Chapter 2

Accurate Calculation of Free Energy Changes upon Amino Acid Mutation

Matteo Aldeghi, Bert L. de Groot, and Vytautas Gapsys

Abstract

Molecular dynamics based free energy calculations allow for a robust and accurate evaluation of free energy changes upon amino acid mutation in proteins. In this chapter we cover the basic theoretical concepts important for the use of calculations utilizing the non-equilibrium alchemical switching methodology. We further provide a detailed step-by-step protocol for estimating the effect of a single amino acid mutation on protein thermostability. In addition, the potential caveats and solutions to some frequently encountered issues concerning the non-equilibrium alchemical free energy calculations are discussed. The protocol comprises details for the hybrid structure/topology generation required for alchemical transitions, equilibrium simulation setup, and description of the fast non-equilibrium switching. Subsequently, the analysis of the obtained results is described. The steps in the protocol are complemented with an illustrative practical application: a destabilizing mutation in the Trp cage mini protein. The concepts that are described are generally applicable. The shown example makes use of the `pmx` software package for the free energy calculations using Gromacs as a molecular dynamics engine. Finally, we discuss how the current protocol can readily be adapted to carry out charge-changing or multiple mutations at once, as well as large-scale mutational scans.

Key words Molecular dynamics, free energy calculations, alchemistry, amino acid mutation, pmx, hybrid structure, hybrid topology, non-equilibrium transitions

1 Introduction

Due to the central role of the free energy in thermodynamics and kinetics, the accurate prediction of free energy changes upon amino acid mutation is one of the central goals in computer-aided molecular design, with potential applications ranging from the engineering of thermostable proteins [1] to that of biosensors [2, 3], sequestrants [4], and protein–protein interactions [5–7]. Predicting mutation effects allows understanding the causes of drug resistance

Electronic supplementary material: The online version of this chapter (https://doi.org/10.1007/978-1-4939-8736-8_2) contains supplementary material, which is available to authorized users.

Tobias Sikosek (ed.), *Computational Methods in Protein Evolution*, Methods in Molecular Biology, vol. 1851,
https://doi.org/10.1007/978-1-4939-8736-8_2,

[8, 9]. Engineered stable proteins with high affinity and specificity toward their binding targets may also serve as biopharmaceuticals [10, 11]. Accurate and robust estimation of the free energy differences between protein sequence variants, thus, is crucial to the successful design of proteins with the desired thermodynamic features.

Different approaches have thus been developed that can return an estimate of free energy changes that relate to the different stabilities or binding affinities of wild-type and mutant proteins. These include fast scoring methods [12–16], implicit-solvent approaches based on the post-processing of molecular dynamics (MD) simulations [17–19], and the computationally more expensive but theoretically rigorous (from a statistical mechanics viewpoint) *alchemical* free energy methods [1, 20]. In this chapter, we focus on the latter category of calculations, which are based on all-atom computer simulations that correctly sample the Boltzmann distribution of microstates and inherently take into account entropic and discrete solvent effects.

In alchemical free energy calculations, an amino acid can be transformed into another one via a non-physical path, hence the name that is reminiscent of the ancient practice that aimed at the transmutation of lead into gold. The amino acid transformation can be carried out reversibly, in what are referred to as *equilibrium* free energy calculations, or irreversibly, in *non-equilibrium* calculations [21]. In both cases, the amount of work needed for the transformation and free energy difference between the initial and final states can be recovered. However, the setup of the calculations differs. In this chapter, we discuss non-equilibrium approaches that carry out this transformation irreversibly and describe protocols that can be used for the accurate estimation of free energy changes upon amino acid mutation. In the text, we use the prediction of protein stability changes upon an amino acid mutation as an example application. The methodology and protocol presented here are of generic character and can be applied to study other biophysical processes, assuming a suitable thermodynamic cycle can be built, e.g., changes in protein–protein, protein–DNA, or protein–ligand binding affinities.

In this chapter, we first provide some background concepts that are at the foundation of the non-equilibrium alchemical free energy method; for a more detailed description we give references to more specialized literature sources. Further, we concentrate on the description of the practical steps involved in preparing and subsequently carrying out the free energy calculations following a general protocol. As an example, we use a Trp cage mini protein [22] that provides a real case on which we illustrate setting up and running alchemical free energy calculations of protein mutation. We assume the reader is familiar with the general principles of molecular dynamics simulations. Throughout this chapter, we

discuss the potential caveats and solutions for some of the frequently encountered issues. In the last section of the chapter, we describe how the protocol can be easily modified and expanded to perform large-scale mutational scans or to calculate other free energy changes of interest, such as changes in protein–protein or protein–ligand affinities upon protein mutation. Finally, in the Notes section, we provide a few technical remarks that may prove helpful when setting up alchemical free energy calculations using Gromacs 2016 [23] and the `pmx` python library with the specialized set of scripts [24].

2 Theory

In this section, we briefly review some of the central concepts that allow the estimation of free energy differences from physics-based computer simulations, like Monte Carlo or molecular dynamics (MD) simulations. We place particular focus on the theoretical foundations of non-equilibrium work (NEW) calculations and how they can be used to estimate free energy differences along alchemical (i.e., non-physical) paths. The interested reader can find a broader appraisal of theoretical aspects, also including equilibrium free energy calculations and geometrical transformations, in the numerous excellent reviews that have been written on the subject, [21, 25–29] as well as in the publications by Jarzinski [30, 31], Crooks [32–34], and Hummer [35–37].

2.1 Definition of Free Energy and Irreversible Work

The free energy surface of a system determines its thermodynamic and kinetic properties and, as such, it provides access to understanding biophysical processes, including protein folding, ligand binding, protein–protein association, etc. For instance, a polypeptide chain in solution may be found in many disordered conformations, or in ordered conformations with well-defined secondary and tertiary structure. We can define the set of disordered conformations as the unfolded state of the system (state A), and the set of ordered conformations as the folded state (state B). It is rarely possible to sample the whole phase space of a protein, which would require observing all the folded and unfolded conformations multiple times. However, in practice free energy differences rather than free energies are typically of interest. The difference between the free energy of state A and B alone will give the relative equilibrium probability of finding the protein in its unfolded form with respect to the folded form; i.e., the free energy difference ΔG is proportional to the ratio of probabilities of finding the system in state A or B:

$$\frac{p_A}{p_B} = \frac{e^{-\beta G_A}}{e^{-\beta G_B}} = e^{-\beta(G_A - G_B)} \tag{1}$$

$$\Delta G = G_A - G_B = -k_{\mathrm{B}} T \ln \frac{p_A}{p_B} \tag{2}$$

where G is the free energy of the whole phase space of the system for an ensemble with a fixed number of particles, constant pressure and temperature (T), i.e., isothermal–isobaric conditions. G_A is the free energy of the unfolded state, G_B is the free energy of the folded state, and $\beta = 1/k_{\mathrm{B}}T$, with k_{B} is the Boltzmann constant with T denoting the absolute temperature.

This free energy difference also determines the maximum amount of work that can be extracted from the closed system during a thermodynamic process, which can only be achieved in the limit of reversibility. During a reversible process, the system is always in thermodynamic equilibrium, which implies that only infinitesimal changes are applied to it and the transformation is infinitely slow. However, for any finite time interval τ, the system will be driven out of equilibrium, resulting in heat dissipation and hysteresis effects, so that the process will be irreversible. In fact, in accordance to the second law of thermodynamics, the work done during a process is on average equal or larger, due to dissipative work, than the free energy difference between the initial and final state:

$$\langle W(\tau) \rangle \geq \Delta G \tag{3}$$

The equality holds only in the limiting case of a reversible process where ($\tau \to \infty$), whereas for finite τ, the difference between $\langle W(\tau) \rangle$ and ΔG is caused by dissipative work and its magnitude will also depend on the chosen thermodynamic path.

