Chapter 11

Absolute Alchemical Free Energy Calculations for Ligand Binding: A Beginner's Guide

Matteo Aldeghi, Joseph P. Bluck, and Philip C. Biggin

Abstract

Many thermodynamic quantities can be extracted from computer simulations that generate an ensemble of microstates according to the principles of statistical mechanics. Among these quantities is the free energy of binding of a small molecule to a macromolecule, such as a protein. Here, we present an introductory overview of a protocol that allows for the estimation of ligand binding free energies via molecular dynamics simulations. While we focus on the binding of organic molecules to proteins, the approach is in principle transferable to any pair of molecules.

Key words Free energy, Computer simulations, Molecular dynamics, Alchemical transitions, Protein–ligand binding, Binding free energy, Binding affinity, Drug design, Molecular modeling

1 Introduction

The accurate prediction of the affinity of a drug for its target protein has long been a central objective of structure-based drug design. As such, many computational approaches that try to calculate or approximate binding free energy have been developed [1, 2]. These range from fast scoring functions [3], to implicit-solvent approaches based on the postprocessing of simulation snapshots [4], to the more rigorous yet computationally expensive free energy methods [2]. In particular, alchemical free energy calculations based on all-atom molecular dynamic (MD) simulations in explicit solvent are one of the approaches that operate at the highest level of theoretical rigor, calculating free energy differences from well-founded statistical mechanics principles and naturally including entropic and solvent effects. These calculations are based on a thermodynamic cycle that include a series of nonphysical intermediate states (hence the name *alchemical*), from which the free energy difference between two physical end states can be recovered

Mohini Gore and Umesh B. Jagtap (eds.), *Computational Drug Discovery and Design*, Methods in Molecular Biology, vol. 1762, https://doi.org/10.1007/978-1-4939-7756-7_11,

as the sum of the free energy differences between all pairs of alchemical intermediate states [5]. Here, we will focus in particular on absolute binding free energy (ABFE) calculations that make use of such an alchemical cycle, in which the ligand is nonphysically "removed" from the solution environment and "inserted" into the protein's binding site. Note that the term *absolute* refers to the fact that a binding free energy ΔG_b, rather than a binding free energy difference ($\Delta\Delta G_b$), is being calculated.

The advantage of the alchemical approach as compared to computationally cheaper ones is that it is expected to be more accurate due to its ability to return the exact binding free energy for the physical model used [1, 2, 6]. As will become evident in this chapter, however, this comes at a high computational cost. The method is also in principle general, meaning it can be applied to any protein–ligand pair, as long as the sampling challenges can be addressed. Potential applications of the approach are the ranking of ligands with different scaffolds binding to a specific protein, the accurate rescoring of docking poses, and the prediction of ligand selectivity [7, 8].

Here, we present a guide on how to carry out alchemical ABFE calculations based on a standard protocol that uses equilibrium MD simulations. In particular, we focus on the more practical aspects of setting up the calculations alongside a brief overview of the theory and technical details. Throughout the text, we direct the reader who would like to further explore certain technical topics toward more specialized reviews and chapters. Despite this being a guide that aims to explain how to set up and run the calculations in simple terms, at this point in time these calculations should still be considered an advanced approach. Consequently, we recommend that the reader first familiarize themselves with ligand parametrization and simulations of protein–ligand complexes. Some experience with solvation free energy calculations may also be beneficial, as these represent part of the thermodynamic cycle used for the binding free energy calculations.

The chapter is organized as follows. First, we provide an overview of the theoretical aspects of binding free energy, including its definition, how it can be extracted from computer simulations, and the alchemical thermodynamic cycle we will use to calculate it. Then, we touch upon the software and hardware requirements for the calculations. Finally, we discuss the protocol step by step, pointing out potential pitfalls and issues. In Subheading 5, we show how certain steps can be carried out using the Gromacs 2016 simulations package. This chapter is accompanied by a tutorial (Absolute Binding Free Energy—Gromacs 2016) available on www.alchemistry.org, where the reader can find the input files needed to practice with an example calculation.

2 Theory

Here we review some theoretical and methodological concepts that underlie the use of computer simulations for the calculation of binding free energies. We provide only a brief overview of some key concepts and, due to the practical nature of the chapter, this section is not meant to be exhaustive. The reader can find a more extensive appraisal of theoretical aspects in one of the many excellent reviews and textbooks written on the subject and here referenced [9–13]. The book by Chipot and Pohorille [9] is particularly comprehensive.

2.1 Definition of Binding Free Energy

The reversible binding of a ligand (L) to a protein (P) to form a complex (PL) can be described by the following chemical reaction:

$$P + L \leftrightharpoons PL$$

At equilibrium, the binding constant K°_b defines the ratio of the product and reactants concentrations in solution:

$$K^\circ_b = c^\circ \frac{[PL]}{[L][P]} \tag{1}$$

where square brackets indicate a concentration, and c° is the standard state concentration, which is typically defined as 1 mol/L. Since this definition of standard state does not change the numerical value of K°_b (as it is multiplied by one) it is customary to omit c° when discussing equilibrium constants. However, it is important to remember that the binding constant is a dimensionless quantity (without c° it would have units of inverse concentration), and it is dependent on the chosen standard state. For a single ligand binding event, K°_b is associated to the binding free energy by the following well-known equation.

$$\Delta G^\circ_b = -k_B T \ln K^\circ_b \tag{2}$$

where k_B is the Boltzmann constant, and T is the temperature. For a mole of ligand, the gas constant ($R = N_A k_B$, where N_A is the Avogadro constant) is used instead of k_B. From the above discussion of the standard state, it follows that the binding free energy is also thus defined with respect to the chosen standard state.

The equilibrium constant can also be expressed as the ratio of probabilities ($\mathcal{P}$) for the system being in either the bound or unbound state, so that $K^\circ_b = \mathcal{P}_1/\mathcal{P}_0$, where "1" denotes the bound (PL) state, and "0" the unbound ($P + L$) state. The probability of finding the system in the bound versus unbound state is determined by the ratio of their partition functions Q_1 and Q_0, where the partition function of a system in the isothermal-isobaric (NPT) ensemble is defined as follows.

$$\mathcal{Q}_{\mathrm{NPT}} = \frac{1}{N!h^{3N}} \iiint e^{-\beta[H(\boldsymbol{x},\boldsymbol{p})+pV]} dV d\boldsymbol{x} d\boldsymbol{p} \tag{3}$$

with N being the number of particles, $\beta = 1/(k_{\mathrm{B}}T)$, $\boldsymbol{x}$ and $\boldsymbol{p}$ the coordinates and momenta, H the Hamiltonian, p the pressure, V the volume, and h the Planck's constant. The Hamiltonian of a system consists of a potential energy term $U(\boldsymbol{x})$ that depends on the particles' coordinates, and a kinetic term $K(\boldsymbol{p})$ that depends on their momenta. However, since we are considering the free energy difference of a process in which neither the temperature or particles' masses change, the kinetic contribution to the Hamiltonian is constant. Therefore, only the configurational part of the partition function needs to be considered in this case, and the Gibbs binding free energy can be expressed as follows.

$$\Delta G_{\mathrm{b}}^{\circ} = -k_{\mathrm{B}}T \ln \frac{\int_{V_1}\int_{\Gamma_1} e^{-\beta[U_1(\boldsymbol{x})+pV_1]} dV d\boldsymbol{x}}{\int_{V_0}\int_{\Gamma_0} e^{-\beta[U_0(\boldsymbol{x})+pV_0]} dV d\boldsymbol{x}} \tag{4}$$

where V_1 and V_0 are the container volumes, while Γ_1 and Γ_2 are the *phase space volumes* of the bound and unbound states, respectively. However, due to the limited compressibility of water, at 1 atm the effect of changes in average volume on the binding free energy is negligible [11, 14]. This means that the pV component of the free energy can be ignored without major effects on the results, and that the Helmholtz free energy closely approximates the Gibbs free energy.

$$\Delta G_{\mathrm{b}}^{\circ} \cong \Delta A_{\mathrm{b}}^{\circ} = -k_{\mathrm{B}}T \ln \frac{\int_{\Gamma_1} e^{-\beta U(\boldsymbol{x})} d\boldsymbol{x}}{\int_{\Gamma_0} e^{-\beta U(\boldsymbol{x})} d\boldsymbol{x}} \tag{5}$$

This definition assumes the existence of separate and well-defined "bound" and "unbound" states, which is valid for tight and specific binders, but might not be justified in the case of very weak and nonspecific binders [11]. The partition function for a complex system has no analytical solution and thus simulations need to be used to sample the accessible phase space. The whole phase space is computationally difficult to sample but, when calculating free energy *differences*, inaccessible high-energy regions will quite often not be sampled for either state of interest, resulting in a cancellation of errors that allow ΔG to be calculated.

When comparing binding free energies, it is important to keep in mind that, as mentioned previously, the binding constant depends upon a reference concentration. This dependence is due to the connection between available volume and entropy [15]. It is therefore necessary to refer to the same standard state when comparing binding free energies. A standard concentration $c^{\circ} = 1$ mol/L corresponds to a standard volume $V^{\circ} = 1660$ Å^3 (the volume occupied by one molecule at the concentration of

1 mol/L). As the 1 M standard state is the most widely adopted, it follows that it is simplest to calculate the binding free energy with respect to this standard state. However, during a free energy calculation the protein–ligand system is not simulated at c°, and thus a correction is needed to recover the standard binding free energy. If ΔG_b is the binding free energy calculated using a simulated box of volume V, the standard binding free energy ΔG°_b can be recovered as follows [15]:

$$\Delta G^\circ_b = \Delta G_b - k_B T \ln \frac{V}{V^\circ} \tag{6}$$

where V° is the standard volume of 1660 Å^3, which corresponds to a standard concentration $c^\circ = 1$ mol/L.

2.2 Estimating Free Energy Differences from Equilibrium Simulations

Given the relationship $K^\circ_b = \mathcal{P}_1/\mathcal{P}_0$, one could think of straightforwardly calculating the difference in free energy between two states of a system by counting the number of configurations in both states. That is, for binding free energies, by counting the number of bound versus unbound configurations during a simulation. This approach is however only feasible when sufficient statistics can be collected, i.e., when the system can transition between the two states of interest many times within the timeframe of the simulation. Despite great improvements in simulation performance in the last few decades, this is still not computationally feasible with unbiased simulations due to the timescales involved in the binding/unbinding process. Several other approaches have thus been developed to estimate free energy differences using data that can be collected via molecular dynamics simulations. Here, we focus only on the main approaches used to estimate free energy differences from equilibrium simulations and that are relevant for alchemical pathways. Approaches that estimate free energy from nonequilibrium transitions between end states are also available, and reviews of such methods can be found elsewhere [9, 16].

2.2.1 Perturbation Approaches

One of the most well-known methods to estimate free energy differences is based on perturbation theory, and relies on the following formula introduced by Zwanzig [17].

$$\Delta G_{0,1} = -k_B T \ln \left\langle e^{-\beta(U_1(\boldsymbol{x}) - U_0(\boldsymbol{x}))} \right\rangle_0 \tag{7}$$

The equation shows that the free energy difference can be calculated as the logarithm of the ensemble average of the exponential of the Boltzmann weighted potential energy difference between the two states. From the subscript of the ensemble average it is possible to note how the potential energy difference is evaluated for the same reference ensemble; i.e., equilibrium sampling is carried out for one state, here labeled "0," and the energies are

computed for both thermodynamic states "0" and "1" over the same configurations. The above formula applies to the *forward* transformation $0 \rightarrow 1$; it is also possible to calculate the free energy difference for the *backward* transformation $1 \rightarrow 0$. While the two expressions are equivalent, in practice their convergence properties may not be the same [9, 18], and the difference in the two resulting ΔG values is referred to as *hysteresis*. As mentioned previously, the kinetic part of the Hamiltonian does not contribute toward the free energy as temperature and masses are unvaried, and the pressure-volume contribution is only marginal, so that here we consider only the potential energy contribution to the Hamiltonian for simplicity.

