, SIRE (`http://siremol.org/Sire/Authors.html`), and AMBER [6] data files, with examples. However, except for reading the input data, our code is independent of the specific simulation package, and can easily be adapted to work with any data format containing the quantities needed by the free energy estimators used here ($\partial U/\partial \lambda$ for TI- and

$\Delta U_{i,j}$ for FEP-based estimators). Thus, while our example here uses the case of the GROMACS simulation package, our prototype tool, freely available on GitHub `https://github.com/choderalab/pymbar-examples`, can easily be modified to work with other simulation packages.

2.1 Obtaining input data

As noted, the key input information needed for full, general analysis of free energy calculations includes potential energy differences between (at least) adjacent lambda values, as well as $\partial U/\partial \lambda$ values at all lambda values. To be specific, we will give an example of the calculation with GROMACS-formatted input files. In particular, GROMACS currently (v3.3 through v5.0) store all energies to binary energy files, but also write out all the potential energies and differences thereof needed for analysis to human-readable text files with the `.xvg` file format. In GROMACS, these are formatted as shown in Fig. 3. For standard simulations (in contrast to *expanded ensemble* simulations discussed below), there are several such files - one for each λ value. In GROMACS, the precise number of $\partial U/\partial \lambda$ fields varies with the number of different types of λ value which are utilized, which corresponds to the number of dimensions in the λ vector. [2]

Expanded ensemble simulations [22, 21, 24, 29, 26] are an approach which allows for simultaneous exploration of both λ and coordinate space in a single simulation, potentially allowing for faster sampling across alchemical states provided that the kinetic barriers that divide conformations important are lower at other λ states. In expanded ensemble simulations, a single simulation samples all states, and thus produces a single energy file. At each time step, the simulation is in one specific λ state, which is stored in the second field of the output energy file only in the expanded ensemble case. This allows determination of which λ state stored $\partial U/\partial \lambda$ and $\Delta U_{i,j}$ values belong to.

## 2.2 Alchemical analysis techniques can be divided into families

As noted, a variety of methods can take output from alchemical free energy calculations and yield free energy differences. Conceptually, these methods can be divided into two categories based on the quantity used to compute ΔG: thermodynamic integration [17] (TI) methods and free energy perturbation [44] (FEP) methods.

In TI, the free energy change along the path composed of K states is computed as a weighted sum of the ensemble

---

[2] Whenever there is an additional field corresponding to the pV energy term it will be added to the potential energy of corresponding state.

Time	Energy	$\partial U/\partial\lambda_{Coul}$	$\partial U/\partial\lambda_{vdW}$	$\Delta U_{1,0}$	$\Delta U_{1,1}$	$\Delta U_{1,2}$	...	$\Delta U_{1,10}$
0.0000	-28935.719788	79.8439	-169.8430	-15.9688	0.0000	23.9532	...	95.7991
1.0000	-28999.753933	64.3186	-73.0815	-12.8637	0.0000	19.2956	...	104.3823
2.0000	-29076.020596	38.3567	-83.7201	-7.6713	0.0000	11.5070	...	80.3175
3.0000	-29122.694521	73.6638	-213.0380	-14.7328	0.0000	22.0991	...	79.2340
4.0000	-29133.722682	38.2535	-70.6495	-7.6507	0.0000	11.4761	...	80.3560
5.0000	-29147.649111	43.7478	-109.7610	-8.7496	0.0000	13.1244	...	78.1552
6.0000	-29161.004598	79.9055	-82.7909	-15.9811	0.0000	23.9717	...	110.6894
7.0000	-29144.287730	59.0886	-97.2498	-11.8177	0.0000	17.7266	...	95.7572
8.0000	-29109.173478	27.1675	-62.3135	-5.4335	0.0000	8.1502	...	76.0401
9.0000	-29064.050011	57.4257	-91.3534	-11.4851	0.0000	17.2277	...	84.2167

Fig. 3 Sample GROMACS free energy calculation data for the first ten snapshots of 3-methylindole in water, as the data appears in the `dhdl.1.xvg` file, with the names for each field of a row given in the header. These are: time in picoseconds, the energy of the system (either potential or total, depending on the option used), the total energy derivative with respect to all the lambda types employed in the perturbation ($\partial U/\partial\lambda_{Coul}$ and $\partial U/\partial\lambda_{vdW}$), the total energy differences evaluated between the current lambda state of index 1 and the other states ($\Delta U_{i=1,j}$, j=0, 1, ..., 10 in this case), which reduces to the difference in potential energy when there is no mass perturbation. Effective in GROMACS version 4.0, the default setting is to evaluate the $\Delta U_{i,j}$ between the adjacent states only (for our example here – $\Delta U_{1,0}$, $\Delta U_{1,1}$, and $\Delta U_{1,2}$), and whenever all the differences are needed (which is essential for MBAR) the `.mdp` option `calc-lambda-neighbors` should be set to −1.
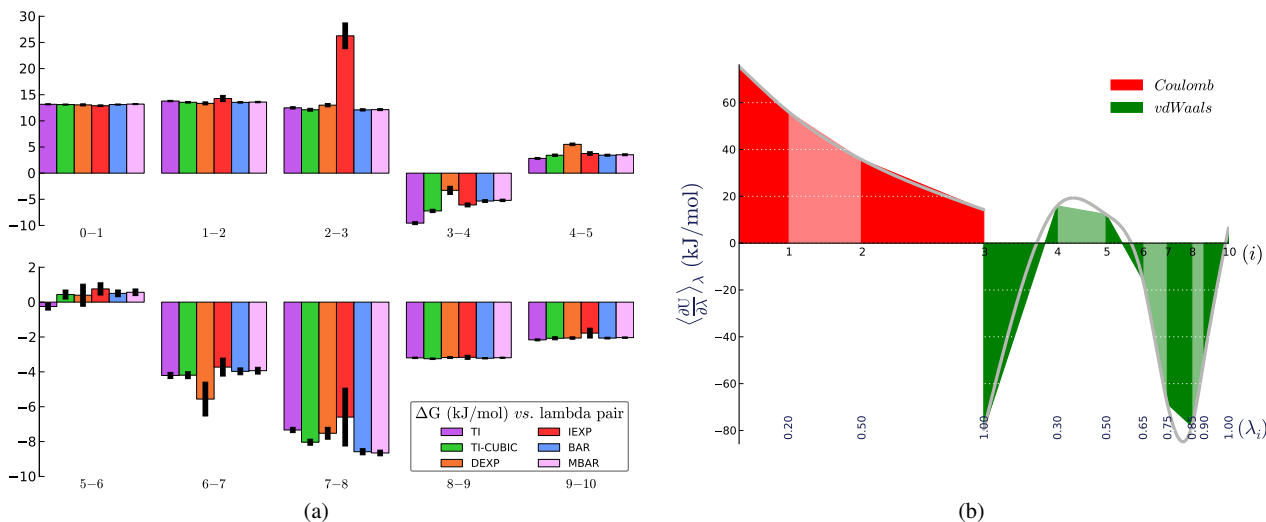
Fig. 4 The hydration free energy calculation for 3-methylindole, analyzed by various methods. (a) A bar plot of the free energy differences evaluated between pairs of adjacent states via several methods, with corresponding error estimates for each method. (b) A plot of $\langle\frac{\partial U}{\partial\lambda}\rangle$ vs λ for thermodynamic integration, with filled areas indicating free energy estimates from the trapezoid rule, and silver curve indicating interpolation via cubic spline. Different ΔG components are shown in distinct colors: in red is the electrostatic ΔG component (λ indices 0 to 3), while in green is the van der Waals ΔG component (λ indices 3 to 10). Color intensity alternates with increasing λ index. Alternate interpolation schemes disagree most around λ points where the slope of $\langle\frac{\partial U}{\partial\lambda}\rangle$ changes suddenly, so these are regions where a more dense lambda spacing is desirable.

averages of the derivative of potential energy function with respect to the coupling parameter λ:

$$\Delta G = \sum_{i=1}^{K} W_i \left\langle \frac{\partial U}{\partial\lambda} \right\rangle_{\lambda_i} \tag{1}$$

where W_i are the weighting factors that depend on the numerical integration scheme used [28].