If we use a parameter λ to drive a non-equilibrium process along a certain path, such that the process is started at $\lambda = 0$ and it is concluded at $\lambda = 1$, with λ being constantly modified at each time step, one can calculate the work performed on the system by integrating the energetic cost required to modify it:

$$W(\tau) = \int_{\lambda=0}^{\lambda=1} \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{v}, \lambda)}{\partial \lambda} d\lambda \tag{4}$$

where $\mathcal{H}$ is the Hamiltonian of the system, which depends on the phase space coordinates $\boldsymbol{x}$ and velocities $\boldsymbol{v}$ of the system and the coupling parameter λ.

2.2 Estimating Free Energy Differences from Non-equilibrium Simulations

From the considerations above, it is possible to derive estimators that allow calculating free energy differences from equilibrium and non-equilibrium simulations. Both, the Zwanzig's formula [38], which lies at the basis of free energy perturbation (FEP) approaches, and thermodynamic integration (TI) [39] make use of ensemble averages obtained from equilibrium simulations for the

estimation of free energy differences. More recently, Jarzynski has shown how one can derive an identity from the inequality in Eq. 3, such that a free energy difference can also be obtained from an ensemble of non-equilibrium simulations in which the system is driven irreversibly from one state to another [30, 31]. In fact, it is possible to show that both FEP and TI are limit cases of Jarzynski's equality, in which the non-equilibrium transformation is performed instantaneously (infinitely fast: $\tau \to 0$) or reversibly (infinitely slowly: $\tau \to \infty$), respectively [21]. The Crooks Fluctuation Theorem (CFT) [32–34] has further generalized the Jarzynski's equality by relating the equilibrium free energy difference to the ratio of non-equilibrium work distributions collected by performing the process in the forward and reverse directions. In the following, we focus on non-equilibrium work (NEW) approaches. More specifically, we review the free energy estimators based on the Jarzynski's equality and Crooks Fluctuation Theorem (Crooks Gaussian Intersection and Bennet's Acceptance Ratio).

2.2.1 Jarzynski's Equality

The equality derived by Jarzynski in 1997 [30, 40] relates the uni-directional non-equilibrium work average to the equilibrium free energy difference:

$$\left\langle e^{-\beta W(\tau)} \right\rangle = e^{-\beta \Delta G} \tag{5}$$

The work W depends on the chosen path connecting the initial ($\lambda = 0$) and final ($\lambda = 1$) states. The parameter λ controls the time evolution of a system with a time-dependent Hamiltonian. The average on the left-hand side of the equation is an ensemble over both equilibrium initial conditions and non-equilibrium transformations. In fact, the above equality requires the non-equilibrium transitions to be started from an equilibrium ensemble; on the other hand, there is no such requirement for the final state of the system at the end of the transition [21, 30]. The non-equilibrium trajectories are then weighted with the Boltzmann factor of the external work done on the system. The work W can be calculated from Eq. 4 by numerical integration; note how instantaneous, rather than ensemble average (as done in TI), $\partial \mathcal{H}/\partial \lambda$ values are evaluated. In the limit of an infinitely fast ($\tau \to 0$) or slow ($\tau \to \infty$) transformation, Eq. 5 reduces to the Zwanzig equation and TI, respectively [21]. In fact, if the system is brought from $\lambda = 0$ to $\lambda = 1$ instantaneously, its configurations at both end states are the same and W simply corresponds to the change in Hamiltonian (which, for transformations that conserve the kinetic energy of the system, corresponds to the change in potential energy). On the other hand, for an infinitely slow transformation, the system is always in equilibrium so that $\langle W \rangle = \Delta G$.

From Eq. 5 one can directly estimate the free energy difference as follows, with N being the number of non-equilibrium trajectories sampled:

$$\widehat{\Delta G} = -k_{\mathrm{B}} T \ln \left[\frac{1}{N} \sum_{i}^{N} e^{-\beta W_i} \right] \tag{6}$$

However, in practice, this exponential estimator is affected by statistical and systematic errors. In fact, due to the exponential weight, the average will mostly depend on values at the tail of the work distribution. This means that rare events where little work is dissipated will dominate the estimate; consequently, the free energy will converge slowly to the true value given that rare events are most likely poorly sampled. Furthermore, it has been shown that this estimator is biased [41, 42], i.e., it introduces a systematic error in the free energy estimate for finite numbers of N.

2.2.2 Crooks Fluctuation Theorem

Jarzynski's equality considers the transitions in one direction only, e.g., from $\lambda = 0$ to $\lambda = 1$. The Crooks Fluctuation Theorem (CFT) takes into account the work values obtained from performing the process in both forward (λ: $0 \rightarrow 1$) and reverse (λ: $1 \rightarrow 0$) directions. According to the CFT, the forward and reverse work distributions relate to the free energy difference as follows:

$$\frac{P_f(W)}{P_r(-W)} = e^{\beta(W - \Delta G)} \tag{7}$$

where $P_f(W)$ and $P_r(-W)$ are the normalized probability distributions of work values obtained from the forward and reverse transformation paths. Note that Jarzynski's equality can be derived from Eq. 7 by integration over W [21]. With enough overlap between the forward and reverse work distributions, the free energy difference can be estimated directly from Eq. 7 as follows:

$$\widehat{\Delta G} = W + k_{\mathrm{B}} T \ln \frac{P_f(W)}{P_r(-W)} \tag{8}$$

with $\widehat{\Delta G} = W$ at the intersection of the work distributions. However, this approach has known limitations: firstly, for certain paths it might be difficult to obtain substantial overlap between $P_f(W)$ and $P_r(-W)$. Secondly, mainly the tails of the distributions, which are defined by rare events of low work dissipation, will contribute to the free energy difference.

To partly alleviate these problems, one can approximate the work distributions with an analytical function [43]. One such strategy, which leads to accurate free energy estimates, was proposed by Goette and Grubmüller [29]. By using a Gaussian approximation, a Crooks Gaussian Intersection (CGI) estimator was derived:

$$\widehat{\Delta G} = \frac{\frac{\langle W_f \rangle}{\sigma_f^2} - \frac{\langle W_r \rangle}{\sigma_r^2} \pm \sqrt{\frac{1}{\sigma_f^2 \sigma_r^2}\left(\langle W_f \rangle + \langle W_f \rangle\right)^2 + 2\left(\frac{1}{\sigma_f^2} - \frac{1}{\sigma_r^2}\right)\ln\frac{\sigma_r}{\sigma_f}}}{\frac{1}{\sigma_f^2} - \frac{1}{\sigma_r^2}} \quad (9)$$

where σ_f and σ_r are the variances of the forward and reverse work distributions. Note that the accuracy of this estimator relies on the Gaussian approximation. Thus, it is advised to check this assumption by, for instance, using a statistical test like the Kolmogorov–Smirnov test [44]. The CGI estimator does not have an analytical error estimate, but the error can be estimated by the bootstrap approach [45].

Another ΔG estimator, termed BAR (Bennet's Acceptance Ratio), does not require an analytical approximation for the work distributions. Originally, the BAR relation was derived in 1976 by Bennet for a system sampling two states at equilibrium and performing instantaneous transformations between the states. Bennet showed that the information from the forward and reverse distributions of the potential energy difference (ΔU) could be combined in order to obtain an optimal estimate of the free energy difference [46]. For a non-equilibrium process carried out during a finite amount of time, the same derivation holds by substituting ΔU with the non-equilibrium work W. In 2003, Shirts and coworkers showed how the same estimator can be derived starting from the Crooks Fluctuation Theorem using maximum-likelihood arguments [47]. The BAR estimates the free energy difference by satisfying the following relation:

$$\sum_{i=1}^{N_f} \frac{1}{1 + \frac{N_f}{N_r} e^{\beta(W_i - \widehat{\Delta G})}} = \sum_{j=1}^{N_r} \frac{1}{1 + \frac{N_r}{N_f} e^{-\beta(W_j - \widehat{\Delta G})}} \quad (10)$$

where N_f and N_r are the number of forward and reverse trajectories. The BAR equation needs to be solved numerically, for instance, by using a Newton–Raphson or Nelder–Mead solver [48]. This estimator is asymptotically unbiased and an analytical expression for its variance is available [47]. Furthermore, a convergence criterion for BAR has been proposed [49].