The approach based on Eq. 7 can be referred to as *exponential averaging* (EXP), however, the term *free energy perturbation* (FEP) is often used too. Note that FEP is also at times used to refer to alchemical free energy methods in general, where *perturbation* in this case refers to the perturbation in the chemical identity and interactions of the atoms themselves. Despite the above equation being exact, it has been shown that EXP converges only slowly with the amount of data collected, and an average that appears to have converged may only indicate poor overlap between the two states studied [19, 20].

The free energy obtained via EXP for either the forward or reverse direction converges to the same result in the limit of infinite sampling. A simple way to improve EXP is thus to simply perform the calculation in both directions and average the results. However, because of a direct relationship between the distributions of potential energy differences in the forward and reverse directions, Bennet could derive a more robust and statistically optimal way to use information from both directions [21]. The *Bennett's Acceptance Ratio* (BAR) provides a maximum likelihood estimate of the free energy given the samples from the two states [22, 23]. Studies have shown the superiority of the BAR over EXP in molecular simulations: significantly less phase space overlap between states is required in order to converge results as compared to EXP [19, 20]. Note, however, that BAR requires sampling and energy evaluation of the system configurations from both states to estimate the free energy difference.

As phase space overlap affects the reliability of the estimate, free energy differences are most often calculated by simulating several intermediate states in addition to the two end states, in order to increase the overlap between each pair of states. A multistate extension of BAR, called the *multistate Bennett's Acceptance Ratio* (MBAR), has been proposed by Shirts et al. [24]. In this approach, a series of weighting functions are derived to minimize the uncertainties in free energy differences between all states considered simultaneously. MBAR reduces to BAR when only two states are considered, and it can also be interpreted as a zero-width weighted

histogram analysis method (WHAM) [11, 24]. MBAR has the lowest variance among the methods discussed here, and is likely the most reliable estimator for the type of free energy calculations described in this chapter [24, 25].

EXP and BAR are available in popular simulation packages such as Gromacs [26], Amber [27], and NAMD [28]. A python implementation of the MBAR estimator is instead provided by the authors of the original publication at https://github.com/choderalab/pymbar.

2.2.2 Thermodynamic Integration

In the thermodynamic integration (TI) approach, rather than potential energy differences, the data needed to estimate the free energy difference is the derivative of the potential energy with respect to the coupling parameter λ, a continuous variable that describes the series of intermediates states between the two end states $\lambda = 0$ and $\lambda = 1$. From this observable, it is possible to recover the free energy difference with the following formula.

$$\Delta G_{0,1} = \int_{\lambda=0}^{\lambda=1} \frac{\partial U(\lambda, \boldsymbol{x})}{\partial \lambda}_{\lambda} \tag{8}$$

The ensemble average $\langle \partial U/\partial \lambda \rangle$ is obtained from an equilibrium simulation performed at a certain value of λ. Numerical integration is then needed to recover the free energy difference between $\lambda = 0$ and $\lambda = 1$. The trapezoidal rule is often used for simplicity, but any numerical integration method can be employed. While λ is a continuous variable, only discrete values of it are sampled, so that there will be a bias in the estimate that depends on how well the chosen values of λ allow for an accurate quadrature. Therefore, while the accuracy of perturbation approaches depends on the overlap of energy distributions, the accuracy of TI depends on the smoothness of the integrand [29, 30].

TI is a popular method for the estimation of free energy differences, as it is robust and accurate while also easy to implement if the simulation code provides $\partial U/\partial \lambda$ values. The *alchemical analysis* tool [30] available at https://github.com/MobleyLab/alchemical-analysis already implements the automated analysis of simulation data collected with Gromacs [26], Amber [31], or Sire (http://siremol.org/), and the estimation of free energy differences using TI, EXP, BAR, and MBAR.

2.3 The Thermodynamic Cycle

For both perturbation and thermodynamic integration methods, several intermediate states are needed in order to obtain a reliable estimate of their free energy difference. In fact, when the two end states are the protein-bound and protein-unbound ligand, it is not possible to obtain a reliable binding free energy estimate by simulating only these two, and a pathway of intermediate states is needed. The computation of a free energy difference thus involves

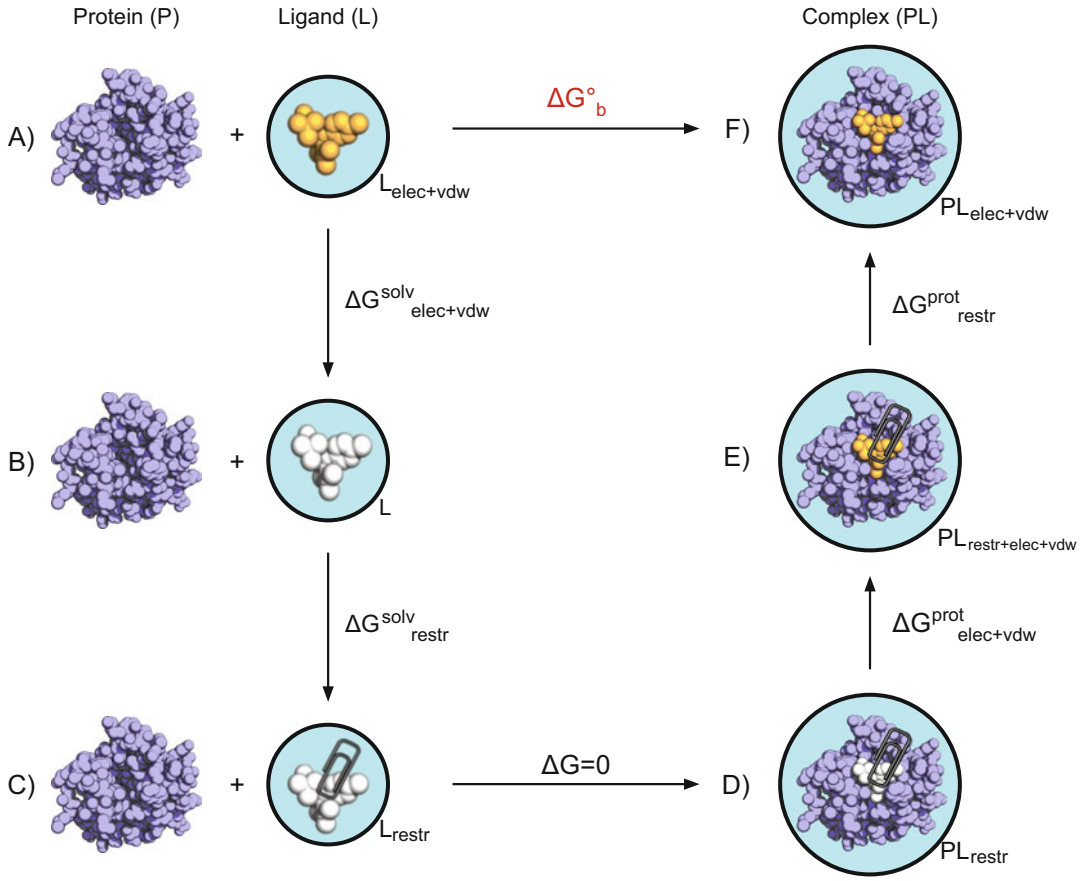

Fig. 1 Thermodynamic cycle used in absolute binding free energy calculations. The fully interacting ligand (orange) in solution at the top left (A) is transformed into a noninteracting solute (B, white) during a series of equilibrium simulations where its electrostatic and van der Waals interactions are scaled to zero, providing the term $\Delta G^{\text{solv}}_{\text{elec+vdW}}$. The ligand is then restrained while still noninteracting with the environment (C), calculating $\Delta G^{\text{solv}}_{\text{restr}}$. This state is equivalent to having the noninteracting ligand restrained within the protein cavity (D). The restrained and noninteracting ligand in complex with the protein has its electrostatic and vdW interactions turned back on again (E), giving $\Delta G^{\text{prot}}_{\text{elec+vdW}}$. The restraints between ligand and protein are then removed ($\Delta G^{\text{prot}}_{\text{restr}}$), closing the cycle, and the final state is the unrestrained and fully interacting ligand in complex with the protein (F). Reproduced from Aldeghi et al. [7] with permission from The Royal Society of Chemistry

the definition of the path that connects the two end states of interest. Since the free energy is a state function, the nature of the path is unimportant, and we can choose to use a thermodynamic cycle that connects the bound and unbound states through several nonphysical intermediate ones, as shown in Fig. 1. The nonphysical nature of the cycle used is the reason why this type of calculations is typically referred to as *alchemical*. The cycle depicted in Fig. 1 can be discretized in N states, which are independently simulated. These independent simulations are also often referred to as *windows*. Then, the free energy difference between each state i and its

successor $i + 1$ can be calculated, and the binding free energy can be recovered as the sum of all these $\Delta G_{i,i+1}$.

$$\Delta G_{\mathrm{b}} = \sum_{i=0}^{i=N-2} \Delta G_{i,i+1} \tag{9}$$

There are six states along the absolute binding free energy cycle that are conceptually helpful to think of: the two physically meaningful end states (i.e., the bound and unbound states), and four alchemical intermediate states where the ligand is *decoupled* from the environment, i.e., it does not interact with any other molecular species in the simulation. In general, we use the term *decoupled* to indicate a state in which the intermolecular interactions of the ligand have been removed, while the intramolecular interactions are still present; i.e., the atoms in the ligand feel the forces resulting from electrostatic and van der Waals (vdW) interactions with the other atoms in the same molecule. On the other hand, we use the term *annihilated* when also the intramolecular interactions have been removed. Figure 1 shows these intermediate states visually, to help understand their nature and how they connect the two end states together. In the unbound end state (state A), we are considering a ligand that is free in solution. Simulating a box containing only the ligand is computationally efficient and ensures there are no interactions with the protein. This fully interacting ligand in solution (Fig. 1. State A; in orange) is then transformed into a noninteracting solute (Fig. 1. State B; in white) by scaling its electrostatic and vdW interactions to zero through several nonphysical states that can be simulated independently. The ligand is then restrained to limit its accessible sampling volume while still not interacting with the environment (Fig. 1, State C; in white with a paper clip). Restraining the ligand substantially aids the convergence of the calculations. In fact, if the ligand was left unrestrained when decoupled, it could leave the binding pocket and float around the whole simulation box. Then, once its interactions with the environment were turned back on, it would have to go through a physical binding process in order to find its position in the protein again. State C is equivalent to having the noninteracting and restrained ligand within the protein cavity (Fig. 1, State D), since no work is needed to change the relative positions of the completely noninteracting protein and ligand. The decoupled and restrained ligand in complex with the protein has then its electrostatic and vdW interactions turned back on again (Fig. 1, State E). The restraints between ligand and protein are then finally removed, closing the cycle, and reaching the other physical end state, that is, the bound protein–ligand state (Fig. 1, State F).