Several different schemes are available for numerical integration in TI. In our provided tool, alchemical-analysis.py, we implement TI-1 and TI-3 [28] which differ in how they interpolate between data points for integration. TI-1 uses the trapezoidal rule (a first-order polynomial), while TI-3 uses

a (natural) cubic spline. The relative performance of these different TI methods will depend on the nature of the underlying data and the shape of the $\frac{\partial U}{\partial\lambda}$ curve being integrated – and thus, it depends on the alchemical path chosen.

Perturbation-based methods include a broad range of techniques loosely related to FEP. In our prototype tool, these include Deletion Exponential Averaging (DEXP), Insertion Exponential Averaging (IEXP), Gaussian Deletion (GDEL), Gaussian Insertion (GINS), Bennett Acceptance Ratio (BAR), Unoptimized Bennett Acceptance Ratio (UBAR), Range-based Bennett Acceptance Ratio (RBAR), and Multistate Bennett Acceptance Ratio (MBAR). Some of them, like BAR [1] and MBAR [35], are in common use and are deeply

entrenched in the parlance of the field, while others either do not have customary, generally accepted names, or remain little-known. For these methods we will use naming conventions suggested by Paliwal and Shirts [28]. The first two, DEXP and IEXP, are based on the exponential averaging scheme of the potential energies between two adjacent states—the so-called Zwanzig relationship [44]:

$$\Delta G_{ij} = -\frac{1}{\beta} \ln \langle \exp(-\beta \Delta U_{ij}) \rangle_i \qquad (2)$$

Depending on the direction of the transformation the process can be interpreted as either "deletion" or "insertion", hence the first letters in the acronyms. Typically, one of these processes, DEXP, proceeds in the direction of increasing entropy, while the other, IEXP, proceeds in the direction of decreasing entropy.

If potential energy differences are distributed in a Gaussian manner (GINS and GDEL [28]), then the Zwanzig relationship reduces to [14, 13]:

$$\Delta G_{ij} = \langle \Delta U_{ij} \rangle - \frac{\beta}{2} \sigma^2_{\Delta U_{ij}} \qquad (3)$$

where $\sigma^2_{\Delta U_{ij}} = \langle \Delta U_{ij}^2 \rangle - \langle \Delta U_{ij} \rangle^2$. Again, here, the estimator is referred to as either GDEL when ΔU_{ij} is used in the direction of increasing entropy, or GINS when ΔU_{ij} is used in the direction of decreasing entropy.

Methods based directly on the Zwanzig relationship, such as those just discussed, yield alternate estimations for the free energy difference depending on the direction of the transformation. These discrepancies originate from undersampling in the tail regions of the ΔU_{ij} distributions [32], which yields biased free energy estimates. BAR [1] eliminates the bias in ΔG estimation by including both forward, ΔU_{ij}, and reverse, ΔU_{ji}, potential energy differences in the analysis. In BAR, the free energy change between intermediate states i and j (comprised of N_i and N_j microstates, respectively) is found by solving numerically the implicit function of ΔU_{ij}:

$$\frac{1}{\langle 1 + \exp[+\beta(\Delta U_{ij} - C)] \rangle_i} = \frac{1}{\langle 1 + \exp[-\beta(\Delta U_{ji} - C)] \rangle_j} \qquad (4)$$

where $C = \Delta G_{ij} + \frac{1}{\beta} \ln \left(\frac{N_j}{N_i} \right)$.

In addition to this full version of BAR, there are two BAR-related methods that are advantageous in that there is no need to retain all potential energy differences for postprocessing. These methods focus on accumulating the averages in eq 4 as the simulation progresses. This is achieved by either setting the constant $C = \beta^{-1} \ln(N_j/N_i)$ and thus avoiding the self-consistency procedure entirely (UBAR [28]), or picking a range of starting values of C and obtaining a

range of ΔG_{ij} estimates from which the one having minimum variance is chosen as an input value for the constant C thus making the self-consistent solution essentially precalculated (RBAR [28]). UBAR can have issues when the free energy is significantly different from zero, while RBAR is essentially as accurate as BAR as long as the true value of C is within $\approx 1 - 2k_B T$ of the one of the range of trial C values.

MBAR [35] constitutes a further development of the BAR method. In BAR, the free energy change between the two adjacent states is computed to yield the minimum variance given data collected at that single pair of states alone, while MBAR finds the best estimate of free energy changes between all states simultaneously by optimizing the matrix of the ΔG variances, thus making use of all available data. MBAR can also be considered [35] as a limiting case of the WHAM [19] method in which the histogram width is set to zero.

2.3 Cross-comparison of different analysis techniques can highlight problems

TI and perturbation-based analysis techniques have different limitations. Specifically, the accuracy of TI is not a direct function of overlap in energy distributions but instead is a function of the average curvature $\langle \frac{\partial^2 U}{\partial \lambda^2} \rangle$ [30]. On the other hand, perturbation-based techniques do not depend on smoothness of the integrand, but rather on overlap in the sampled energy distributions. Given these differences in input information and limitations, consistency checks across these method families can be a valuable tool for identifying analysis or sampling problems.

In our experience, comparing results from different methods can serve as a warning sign of either insufficient sampling or a λ spacing which is too wide, so it is useful to have a *family* of analysis methods. Fig. 4 shows hydration free energies for 3-methylindole computed via a variety of methods. As seen from Fig 4(b), the discrepancies between results from the alternate interpolation schemes are most prominent in the vicinities of the rapid change in the $\langle \frac{\partial U}{\partial \lambda} \rangle$ derivative (the van der Waals lambda states 3-4, 4-5, 5-6, and 7-8). A good practice, thus, is to ensure dense lambda spacing for these regions.

In general, approaches based on Zwanzig's relation are expected to break down earlier (in terms of phase space overlap) than other approaches, so IEXP and DEXP will tend to become inconsistent even when data may still be sufficient to obtain accurate free energy estimates from many of the other methods (see Fig. 3). In our work, we primarily look for disagreement between TI-based methods and BAR-based methods.

2.4 The free energy change is often broken down into components

Alchemical transformations are usually comprised of several conceptual steps which modify different terms in the potential. For example, solvation or binding free energy calculations are often separated at least into electrostatic and Lennard-Jones components. Thus, the free energy associated with modifying each of these terms in the potential can be computed separately, which can provide some qualitative insight. However, it is important to note that the free energies of each component are path-dependent observables. ΔG_{Coul} will change in value depending on in what order the electrostatic transformation is carried out, so it cannot be directly considered the electrostatic component of the free energy. In GROMACS, since λ is a vector, multiple steps can be handled within a single set of simulations, and our alchemical-analysis.py tool automatically handles separation of different free energy components. It automatically identifies the number of charging states and prints out along with the total free energy change its breakdown into free energy components, ΔG_{Coul} and ΔG_{vdW}. The van der Waals contribution is computed as $\Delta G_{Total} - \Delta G_{Coul}$, where ΔG_{Coul} is the free energy change between states differing only in their charge state (in GROMACS, controlled by `coul-lambda` or `fep-lambda`). Whenever there are other types of transformations involved (for example, λ controlling restraints holding a ligand in a binding site in a binding free energy calculation [4]) the difference $\Delta G_{Total} - \Delta G_{Coul}$ will not equal ΔG_{vdW}.[3]

2.5 Analysis should be carried out on uncorrelated samples

In general, the free energy expressions above are intended to be applied to a set of uncorrelated samples of the relevant observables, but simulation data will be stored more frequently. Thus, we typically need to re-sample the relevant energy derivatives $\partial U/\partial \lambda$ and/or energy differences $\Delta U_{i,j}$ to obtain uncorrelated samples. Particularly, samples used for computing the averages in eqs (1)–(4) are *statistically independent* samples. There are several techniques to identify and retain independent samples [11, 40, 10] and we find analysis of autocorrelation times to be particularly useful in this regard.