The main assumption in the BAR derivation is that the work values are statistically independent [46, 47]. It is thus important to bear this in mind, because if initial configurations are selected from an equilibrium simulation with high frequency, the resulting work values may be correlated [21].

2.3 Free Energy Differences Upon Protein Mutation: The Alchemical Path

To calculate a free energy difference, firstly we need to define the initial and final states of interest, and secondly the path connecting them. If we consider the folding example already used, then the initial state would be the unfolded protein and the final state would be the folded protein, with the free energy difference we want to calculate being the protein folding free energy. If the structure of

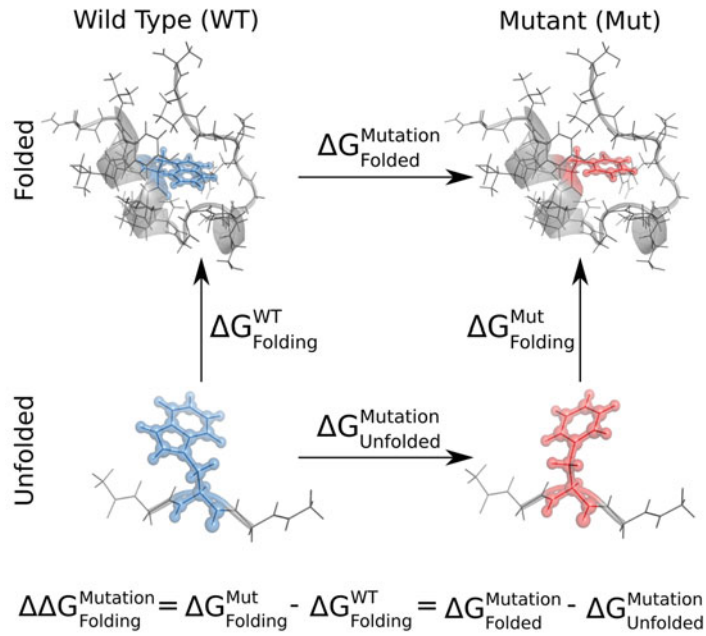

Fig. 1 Schematic representation of a thermodynamic cycle to calculate changes in protein folding free energy upon mutation ($\Delta\Delta G^{Mutation}_{Folding}$). The left column shows the folding process of a wild-type protein, with the associated folding free energy $\Delta G^{WT}_{Folding}$; the right column shows the same folding reaction but for a mutated protein, resulting in the folding free energy $\Delta G^{Mut}_{Folding}$. The process depicted in the bottom row corresponds to the alchemical transformation of the wild-type unfolded protein into the mutant with the associated free energy difference $\Delta G^{Mutation}_{Unfolded}$. The reaction in the top row corresponds to the same alchemical transformation but done on the folded protein, so that the free energy difference between the two mutants is $\Delta G^{Mutation}_{Folded}$. The free energy differences for the vertical processes are computationally demanding to compute, but those for the horizontal transformations are more accessible. Thus, $\Delta\Delta G^{Mutation}_{Folding}$ can be calculated from the difference between $\Delta G^{Mutation}_{Folded}$ and $\Delta G^{Mutation}_{Unfolded}$

the folded protein is known, we can then transform state B into state A (via a reversible or irreversible process) by, for instance, pulling the N- and C-termini apart and measuring the work needed to unfold the protein. Although this is in principle possible, such a large perturbation of the system will likely require a lot of computation in order to achieve convergence. However, if the interest is in evaluating *changes* in folding free energy upon protein mutation ($\Delta\Delta G_{\text{folding}}$), it is possible to build a thermodynamic cycle (Fig. 1) that allows to calculate this quantity via alchemical (i.e., non-physical) paths that introduce smaller perturbations in the system, and which are easier to converge. Thus, thanks to the fact that computationally we have control over the topology and potential energy function describing the system, we can take full advantage of the better convergence properties of the non-physical transformation over physical ones; i.e., it is easier to obtain accurate

results by alchemically mutating a wild-type protein into its mutant, rather than (un)folding both of them. As the free energy is a state variable, obtained free energy changes are path-independent. It is therefore unproblematic to choose unphysical pathways.

2.3.1 The Thermodynamic Cycle

As shown in Fig. 1, one can define a cycle where for both the initial (unfolded) and final (folded) states the wild-type protein is transformed into a mutant of interest via a non-physical path. The free energy difference of protein folding upon an amino acid mutation ($\Delta\Delta G^{\text{Mutation}}_{\text{Folding}}$) can be recovered by following both, the physical paths of folding the WT and mutant protein ($\Delta G^{\text{Mut}}_{\text{Folding}} - \Delta G^{\text{WT}}_{\text{Folding}}$), and the alchemical paths of morphing the amino acids in the folded and unfolded states ($\Delta G^{\text{Mutation}}_{\text{Folded}} - \Delta G^{\text{Mutation}}_{\text{Unfolded}}$).

From the thermodynamic cycle in Fig. 1 it is clear that in order to calculate $\Delta\Delta G^{\text{Mutation}}_{\text{Folding}}$ we need to be able to simulate the protein's unfolded state. However, the unfolded state of the full-length protein is by its nature poorly defined and would be challenging to simulate [50, 51]. Therefore, short protein fragments have been typically used [52–54]. In particular, it has been observed that capped sequence context independent tripeptides (GXG, where X is the mutated residue) serve as a good approximation of the unfolded state for estimating changes in protein thermostability [20]. In practice, the context independent are convenient to use, as they allow to systematically precompute all possible residue mutations. In such a way, one only needs to calculate $\Delta G^{\text{Mutation}}_{\text{Folded}}$, while $\Delta G^{\text{Mutation}}_{\text{Unfolded}}$ can be found in a precomputed table.

Although here we take protein folding as an example, the same alchemical approach can easily be used to build other thermodynamic cycles by changing the end states; for instance, differences in ligand–protein, protein–protein, or protein–DNA/RNA binding free energy can be calculated by using the apo protein as the initial state and the complex as the final one. Note that while the $\Delta G_{\text{Mutation}}$ values refer to non-physical transformations, the final $\Delta\Delta G$ value obtained from such cycles is that of a physical process (e.g., folding, association, etc.) and can be directly compared to experimental values that measure the same free energy differences.

2.3.2 Single and Dual Topology

We have described how alchemical transformations can be used to build thermodynamic cycles that allow one to calculate changes in free energy differences upon an amino acid mutation. However, how can one alchemically mutate one residue into another during a simulation? Given the separate Hamiltonians at the two end states, it is necessary to define a hybrid topology that contains both physical states. In the specific case of mutating an amino acid into another one, the residue being mutated must be able to represent both the wild-type and mutant residue. This is typically achieved using the single or dual topology approach [55–57].