If the cycle just described is discretized into N intermediate states, it is then possible to recover the binding free energy ΔG_{b}. Following from the discussion above, the cycle can be split into four main steps, each of them corresponding to the free energy

difference of transitioning between the intermediate states highlighted in Fig. 1. $\Delta G^{\text{solv}}_{\text{elec+vdW}}$ is the free energy of decoupling the ligand from the solution (state A → B), effectively bringing it to gas phase. $\Delta G^{\text{solv}}_{\text{restr}}$ is the free energy of restraining the ligand to a certain portion of phase space while still not interacting with the environment (state B → C). The free energy of placing the noninteracting and restrained ligand into the protein binding pocket (state C → D) is zero, so that it needs not be calculated. $\Delta G^{\text{prot}}_{\text{elec+vdW}}$ is then the free energy of coupling the restrained ligand to the environment again (state D → E), basically bringing the ligand back into the solution while being kept into the protein's binding pocket. Then, finally, $\Delta G^{\text{prot}}_{\text{restr}}$ is the free energy of removing the restraints that kept the ligand in place when not interacting with the environment. Thus, the binding free energy can be recovered as the sum of these four major steps, with the addition of a correction for the 1 M standard state:

$$\Delta G^{\circ}_{\text{b}} = \Delta G^{\text{solv}}_{\text{elec+vdW}} + \Delta G^{\text{solv}}_{\text{restr}} + \Delta G^{\text{prot}}_{\text{elec+vdW}} + \Delta G^{\text{prot}}_{\text{restr}} - k_{\text{B}} T \ln \frac{V}{V^0} \tag{10}$$

2.3.1 Restraints

The use of restraints is important in the protocol here described as it prevents the ligand from leaving the protein binding pocket while it is not interacting with the environment. This is necessary to make sure that the conformations sampled during the simulations correspond to a well-defined bound state. If the ligand were to leave the binding pocket in the windows where it is partially or completely decoupled, and started sampling the whole volume of the box, it would have a large configurational phase space available, leading to convergence issues. The use of restraints aids good phase space overlap between windows and faster convergence [32, 33].

In theory, any type of restraint that keeps the ligand in its bound pose can be used if its free energy contribution is properly accounted for. Also note that the use of restraints somewhat complicates the standard state correction, since the volume V does not correspond anymore to the volume of the whole box [15]. In practice, we find the set of restraints proposed by Boresch et al. [32] to be particularly convenient. In summary, this set of restraints not only allows to keep the ligand in a specific orientation relative to the binding pocket [33], but also provides an analytical solution for $\Delta G^{\text{solv}}_{\text{restr}}$, thus reducing the number of simulations to be run. Furthermore, the analytical solution also already includes the standard state correction. This set of restraints needs to be harmonic and is comprised by one distance, two angles, and three dihedrals, to be applied between three atoms of the ligand and three of the protein, as shown in Fig. 2. The authors also showed how the exact value of the force constant used for the harmonic restraints should not

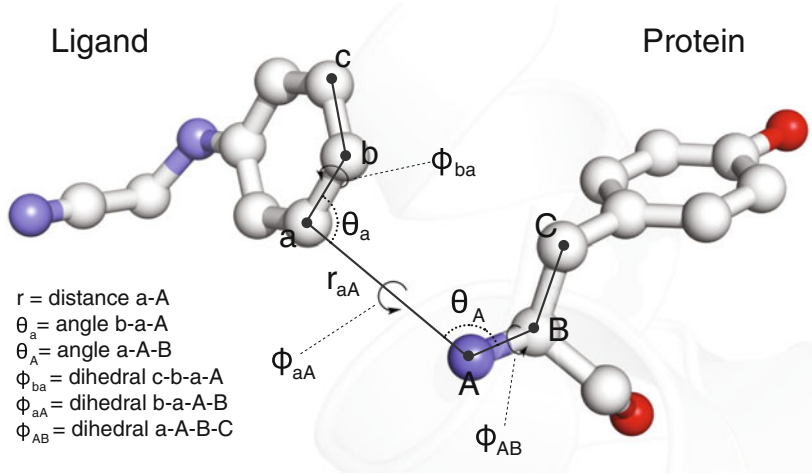

Fig. 2 Set of restraints proposed by Boresch et al. [32] for use in free energy calculations. The atoms and terms involved in this set of restraints are shown. Atoms "a," "b," and "c" belong to the ligand (on the left), while atoms "A," "B," and "C" belong to the protein (on the right)

affect the result, as the effects of different restraint strengths cancels out between $\Delta G^{\text{solv}}_{\text{restr}}$ and $\Delta G^{\text{prot}}_{\text{restr}} + \Delta G^{\text{prot}}_{\text{elec+vdW}}$.

$\Delta G^{\text{prot}}_{\text{restr}}$ needs to be estimated via simulations, running multiple intermediate states where the force constants of the six harmonic restraints are interpolated between their chosen value and zero. On the other hand, $\Delta G^{\text{solv}}_{\text{restr}}$ can be calculated analytically using the following formula (Eq. 32 in Boresch et al. [32]), which also includes the standard state correction:

$$\begin{aligned} &\Delta G^{\text{solv}}_{\text{restr}} - k_{\text{B}} T \ln \frac{V}{V^{\circ}} \\ &\quad = k_{\text{B}} T \ln \left[\frac{8\pi^2 V^{\circ}}{r_{\text{aA}}^2 \cdot \sin\theta_{\text{a}} \cdot \sin\theta_{\text{A}}} \frac{\left(k_{r_{\text{aA}}} k_{\theta_{\text{a}}} k_{\theta_{\text{A}}} k_{\phi_{\text{ba}}} k_{\phi_{\text{aA}}} k_{\phi_{\text{AB}}}\right)^{1/2}}{\left(2\pi k_{\text{B}} T\right)^2} \right] \end{aligned} \tag{11}$$

where V° is the standard volume of 1660 Å^3, r_{aA} is the reference value chosen for the distance restraint, θ_{a} and θ_{A} are the reference values of the two angle restraints, and k are the force constants of the harmonic restraints for the one distance (r_{aA}), two angle (θ_{a} and θ_{A}), and three dihedral (ϕ_{ba}, ϕ_{aA}, and ϕ_{AB}) restraints. Later in the text, we will refer to Eq. 11 simply as $\Delta G^{\text{solv}}_{\text{restr}}$, while implicitly assuming it also contains a correction for the standard state.

3 Materials

In this section, we summarize the information and tools that are needed before one can obtain binding free energy predictions as described in this chapter.

3.1 Starting Model Structure

The calculations here described require the three-dimensional coordinates of the protein–ligand complex as a starting point. The structure can be obtained experimentally (e.g., from X-ray crystallography or NMR) or from any modeling approach (e.g., from docking), depending on the objective of the calculations.

Considering that currently only limited sampling of phase space is possible through simulation, the closer the starting protein–ligand model is to the "true" structure, the more likely it is that the calculations will return an accurate binding free energy. As such, a high-resolution X-ray structure of the complex would probably be the most desirable starting point. Nonetheless, it is rarely the case that one has such a structure in advance, as at this stage free energy calculations are likely not needed anymore. However, it is possible to take the protein in complex with another ligand, and model the compound of interest into the binding pocket especially if conserved binding patterns are present and known. Alternatively, it is possible to use docking to generate hypotheses about the binding pose of the ligand of interest, and then use free energy calculations in order to accurately rescore them and identify the most stable pose [7, 34].

In some cases, the structure of the protein might not be experimentally resolved. In this situation, it is still possible to resort to homology modeling. However, the chances of starting from a structure far from equilibrium and possibly trapped in some metastable state are higher, resulting in calculations more likely to return inaccurate results. Good performance in relative binding free energy (RBFE) calculations using homology models has been recently reported [35]. However, ABFE calculations do not benefit from the same error cancellations present in RBFE methods and this usually manifests itself in ways that are indicative of a more pronounced sampling problem [36–38]. In any case, it is evident that the performance of the calculations would be highly dependent on the quality of the model, in particular in the proximity of the binding pocket. Thus, we would suggest extreme care in the interpretation of the results when the confidence on the quality of the starting protein–ligand structure is low—whether this comes from experiments or modeling.

3.2 Software Requirements

As mentioned in Subheading 2, simulations that sample a correct statistical ensemble of system configurations need to be performed. Moreover, depending on the free energy estimator we plan to use, we need to be able to extract the data that will be used for the free energy estimate. There are a number of simulation packages that satisfy these two requirements and are freely available, among which are Gromacs [26], Amber [31], NAMD [39], Sire (http://sire.org) and ProtoMS (http://www.essexgroup.soton.ac.uk/ProtoMS/index.html). Here, we often refer specifically to the setup in Gromacs, as it the code the authors are most familiar with.

3.3 Hardware Requirements

Considering that many intermediate states (windows) need to be simulated, obtaining well-converged ABFE calculations is computationally demanding, despite the calculations being highly parallelizable. The hardware requirements will depend on the details of the system simulated, the specifications of the hardware used, the simulation code, and on how long one is willing to wait for an answer. Here, we assume that a reasonable timeframe for a single calculation is not more than one or 2 days. At the time of writing of this protocol (early 2017), such deadline cannot be met if running the calculations on a modern desktop machine for most protein–ligand systems. Thus, typically, ABFE calculations need to be run on CPU or GPU clusters where at least a few hundred CPUs, or a few tens GPUs, or a mix of those, are available. Nevertheless, algorithmic and hardware improvements might mean that such recommended requirements might soon not apply anymore.

4 Methods

4.1 System Preparation

The steps for the preparation of the system to be simulated are not different from the ones needed for any simulation of a protein–ligand complex. First, the protein model typically needs to be refined. X-ray structures may contain missing residues and atoms, which need to be modeled; these include hydrogen atoms, which need to be added at the pH of interest. Similar considerations apply to the ligand, for which pK_a calculations might reveal the protonation state that is dominant in solution. Care should be taken also in checking for the presence of multiple tautomeric states.

4.2 Force Field Choice and Ligand Parameterization

Once the simulation box (including water molecules and ions) is prepared, a potential energy function (force field) needs to be chosen. Among the most commonly used force fields for protein–-ligand simulations are the ones from the Amber [40–42] and CHARMM [43–45] families. Although this is not necessarily the case, generally the more recent the force field the more likely it will be accurate given the additional experience collected through its use by the community and consequent refinement by the developers. The standard Amber and CHARMM biomolecular force fields do not contain parameters for organic molecules. Thus, in order to obtain the parameters for these small molecules, the complementary General Amber Force Field (GAFF) [46] and CHARMM General Force Field (CGenFF) [47] need to be used. For both these force fields there are tools that allow automated atom typing, assignment of parameters, and charge derivation [48–50]. It goes without saying that the quality of the ligand parameters is very important for the accurate estimation of their binding free energies. The user can therefore also use more

advanced protocols in order to refine the model of the organic molecule if the available parameters are suspected to be inadequate [51, 52].

4.3 Defining the Intermediate States

The core step in the setup of the calculations is possibly the definition of the alchemical pathway. As depicted in Fig. 1, several intermediate states need to be used to link the bound and unbound states and recover a precise estimate of the binding free energy. Since we will be sampling discrete alchemical states along the path, we need to choose how many intermediate states to have, how they should be distributed along the alchemical path, and for how long to simulate them. From Fig. 1 it is also possible to gather how there will be two sets of simulations during the calculations: one set in which the ligand in solution is simulated (left leg of the cycle in Fig. 1), and one in which the protein–ligand complex is simulated (right leg of the cycle in Fig. 1).

The coupling parameter λ defines the thermodynamic state of the system along the alchemical pathway. This parameter can take any value between zero and one and is used to scale coulombic charges, Lennard–Jones parameters and restraint force constants. Since these are the three sets of parameters that need to be scaled, it is convenient to have distinct coupling parameters: λ_{coul}, λ_{vdw}, λ_{restr}. In fact, it is often easier to carry out these three transformation separately, i.e., using a set of windows where only the charges are changed, then a set where only the LJ are changed, and then a third set where only the restraints are modified, as exemplified in Fig. 3. However, as mentioned in Subheading 2.3.1, $\Delta G^{\text{solv}}_{\text{restr}}$ can be calculated analytically so that λ_{restr} apply only to the protein–ligand complex simulations on the right-hand side of the cycle in Fig. 1. For convenience, and to avoid confusion, in Fig. 3 and throughout the text $\lambda = 0$ always indicates the state where the ligand is unrestrained and fully coupled (both in solution and the complex), and $\lambda = 1$ always indicate the state where the ligand is restrained and fully decoupled. We can see in Fig. 1 that for the protein–ligand complex simulations we need to calculate the free energy of coupling the ligand to the environment, i.e., going from the decoupled and restrained state to the coupled and unrestrained state. Nonetheless, it does not matter what is defined as $\lambda = 0$ and $\lambda = 1$, since the equilibrium free energy of coupling (state D → F in Fig. 1) the ligand is simply the opposite of the free energy of decoupling it (state F → D in Fig. 1); we just need to make sure the correct signs are used. When using the EXP estimator, there are also considerations about the forward and reverse calculations, as mentioned in Subheading 2.2.1. However, this is not a concern when using the more robust BAR and MBAR estimators that consider information from multiple states at once.