Autocorrelation analysis begins with calculation of the autocorrelation function, which is fairly standard procedure

in time series analysis. For a discrete set of N samples occurring time δt apart, the autocorrelation function of the observable A at a given point i is found as

$$C_A(i) = \frac{\sum_{j=0}^{N} \delta A(j) \delta A(j+i)}{\sum_{j=0}^{N} \delta A^2(j)} \tag{5}$$

where $\delta A(i)$ defines the deviation of the current observable from its mean

$$\delta A(i) = A(i) - \frac{1}{N} \sum_{i=0}^{N} A(i) \tag{6}$$

In our script we use $\partial U/\partial \lambda$ as the observable A, although any of the potential energy differences $\Delta U_{i,j}$ can be used as well.

The autocorrelation time τ is defined as the integral of $C_A(i)$, and becomes noisy at long times, especially at more than half of the simulation time. To obtain reliable estimate of τ, the simulation time should, as a general rule of thumb, exceed 50τ. Once the correlation time is found, a set of independent samples is built up by picking every gth (where $g = 1+2\tau$) sample out of the original set. A detailed derivation can be found elsewhere [8].
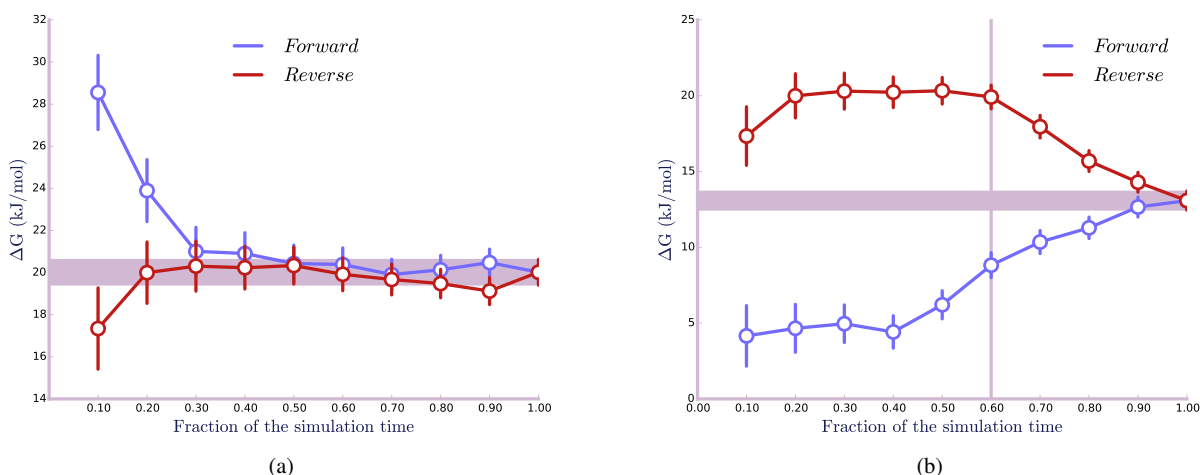
3 Analysis outputs and visualization

3.1 Time-reversed convergence plots reveal non-equilibrated regions

The trajectory snapshots to be analyzed must be sampled at equilibrium. To get rid of any non-equilibrated region of the trajectory its location should first be identified. Automatic identification of non-equilibrated regions remains a major research challenge, and in fact equilibration may have occurred prior to start of data collection depending on the equilibration protocol. So assessment of equilibration remains a major task for the researcher conducting free energy simulations, but automated tools can provide at least some help. A standard approach is to look at convergence of the free energy estimate as a function of simulation time, as in Fig. 5.

While convergence analysis is normally applied on data in the order in which it was collected, the same analysis can also be applied to time-reversed data, as in the work of Yang *et al.* [41]. Yang *et al.* focused on automatic detection of equilibration based on reverse cumulative averaging starting with the end point of the simulation. Our approach is slightly different, and we simply compare forward and reverse free energy estimates. That is, we might compare an estimate of the free energy change based on the first 10% of the data with an estimate based on the last 10% of the data. This results in two ΔG estimates for each observation time, one using data collected starting from the beginning

---

[3] Our Python tool does not currently separate out a restraining component of the free energy, because restraining transformations are not always separable from other transformations. Unlike Coulombic transformations, most of the other transformation types can be (and are [25, 34]) performed simultaneously (to decrease the number of the simulation runs), i.e. they are coupled, which makes component separation impossible.

Fig. 5 Free energy convergence with time. (a) Computed free energy differences (with error bars) are shown as a function of time to help assess equilibration and convergence. Here, we show free energy estimates resulting both from the normal ("forward") time series, and what we would obtain if the data were analyzed in a time-reversed manner ("reverse") using the same amount of data, as discussed in the text. If all of the data is collected at equilibrium from the same distribution, the reverse free energy estimates ought to agree within error (at least after they are converged) which is the case for the figure depicted on the left panel after time = 0.3. (b) Here, Gaussian noise is added to the first 40% of the data to mimic including non-equilibrated samples. Now, the reverse set of the free energy estimates shows an initial steady plateau (indicating data which is consistent with time) but then at later times, the estimate begins to differ from this initial estimate and converges to the wrong ΔG value. The vertical purple line indicates the boundary between the equilibrated and non-equilibrated regions.

of the trajectory and moving forwards, and the other using data collected starting from the end and moving backwards (Fig. 5).

Fig. 5 and 6 show convergence of free energy estimates obtained both from normal and time-reversed data. We include the forward estimate as this is the conventional way to examine the data and assess convergence, but we actually find the estimate with time-reversed data more helpful. In a favorable case, both sets of data should converge rapidly to within uncertainty of the final value (Fig. 5(a)). However, consider a hypothetical case where the free energy calculations were begun before the system is at equilibrium, and the first 40% of the data is in fact non-equilibrium. What would we expect in this case?

In the case where a substantial amount of data at the beginning of a set of simulations is not equilibrated, it is reasonable to assume that any free energy estimate based on this data would differ substantially from an estimate based on data from the equilibrated region. Thus as we examine free energy estimates from both normal and time-reversed data, both will reach the same *final* value but from opposite directions. We would expect that the time-reversed estimate will be steady (within uncertainty) around some value and then leave this value to reach a different final ΔG value as un-equilibrated data begins being included in the free energy estimate, while the forward estimate will exhibit some overall trend as the un-equilibrated data from the beginning of the simulation starts to carry less and less weight. This also means that forward and reverse ΔG estimates will tend to

approach the final value from opposite directions - i.e. if the forward estimate ascends to the final value, the reverse estimate will descend to it. Thus the two free energy estimates have at least partial reflection symmetry around the line $\Delta G = \Delta G_{final}$.

To illustrate this, we created a hypothetical case of un-equilibrated data by taking our well-converged simulations of 3-methylindole and adding Gaussian noise to the first 40% of the time series at all lambda values. Fig. 5(b) shows this case where the $\Delta U_{i,j}$ distributions of the first 40% of the time series have been contaminated with the Gaussian noise centered at the original mean of each distribution but with a standard deviation of 5 $k_B T$ units. This is a deliberately artificial example, but it provides a clear demonstration of how this analysis can be useful. This "un-equilibrated" data at the beginning of of the timeseries changes the behavior of the $\Delta G_{reverse}(t)$ function in the expected way – it remains stable essentially within uncertainty of a constant value (20 kcal/mol) over a plateau region extending to around $t = 0.6 t_{total}$ (i.e., until the reverse estimate begins to include the first 40% of the data). After this point, the non-equilibrated data adversely affects the reverse ΔG estimates and *both* the normal and time-reversed free energy estimates converge to the wrong ΔG value (the correct value is seen in Fig. 5(a)). This graph then suggests a simple solution: We can recover the correct free energy estimate if we simply recognize the region of non-equilibrated data and discard it. Fig. 6 shows the convergence plot obtained via this procedure, with the first 40% of the data removed from the

analysis. As expected, the elimination of the non-equilibrium region from the analysis restores the well-behaved character of the $\Delta G_{reverse}(t)$ function, with all data points in both directions essentially lying within uncertainty of the final value.