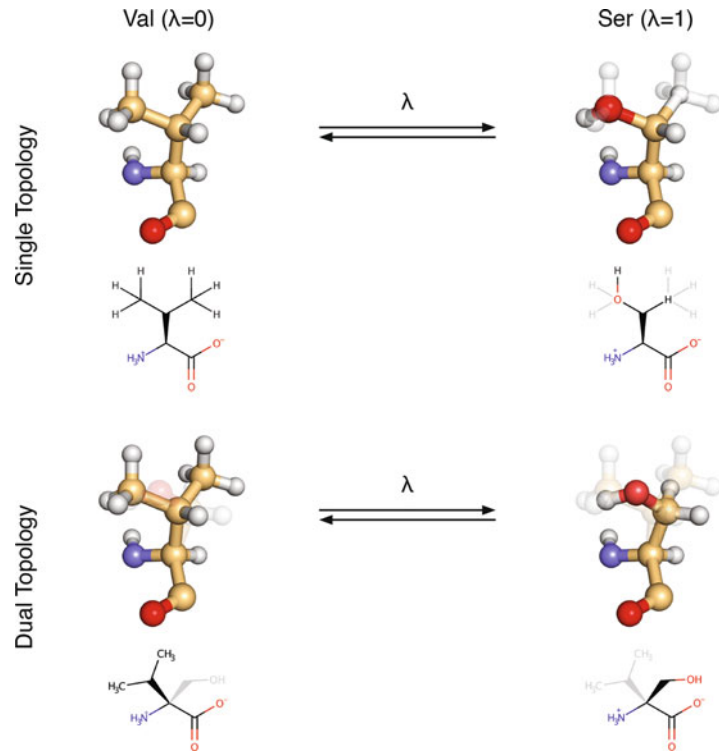

Fig. 2 Example of the single and dual topology setup for the mutation of valine into serine. Dummy atoms in the three-dimensional rendering are shown as transparent balls and sticks, whereas in the chemical structure drawings they are shown in grey. In the single topology approach, a methyl part of valine's side chain is transformed into serine's hydroxyl group, with a carbon becoming an oxygen, while two hydrogens are turned into non-interacting dummy particles; all hydrogens of the second methyl are decoupled as well, while the carbon becomes a C_β hydrogen. In the dual topology approach, no element mutation occurs, because both valine and serine side chains are present in both states, where, however, only one of the two is coupled to the system, with the other one being non-interacting

In the single topology approach (Fig. 2), a number of atoms of state *A* is mapped onto the atoms of state *B*. This means that not only a particle's partial charge, but also its chemical element (i.e., atom type) can change according to the λ parameter. For instance, in the example shown in Fig. 2 for a valine being mutated into a serine, one of valine's carbon atoms at $\lambda = 0$ becomes a serine's oxygen at $\lambda = 1$. Effectively, this means that along the alchemical path controlled by λ, the Lennard-Jones and bonded parameters for that particle are interpolated between those of a carbon atom and those of an oxygen atom. Note that such change in chemical identity implies that also the associated equilibrium bond lengths will be modified (e.g., a C–H bond will be shorter than a C–C bond) [55, 58, 59]. Often, the number of atoms in the two end

states is not equal, thus not all atoms of the states A and B can be matched. Therefore, non-interacting particles are used either in state A or B. These dummy atoms do not have electrostatic and van der Waals (vdW) interactions with the system; however, they maintain their bonded interactions, so that they effectively are in a vacuum-like state. In the example in Fig. 2, five of valine's hydrogen atoms are turned into dummy atoms.

In the dual topology approach, atoms that are different between the two end states are not morphed directly, but rather transformed into dummy particles [26, 56, 57]. For amino acids, this effectively means that the side chains of both residues are present at the same time. However, at $\lambda = 0$ the side chain of the initial state is interacting with the system and the side chain of the final state is present as non-interacting particles. On the other hand, at $\lambda = 1$ the side chain of the final state is interacting and that of the initial state is turned into non-interacting dummy atoms. This can be seen in Fig. 2: in the initial state, the methanol side chain of serine is decoupled, whereas in the final state it is the propyl side chain of valine being turned off.

In practice, there does not need to be a clear separation between a single and dual topology setup. While some atoms may be morphed between the states following a single topology approach, other atoms in the same system may be turned into dummies according to a dual topology approach.

It is important to bear in mind that the free energy change (ΔG) of the mutation differs depending on whether the single or dual topology approach is used. This is due to the fact that the end states are effectively different due to different dummy atom constructions. In addition, in the single topology approach there is a contribution to the free energy difference from the change in bond lengths. However, the contributions to the free energy difference resulting from the details of the atom mapping between the end states cancel out in a thermodynamic cycle like the one in Fig. 1, such that the final $\Delta\Delta G$ value is independent of how the hybrid topology is implemented [57, 59].

Using dummy particles in alchemical transitions requires introduction and annihilation of particles into the system. Such transformations impose a large perturbation, e.g., creating a particle interacting with the environment in a place of a non-interacting dummy atom results in strong van der Waals repulsions and Coulombic interactions. In turn, large forces are exerted on the atoms which leads to instabilities in dynamics and integration artifacts. To circumvent these issues, it is a common practice to modify, "soften," the non-bonded interactions during the alchemical transformations. A number of functional forms and parameter sets to such soft-cored interactions have been proposed [60–64]. Altering the non-bonded interactions along the alchemical pathway does not affect the final free energy estimates, because the physical end

states are still described by the correct unmodified Hamiltonian. The official release of Gromacs 2016 implements a soft-core variant [60] allowing to modify both the van der Waals and Coulombic interactions (*see* **Note 1**).

3 Alchemical Amino Acid Mutations

In this section we use a Trp cage mini protein [22] as a model system to illustrate the process of performing a single amino acid mutation. The `pmx` [20, 26] software package will be used to introduce a point mutation in this 20 amino acid peptide. `pmx` provides a single topology-based setup of the alchemical calculations allowing for an automated generation of hybrid amino acid structures and topologies compatible with the Gromacs [23] MD simulation engine.

In this example we will describe in detail the steps needed to prepare the alchemical simulations (Fig. 3) and calculate the free energy difference upon a tryptophan, W6, to phenylalanine mutation (W6F) in the Trp cage protein. W6 is the key residue in the hydrophobic core of this mini protein providing stability to its fold.

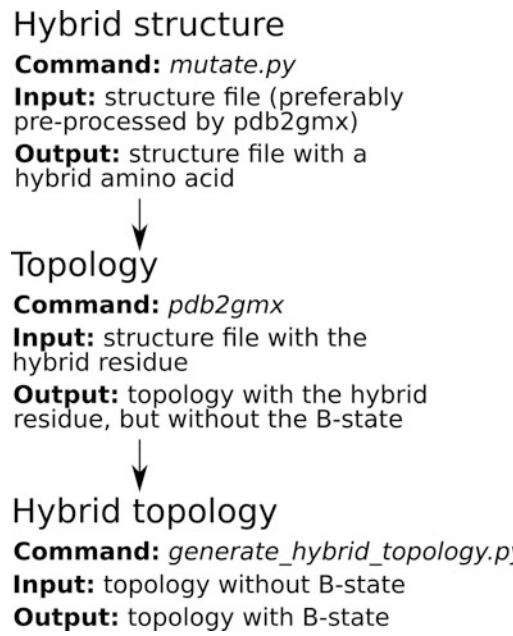

Fig. 3 A schematic depiction of the main steps in generating hybrid structures and topologies for the alchemical simulations using `pmx`. Firstly, `pmx` is used to introduce a mutation into the protein. Afterwards, the Gromacs tool *pdb2gmx* generates a topology for the protein with the hybrid residue in a user chosen molecular mechanics force field. In the last step, `pmx` is used again to add the *B*-state parameters to the topology file

We will assess the change in the thermodynamic stability by calculating the double free energy difference ($\Delta\Delta G$) for the W6F mutation in the folded Trp cage and its unfolded variant approximated by a capped tripeptide (Fig. 1).

For the results of more alchemical mutations in the Trp cage protein, *see* [65].

3.1 Setting Up pmx

pmx is a python library that allows the convenient manipulation of biomolecular structure and topology files. Within the framework of pmx, a number of scripts have been developed and specifically designed to prepare and analyze alchemical free energy calculations. pmx generates topology files that are compatible with the Gromacs simulation engine.

Mutations in a number of contemporary molecular mechanics force fields are supported. This is achieved by means of pre-generated mutation libraries compatible with the Gromacs force field organization. After installing Gromacs and pmx, the GMXLIB environmental variable needs to be set to specify the path to the mutation libraries that come with the pmx package (*see* **Note 2**).