As mentioned, there are two main sets of simulations to be run: the ligand and complex simulations. Figure 3 shows a simple

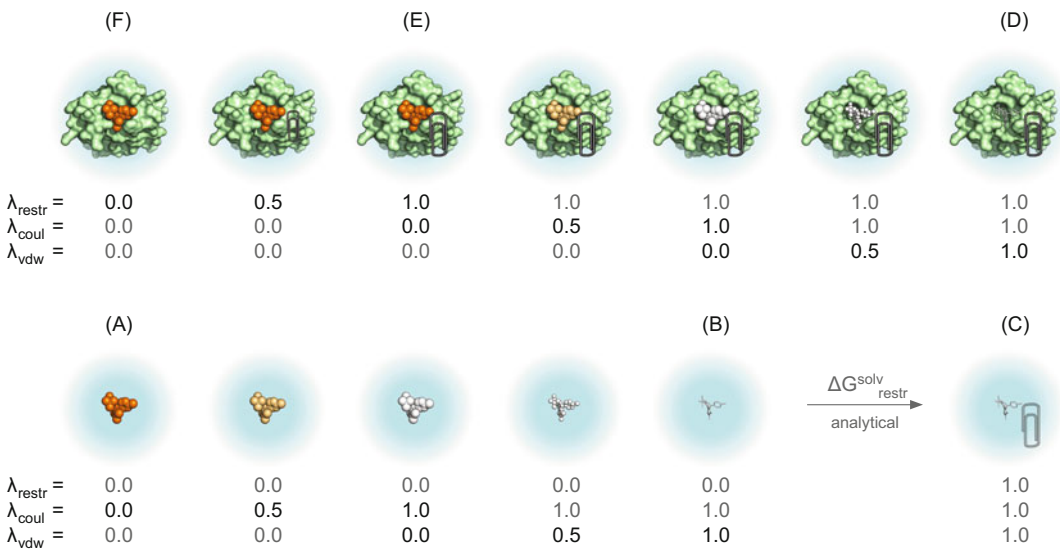

Fig. 3 Simplified example of alchemical path and states controlled by the λ parameters. The top row represents the decoupling and restraining of the ligand from the solvated protein–ligand complex; the bottom row represents the decoupling of the ligand from solution. The thermodynamic states (A)–(F) match the states shown in the thermodynamic cycle in Fig. 1. An orange color for the ligand indicates the presence of coulombic interactions with the environment, while a white color their absence; when the vdW surface of the ligand is shown, it indicates the presence of vdW interactions with the environment, while a stick representation of the ligand indicates the absence of vdW intermolecular interactions; a paper clip represents the presence of restraints, with its size proportional to the force constant used. The values of λ_{restr}, λ_{coul}, and λ_{vdw} define the thermodynamic state of the simulation

example of alchemical path that can be used for the calculations. It considers only a very small number of windows just for illustration purposes; in reality many more intermediate states should be used. At the top of this figure, the ligand is decoupled/annihilated from the protein environment, while at the bottom it is decoupled/annihilated from the solution environment. The columns of λ values for the restraints, coulombic, and Lennard–Jones interactions define each thermodynamic state, that is, each simulation that is carried out. For instance, in state F (corresponding to state F in Fig. 1 too) nothing has changed yet, since all λ values are zero and the simulation is a standard unbiased protein–ligand simulation. Then, in the second complex simulation, the restraints have been turned on partially, while the coulombic and LJ interactions are still fully on; in the third simulation, the restraints have been turned on completely ($\lambda_{restr} = 1$). Then, the coulombic interactions of the ligand start being modified, and between the fourth and fifth simulation of the complex, the ligand charges have been completely decoupled/annihilated. Finally, in the sixth and seventh simulation, the Lennard–Jones interactions of the ligand have also now been decoupled/annihilated, reaching state D where the ligand is fully decoupled and restrained. *See* **Note 1** for how to set up such an

alchemical path in Gromacs. The same procedure is carried out for the ligand in solution. In this case, however, the restraints are turned on after the decoupling of the ligand intermolecular interactions. However, as discussed in Subheading 2.3.1, when using the set of restraints proposed by Boresch et al. [32] and described here, these restraining steps do not need to be simulated since this free energy difference can be calculated analytically.

In general, it is difficult to know a priori how many windows should be run and for how long for a specific system that has not been tested before. Test runs are then useful in order to have an idea about phase space overlap, convergence, and precision of the calculations; we will discuss how one can assess these in Subheading 4.6. In practice, the computational resources available play a role as well: the more sampling (i.e., simulation length or number of repeat runs) the better, but if computational efficiency is required (either due to large scale calculations, or limited resources) it becomes even more important to test the calculations in order to find the setup that maximizes precision for the resources available. To provide an idea of what the calculations might involve, for systems containing drug-like ligands binding to small (~110 residues) and fairly rigid proteins, we typically employed 42 windows for the complex and 31 for the ligand simulations, each lasting 10–15 ns.

Ideally, the λ values for the intermediate states should be spaced in such a way that the statistical uncertainty in the free energy difference between neighboring states is equal, as this results in the lowest variance path [53]. In practice, this might be hard to achieve; yet, if the uncertainty of the ΔG between two states is particularly large, it is evident that more windows or tighter spacing is needed. Whether tighter lambda spacing is necessary can also be visually evaluated by looking at the plot of $\langle \partial U/\partial \lambda \rangle$ versus λ used for TI: where the slope of the curve changes more rapidly more windows are needed (an example is shown Fig. 4). The windows used for decoupling the ligand charges can generally be spaced linearly, e.g., $\lambda_{\text{coul}} = [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$. Using a LJ soft-core potential, λ_{vdw} can also be spaced linearly to start with, and adjusted as needed. However, more windows are typically needed for the LJ than for the charge decoupling. For the restraints, tighter lambda spacing is typically needed when inserting the harmonic potentials [33]. For instance, with $\lambda_{\text{restr}} = 0$ being the state without restraints and $\lambda_{\text{restr}} = 1$ the state with fully coupled restraints, a possible spacing is $\lambda_{\text{restr}} = [0.0, 0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.35, 0.5, 0.75, 1.0]$.

While the number, spacing, and length of the windows will need to be adapted to the specific system of interest and the desired precision, there are a few rules that should always be followed. Some of these have already been mentioned or alluded to in the

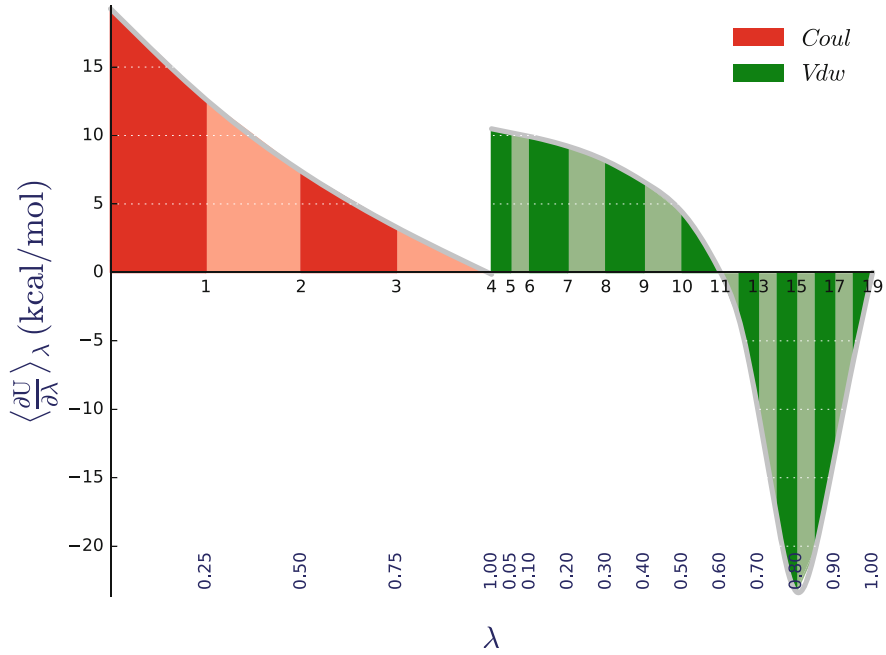

Fig. 4 Thermodynamic integration plot for the decoupling of a ligand from solution. More specifically, it shows the $\langle \partial U/\partial \lambda \rangle$ profile for the decoupling of n-phenylglycinonitrile [54], the ligand used as an example in the tutorial that accompanies this chapter and available on the alchemistry.org website. It is possible to notice the separate decoupling of coulombic and vdW interactions, as well as the fact that the coulombic transformation is smoother than the vdW one

previous paragraphs; here we report them more explicitly and with a brief explanation:

- Use a soft-core potential for decoupling/annihilating the LJ interactions (*see* **Note 2**) [55–58]. Simple scaling schemes for the LJ parameters result in unequal phase space overlap between lambda windows, as well as large forces and numerical instabilities due to a singularity at $r_{ij} = 0$, where i and j are two particles and r the distance between them. Soft-core potentials resolve the instabilities observed during the decoupling of the van der Waals parameters by modifying the LJ equation so that the interaction energy is finite for any configuration.
- Decouple charges and LJ separately (as shown in **Note 1**). Imagine two atoms with opposite partial charges and little repulsion. Even if the charges are small, in the limit of $r_{ij} = 0$, the potential energy of the interaction goes to infinity. This results in large forces and simulation crashes [59, 60]. Thus, a simple way to resolve this issue is to make sure the LJ parameters are decoupled after the charges, or coupled before them, so that when charges are present the repulsive part of the LJ interactions is present too. Another way around this problem is to use a soft-

core potential for the electrostatic interactions as well (*see* **Note 3**) [60, 61].

- Avoid changing the net charge of the system across the alchemical path; if you need to, as in the case of decoupling/annihilating charged ligands, take particular care in correcting for artifacts [11, 62]. If you are decoupling/annihilating a ligand that bears a net charge of +1 (which you have neutralized using counter ions), at $\lambda_{\text{coul}} = 1$ the system in the condensed phase will have a net charge of −1. Currently, most molecular dynamics simulations use a periodic treatment of electrostatics via Ewald summation or related mesh methods. These methods require the box to be neutral; therefore counter ions are added in order to neutralize the system. When the ligand is brought to a separate vacuum state, the sum of remaining particle charges will no longer be neutral. A homogeneous compensating background charge is then added in order to keep the neutrality of the system, resulting in spurious interactions and artifacts [62, 63]. Rocklin et al. [62] have proposed a correction that applies exactly to this situation (*see* **Note 4**).

For a more detailed overview of the current best practices related to constructing the alchemical path, we highly recommend the chapter by Shirts [11] and the relevant pages on the Alchemistry website (www.alchemistry.org).