Our discussion here has assumed that all of the data, after equilibration, is collected at equilibrium from the same distribution so that the forward and reverse free energy estimates agree within error after convergence. But what if there samples are instead collected from multiple metastable states, such as if the system undergoes a conformational transition? One can imagine a situation where the first and last halves of the data were sampled in two different metastable states. This could result in a convergence plot which displays the same behaviors as in the case of unequilibrated data. Thus, our analysis in this subsection refers to the case of adequate sampling, where all relevant metastable states are covered and there is sufficient number of transitions between them within each time block considered in the convergence plot. When this is the case, the deviation of the free energy estimates is entirely due to the presence of unequilibrated data. However, if the populations of different states differ drastically from block to block (i.e. if the transition time between metastable states is a reasonable fraction of the total simulation time) the technique described here will not discriminate between discrepancies caused by unequilibrated data and those caused by an unconverged free energy estimate. In other words, in the limit of adequate sampling, this technique can help to identify unequilibrated data. However, when sampling is *not* adequate, it will simply indicate a problem which can be due to slow convergence/inadequate sampling of metastable states (slow transitions between states), or to unequilibrated data.

3.2 Practical recommendations for using the convergence plot

In our view, then, the convergence plot is of great value for detecting potentially un-equilibrated data, provided the data to be analyzed was adequately sampled, i.e. with an adequate number of transitions between metastable states and reasonably correct populations in all time blocks. Whenever the time-reversed free energy estimate shows an extended plateau, then after some time begins to exhibit an overall trend in time (as in Fig. 5(b)), it suggests that the new data being included which leads to that trend may be nonequilibrated. Thus this should be considered a substantial warning sign, and data collected prior this point should perhaps be discarded. [4] In contrast, if both forward and reverse

---

[4] Remember, the time-reversed ΔG estimates are plotted in a backward manner, so that if the point in question is encountered at the time $t' = 0.6t_{total}$, the portion of the data to be discarded as non-equilibrated is from $t = 0$ up to $t = t_{total} - t' = 0.4t_{total}$.
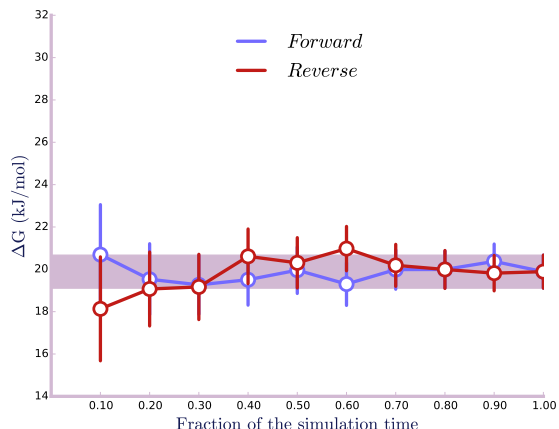
Fig. 6 Free energy convergence with time with the first 0.4 of the time series data skipped as non-equilibrated. Discarding the corrupted data restores the agreement between the free energy estimates within the reverse data set.

estimates quickly approach the same value as in Fig. 5(a), and there is no plateau in the time-reversed data from which the free energy estimate departs as more data is included, then this measure suggests no concern.

3.3 The overlapping distribution method identifies regions with poor phase-space overlap

In addition to looking for consistency between free energy estimates from several methods discussed in Section 2.3, the Overlapping Distribution Method (ODM) [11, 32]—introduced by Bennett [1] under the name of the Curve-Fitting Method— provides another useful technique for spotting trouble. It is a helpful tool for assessing consistency, when combined with another free energy estimator.

We start with the equation 19 from the Bennett paper [1]:

$$\frac{P_{i+1}(\Delta U)}{P_i(\Delta U)} = \exp(\Delta G_{i,i+1} - \Delta U) \tag{7}$$

where $P_{i+1}(\Delta U)$ and $P_i(\Delta U)$ are the distributions of the potential energy differences between adjacent states obtained when sampling at state $\lambda = i + 1$ and $\lambda = i$, respectively. The free energy change between the states, $\Delta G_{i,i+1}$, and corresponding potential energy difference, ΔU, are written in reduced units.

This equation can be rewritten as

$$g_{i+1}(\Delta U) - g_i(\Delta U) = \Delta G_{i,i+1} \tag{8}$$

where the new distribution functions g_{i+1} and g_i were obtained by taking natural logarithm of both sides of Eq. 8 and splitting the ΔU term

$$g_{i+1}(\Delta U) = \ln P_{i+1}(\Delta U) + C\Delta U \tag{9}$$
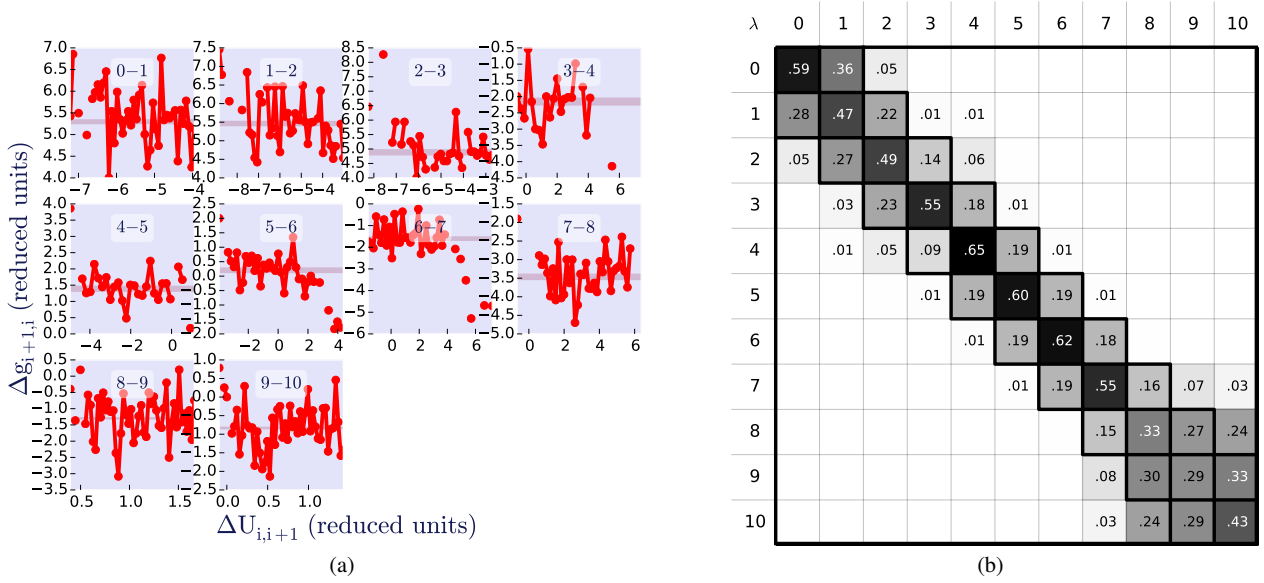
Fig. 7 The overlapping distribution method and the overlap matrix. (a) When the difference $\Delta g_{i,i+1} = g_{i+1} - g_i$ of Eq. 8 is plotted as a function of $\Delta U_{i,i+1}$, the resulting graph can be used as a consistency inspector. Here, the function should oscillate about the horizontal purple strip (represents the BAR free energy change estimate; the width is dictated by the estimate error) over substantial range of abscissa values proving a good overlap between the ΔU distributions of two adjacent states. The overlap may not be continuous due to the discontinuity of either probability distribution function resulting in the lack of connection between the $\Delta g_{i,i+1}$ points. (b) Overlap between the distributions of potential energy differences is essential for accurate free energy calculations and can be quantified by computing the overlap matrix $\mathbf{O}$ discussed in the text and visualized here. Its elements O_{ij} are the probabilities of observing a sample from state i (ith row) in state j (jth column). According to the figure, the probability of observing, for example, a sample collected from state 2 having come from a simulation state 3 is 0.14. As discussed in the text, for the terminal states (here states 0 and 10) to be interconnected, the overlap matrix should be at least tridiagonal.
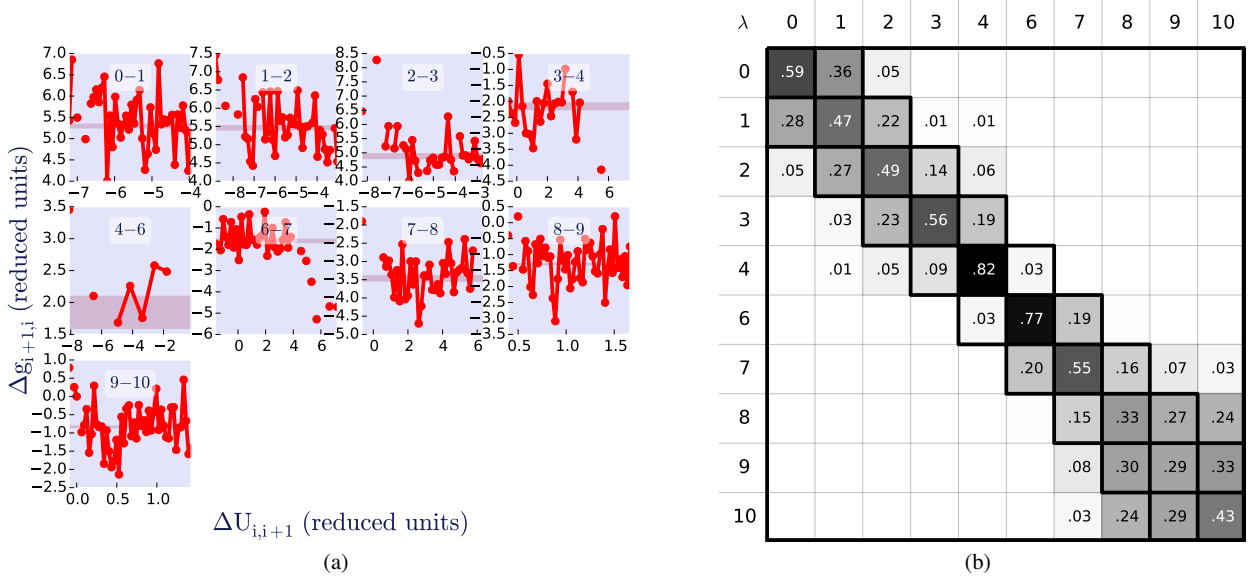
Fig. 8 The overlapping distribution method and the overlap matrix for the case when the lambda state 5 is left out from the analysis. (a) Excluding lambda state 5 from the analysis results in the deterioration of the phase space overlap between states 4 and 6 which are now neighboring states. The $\Delta g_{i,i+1}$ function is only partially continuous and does not exhibit oscillations or saw-like behavior over a substantial interval of the $\Delta U_{i,i+1}$ values as it does in Fig. 7(a). (b) Overlap matrix elements associated with the lambda states in question are significantly smaller than any other elements. The probability of finding a microstate sampled from state 4 in state 6 is 0.03; so is the probability of finding a microstate sampled from state 6 in state 4. This magnitude for the phase space overlap seems to be still tolerable in this particular case, as it does not affect the free energy estimate other than by increasing the estimated uncertainty.