3.2 Hybrid Structure

The first step in the setup comprises the generation of the hybrid structure for the amino acid to be mutated (Fig. 3). The only file required for this step is the protein structure in .pdb or .gro format. The protein structure needs to be complete, i.e. all heavy and hydrogen atoms need to be present. In order to add missing heavy atoms, external software needs to be used, e.g., Rosetta [15], Modeller [66], or PyMol [67]. Furthermore, given that structures resolved by means of X-ray crystallography usually contain no hydrogen atoms, these need to be added as well. Various software packages, like WhatIf [68] or Rosetta, offer assignment of hydrogen coordinates for protein structures. The Gromacs tool *pdb2gmx* can do this too. In fact, it is convenient to pre-process a .pdb file with *pdb2gmx* because it produces a structure file with atom names already compatible with the Gromacs internal atom naming given the selected force field. *pdb2gmx* also identifies whether any heavy atoms in a protein are missing, so that the tool can be used to identify incomplete residues. While *pdb2gmx* will not model missing heavy atoms, it will inform about such deficiencies. Note that *pdb2gmx* will fail if the input structure contains molecules that are not readily recognized by Gromacs. Therefore, molecules that are not present in the force field file have to be removed from the structure at this stage and processed independently.

For the Trp cage model system we use an NMR structure (PDB-ID 1L2Y) [22] that was deposited with 38 conformers. After manually extracting conformer #2, we pre-process the structure by running it through *pdb2gmx*:

```
gmx pdb2gmx -f 1l2y_conf2.pdb -o 1l2y_conf2_pdb2gmx.pdb
-ff amber99sb-star-ildn-mut -water none -ignh
```

In this example we have selected an updated version of the Amber99sb*ILDN force field [69–71] for which the mutation library has been pre-generated. No water model needs to be chosen at this stage, because with this step we only want to obtain a pre-processed structure file with added hydrogens and Gromacs compatible atom names. The "-ignh" flag ignores the hydrogen atoms already present in the structure and adds them again using the *pdb2gmx* logic, ensuring the names of the hydrogen atoms are compatible with Gromacs and the selected force field (*see* **Note 3**).

The output structure file obtained as described above is then used as an input for the `pmx` script *mutate.py*:

```
python mutate.py -f 1l2y_conf2_pdb2gmx.pdb -o mut.pdb -ff amber99sb-
star-ildn-mut
```

Upon execution, the command prompts for an interactive selection of a residue to mutate (W6) and a target amino acid (Phe or F). When developing a workflow for a large-scale mutation scan, it may be convenient to provide the information about the amino acid mutations as a text file. For this purpose a "-script" flag in *mutate.py* is available: this option expects a text file with an amino acid number and the name of the residue to mutate into. In the case of the Trp cage example: `6 Phe`.

3.3 Topology

At this point we use the hybrid structure from the previous step ("mut.pdb") as an input to *pdb2gmx* (Fig. 3). This time we want to obtain the topology file containing all the information needed by Gromacs to run the simulations. The topology file will also include the description of the hybrid mutated residue, however, parameters only for one physical state (state *A*) are defined in the output topology file. It is also important to note that at this step the "-ignh" flag should not be set, since the hydrogen atoms have already been added in the previous step.

```
gmx pdb2gmx -f mut.pdb -o mut_pdb2gmx.pdb -ff amber99sb-star-ildn-mut
-water tip3p -p topol.top
```

If one wants to include a ligand that has been parameterized separately, this can be added to the structure ("mut_pdb2gmx.pdb") and topology file ("topol.top") at this stage.

3.4 Hybrid Topology

The generated topology file ("topol.top") has the hybrid residue W2F incorporated. However, it is a non-standard hybrid amino acid with two physical states (*A* and *B*). While state *A* is included in the topology, state *B* still needs to be included explicitly. The

required topology parameters for state *B* can be added by the `pmx` script *generate_hybrid_topology.py* (Fig. 3):

```
python generate_hybrid_topology.py -p topol.top -o hybrid.top -ff
amber99sb-star-ildn-mut
```

3.5 Webserver

The procedure detailed above (and summarized in Fig. 3) can also be executed via a webserver interface: http://pmx.mpibpc.mpg.de. Provided with a protein structure file, the pmx webserver will perform a user-selected mutation in one of the supported molecular mechanics force fields.

The webserver runs a number of additional structure pre-processing steps that simplify the setup procedure. While broken or incomplete proteins will not be repaired, a number of other useful modifications are applied: residue and atom names are matched to the force field nomenclature, terminal residues are dealt with, and if needed hydrogen atoms may be added via *pdb2gmx*. Optionally, the structure may be checked before the mutation is performed, so that the user is informed about any potential deficiencies in the input file. In addition, the setup offered by the webserver is not limited to single amino acid mutations, but also allows to prepare files for mutation scans over selected protein chains.

3.6 Alchemical Simulations

The hybrid structures and topologies we just obtained can readily be used for MD simulations and to calculate free energy differences. Numerous protocols for relative alchemical free energy calculations are currently available: equilibrium approaches (TI, FEP) as well as non-equilibrium methods. Here, we employ non-equilibrium calculations based on the Crooks Fluctuation Theorem.

3.6.1 System Preparation

Firstly, the hybrid structure and topology are used in preparing the system for molecular dynamics simulations following a standard procedure. The protein needs to be placed in a simulation box and solvated. Then ions need to be added to neutralize the system and, optionally, reach a desired salt concentration. These are conventional steps used to prepare an ordinary MD simulation: for a more detailed description of this procedure in Gromacs we refer the reader to a specialized protocol [72].

3.6.2 Equilibrium Simulations

Next, we set up two equilibrium simulations: one for the WT Trp cage (W6, state A, $\lambda = 0$) and another for the mutated protein (F6, state B, $\lambda = 1$) (Fig. 4). We start with an energy minimization performed on both states separately. The parameters for the energy minimization (.mdp) are the same as those used in non-alchemical simulations, with the exception of two flags. The `free-energy` flag has to be set to `yes`. This indicates that the free energy code in