4.4 Defining the Restraints

When using the set of restraints proposed by Boresch et al. [32], one distance, two angle, and three dihedral harmonic restraints between the ligand and the protein need to be defined. Thus, one needs to choose three protein and three ligand atoms to be involved in the restraints, the reference values of the harmonic potentials and their force constants. Since the objective is to keep the position and orientation of the decoupled ligand roughly similar to its known or hypothetical binding pose, we suggest choosing atoms from rigid parts of the ligand and the protein: e.g., three atoms from the protein backbone and three atoms from an aromatic ring in the core of the small molecule (*see* **Note 5**).

The equilibrium values of the distance and angles restraints can be obtained from a small preliminary MD simulation, or also simply set equal to the distance and angles observed in the X-ray or docked ligand structure. In fact, these equilibrium values do not need to represent exactly the minimum energy orientation of the ligand; if the ligand is restrained to a slightly unfavorable orientation, $\Delta G^{\text{prot}}_{\text{restr}}$ will be larger as more work is necessary to fix the ligand in that position, but it will be compensated by $\Delta G^{\text{prot}}_{\text{elec}}$, $\Delta G^{\text{prot}}_{\text{vdw}}$, and $\Delta G^{\text{solv}}_{\text{restr}}$. However, it is important that the target-restrained orientation is not a high-energy state and it is easily accessible from the starting orientation of the ligand. If the target orientation is chosen in such a way that the ligand clashes into the protein, or if the starting

orientation is kinetically trapped in a different conformation as compared to the target one, large and unconverged free energy values of $\Delta G^{\text{prot}}_{\text{restr}}$ will be obtained [33, 54]. In our experience we find that, using force constants of 10 kcal/mol/Å^2 [rad^2], restraining free energies are typically below 2 kcal/mol; in some case they can be larger, but we would suggest that if they exceed 3–4 kcal/mol the simulations should be carefully checked, as this may be symptomatic of either a trapped ligand conformation or an error in the definition of the target orientation.

In their example, Boresch et al. [32] used restraint force constants ranging from 5 to 50 kcal/mol/Å^2 [rad^2], showing how the correctness of the approach is independent of the stiffness of the restraints. The precision of the restraining free energy may however be affected by restraints that are too loose of too stiff [32]. Thus, in practice, intermediate force constants of about 10 kcal/mol/Å^2 [rad^2] have typically been used [7, 8, 54, 64]. **Note 6** shows how protein–ligand restraints can be applied in Gromacs.

4.5 Running the Simulations

At this point the simulations can be started. The rest of the setup is equivalent to a standard unbiased MD simulation, and it is necessary to make sure the correct ensemble is sampled [65]. Remember that anything that can affect the potential energy of the system and the resulting thermodynamic ensemble can also affect the free energy calculations. For instance, the correct Boltzmann distributions of kinetic energies should be generated for both the coupled and decoupled states. Thus, stochastic thermostats that ensure ergodicity and the generation of the correct ensemble, such as the Andersen or Langevin thermostats [66–68], are typically used [69–73]. Similar considerations apply to pressure coupling, so that the Berendsen barostat [74] is avoided in favor of Parrinello–Rahman [75] for the production runs from which data is collected.

In order to be able to analyze the data and estimate the free energy differences at the end of the simulations, it is important that the code stores this information for postprocessing. What data is needed depends upon the free energy estimator of choice (e.g., $\partial U/\partial\lambda$ for TI or ΔU_{ij} for perturbation approaches). If the software allows it, one can also save the data used by all estimators and then compare the results from different approaches (*see* **Note 7**).

Each state, as defined by its λ value, needs then to be minimized and equilibrated independently. This means that if there is a total of N windows, we will need to run N separate minimization, equilibration, and production simulations. This is important because the ligand is present in some simulations, but effectively absent from others, so that the protein and solvent need to adapt to the different environments. All production runs can be run independently, making it easy to parallelize the calculations, or with Hamiltonian replica exchange (HREX) in order to enhance the mixing between

states [76, 77] (*see* **Note 8**). The HREX approach has low computational overhead and can result in enhanced sampling, improved phase space overlap and faster convergence, while in the worst case scenario is no different from running the calculations independently [76, 78–80].

4.6 Analyzing the Data

The data collected from all simulations is finally analyzed in order to obtain an estimate of the binding free energy as the sum of the smaller free energy differences along the path. In particular, we need to estimate the two free energy differences for the decoupling of the ligand from solution ($\Delta G^{\text{solv}}_{\text{coul+vdw}}$) and from the protein–ligand complex ($\Delta G^{\text{prot}}_{\text{elec+vdw}} + \Delta G^{\text{prot}}_{\text{restr}}$); these contributions can then be added along with the $\Delta G^{\text{solv}}_{\text{restr}}$ that was obtained analytically to recover the final binding free energy (*see* **Note 9**). Each separate free energy estimate will have its associated uncertainty, which will need to be propagated into the final $\Delta G^{\circ}_{\text{b}}$. Simulation packages that support alchemical free energy calculations also provide tools for their analysis, using one or more of the estimators previously discussed. Alternatively, the *alchemical-analysis* tool (https://github.com/MobleyLab/alchemical-analysis) is a Python program that implements the automated analysis of free energy calculations performed with Gromacs [26], Amber [31], Sire (http://sire.org), and Desmond [81], and allows easy access to a number of estimators, including MBAR, and the best practices mentioned below (*see* **Note 10**) [24, 30]. An example plot for TI obtained with this tool is shown in Fig. 4**.**

When analyzing the data obtained from the simulations, it is first important to make sure the samples are not correlated. In practice, $\partial U/\partial\lambda$ and ΔU_{ij} values are typically printed to file frequently, and their values are likely correlated. One could set a particularly low output frequency when setting up the calculations, but this is likely to result in the loss of potentially useful information from the simulations. What is often done is then to calculate the autocorrelation time τ of the time series, and then subsample the data by picking a sample every $1 + 2\tau$ [82]. Once uncorrelated $\partial U/\partial\lambda$ or ΔU_{ij} values have been obtained, they can be fed into different estimators, such as TI and MBAR. Since TI and perturbation approaches use different information for the free energy estimation and have different limitations, comparing the results obtained with the two can be a simple way to check for potential analysis or sampling issues with the calculations.

It is common to exclude from the analysis an initial portion of the simulations, as it is expected to contain nonequilibrated samples. The exact determination of the nonequilibrated region of the simulations is however not trivial. In fact, if extensive equilibration is performed prior the production runs, some system may reach equilibrium before the data needed for the analysis even starts

being collected. Nonetheless, it is important to try to exclude nonequilibrated regions of the simulations from the analysis and two main approaches have been proposed for this task. One approach is to plot the convergence of the free energy results in both the "forward" and "reverse" directions, that is, plotting the free energy estimate as a function of the simulated time by including more data while going both from t_0 to t_f, and from t_f to t_0, where t_f is the time of the final snapshot in the simulations [30]. When all data is included, the forward and reverse estimates are the same. If the data have been collected at equilibrium and from the same distributions, and if the calculations have converged, the forward and reverse plots should agree within error. If, when considering the data from the two ends of the simulations, the "forward" and "reverse" free energy estimates converge to two separate and well-defines values, this indicate that nonequilibrated samples have likely been included in the analysis [30]. A different approach, which detects the equilibrated region automatically, has instead recently been proposed by Chodera [83]. In this automated approach, the autocorrelation time is calculated while removing larger portions of the simulation data, and the equilibration time is chosen as the time that maximizes the number of effective uncorrelated samples [83].

Phase space overlap is another property that should be checked when analyzing the results. Poor overlap in specific regions can be resolved by using more, or differently spaced, λ windows. Little overlap does not necessarily lead to wrong free energy estimates, but does result in increased uncertainty; the user should decide what level of uncertainty is acceptable. However, when using perturbation approaches, very poor overlap can result in an underestimate of the variance and an inaccurate free energy estimate [30]. If the MBAR estimator is used, it is possible to obtain an overlap matrix that provides a quantitative estimate of the phase space overlap between simulations [24]. This matrix shows the probability of a sample from state i having been generated in state j, thus providing an indication of the degree of phase space overlap. An example of such an overlap matrix is shown in Fig. 5. The overlap matrix should be tridiagonal, which means that all elements of the main diagonal, as well as the diagonal below and above it, should be nonzero. A value of 0.03 for the tridiagonal elements has been suggested as a threshold to highlight potential phase space overlap issues between two windows [30]. If phase space overlap issues are identified during testing of the free energy protocol for a specific system, it might be possible to adjust the spacing of the λ windows so to increase overlap and reduce the uncertainty of the free energy estimate. If phase space overlap issues are identified instead after running extensive simulations, it is still possible to run additional λ windows in the problematic area of the path, which will be more cost-effective than rerunning the whole calculation with a different

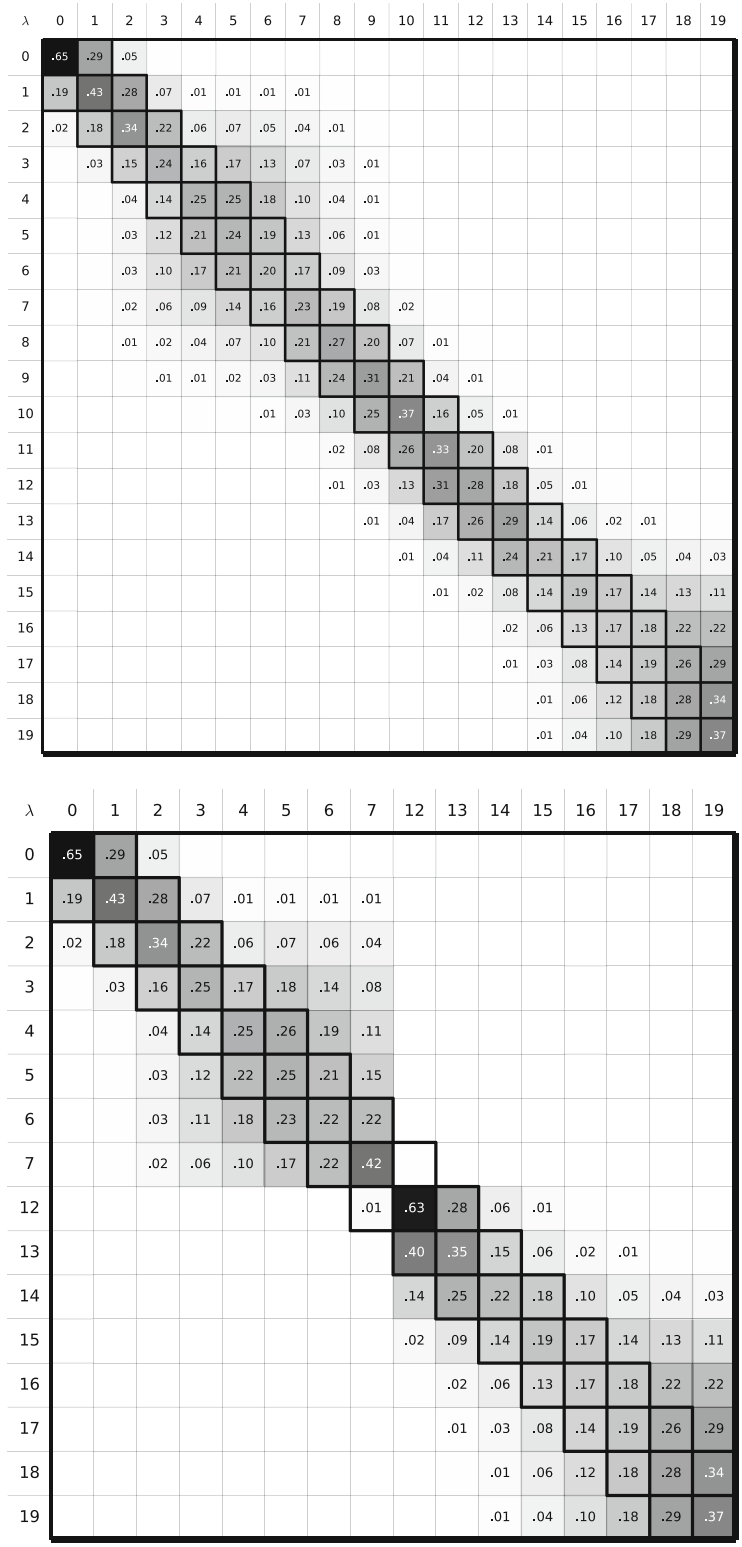

Fig. 5 Overlap matrices for the decoupling of a ligand from solution. The overlap matrix at the top was obtained for the decoupling of n-phenylglycinonitrile from solution using a total of 20 windows, and shows good overlap

spacing. In general, a good phase space overlap is attainable for the decoupling of the ligand from solution, but can be more challenging for the decoupling of the ligand from the complex.