$$g_i(\Delta U) = \ln P_i(\Delta U) - (1 - C)\Delta U \qquad (10)$$

where C is an arbitrary constant.

If the difference $\Delta g_{i,i+1} = g_{i+1} - g_i$ is plotted versus ΔU, there should be a range of the ΔU values over which $\Delta g_{i,i+1}$ oscillates about the free energy estimate obtained by a standard technique such as BAR, provided there is an overlap between the two distributions and the sampling was sufficient.

This analysis, shown in Fig. 7(a), means that we can graphically inspect the difference $\Delta g_{i,i+1}$ over a range of the ΔU values and it should appear relatively constant. If not, it is a substantial warning sign. All the subplots of Fig. 7(a) exhibit good behavior: the $\Delta g_{i,i+1}$ function oscillates about the ΔG_{BAR} estimate (shown in purple, with a width corresponding to the uncertainty in ΔG_{BAR}). In contrast, the $\Delta g_{i,i+1}$ function depicted on the subplot 4-6 of Fig. 8(a) behaves abnormally. Figure 8 was obtained by processing exactly the same data but with the lambda state 5 left out from the analysis. This resulted in the deterioration of the distribution overlap between states 4 and 6, which the ODM helps highlight: the $\Delta g_{i,i+1}$ function is not continuous and does not exhibit saw-like behavior over a substantial interval of the $\Delta U_{i,i-1}$ values as it does in Fig. 7(a). This tells us that the estimated free energy change for the 4-6 pair of states is getting less reliable, which is a problem.

To further highlight the problem, we exacerbate the overlap by throwing away yet another intermediate state, lambda state 6, from the analysis and re-examine the $\Delta g_{i,i+1}$ function (Fig. 10). This time, there is only one point on the graph for subplot 4-6, and the point is far from the unusually wide (because of the greater uncertainty) ΔG_{BAR} estimate, indicating clear and severe problems with overlap in this case.

In general, low overlap does not necessarily make the ΔG estimate completely wrong or substantially untrustworthy. However, it does substantially increase the corresponding variance, as only few samples contribute to the free energy estimate. Users themselves must decide what level of uncertainty in the free energy estimate can be tolerated, and whether the poor efficiency identified by this method might suggest restructuring the spacing or number of intermediate states before collecting additional data.

Figure 9 shows the free energies obtained from the analysis with different states omitted. The free energy change estimated between states 4 and 7 through the intermediate states 5 and 6 (-0.028 ± 0.332 kJ/mol) and 6 only (0.601 ± 0.651 kJ/mol) are within statistical noise. However, the 4-7 ΔG estimate without any intermediate states yields 3.986 ± 2.512 kJ/mol which is almost two standard deviations away from the original value of -0.028 ± 0.332 kJ/mol. On the case of extremely poor overlap (as in Fig. 10), the FEP-based methods tend to underestimate the variance, making free energy estimates untrustworthy.

Despite its ability to identify lambda regions with poor and good overlap, the overlapping distribution remains a qualitative method. If one needs to translate "poor overlap" and "good overlap" into concrete numbers, the overlap matrix, discussed in the next subsection, should be employed.

3.4 The overlap matrix is a quantitative estimator of the phase-space overlap

The overlap matrix is a helpful tool in finding the magnitude of the phase space overlap and we recommend using it as a consistency check whenever the free energy estimate relies on the overlap, as in the case of the FEP-based methods.

If we define the weight of each of the N samples x_n (collected from all K states) in the ith lambda state as:

$$W_{n,i}(x_n) = \frac{e^{\beta G_i - \beta U_i(x_n)}}{\sum_{k=1}^{K} N_k e^{\beta G_k - \beta U_k(x_n)}} \qquad (11)$$

then the covariance matrix $\mathbf{\Theta}$ of the vector of reduced free energies βG can be written (From Eq. D6 of Ref. [35]) as

$$\mathbf{\Theta} = \left(\mathbf{O^{-1}} - \mathbf{I}\right)^+ \mathbf{N^{-1}} \qquad (12)$$

where $^+$ is the Moore-Penrose pseudoinverse, $\mathbf{N}$ is a diagonal matrix with its elements N_{ii} the number of samples collected in state i, and $\mathbf{O}$ is the overlap matrix, defined as:

$$\mathbf{O} = \mathbf{W^T W N} \qquad (13)$$

We note that $e^{\beta G_i - \beta U_i(x_n)}$ in eq (11) is the probability $p_i(x_n)$ of sample x_n occuring when simulating state i. The unnormalized probability of sample x_n is the Boltzmann weight $e^{-\beta U_i(x_n)}$, and since $e^{-\beta G_i} = \int e^{-\beta U_i(\mathbf{x})} d\mathbf{x}$, $e^{-\beta G_i}$ is the normalizing constant.