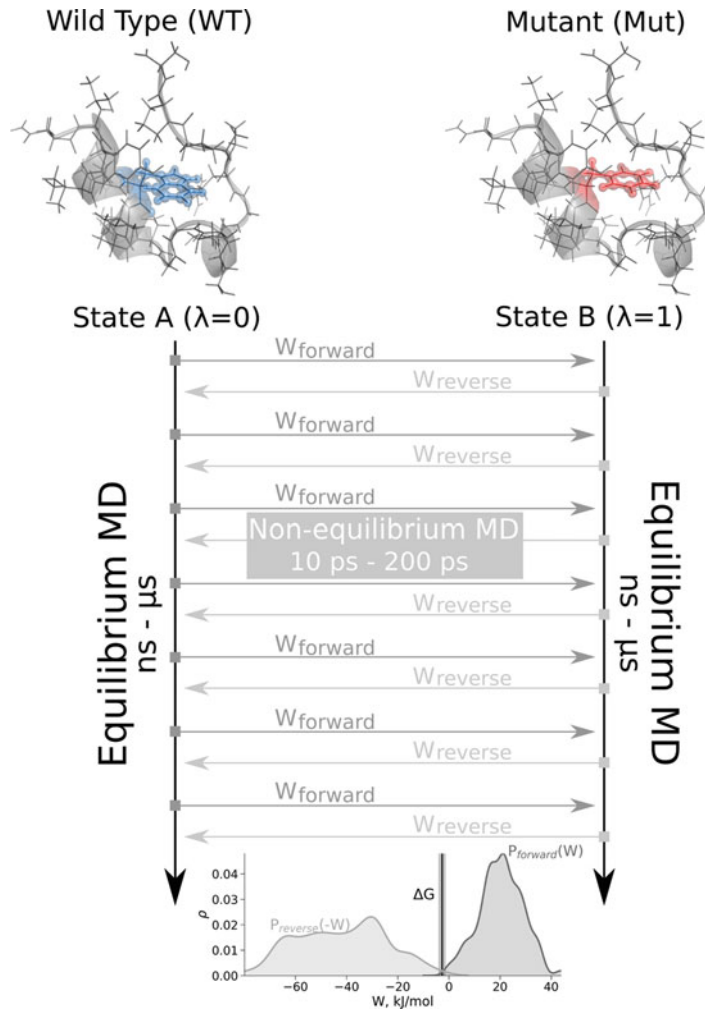

Fig. 4 The procedure of non-equilibrium alchemical simulations for one leg of the thermodynamic cycle: mutation in the folded state of a protein. Two independent equilibrium simulations are performed by keeping the system in its physical states: WT ($\lambda = 0$) and mutant ($\lambda = 1$). These simulations need to sufficiently sample the end state ensembles, as the accuracy of the free energy estimate will depend on the convergence of the equilibrium sampling. Typically, the equilibrium simulations are in the nanosecond to microsecond time range. From the generated trajectories, snapshots are selected to start fast (typically 10–200 ps) transitions driving the system in the forward (λ: $0 \rightarrow 1$) and reverse (λ: $1 \rightarrow 0$) directions. The work values required to perform these transitions are collected and the Crooks Fluctuation Theorem is used to calculate the free energy difference between the two states

Gromacs will be activated for those interactions that have two sets of parameters (states *A* and *B*) in the topology file. In addition, the `init-lambda` flag has to be set to `0` for the simulation in state *A* (WT Trp cage) and to `1` for state *B* (mutated Trp cage).

After the energy minimization runs for the A and B states are complete, MD simulations can be started from the energy minimized conformations. Similarly to the energy minimization, the simulation parameters are identical to the conventional MD runs, except for setting the `free-energy` and `init-lambda` flags for the simulations in state *A* and *B*, respectively (Fig. 4). These equilibrium runs are used to sample the relevant phase space volumes, i.e., the conformational changes in the WT and mutated variant of Trp cage. Therefore, the ensembles generated during the equilibrium runs will define how accurately the free energy difference will be estimated. This consideration dictates the sampling time: the simulation time should be sufficient to sample the transitions that are considered to be relevant. For example, if a protein is known to undergo large-scale conformational changes and the introduced mutation may be affecting the populations of these conformers, the simulation time has to be long enough to properly sample such transitions. Equilibrium simulations in this case could require microseconds or longer to converge. On the other hand, it is often important to estimate the free energy difference for a structure that would remain close to its experimentally resolved structure. In this scenario, it is sufficient to sample smaller changes in rotameric states of the side chains and minor backbone motions. In previous large-scale amino acid scans investigating protein thermodynamic stabilities, we have observed good agreement with experimental data when using 10–20 ns of equilibrium sampling [1, 20].

Another issue to consider when choosing the sampling time is the definition of states for which the free energy difference will be calculated. In the Trp cage example, we are aiming to estimate the mutation-induced free energy difference in folding free energy. This implies that one of the end states that we need to simulate needs to be the *folded* state, while the other needs to be the *unfolded* state. If we were to introduce a destabilizing mutation (in fact W6F has been shown to strongly destabilize Trp cage [73, 74]), over a longer simulation time the protein would unfold. Thus, the definition of the *folded* state used in the free energy calculation would be violated, rendering the calculated free energy differences inaccurate. For the Trp cage W6F mutation example, we will use equilibrium simulations of 10 ns: short enough such that no spontaneous unfolding occurs.

3.6.3 Non-equilibrium Transitions

Once the equilibrium simulations are completed, we can proceed to the non-equilibrium part of the simulation protocol. Fast non-equilibrium transitions serve the purpose of connecting the two physical states (*A* and *B*) and allow obtaining the free energy difference between them. These transitions are started from snapshots extracted from the two equilibrium trajectories. From each equilibrium trajectory, we discard the first 2 ns as equilibration

time. Then we use the last 8 ns to extract 100 frames equidistant in time (1 frame every 80 ps) representing an equilibrium ensemble of starting conformations for the non-equilibrium transitions. These frames can be conveniently extracted using the Gromacs tool *gmx trjconv* (*see* **Note 4**). Note that the number of non-equilibrium transitions performed will influence the accuracy of the free energy estimate. More transitions allow acquiring more work values, which consequentially allow for a more accurate ΔG estimate. In previous investigations we have observed that 50–100 transitions are generally sufficient to obtain accurate estimates of folding free energy changes upon protein mutation [1, 20].

Another parameter influencing the accuracy of the ΔG estimate is the transition time. Over the course of a shorter transition, the system is driven further away from equilibrium, and more work is dissipated along the alchemical path. In turn, the free energy estimate becomes less accurate. The optimal transition time depends on the size of the perturbation and the nature of the system, e.g., replacing a small residue with a large one represents a larger perturbation than a small-to-small residue mutation. Larger perturbations may require slower transitions to obtain free energies at the desired level of accuracy. Transition times ranging from tens to hundreds of picoseconds are usually enough to return accurate results [1, 20].

For the Trp cage example considered here, we use non-equilibrium transitions of 50 ps. Using a 2 fs time step, this means running 25,000 integration steps. Therefore, the λ value will be changed at a speed of 1/25,000=4e−5. For the transition simulations in the forward (state *A* to *B*) direction, we set the following parameters in the .mdp file:

```
nsteps          = 25000
nstcalcenergy = 1
nstdhdl         = 1
free-energy     = yes
init-lambda     = 0
delta-lambda    = 4e-5
sc-alpha        = 0.3
sc-sigma        = 0.25
sc-power        = 1
sc-coul         = yes
```

`nsteps` defines the number of steps. The system starts at `init-lambda=0` and is morphed into the $\lambda = 1$ state over the course of a transition in `nsteps` number of steps. The energy and $\partial H/\partial\lambda$ are calculated at every integration step (`nstcalcenergy` and `nstdhdl` set to 1). The parameters starting with a prefix `sc-` control the soft-core interactions. In this non-equilibrium protocol

we soften both the van der Waals and Coulombic interactions: `sc-coul=yes` (*see* **Note 1**).

For the transitions in the reverse direction (state B to A), the .mdp parameters are the same, with the exception of the starting state and the direction of the transition:

```
init-lambda    = 1
delta-lambda   = -4e-5
```

For each of the 100 transitions in both directions, the data we need to collect are the $\partial H/\partial\lambda$ values, which are stored in the ".dhdl.xvg" files. Integration over these values gives the work performed during the non-equilibrium transitions in the forward and reverse directions (Fig. 4).

3.7 Analysis

The integration over the $\partial H/\partial\lambda$ curves and the free energy difference estimation can be performed with the `pmx` script *analyze_dhdl.py*:

```
python analyze_dhdl.py -fA stateA/dhdl*.xvg -fB stateB/dhdl*.xvg
```

The script will output the summary of results in a text file containing the estimate of the free energy difference using three estimators: Crooks Gaussian Intersection (CGI), Bennet's Acceptance Ratio (BAR), and Jarzynski's equality. While CGI and BAR use the work distributions generated in both, forward and reverse, directions, Jarzynski's estimator is one-directional. We recommend using the BAR estimation for the ΔG value, as it utilizes all the available work values from both directions and makes no assumptions about the shape of the work distributions. Conveniently, the script also generates plots of the work values over time and of their distributions (Fig. 5), which are useful to detect potential sampling or lack of the work distribution overlap issues.

The convergence of the results can be assessed in various ways. Firstly, if a systematic drift of the work values over time is observed, it usually indicates lack of convergence during the equilibrium sampling stage. The work values are likely to drift due to a conformational change and it may be important to thoroughly sample the significant conformational motions in the protein. Lack of convergence may also be deduced from the error values provided together with the free energy estimates. The uncertainties of the CGI and BAR estimators are sensitive to the lack of the overlap between the forward and reverse work distributions (*see* **Note 5**). A large uncertainty in the ΔG estimate indicates that the overlap between the work distributions might be insufficient. Slower transitions keep the system closer to equilibrium, so that less work is dissipated along the path and the overlap between work distributions generally increases. Running more non-equilibrium transitions increases the probability of observing work values with low dissipation, which also contributes toward good overlap of the work distributions.