Finally, in some cases it might be necessary to correct for simulation artifacts. We have already mentioned how charged ligands can be problematic, as the net charge of the system changes across windows during the charge annihilation process. This introduces a series of artifacts when using Ewald summation methods for the treatment of electrostatics in periodic systems [62]. In this case, the approach proposed by Rocklin et al. [62], which consists of a few analytical corrections and an implicit-solvent calculation, can be used. Sometimes, a correction for the long-range Lennard–Jones interactions is included too. Some simulation packages, like Gromacs, already include an analytical correction for the long-range LJ interactions, which are generally excluded during the simulations by the use of LJ cutoffs. However, these analytical corrections assume an isotropic system outside the cutoff, which is not valid when simulating protein–ligand complexes (as the cutoff is generally smaller than the largest dimension of the complex). Shirts et al. [84] have proposed a numerical correction to the free energy estimate for such cases. The correction is based on the postprocessing of some of the simulations using a larger LJ cutoff, effectively building an additional thermodynamic cycle on top of the one already used in order to calculate the $\Delta\Delta G$ of going from a system with a short LJ cutoff (where the isotropic assumption fails) to a system with a long LJ cutoff (where the isotropic assumption holds). Note that since the long-range part of the LJ interactions is always attractive, the correction results in the prediction of slightly stronger affinity values; in our experience, for drug-like ligands and using a LJ cutoff of 1.0 nm, this typically amounts to an additional 0.3–0.8 kcal/mol to the binding free energy. Taking these additional interactions into account does not necessarily result in more accurate binding affinity predictions, but it does increase reproducibility by removing the dependence on the cutoff value used. An alternative option to the correction by Shirts et al. [84] is to run the simulation using lattice summation methods also for the LJ interactions, such as the recently proposed LJ-PME approach [85]. In this way, there is no need for a post hoc correction, and the forces arising from these long-range interactions are considered already during the simulation. However, this comes at additional computational cost [85].

Fig. 5 (continued) between each pair of states. The matrix at the bottom was obtained by disregarding the data from four windows (8, 9, 10, and 11), and shows poor overlap between states 7 and 12. In the latter case, it becomes evident that additional simulations or a different λ spacing is needed. The overlap plots have been obtained with the *alchemical analysis* python tool [30]

In this section, we provided a brief overview of how to analyze alchemical free energy results while also mentioning some recommended procedures to check for possible issues affecting the reliability of the estimate. For a more detailed explanation of the best practices and tools available for the analysis of equilibrium free energy calculations, we highly recommend reading the article by Klimovich et al. [30].

4.7 Sampling and Modeling Challenges

As in all MD simulations, issues may arise as a consequence of limited sampling and model inaccuracies. Inadequate sampling might manifest itself as poorly converged free energy estimates, which can be identified through convergence plots and the size of the uncertainty in estimates. Repeating the calculations, possibly using different starting structures, is a simple yet appropriate way to assess the precision of the calculations and highlight possible convergence problems, confirming that the results are reproducible. It is important to assess whether converged results can be obtained within the timescale of the planned calculations for the specific system of interest. Alchemical free energy calculations are expected to be more accurate than implicit solvent end-point methods and scoring functions, [6, 8, 9]. However, they are also sensitive to sampling issues [86] such that they can become unreliable if the phase space explored by the simulations is inadequate for capturing the conformational ensembles that largely determine the binding free energy. Moreover, severe sampling issues due to very slow degrees of freedom that cannot be sampled during the timescale of the simulations can result in apparently converged calculations despite affecting the predicted affinities. This can be the case when the ligand induces conformational changes in the protein. If these changes are known, the problem may be tackled by separately calculating the potential of mean force (PMF) for the conformational change with methods such as umbrella sampling. Mobley et al. [87] used such an approach to take into account the free energy contribution of a valine side chain rearrangement that was not sampled during the alchemical calculations. Lin et al. [88] have used umbrella sampling to calculate the free energy change involved in the large loop rearrangement that is associated with the binding of type-II kinase inhibitors. In many instances, however, similar slow degrees of freedom might be unknown, and left unsampled. The position of structural water molecules can be important as well. If these water molecules are not present in the starting structure for the calculations, but are important for binding, and if diffusion to their most stable location is kinetically hindered, this too can result in convergence issues and biased results. Sampling schemes that try to tackle these issues have been developed or are subject of active research [89–92], but are outside the scope of this discussion.

Considerations about the quality of the starting structure and the modeling of the system's chemistry are important too. For instance, the ligand might have multiple protonation and tautomer states in solution, without a single one being necessarily largely dominant. In addition, the solvent and the protein represent two very different environments, so that pK_a shifts are possible and the protonation states of the ligand or protein residues may change between the unbound and bound states. Currently, this is not inherently captured by typical simulations. These complications might thus require multiple calculations or more advanced protocols that correctly take into account the contributions arising from the presence of multiple molecular species in equilibrium. Furthermore, the suitability of the force field for the molecules studied can clearly have important effects on the accuracy of the calculated affinities. Given the vast chemical space of small drug-like molecules, the transferability of certain parameters, such as those for the torsions, can be problematic and may result in inaccurate energies and relative conformational populations. This, in turn, may affect the binding free energy results in ways that are hard to foresee. Small molecule parameters can be refined, for instance, by targeting quantum mechanical energies [8, 51] but it is up to the user to decide how much human and computational effort to invest in order to validate the quality of the parameters.

5 Notes

In these Notes we describe how specific steps can be carried out in Gromacs 2016. It is however recommended to also refer to the Gromacs manual for additional details, in particular considering possible changes/developments in future versions of the software.

1. In Gromacs, the fictitious alchemical path depicted in Fig. 3 and pertaining to the simulations of the protein–ligand complex could be implemented via options in the mdp file as follows.

```
free-energy     = yes
couple-moltype  = ligand
couple-lambda0  = vdw-q
couple-lambda1  = none
bonded-lambdas  = 0.0  0.5  1.0  1.0  1.0  1.0  1.0
coul-lambdas    = 0.0  0.0  0.0  0.5  1.0  1.0  1.0
vdw-lambdas     = 0.0  0.0  0.0  0.0  0.0  0.5  1.0
```

The `free-energy` flag tells the code we are performing free energy calculations; `couple-moltype` defines the molecule type, as defined in the topology, that is to be (de)coupled.

`couple-lambda0` and `couple-lambda1` define the state of the intermolecular interactions at λ values of zero and one, respectively; in this case we are defining that at lambda zero both coulombic and vdW interactions are on, while they are off at lambda one (i.e., the ligand is decoupled while going from lambda zero to one). `bonded-lambdas` defines the vector of λ_{restr} values, since the restraints can be applied as additional bonded terms between different molecules, as shown in **Note 6**. This vector can be omitted when decoupling the ligand from solution, since for that leg of the cycle the contribution of the restraints can be obtained analytically. `coul-lambdas` and `vdw-lambdas` define instead the lambda vectors for the decoupling of the coulombic and Lennard–Jones interactions. In this example, there are seven states to be simulated, each defined by a column of the vectors above. `init-lambda-state` is then used in the mdp file in order to choose which column is being simulated; for instance, `init-lambda-state = 0` means that the index 0 of the lambda arrays is simulated ($\lambda_{\text{restr}} = 0.0$, $\lambda_{\text{coul}} = 0.0$, $\lambda_{\text{vdw}} = 0.0$), while `init-lambda-state = 3` means that index 3 of the arrays is simulated ($\lambda_{\text{restr}} = 1.0$, $\lambda_{\text{coul}} = 0.5$, $\lambda_{\text{vdw}} = 0.0$). Additionally, the `couple-intramol` flag can be used to decide whether the ligand should be decoupled or annihilated.

2. In Gromacs, the Lennard–Jones soft-core potential by Beutler et al. [55] is implemented and can be used by specifying the following in the mdp file.

```
sc-alpha = 0.5
sc-power = 1
sc-sigma = 0.3
```

These three parameters define the soft-cored Lennard–Jones function [55]. In particular, the alpha parameter needs to be larger than zero for the LJ interactions to be soft-cored. An `sc-alpha` of 0.5, `sc-power` of 1, and `sc-sigma` of 0.3 are typically used values [33, 54, 56, 69, 93].

3. A soft-core potential for the coulombic interactions can be activated in Gromacs with the mdp option `sc-coul`. However, for equilibrium ABFE calculations, it is not clear whether performing the coulombic and vdW transformations simultaneously would result in much better efficiency as compared to performing the transformation separately. In fact, since the charge decoupling process typically shows good overlap between windows and smooth $\langle \partial U/\partial \lambda \rangle$, few windows are usually necessary [61, 93, 94].

4. One might think that in order to keep the neutrality of the box it could also be possible to (de)couple an ion from solution

together with the ligand, so that the change in net charge would be compensated. This is however likely to introduce spurious free energy contributions, due to the fact that the free energy of decoupling an ion from the water box is not the same as the free energy of decoupling it from the box that contains also the protein. Since the environments in the two legs of the ABFE cycle are very different, the two ion decoupling free energies are unlikely to cancel out. Alternatively, the two legs of the cycle could be performed in a single simulation box. However, for the ligand in solution to avoid strong interactions with the protein–ligand complex a very large simulation box is needed. When considering the scaling of MD simulations with the number of particles, it can be seen how the calculations would become computationally much more expensive. In addition, there could still be residual finite size effects that remain unaccounted for [95, 96].

5. In our experience, we have found that including hydrogen atoms (which bonds are typically constrained with the LINCS algorithm) as part of the set of restrained atoms can cause stability issues during the simulation. Therefore, we would suggest choosing a set of heavy atoms only for the protein–ligand restraints.