The overlap matrix is a $K \times K$ matrix with entries:

$$O_{ij} = \sum_{n=1}^{N} \frac{N_i e^{\beta G_i - \beta U_i(x_n)}}{\sum_{k=1}^{K} N_k e^{\beta G_k - \beta U_k(x_n)}} \frac{e^{\beta G_j - \beta U_j(x_n)}}{\sum_{l=1}^{L} N_l e^{\beta G_l - \beta U_l(x_n)}}$$
$$= \sum_{n=1}^{N} \frac{N_i p_i(x_n)}{\sum_{k=1}^{K} N_k p_k(x_n)} \frac{p_j(x_n)}{\sum_{l=1}^{L} N_l p_l(x_n)} \qquad (14)$$

The N samples were collected with N_1 samples from the $p_1(x)$ distribution, N_2 samples from the $p_2(x)$ distribution, and so forth. This combination of the K distributions is known as a mixture distribution, and can be written mathematically as:
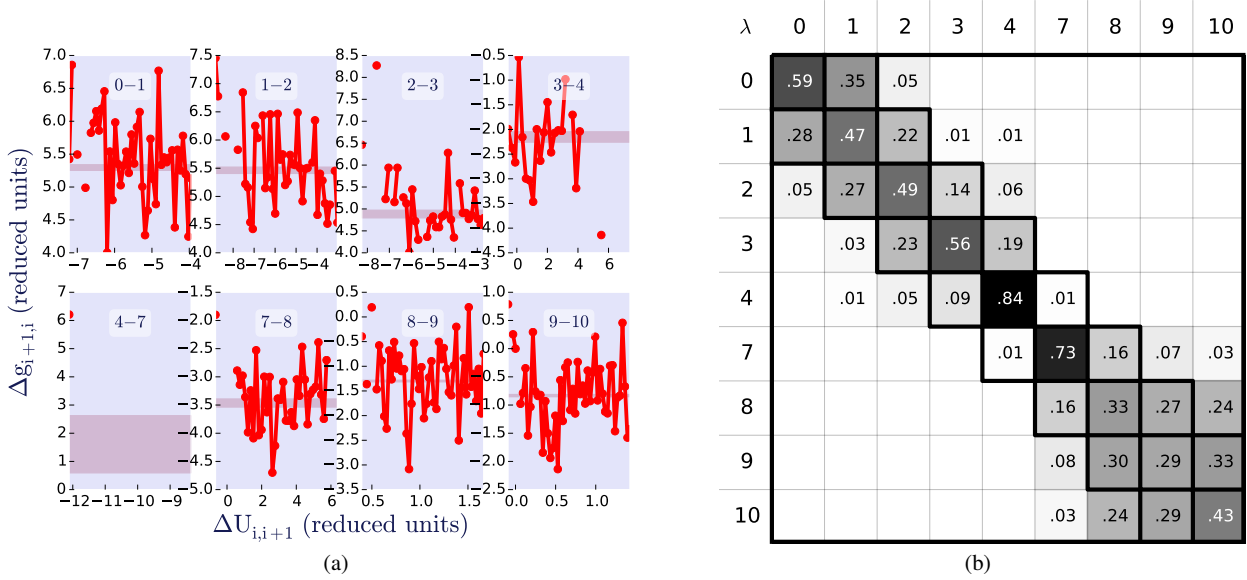
$$p_{\text{mix}}(x) = \sum_{k=1}^{K} \frac{N_k}{N_{tot}} p_k(x) = \frac{1}{N_{tot}} \sum_{l=1}^{K} N_k p_k(x_n)$$

where $N_{tot} = \sum_{k=1}^{K} N_k$.

We next note that the Monte Carlo estimate of the integral $\int A(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$, where A is some function of $\mathbf{x}$, is $\frac{1}{N} \sum_{n=1}^{N} A(x_n)$, where the samples x_n are drawn from the

$\Delta G_{4-5} = 3.439 \pm 0.187$
$\Delta G_{5-6} = 0.504 \pm 0.193 \qquad \Delta G_{4-6} = 4.572 \pm 0.621$
$\Delta G_{6-7} = -3.971 \pm 0.196 \qquad \Delta G_{6-7} = -3.971 \pm 0.196 \qquad \Delta G_{4-7} = 3.986 \pm 2.512$

$\Delta G_{4-7} = -0.028 \pm 0.332 \qquad \Delta G_{4-7} = 0.601 \pm 0.651 \qquad \Delta G_{4-7} = 3.986 \pm 2.512$

Fig. 9 The free energy estimates (in kJ/mol) obtained from the analysis of all lambda states (left), with the lambda state 5 left out from the analysis (center), and with the lambda states 5 and 6 left out from the analysis (right). While the excluding of state 5 results in the increasing of the uncertainty for the ΔG_{4-7} estimate, the excluding of states 5 and 6 makes the ΔG_{4-7} estimate untrustworthy.

λ	0	1	2	3	4	7	8	9	10
0	.59	.35	.05						
1	.28	.47	.22	.01	.01				
2	.05	.27	.49	.14	.06				
3		.03	.23	.56	.19				
4		.01	.05	.09	.84	.01			
7					.01	.73	.16	.07	.03
8						.16	.33	.27	.24
9						.08	.30	.29	.33
10						.03	.24	.29	.43

(a)   (b)

Fig. 10 The overlapping distribution method and the overlap matrix for the case when two lambda states in a row (5 and 6) are left out from the analysis. (a) Excluding lambda states 5 and 6 from the analysis results in the significant deterioration of the phase space overlap between states 4 and 7 which are now neighboring states. There is not even a hint of oscillation or saw-like behavior of the $\Delta g_{i,i+1}$ function. (b) The matrix elements associated with the lambda states in question are almost negligible. The probability of finding a microstate sampled from state 4 in state 7 is 0.01; so is the probability of finding a microstate sampled from state 7 in state 4. Such a low phase space overlap makes the free energy estimate ΔG_{4-7} untrustworthy

probability distribution $p(x)$. This means that each element of the overlap matrix can be seen as a Monte Carlo estimate of the integral:

$$O_{ij} = \int \frac{N_i p_i(\mathbf{x})}{\sum_{k=1}^{K} N_k p_k(\mathbf{x})} \left(\frac{p_j(\mathbf{x})}{\sum_{l=1}^{L} N_l p_l(\mathbf{x})} \right) \sum_{l=1}^{L} N_l p_l(\mathbf{x}) d\mathbf{x}$$

$$= \int \frac{N_i p_i(\mathbf{x})}{\sum_{k=1}^{K} N_k p_k(\mathbf{x})} p_j(\mathbf{x}) d\mathbf{x}$$

$$= \left\langle \frac{N_i p_i(\mathbf{x})}{\sum_{k=1}^{K} N_k p_k(\mathbf{x})} \right\rangle_j$$

where the averages are over the probability distribution of samples in state j. O_{ij} can therefore be interpreted as the average probability of a sample generated in state j being observed in the ith state. This average is computed over samples collected from all of the states, not just the samples from state j. We can easily see that we must have $\sum_i O_{ij} = 1$.

Because O_{ij} is a stochastic (or Markov) matrix, its eigenvalues are all real and positive, and the largest is 1. In fact, since it is a diagonal matrix ($\mathbf{N}$) times a symmetric matrix ($\mathbf{W}^T \mathbf{W}$), then all the eigenvalues are real and positive.

We can then write (using the standard notation for the eigenvalue decomposition):

$$\boldsymbol{\Theta} = \left(\mathbf{O^{-1}} - \mathbf{I} \right)^+ \mathbf{N^{-1}}$$

$$= \left((\mathbf{Q \Lambda Q^{-1}})^{-1} - \mathbf{I} \right)^+ \mathbf{N^{-1}}$$

$$= \left(\mathbf{Q \Lambda^{-1} Q^{-1}} - \mathbf{Q Q^{-1}} \right)^+ \mathbf{N^{-1}}$$

$$= \left(\mathbf{Q}(\mathbf{\Lambda^{-1}} - \mathbf{I}) \mathbf{Q^{-1}} \right)^+ \mathbf{N^{-1}}$$

$$= \mathbf{Q}(\mathbf{\Lambda^{-1}} - \mathbf{I})^+ \mathbf{Q^{-1} N^{-1}}$$

In this case, because it is a diagonal matrix, we can give a simple formula for the pseudoinverse; $(\mathbf{\Lambda^{-1}} - \mathbf{I})^+$ is a diagonal matrix with one zero diagonal entry (corresponding to the largest eigenvalue, which is 1), and the other entries corresponding to the ith eigenvalue λ_i being $\lambda_i/(1 - \lambda_i)$.