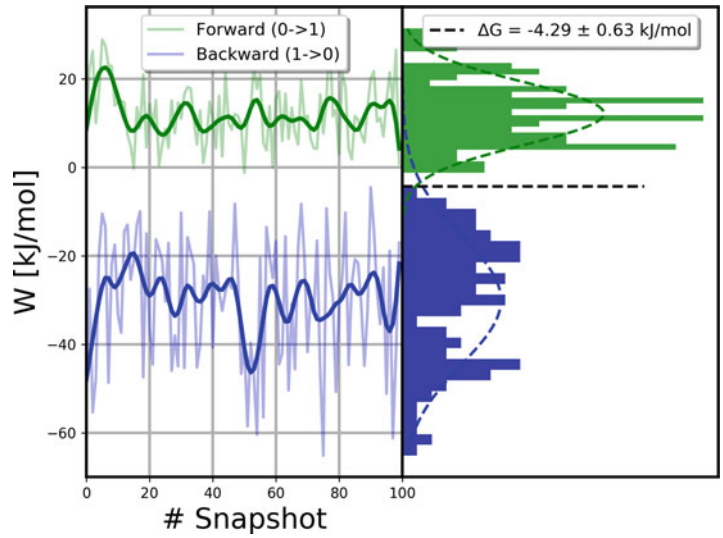

Fig. 5 A standard output generated by the `pmx` *analyze_dhdl.py* script. On the left, the work values for the forward and reverse transitions are depicted for every starting structure. On the right, the distributions of these work values are shown as histograms, the intersection of which allows obtaining the free energy difference. In the current case, the BAR estimator yields a ΔG value of -4.29 ± 0.63 kJ/mol for the W6F mutation in the folded state of the Trp cage protein

The most reliable way to assess the precision of the free energy estimates obtained is to repeat the whole procedure, including equilibrium and non-equilibrium simulations, multiple times. The calculated ΔG values and their spread obtained from multiple independent calculations more accurately capture under-sampling issues. For the Trp cage W6F mutation, we have obtained a ΔG value of -4.29 ± 0.63 kJ/mol (Fig. 5) from a single calculation. Then, we repeated the whole calculation five times, from the system preparation to the equilibrium and non-equilibrium simulations. The average free energy value we obtained was of -3.73 kJ/mol with a standard error of 0.88 kJ/mol. This result confirms that the ΔG estimate obtained can be considered to be reliable.

3.8 Double Free Energy Difference

So far we have calculated the free energy difference for one leg of the thermodynamic cycle (Fig. 1): mutation in a folded protein. To obtain the final double free energy difference the same procedure needs to be performed for the unfolded Trp cage peptide. It has been demonstrated that in the context of the alchemical free energy calculations the unfolded state can be approximated by a capped tripeptide with the residue of interest surrounded by two glycines [20].

Given the tripeptide approximation, the ΔG values can be pre-calculated for every amino acid combination of interest. The tabulated tripeptide ΔG values can be found on the `pmx` webserver: http://pmx.mpibpc.mpg.de (*see* **Note 6**). The tripeptide's W2F mutation in the Amber99sb*ILDN force field has an associated free energy change ($\Delta G^{\text{Mutation}}_{\text{Unfolded}}$) of -17.96 ± 0.32 kJ/mol. The $\Delta\Delta G$ of interest can thus be calculated as $\Delta G^{\text{Mutation}}_{\text{Folded}} - \Delta G^{\text{Mutation}}_{\text{Unfolded}}$. Thus, in our Trp cage example, the $\Delta\Delta G$ of folding for the W6F mutation is estimated to be 13.67 ± 0.71 kJ/mol. This calculated estimate closely matches the experimentally measured destabilization of 12.5 ± 0.6 kJ/mol [73, 74]. A previous large-scale study compared calculated and experimental $\Delta\Delta G$ values for protein thermostability changes upon mutation for the proteins barnase and Staphylococcal nuclease [1]. It was found that the mean unsigned error in the predictions was of approximately 4 kJ/mol, with the uncertainty due to finite sampling, the force field, and the experimental error equally contributing to the discrepancy between calculated and experimental $\Delta\Delta G$ values. Therefore, the calculated $\Delta\Delta G$ value for the Trp cage W6F mutation falls well within the range of the expected accuracy.

4 Miscellaneous Applications of the Protocol

In the previous section we have outlined a general protocol for the calculation of free energy changes upon an amino acid mutation. However, the protocol can and in some cases should be expanded or modified in order to fit the specific needs of the problem at hand. In particular, charge-changing mutations (e.g., Ala to Asp, Trp to Arg, etc.) need special care as artifacts that affect the accuracy of the calculations are otherwise introduced when applying Ewald summation for long range electrostatics. In addition, one is not limited to a single amino acid mutation per calculation. However, when mutating more residues at the same time, more attention should be paid to possible convergence and work overlap issues. Finally, for protein design applications, large mutational scans need to be set up and run, and with a few precautions the protocol can be executed in a more efficient and convenient manner. All these additional points are discussed in the following paragraphs.

4.1 Charge-Changing Mutations

It is often of interest to calculate free energy differences upon amino acid mutations that cause a net charge change. In principle, there is no fundamental difference whether an alchemical mutation is to be charge-changing or charge-conserving. The issue, however, is of a technical nature and has to do with the treatment of long range electrostatic interactions in molecular dynamics simulations. The state-of-the-art long range Coulombic interaction calculations utilize Particle Mesh Ewald (PME) summation [75, 76]. Due to

the specifics of the PME algorithm, for a non-neutral system any extra charge is neutralized by the implicit introduction of a uniform background charge. Taking into account the contributions of this effect to the free energy difference requires additional, and technically involved, correction schemes [77, 78]. If neglected, significant artifacts may occur [79]. Therefore, conserving a system's charge during the alchemical transition is preferred.

To accomplish this, we suggest using a double-system/single-box setup [24, 26]. In this approach both legs of a thermodynamic cycle, e.g., mutation in the folded and unfolded states in Fig. 1, are placed in the same simulation box. The systems of the separate legs in the thermodynamic cycle are set in different physical states: if the folded state protein has a WT residue at $\lambda = 0$, then the unfolded state must be in its mutated form at $\lambda = 0$. During the alchemical transition, the system goes from $\lambda = 0$ to 1, and the folded state is transformed into the mutant while the unfolded (mutant) state is simultaneously transformed into the WT. In this way, the charge of the system will be conserved during the transformation, and the free energy difference calculated already refers to the $\Delta\Delta G$ across the thermodynamic cycle of interest.

The assumption underlying the double-system/single-box approach is that both legs of the thermodynamic cycle are independent when placed in a single simulation box. Therefore, it is important to place the systems (e.g., the folded protein and unfolded peptide) sufficiently far apart. The distance between the molecules needs to be larger than the short range electrostatic cutoff. We have obtained reliable free energy estimates by setting the distance to be at least 3 nm between any atoms of the molecules from the separate legs of the cycle [1, 24]. To ensure that the proteins do not diffuse and come closer to one another during the course of simulations, position restraints on a single backbone atom close to the center of mass of each protein ought to be used. These position restraints affect only the translational degrees of freedom of the proteins. The contribution of such position restraints to the translational partition function will be the same in both legs of the thermodynamic cycle and will cancel from the $\Delta\Delta G$ estimate.

4.2 More Than One Mutation at Once

The protocol in this chapter described an example of a single amino acid mutation in a protein. `pmx`, however, also allows introducing multiple mutations at once as well. This can be done either by interactively selecting more than one mutation to be applied or by providing an external file with every mutation defined in a new line of a text file. The `pmx` webserver also provides the option to introduce multiple mutations.