6. Protein–ligand restraints can now be applied in Gromacs through the use of the `[intermolecular_interactions]` section at the very end of the topology file. In this section, it is possible to define bonded terms between any atom in the system using global indices (i.e., the indices found in the coordinates file). To generate harmonic restraints, bonds of type 6, angles of type 1, and dihedrals of type 2, can be used (*see* Table 5.5 in the Gromacs manual) as shown in the example below. Furthermore, it is necessary to define the force constants for the restraints in states A and B (i.e., states $\lambda_{\text{restr}} = 0$ and $\lambda_{\text{restr}} = 1$). One of the two force constants needs to be zero, which corresponds to the state where the restraints are effectively absent. The equilibrium distance/angles can be set to the same value for both states instead. The other force constant should be set to the value chosen, as discussed in Subheading 4.4, for example 10 kcal/mol/Å^2 [rad^2]. Remember that Gromacs uses kJ/mol and nm as units for energy and length. In the example below, *ai, aj, ak*, and *al* are global atom indices (atoms 1, 2, 3 can be ligand atoms, and atoms 4, 5, 6 protein atoms, or vice-versa); `type` defines the function type used; b_A, th_A, and phi_A are the equilibrium values of distance and angles for the harmonic restraints in state A ($\lambda_{\text{restr}} = 0$), and b_B, th_B, and phi_B the ones in state B ($\lambda_{\text{restr}} = 1$); k_A is the force constant used in state A ($\lambda_{\text{restr}} = 0$), and k_B the one used in state B ($\lambda_{\text{restr}} = 1$). The bonded-lambdas vector discussed in **Note 1** will

interpolate between the force constant (and equilibrium positions) in state *A* and *B*.

```
[ intermolecular_interactions]
[ bonds ]
; ai  aj  type  bA   kA   bB   kB
  3   4   6     0.5  0.0  0.5  4184.0

[ angles ]
; ai aj ak type  thA   kA   thB   kB
  2  3  4  1     80.0  0.0  80.0  41.84
  3  4  5  1     30.0  0.0  30.0  41.84

[ dihedrals ]
; ai aj ak al type  phiA    kA   phiB    kB
  1  2  3  4  2     -100.0  0.0  -100.0  41.84
  2  3  4  5  2      120.0  0.0   120.0  41.84
  3  4  5  6  2       15.0  0.0    15.0  41.84
```

7. In the Gromacs mdp file, `calc-lambda-neighbors` controls the number of lambda values for which ΔU_{ij} will be calculated. If one wants to use BAR rather than FEP, then the value of `calc-lambda-neighbors` should be set to 1; in order to also use MBAR, a value of −1, which effectively means "all," should be used instead. With the option `dhdl-derivatives` set to "yes" by default when using the free energy code, also the $\partial U/\partial\lambda$ values (needed for TI) will be calculated and stored with the frequency specified by `nstdhdl`. By setting `separate-dhdl-file = yes`, the $\partial U/\partial\lambda$ and ΔU_{ij} values will be printed to a file with default name "dhdl.xvg", which can later be processed with the Gromacs tool `gmx bar` or with the `alchemical_analysis` Python tool [30].
8. Replica exchange can be activated when using `mdrun` in Gromacs with the `--replex` flag, which specifies the frequency of exchange attempts. It has been suggested to set the frequency of the exchanges as high as possible, as long as this does not result in substantial simulation slowdown [76]. With only the `--replex` flag, swaps only between neighboring windows are attempted. Using also the `--nex` command, multiple exchange are performed between any pair of lambda windows, resulting in further mixing between states. A number of swaps between N^3 and N^5, where N is the number of replicas, has empirically been found to provide sufficient mixing [76]. Thus, as an example, for a calculation involving 30 windows, one could use `mdrun --replex 1000 --nex 500000`.
9. In addition to the free energy terms discussed, in certain cases a contribution due to the symmetry should in principle be taken

into account [33]. A symmetry correction applies when: (a) the ligand is symmetric or contains a symmetric moiety; (b) the equivalent (symmetric) orientations or the ligand are not sampled during the simulations, typically due to large energy barriers between the orientations; and (c) the restraints used break the symmetry, such that the normally equivalent orientations of the molecule have different energies due to the biasing potential employed [33]. For a ligand with a number σ_L of symmetric and thus equivalent orientations, this correction amounts to $\Delta G_{sym} = -kT\ln(\sigma_L)$. Thus, at 300 K, for a benzene molecule $\Delta G_{sym} = -kT\ln(12) = -1.48$ kcal/mol; while for a ligand containing a phenyl group $\Delta G_{sym} = -kT\ln(2) = -0.41$ kcal/mol.

10. The *alchemical_analysis.py* script may be run as follows:

```
alchemical_analysis.py -d dhdl_files/ -s 1000 -t 298 -w -g
```

Where the `--d` flag specifies the directory containing the files with the $\partial U/\partial\lambda$ and ΔU_{ij} information from all the windows; the `--s` flag specifies the amount of simulation time (in ps) to discard as equilibration period; `–t` specifies the temperature in Kelvin; `–w` asks for the overlap matrix to be plotted; and `--g` asks for the ΔG between each adjacent state to be plotted. The code defaults to the analysis of Gromacs files, so that in the *dhdl_files* folder it expects to find files that start with "dhdl" and have ".xvg" extension (e.g., *dhdl_1.xvg*, *dhdl_2.xvg*, *dhdl_3.xvg*, etc.); this can however be changed with the `--p` and `--q` flags. Moreover, free energy calculations run with Amber [31], Sire (http://sire.org), or Desmond [81], can still be analyzed by choosing the software with the `--a` flag. The `--f` flag allows instead to plot the free energy estimate as a function of time in both directions. These are just some of the options available, and the reader can find an explanation for all of them by displaying the help message of the script.

Acknowledgments

The EPSRC and Evotec via the Systems Approaches to Biomedical Sciences Doctoral Training Centre (EP/G037280/1) support M.A. J.B. is supported by the EPSRC/MRC via the Systems Approaches to Biomedical Sciences Doctoral Training Centre (EP/G037280/1) with additional support from GSK. We thank David Mobley (University of California, Irvine), John Chodera (MSKCC), and Michael Shirts (University of Colorado, Boulder) for sharing their extensive experience on alchemical free energy calculations through publicly available platforms and personal communications. Work in PCB's laboratory is currently supported by

the MRC, BBSRC, and the John Fell Fund. We thank the Advanced Research Computing (ARC) facility, the EPSRC UK National Service for Computational Chemistry Software (NSCCS) at Imperial College London (grant no. EP/J003921/1), and the ARCHER UK National Supercomputing Services for computer time granted via the UK High-End Computing Consortium for Biomolecular Simulation, HECBioSim (http://www.hecbiosim.ac.uk), supported by EPSRC (grant no. EP/L000253/1).

References

1. Mobley DL, Dill KA (2009) Binding of small-molecule ligands to proteins: "what you see" is not always "what you get". Structure 17:489–498
2. Chipot C (2014) Frontiers in free-energy calculations of biological systems. Wiley Interdiscip Rev Comput Mol Sci 4(1):71–89
3. Kitchen DB, Decornez H, Furr JR, Bajorath J (2004) Docking and scoring in virtual screening for drug discovery: methods and applications. Nat Rev Drug Discov 3:935–949
4. Genheden S, Ryde U (2015) The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. Expert Opin Drug Discov 10 (5):449–461
5. Shirts MR, Mobley DL, Chodera JD (2007) Free energy calculations: ready for prime time? Annu Rep Comput Chem 3:41–59
6. Aldeghi M, Bodkin MJ, Knapp S, Biggin PC (2017) A statistical analysis on the performance of MMPBSA versus absolute binding free energy calculations: bromodomains as a case study. J Chem Inf Model 57:2203–2221
7. Aldeghi M, Heifetz A, Bodkin MJ, Knapp S, Biggin PC (2016) Accurate calculation of the absolute free energy of binding for drug molecules. Chem Sci 7:207–218
8. Aldeghi M, Heifetz A, Bodkin MJ, Knapp S, Biggin PC (2017) Predictions of ligand selectivity from absolute binding free energy calculations. J Am Chem Soc 139:946–957
9. Chipot C, Pohorille A (2007) Free energy calculations: theory and applications in chemistry and biology, vol 86. Springer series in chemical physics. Springer, New York
10. Pohorille A, Jarzynski C, Chipot C (2010) Good practices in free-energy calculations. J Phys Chem B 114(32):10235–10253
11. Shirts MR (2012) Best practices in free energy calculations for drug design. Methods Mol Biol 819:425–467
12. Shirts MR, Mobley DL (2013) An introduction to best practices in free energy calculations. In: Monticelli L, Salonen E (eds) Biomolecular simulations: methods and protocols. Humana Press, Totowa, NJ, pp 271–311
13. Zhou H-X, Gilson MK (2009) Theory of free energy and entropy in noncovalent binding. Chem Rev 109(9):4092–4107
14. Gilson MK, Given JA, Bush BL, McCammon JA (1997) The statistical-thermodynamic basis for computation of binding affinities: a critical review. Biophys J 72(3):1047–1069
15. General IJ (2010) A note on the standard state's binding free energy. J Chem Theory Comput 6(8):2520–2524
16. Gapsys V, Michielssens S, Peters JH, de Groot BL, Leonov H (2015) Calculation of binding free energies. In: Kukol A (ed) Molecular modeling of proteins. Springer New York, New York, NY, pp 173–209
17. Zwanzig RW (1954) High-temperature equation of state by a perturbation method. I. Nonpolar gases. J Chem Phys 22 (8):1420–1426
18. Widom B (1963) Some topics in the theory of fluids. J Chem Phys 39(11):2808–2812
19. Lu N, Singh JK, Kofke DA (2003) Appropriate methods to combine forward and reverse free-energy perturbation averages. J Chem Phys 118(7):2977–2984
20. Shirts MR, Pande VS (2005) Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. J Chem Phys 122(14):144107
21. Bennett CH (1976) Efficient estimation of free energy differences from Monte Carlo data. J Comput Phys 22(2):245–268
22. Shirts MR, Mobley DL, Brown SP (2010) Free-energy calculations in structure-based drug design. In: Merz KM, Ringe D, Reynolds CH (eds) Drug design. Cambridge University Press, Cambridge, pp 61–86
23. Shirts MR, Bair E, Hooker G, Pande VS (2003) Equilibrium free energies from

nonequilibrium measurements using maximum-likelihood methods. Phys Rev Lett 91(14):140601

24. Shirts MR, Chodera JD (2008) Statistically optimal analysis of samples from multiple equilibrium states. J Chem Phys 129:124105
25. Paliwal H, Shirts MR (2011) 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
26. Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E (2015) GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 1–2:19–25
27. Homeyer N, Gohlke H (2013) FEW: a workflow tool for free energy calculations of ligand binding. J Comput Chem 34(11):965–973
28. Liu P, Dehez F, Cai W, Chipot C (2012) A toolkit for the analysis of free-energy perturbation calculations. J Chem Theory Comput 8 (8):2606–2616
29. Pham TT, Shirts MR (2011) Identifying low variance pathways for free energy calculations of molecular transformations in solution phase. J Chem Phys 135(3):034114
30. Klimovich P, Shirts M, Mobley D (2015) Guidelines for the analysis of free energy calculations. J Comput Aided Mol Des 29 (5):397–411
31. Case DA, Betz RM, Cerutti DS, Cheatham TE III, Darden TA, Duke RE, Giese TJ, Gohlke H, Goetz AW, Homeyer N, Izadi S, Janowski P, Kaus J, Kovalenko A, Lee TS, LeGrand S, Li P, Lin C, Luchko T, Luo R, Madej BD, Mermelstein D, Merz KM, Monard G, Nguyen H, Nguyen HT, Omelyan I, Onufriev A, Roe DR, Roitberg A, Sagui C, Simmerling CL, Botello-Smith WM, Swails J, Walker RC, Wang J, Wolf RM, Wu X, Xiao L, Kollman PA (2016) AMBER 2016. University of California, San Francisco
32. Boresch S, Tettinger F, Leitgeb M, Karplus M (2003) Absolute binding free energies: a quantitative approach for their calculation. J Phys Chem B 107(35):9535–9551
33. Mobley DL, Chodera JD, Dill KA (2006) On the use of orientational restraints and symmetry corrections in alchemical free energy calculations. J Chem Phys 125(8):084902
34. Evoli S, Mobley DL, Guzzi R, Rizzuti B (2016) Multiple binding modes of ibuprofen in human serum albumin identified by absolute binding free energy calculations. Phys Chem Chem Phys 18(47):32358–32368
35. Cappel D, Hall ML, Lenselink EB, Beuming T, Qi J, Bradner J, Sherman W (2016) Relative binding free energy calculations applied to protein homology models. J Chem Inf Model 56 (12):2388–2400
36. Mobley DL, Graves AP, Chodera JD, McReynolds AC, Shoichet BK, Dill KA (2007) Predicting absolute ligand binding free energies to a simple model site. J Mol Biol 371 (4):1118–1134
37. Mobley DL (2012) Let's get honest about sampling. J Comput Aided Mol Des 26 (1):93–95
38. Mobley DL, Klimovich PV (2012) Perspective: alchemical free energy calculations for drug discovery. J Chem Phys 137(23):230901
39. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K (2005) Scalable molecular dynamics with NAMD. J Comput Chem 26:1781–1802
40. Hornak V, Abel R, Okur A, Strockbine B, Roitberg A, Simmerling C (2006) Comparison of multiple Amber force fields and development of improved protein backbone parameters. Proteins 65(3):712–725
41. Lindorff-Larsen K, Piana S, Palmo K, Maragakis P, Klepeis J, Dror R, Shaw D (2010) Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins 78:1950–1958
42. Maier JA, Martinez C, Kasavajhala K, Wickstrom L, Hauser KE, Simmerling C (2015) ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. J Chem Theory Comput 11 (8):3696–3713
43. MacKerell AD, Bashford D, Bellott M, Dunbrack RL, Evanseck JD, Field MJ, Fischer S, Gao J, Guo H, Ha S, Joseph-McCarthy D, Kuchnir L, Kuczera K, Lau FTK, Mattos C, Michnick S, Ngo T, Nguyen DT, Prodhom B, Reiher WE, Roux B, Schlenkrich M, Smith JC, Stote R, Straub J, Watanabe M, Wiórkiewicz-Kuczera J, Yin D, Karplus M (1998) All-atom empirical potential for molecular modeling and dynamics studies of proteins. J Phys Chem B 102(18):3586–3616
44. Mackerell AD (2004) Empirical force fields for biological macromolecules: overview and issues. J Comput Chem 25(13):1584–1604
45. Best RB, Zhu X, Shim J, Lopes PEM, Mittal J, Feig M, MacKerell AD (2012) Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone φ, ψ and side-chain χ(1) and