The uncertainty in any free energy difference between states i and j is $\Theta_{ij} = \Theta_{ii} + \Theta_{jj} - 2\Theta_{ij}$ which makes it difficult to write explicit formulas of the variance in terms of the overlap matrix. In general, uncertainties in free energy differences will be larger when the eigenvalues $\lambda_i/(1 - \lambda_i)$ are large, which will occur when the eigenvalues are close to 1. Because of the factor of $\mathbf{N}^{-1}$, any variance can be made arbitrarily low with enough samples. Clearly, the most efficient choices of λ points to simulate will be ones leading to smaller eigenvalues in the overlap matrix.

When will these eigenvalues be close to one? Again, it is difficult to completely generalize, but for stochastic matrices, i.e. matrices with the form of $\mathbf{O}$ with rows summing to 1, the smallest $K - 1$ eigenvalues approach one when the matrices can be nearly decomposed into independent block matrices. The absence of eigenvalues close to one indicates that the matrix is more connected.

3.5 Practical recommendations for using the overlap matrix

We cross-checked the overlap matrix with the overlapping distribution method (Fig. 7, 8, and 10) and arrived at several recommendations for trustworthy free energy calculations. First, trustworthy results should in general have at least a tridiagonal overlap matrix - that is, all pairs of adjacent states should have substantial overlap, and no element should be zero in the main diagonal and the first diagonals above and below the main one. Second, these nonzero elements should be appreciably different from zero. In our experience, with enough samples, values as low as 0.03 (Fig. 8) seems to be tolerable to yield a reliable free energy estimate (as long as the resulting error is sufficiently low), though obviously the number of samples required for an accurate estimate increases with decreasing overlap. Anything below that number should serve as a caution sign as the FEP-based methods tend to underestimate the variance when the phase space overlap is that low; in this case, not only will the estimated variance increase but the free energy estimate itself will likely be substantially incorrect, perhaps by far more than the estimated variance.

4 Alchemical-analysis.py: A sample analysis tool implementing these protocols

We provide a Python tool, alchemical-analysis.py, which implements our recommendations and generates all the plots described above. This tool is versatile in handling several energy file formats from different versions of GROMACS varying in the number of the potential energy differences between the states $\Delta U_{k,j}$, as well as SIRE and AMBER output files. It handles cases when all the potential energy differences are present as well as those with only differences between the adjacent states $\Delta U_{k,k\pm1}$, though in the latter scenario MBAR free energy estimates cannot be computed.

The data file parser is separated from the analysis proper into a subroutine which makes the tool indifferent to the origin of the data, as long as it contains the quantities the free energy estimators rely on, i.e. $\partial U/\partial \lambda$ and/or $\Delta U_{i,j}$. Thus, our tool can easily be adapted to handle data from other simulation packages. As of now, the tool has file parser subroutines for and can analyze, apart from GROMACS `dhdl.xvg` files, the $\partial U/\partial \lambda$ data files generated by SIRE and AMBER.

At minimum, alchemical-analysis.py plots and outputs the free energy differences evaluated for each pair of adjacent states for all methods. The plot (Fig. 4) provides a way of visualizing the results and assisting in locating any λ regions where the free energy changes rapidly. At minimum, two figures are produced, one depicting the bar plot showing the ΔG estimates (differentiated in colors) for all methods (Figure 4(a)), and the other showing free energy differences estimated with the TI methods depicted as an area under the interpolating curve joining the $\langle \frac{\partial U}{\partial \lambda} \rangle_\lambda$ points (Figure 4(b)). All the plots are created by the *matplotlib* software [15], which is a standard plotting package delivered alongside Python proper in scientific Python distributions like, for example, Enthought Canopy (http://enthought.com) and Anaconda (https://store.continuum.io/cshop/anaconda).

4.1 Tool execution and usage

Our tool is executed by the command

`python alchemical-analysis.py [options]`

with the list of options provided below. In general, plots and options are as described above, except when they relate to GROMACS in particular, in which case additional information is provided below.

Options include:

– Simulation temperature
– Directory with the input data
– Datafile prefix and suffix
– Time prior to which the data is to be discarded
– Names of the free energy estimators to be used
– The units the free energies are to be reported in
– Number of decimal places the free energies are to be reported with
– Graphical functionality (discussed above):
    – The $\Delta G_{i,i+1}$ vs. $\lambda_{i,i+1}$ bar plot as in Fig. 4(a)
    – The $\langle \frac{\partial U}{\partial \lambda} \rangle$ vs. λ plot as in Fig. 4(b)
    – The ΔG vs. time plot, as in Fig. 5
    – The overlap matrix calculation, as in Fig. 7
    – The consistency check based on the overlap distribution method, as in Fig. 10

The script outputs a text file, `results.txt`, with a table of free energy differences computed by means of various

methods for each pair of adjacent states, as well as over-all totals. This file also shows the ΔG breakdown into its components discussed in Section 2.4. Full precision data is dumped to a Python pickle file, `results.pickle`, where it is stored in a form a class with multiple instances whose names are self-explanatory.

5 Conclusion

Free energy calculations are still complicated and largely conducted only by experts, in part because even their analysis typically requires substantial expertise. Here we have reviewed a variety of best practices for analysis, and provide a prototype Python tool implementing these best practices.

Here, we highlight ten free energy analysis methods (two based on TI and eight more based on FEP, including MBAR). Running the full suite of analysis techniques provides a valuable consistency check and can help highlight convergence errors, sampling problems, and other issues. We also presented a number of useful ways to graphically evaluate free energy data (and provided examples generated by alchemical-analysis.py), including:

1. a bar plot of the free energy differences evaluated for each pair of adjacent λ states
2. thermodynamic integration as a plot of $\left\langle \frac{\partial U}{\partial \lambda} \right\rangle$ vs. λ
3. free energy estimates as a function of the simulation time in both the forward and reverse directions
4. the overlapping distribution method and the phase space overlap matrix

We believe analysis of calculations in the way highlighted here will provide researchers with a better assessment of the precision and convergence of their calculations, and aid non-experts in getting a better handle on how to successfully understand their results and troubleshoot problems.

6 Acknowledgments

7 Supporting Information

In the Supporting Information, we provide the analysis tool used to generate figures shown here, and .xvg files used in generating the plots/tables presented here.

References

1. Bennett, C.H.: Efficient estimation of free energy differences from Monte Carlo data. J. Comput. Phys. **22**(2), 245–268 (1976)
2. Beutler, T.C., Mark, A.E., van Schaik, R.C., Gerber, P.R., van Gunsteren, W.F.: Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. Chem. Phys. Lett. **222**(6), 529–539 (1994)
3. Beveridge, D.L., DiCapua, F.M.: Free energy via molecular simulation: applications to chemical and biomolecular systems. Annu Rev Biophys Biophys Chem **18**, 431–492 (1989)
4. Boyce, S.E., Mobley, D.L., Rocklin, G.J., Graves, A.P., Dill, K.A., Shoichet, B.K.: Predicting Ligand Binding Affinity with Alchemical Free Energy Methods in a Polar Model Binding Site. J. Mol. Biol. **394**(4), 747–763 (2009)
5. Buelens, F.P., Grubmüller, H.: Linear-scaling soft-core scheme for alchemical free energy calculations. J Comput Chem **33**(1), 25–33 (2012)
6. Case, D.A., Babin, V., Berryman, J., Betz, R.M., Cai, Q., Cerutti, D.S., Cheatham III, T.E., Darden, T.A., Duke, R.E., Gohlke, H., Goetz, A.W., Gusarov, S., Homeyer, N., Janowski, P., Kaus, J., Kolossváry, I., Kovalenko, A., Lee, T.S., LeGrand, S., Luchko, T., Luo, R., Madej, B., Merz, K.M., Paesani, F., Roe, D.R., Roitberg, A., Sagui, C., Salomon-Ferrer, R., Seabra, G., Simmerling, C.L., Smith, W., Swails, J., Walker, R.C., Wang, J., Wolf, R.M., Wu, X., Kollman, P.A.: Amber 14 (2014)
7. Chipot, C.: Frontiers in free-energy calculations of biological systems. Wiley Interdisciplinary Reviews-Computational Molecular Science **4**(1), 71–89 (2014)
8. Chodera, J.D., Swope, W.C., Pitera, J.W., Seok, C., Dill, K.A.: Use of the Weighted Histogram Analysis Method for the Analysis of Simulated and Parallel Tempering Simulations. J. Chem. Theory Comput. **3**(1), 26–41 (2007)
9. Deng, Y., Roux, B.: Computations of Standard Binding Free Energies with Molecular Dynamics Simulations. J. Phys. Chem. B **113**(8), 2234–2246 (2009)
10. Flyvbjerg, H., Petersen, H.G.: Error estimates on averages of correlated data. J Chem Phys **91**(1), 461 (1989)
11. Frenkel, D., Smit, B.: Understanding Molecular Simulation. From Algorithms to Applications. Academic Press (2001)
12. Gapsys, V., Seeliger, D., de Groot, B.L.: New Soft-Core Potential Function for Molecular Dynamics Based Alchemical Free Energy Calculations. J. Chem. Theory Comput. **8**(7), 2373–2382 (2012)