χ(2) dihedral angles. J Chem Theory Comput 8(9):3257–3273

46. Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA (2004) Development and testing of a general amber force field. J Comput Chem 25 (9):1157–1174
47. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, Darian E, Guvench O, Lopes P, Vorobyov I, Mackerell Jr AD (2010) CHARMM general force field: a force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem 31 (4):671–690
48. Wang J, Wang W, Kollman PA, Case DA (2006) Automatic atom type and bond type perception in molecular mechanical calculations. J Mol Graph Model 25(2):247–260
49. Vanommeslaeghe K, MacKerell AD (2012) Automation of the charmm general force field (CGenFF) I: bond perception and atom typing. J Chem Inf Model 52(12):3144–3154
50. Vanommeslaeghe K, Raman EP, MacKerell AD (2012) Automation of the CHARMM general force field (CGenFF) II: assignment of bonded parameters and partial atomic charges. J Chem Inf Model 52(12):3155–3168
51. Huang L, Roux B (2013) Automated force field parameterization for nonpolarizable and polarizable atomic models based on ab initio target data. J Chem Theory Comput 9 (8):3543–3556
52. Betz RM, Walker RC (2015) Paramfit: automated optimization of force field parameters for molecular dynamics simulations. J Comput Chem 36(2):79–87
53. Shenfeld DK, Xu H, Eastwood MP, Dror RO, Shaw DE (2009) Minimizing thermodynamic length to select intermediate states for free-energy calculations and replica-exchange simulations. Phys Rev E Stat Nonlin Soft Matter Phys 80(4):046705
54. Boyce SE, Mobley DL, Rocklin GJ, Graves AP, Dill KA, Shoichet BK (2009) Predicting ligand binding affinity with alchemical free energy methods in a polar model binding site. J Mol Biol 394(4):747–763
55. Beutler TC, Mark AE, van Schaik RC, Gerber PR, van Gunsteren WF (1994) Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. Chem Phys Lett 222:529–539
56. Steinbrecher T, Mobley DL, Case DA (2007) Nonlinear scaling schemes for Lennard-Jones interactions in free energy calculations. J Chem Phys 127(21):214108
57. Zacharias M, Straatsma TP, McCammon JA (1994) Separation-shifted scaling, a new scaling method for Lennard-Jones interactions in thermodynamic integration. J Chem Phys 100 (12):9025–9031
58. Gapsys V, Seeliger D, de Groot BL (2012) New soft-core potential function for molecular dynamics based alchemical free energy calculations. J Chem Theory Comput 8 (7):2373–2382
59. Pitera JW, van Gunsteren WF (2002) A comparison of non-bonded scaling approaches for free energy calculations. Mol Simul 28 (1–2):45–65
60. Anwar J, Heyes DM (2005) Robust and accurate method for free-energy calculation of charged molecular systems. J Chem Phys 122 (22):224117
61. Steinbrecher T, Joung I, Case DA (2011) Soft-core potentials in thermodynamic integration: comparing one- and two-step transformations. J Comput Chem 32(15):3253–3263
62. Rocklin GJ, Mobley DL, Dill KA, Hünenberger PH (2013) Calculating the binding free energies of charged species based on explicit-solvent simulations employing lattice-sum methods: an accurate correction scheme for electrostatic finite-size effects. J Chem Phys 139:184103
63. Hub JS, de Groot BL, Grubmüller H, Groenhof G (2014) Quantifying artifacts in ewald simulations of inhomogeneous systems with a net charge. J Chem Theory Comput 10 (1):381–390
64. Rocklin GJ, Boyce SE, Fischer M, Fish I, Mobley DL, Shoichet BK, Dill KA (2013) Blind prediction of charged ligand binding affinities in a model binding site. J Mol Biol 425 (22):4569–4583
65. Shirts MR (2013) Simple quantitative tests to validate sampling from thermodynamic ensembles. J Chem Theory Comput 9(2):909–926
66. Goga N, Rzepiela AJ, de Vries AH, Marrink SJ, Berendsen HJC (2012) Efficient algorithms for langevin and DPD dynamics. J Chem Theory Comput 8:3637–3649
67. Andersen HC (1980) Molecular dynamics simulations at constant pressure and/or temperature. J Chem Phys 72(4):2384–2393
68. Schneider T, Stoll E (1978) Molecular-dynamics study of a three-dimensional one--component model for distortive phase transitions. Phys Rev B 17(3):1302–1322
69. Shirts MR, Pitera JW, Swope WC, Pande VS (2003) 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

70. Kelly E, Seth M, Ziegler T (2004) Calculation of free energy profiles for elementary bimolecular reactions by ab initio molecular dynamics: sampling methods and thermostat considerations. J Phys Chem A 108(12):2167–2180
71. Hess B, van der Vegt NFA (2006) Hydration thermodynamic properties of amino acid analogues: a systematic comparison of biomolecular force fields and water models. J Phys Chem B 110(35):17616–17626
72. Wang J, Deng Y, Roux B (2006) Absolute binding free energy calculations using molecular dynamics simulations with restraining potentials. Biophys J 91(8):2798–2814
73. Bussi G, Parrinello M (2008) Stochastic thermostats: comparison of local and global schemes. Comput Phys Commun 179 (1–3):26–29
74. Berendsen HJC, Postma JPM, van Gunsteren WF, DiNola A, Haak JR (1984) Molecular dynamics with coupling to an external bath. J Chem Phys 81:3684–3690
75. Parinello M, Rahman A (1981) Polymorphic transitions in single crystals – a new molecular dynamics method. J Appl Phys 52:7182–7190
76. Chodera JD, Shirts MR (2011) Replica exchange and expanded ensemble simulations as Gibbs sampling: simple improvements for enhanced mixing. J Chem Phys 2011 (135):194110
77. Bussi G (2014) Hamiltonian replica exchange in GROMACS: a flexible implementation. Mol Phys 112(3–4):379–384
78. Faraldo-Gómez JD, Roux B (2007) Characterization of conformational equilibria through Hamiltonian and temperature replica-exchange simulations: assessing entropic and environmental effects. J Comput Chem 28 (10):1634–1647
79. Wang K, Chodera JD, Yang Y, Shirts MR (2013) Identifying ligand binding sites and poses using GPU-accelerated Hamiltonian replica exchange molecular dynamics. J Comput Aided Mol Des 27:989–1007
80. Woods CJ, Essex JW, King MA (2003) Enhanced configurational sampling in binding free-energy calculations. J Phys Chem B 107 (49):13711–13718
81. Bowers KJ, Chow E, Xu H, Dror RO, Eastwood MP, Gregersen BA, Klepeis JL, Kolossvary I, Moraes MA, Sacerdoti FD, Salmon JK, Shan Y, Shaw DE (2006) Scalable algorithms for molecular dynamics simulations on commodity clusters. In: Proceedings of the 2006 ACMI/IEEE conference on supercomputing. ACM Press, New York
82. Chodera JD, Swope WC, Pitera JW, Seok C, Dill KA (2007) Use of the weighted histogram analysis method for the analysis of simulated and parallel tempering simulations. J Chem Theory Comput 3:26–41
83. Chodera JD (2016) A simple method for automated equilibration detection in molecular simulations. J Chem Theory Comput 12 (4):1799–1805
84. Shirts MR, Mobley DL, Chodera JD, Pande VS (2007) Accurate and efficient corrections for missing dispersion interactions in molecular simulations. J Phys Chem B 111 (45):13052–13063
85. Wennberg CL, Murtola T, Páll S, Abraham MJ, Hess B, Lindahl E (2015) Direct-space corrections enable fast and accurate lorentz–berthelot combination rule Lennard-Jones lattice summation. J Chem Theory Comput 11 (12):5737–5746
86. Lim NM, Wang L, Abel R, Mobley DL (2016) Sensitivity in binding free energies due to protein reorganization. J Chem Theory Comput 12(9):4620–4631
87. Mobley DL, Chodera JD, Dill KA (2007) Confine-and-release method: obtaining correct binding free energies in the presence of protein conformational change. J Chem Theory Comput 3:1231–1235
88. Lin Y-L, Meng Y, Jiang W, Roux B (2013) Explaining why Gleevec is a specific and potent inhibitor of Abl kinase. Proc Natl Acad Sci U S A 110(5):1664–1669
89. Li H, Fajer M, Yang W (2007) Simulated scaling method for localized enhanced sampling and simultaneous "alchemical" free energy simulations: a general method for molecular mechanical, quantum mechanical, and quantum mechanical/molecular mechanical simulations. J Chem Phys 126(2):024106
90. Wang L, Berne BJ, Friesner RA (2012) On achieving high accuracy and reliability in the calculation of relative protein–ligand binding affinities. Proc Natl Acad Sci U S A 109 (6):1937–1942
91. Wang L, Deng Y, Knight JL, Wu Y, Kim B, Sherman W, Shelley JC, Lin T, Abel R (2013) Modeling local structural rearrangements using FEP/REST: application to relative binding affinity predictions of CDK2 inhibitors. J Chem Theory Comput 9(2):1282–1293
92. Ross GA, Bodnarchuk MS, Essex JW (2015) Water sites, networks, and free energies with Grand Canonical Monte Carlo. J Am Chem Soc 137(47):14930–14943

93. Shirts MR, Pande VS (2005) Solvation free energies of amino acid side chain analogs for common molecular mechanics water models. J Chem Phys 122(13):134508
94. Mobley DL, Dumont ML, Chodera JD, Dill KA (2007) Comparison of charge models for fixed-charged force-fields: small molecule hydration free energies in explicit solvent. J Phys Chem B 111:2242–2254
95. Hünenberger PH, McCammon JA (1999) Effect of artificial periodicity in simulations of biomolecules under Ewald boundary conditions: a continuum electrostatics study. Biophys Chem 78:69–88
96. Lin Y-L, Aleksandrov A, Simonson T, Roux B (2014) An overview of electrostatic free energy computations for solutions and proteins. J Chem Theory Comput 10(7):2690–2709