13. Hummer, G., Pratt, L.R., García, A.E.: Multistate Gaussian Model for Electrostatic Solvation Free Energies. J. Am. Chem. Soc. **119**(36), 8523–8527 (1997)

14. Hummer, G., Pratt, L.R., García, A.E., Berne, B.J., Rick, S.W.: Electrostatic Potentials and Free Energies of Solvation of Polar and Charged Molecules. J. Phys. Chem. B **101**(16), 3017–3020 (1997)

15. Hunter, J.D.: Matplotlib: A 2D graphics environment. Computing in Science & Engineering **9**(3), 90–95 (2007)

16. Kenneth M Merz, J., Ringe, D., Reynolds, C.H.: Drug Design. Structure- and Ligand-Based Approaches. Cambridge University Press (2010)

17. Kirkwood, J.G.: Statistical mechanics of fluid mixtures. J Chem Phys **3**(5), 300–313 (1935)

18. Klimovich, P.V., Mobley, D.L.: Predicting hydration free energies using all-atom molecular dynamics simulations and multiple starting conformations. J. Comput. Aided Mol. Des. **24**(4), 307–316 (2010)

19. Kumar, S., Bouzida, D., Swendsen, R.H., Kollman, P.A., Rosenberg, J.M.: The Weighted Histogram Analysis Method for Free-Energy Calculations on Biomolecules .1. the Method. J Comput Chem **13**(8), 1011–1021 (1992)

20. Lu, H., Schulten, K.: Steered molecular dynamics simulations of force-induced protein domain unfolding. Proteins **35**(4), 453–463 (1999)

21. Lyubartsev, A.P., Laaksonen, A., Vorontsov-Velyaminov, P.N.: Free energy calculations for Lennard-Jones systems and water using the expanded ensemble method A Monte Carlo and molecular dynamics simulation study. Mol Phys **82**(3), 455–471 (1994)

22. Lyubartsev, A.P., Martsinovski, A.A., Shevkunov, S.V., Vorontsov-Velyaminov, P.N.: New approach to Monte Carlo calculation of the free energy: Method of expanded ensembles. J Chem Phys **96**(3), 1776–1783 (1992)

23. Marszalek, P.E., Lu, H., Li, H., Carrion-Vazquez, M., Oberhauser, A.F., Schulten, K., Fernandez, J.M.: Mechanical unfolding intermediates in titin modules. Nature **402**(6757), 100–103 (1999)

24. Martínez-Veracoechea, F.J., Escobedo, F.A.: Variance minimization of free energy estimates from optimized expanded ensembles. J. Phys. Chem. B **112**(27), 8120–8128 (2008)

25. Mobley, D.L., Bayly, C.I., Cooper, M.D., Shirts, M.R., Dill, K.A.: Small Molecule Hydration Free Energies in Explicit Solvent: An Extensive Test of Fixed-Charge Atomistic Simulations. J. Chem. Theory Comput. **5**(2), 350–358 (2009)

26. Monroe, J.I., Shirts, M.R.: Converging free energies of binding in cucurbit[7]uril and octa-acid host–guest systems from SAMPL4 using expanded ensemble simulations. J. Comput. Aided Mol. Des. pp. 1–15 (2014)

27. Naden, L.N., Shirts, M.R.: A Linear Basis Function Approach to Efficient Alchemical Free Energy Calculations. 2. Inserting and Deleting Charged Molecules. Submitted

28. Paliwal, H., Shirts, M.R.: A Benchmark Test Set for Alchemical Free Energy Transformations and Its Use to Quantify Error in Common Free Energy Methods. J. Chem. Theory Comput. **7**(12), 4115–4134 (2011)

29. Paluch, A.S., Mobley, D.L., Maginn, E.J.: Small Molecule Solvation Free Energy: Enhanced Conformational Sampling Using Expanded Ensemble Molecular Dynamics Simulation. J. Chem. Theory Comput. **7**(9), 2910–2918 (2011)

30. Pham, T.T., Shirts, M.R.: Identifying low variance pathways for free energy calculations of molecular transformations in solution phase. J Chem Phys **135**(3), 034,114 (2011)

31. Pitera, J.W., van Gunsteren, W.F.: A Comparison of Non-Bonded Scaling Approaches for Free Energy Calculations. Molecular Simulation **28**(1-2), 45–65 (2002)

32. Pohorille, A., Jarzynski, C., Chipot, C.: Good Practices in Free-Energy Calculations. J. Phys. Chem. B **114**(32), 10,235–10,253 (2010)

33. Pronk, S., Pall, S., Schulz, R., Larsson, P., Bjelkmar, P., Apostolov, R., Shirts, M.R., Smith, J.C., Kasson, P.M., van der Spoel, D., Hess, B., Lindahl, E.: GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics **29**(7), 845–854 (2013)

34. de Ruiter, A., Boresch, S., Oostenbrink, C.: Comparison of thermodynamic integration and Bennett acceptance ratio for calculating relative protein-ligand binding free energies. J Comput Chem **34**(12), 1024–1034 (2013)

35. Shirts, M.R., Chodera, J.D.: Statistically optimal analysis of samples from multiple equilibrium states. J Chem Phys **129**(12), 124,105 (2008)

36. Shirts, M.R., Pitera, J.W., Swope, W.C., Pande, V.S.: Extremely precise free energy calculations of amino acid side chain analogs: Comparison of common molecular mechanics force fields for proteins. J Chem Phys **119**(11), 5740–5761 (2003)

37. Shivakumar, D., Williams, J., Wu, Y., Damm, W.: Prediction of absolute solvation free energies using molecular dynamics free energy perturbation and the OPLS force field. J. Chem. Theory Comput. **6**(5), 1509–1519 (2010)

38. Sotomayor, M., Corey, D.P., Schulten, K.: In search of the hair-cell gating spring elastic properties of ankyrin and cadherin repeats. Structure **13**(4), 669–682 (2005)

39. Steinbrecher, T., Mobley, D.L., Case, D.A.: Nonlinear scaling schemes for Lennard-Jones interactions in free

energy calculations. J Chem Phys **127**(21), 214,108 (2007)

40. Straatsma, T.P., Berendsen, H.J.C., Postma, J.P.M.: Free energy of hydrophobic hydration: A molecular dynamics study of noble gases in water. J Chem Phys **85**(11), 6720–6727 (1986)

41. Yang, W., Bitetti-Putzer, R., Karplus, M.: Free energy simulations: use of reverse cumulative averaging to determine the equilibrated region and the time required for convergence. J Chem Phys **120**(6), 2618–2628 (2004)

42. Zacharias, M., Straatsma, T.P., McCammon, J.A.: Separation-shifted scaling, a new scaling method for Lennard-Jones interactions in thermodynamic integration. J Chem Phys **100**(12), 9025 (1994)

43. Zhao, G., Perilla, J.R., Yufenyuy, E.L., Meng, X., Chen, B., Ning, J., Ahn, J., Gronenborn, A.M., Schulten, K., Aiken, C., Zhang, P.: Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. Nature **497**(7451), 643–646 (2013)

44. Zwanzig, R.W.: High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases. J Chem Phys **22**(8), 1420–1426 (1954)