The caveat of performing an alchemical transformation for several amino acid mutations at once is a slower convergence of the free energy estimate. Having more mutations imposes a larger perturbation to the system. Hence, more work will be dissipated

along the path and the free energy estimate will become less accurate. In such a case, performing the non-equilibrium transitions slower may be necessary.

Another way to calculate the effect of multiple mutations is to perform the mutations sequentially. For example, the free energy difference of introducing the mutations X and Y at once is equal to the combined ΔG of performing the mutation X first and in a separate setup calculating ΔG for the Y mutation in a system where the X mutation is already present. In fact, since free energy is a state function, the sequence of introducing the mutations does not influence the final ΔG estimate, thus the mutation Y can be performed first and then the mutation X can follow. The free energy differences calculated in all three scenarios (X and Y at once, first X then Y, first Y then X) ought to yield the same estimate. Therefore, the spread of these three ΔG values could serve as an indicator of the uncertainty in the calculations.

4.3 Mutation Scan

In protein design studies or large-scale mutation investigations, it is common to perform mutation scans by replacing every amino acid in a protein sequence with another residue; e.g., alanine scans are often employed. The command line scripts described in the current protocol make it easy to build workflows allowing for any number of mutations to be introduced. The `pmx` webserver also allows for an automated generation of the hybrid structures and topologies for a mutation scan with the user-selected amino acids.

When a protein needs to be mutated multiple times, the end state representing the wild-type does not need to be simulated at equilibrium multiple times. The same WT equilibrium simulation can be reused for all mutants to increase the efficiency of the protocol. However, in order to be able to reuse the same equilibrium run, the generation of the hybrid structure and topology files has to be postponed to the step after the equilibrium simulations. Thus, effectively, the steps described in Subheadings 3.2–3.4 need to be performed after the equilibrium simulations have been carried out (Subheading 3.6.2). In this scenario, the equilibrium simulations would be performed without invoking the free energy code and using standard non-hybrid structures and topologies for both the WT and mutant proteins. The hybrid topology can be generated only once, using one of the many frames extracted from the equilibrium simulations. On the other hand, the hybrid coordinates need to be built by the `mutate.py` script for all the frames extracted from the equilibrium trajectories. For example, a custom bash script with a *for* loop, calling `mutate.py` for each of the extracted frames, would add the atoms for state B to all the snapshots extracted. In such a way, the same trajectory can be reused for all of the mutations of interest; for each mutation, one needs to create the topology with the hybrid residues of interest, and generate the corresponding hybrid structures.

5 Summary

In summary, we have presented a step-by-step protocol for the calculation of free energy changes upon amino acid mutation. As an example, we have shown how these calculations can be used to estimate the destabilizing effect of the W6F mutation on the fold of the Trp cage protein using the `pmx` software. Throughout the text, we have also pointed out common issues that may be encountered in alchemical non-equilibrium free energy calculations, as well as their solutions. Furthermore, we discussed how the protocol can be automated and scaled up in order to better fit the requirements of applications that involve large mutational scans, such as protein design. Finally, we remind the reader that while here we focussed on amino acid mutations, `pmx` also allows to set up, run, and analyze alchemical free energy calculations that involve the mutation of nucleic acids and development is in progress to support arbitrary organic molecules (ligands). Thus, overall, `pmx` and the free energy calculation protocol presented here are flexible tools that can find broad application in various fields of computational biophysics and chemistry.

6 Notes

1. When using equilibrium alchemical free energy calculation protocols (equilibrium TI or FEP) it is usually recommended to perform transformations of the van der Waals interactions after turning off the charges on the morphed atoms. In this scenario, only the van der Waals interactions need to be soft-cored, while the Coulombic interactions may be calculated using the unmodified Hamiltonian. In principle, the same procedure could be applied for the non-equilibrium transitions as well, however, it is more convenient to perform the fast alchemical transitions by morphing the Lennard-Jones parameters and charges simultaneously. In this case, both the van der Waals and Coulombic interactions have to be modified using the soft-core potential. If the default Gromacs 2016 soft-core implementation leads to an erratic behavior of the $\partial H/\partial \lambda$ curves (e.g., unreproducible spikes orders of magnitude larger than the average values), an alternative soft-core implementation can be found on the `pmx` webserver's download section, which is more suitable for the non-equilibrium protocol described in this chapter [63].
   Depending on the software version, Gromacs may issue a warning during *grompp* execution regarding the use of the soft-core interactions when van der Waals interactions are not decoupled. In the context of the non-equilibrium free energy calculations it is safe to ignore this warning: flag `-maxwarn`.

2. In order to be able to use the hybrid/alchemical force fields available in `pmx`, the environment variable `$GMXLIB` needs to be set. This is required as Gromacs uses the path specified in `$GMXLIB` to locate additional force field libraries. In `pmx`, all available hybrid force fields can be found in `$PMXHOME/data/mutff45`, where `$PMXHOME` is the absolute path to the `pmx` source folder. Thus, to allow Gromacs to find the `pmx` force fields, you should run the following command (in bash shell):

   ```
   export GMXLIB=$PMXHOME/data/mutff45
   ```

3. The "-ignh" flag tells *pdb2gmx* to ignore the hydrogen atoms present in the input structure. In this way, the tool adds the hydrogen atoms again using its own logic. This can be useful when there are hydrogen atoms in the input structure with atom names that are not recognized by Gromacs and/or not present in the force field of choice. If the flag is not set, *pdb2gmx* will keep the hydrogen atoms present in the structure, which can be useful if external programs were used to determine the protonation states of the protein's residues, or if it is preferred to keep the protonation states determined experimentally (e.g., via neutron diffraction). However, in this case one needs to make sure the names of the hydrogen atoms conform to the naming used in the selected force field, otherwise *pdb2gmx* will raise an error. An alternative is to expand the `aminoacids.arn` file in the force field library of interest to introduce a mapping between the hydrogen atom names in the input structure and in the force field.

4. The Gromacs command `trjconv` allows the user to convert and manipulate trajectory files, and comes handy when one wants to extract single frames to be used as starting points for the non-equilibrium simulations. In particular, the flag `-b` allows to choose the frames to discard before a certain time defined in picoseconds. The flag `-sep` tells the program to write each snapshot as a separate indexed coordinate file. The flag `-skip` tells the program to extract only every n-th frame. The flags `-ur` and `-pbc` keeps molecules intact across the periodic boundaries. For instance, in the example with Trp cage, we ran the following `trjconv` command:

   ```
   gmx trjconv -f equilibrium_sim.xtc -s equilibrium
   _sim.tpr -o frame_.pdb -sep -b 2001 -skip 1 -ur com
   pact -pbc mol
   ```

   In this way, as we saved coordinates to the trajectory file every 80 ps, we obtained 100.pdb files called `frame_`*n*`.pdb`, with `n` from 0 to 99.
   Here, we used an .xtc file to store the trajectory data and .pdb files to extract the snapshots. These file formats contain only the atom coordinates, but no velocities, therefore, when

generating non-equilibrium runs, the flag gen-vel=yes in the .mdp file needs to be set, together with a reference temperature, to generate velocities from a Maxwell distribution. Another option is to use the .trr files for storing the trajectory data from equilibrium simulations. The .trr files can also store the velocities along with the coordinates. If the .trr files are used, the starting snapshots should be extracted as .gro files instead of .pdb, as the .gro file format allows storing both coordinates and velocities. Non-equilibrium runs started from initial structures generated in this way will use velocities as obtained from the equilibrium sampling; thus, the `gen-vel` flag in the .mdp file can be set to `no`.

5. The analytical error for the Bennet's Acceptance Ratio estimator grows very rapidly even for a minor lack of the overlap between the work distributions. The rate of growth for the analytical error often does not match the bootstrapped error estimate, which warrants further investigation into BAR uncertainty estimators. Nevertheless, a large value of the analytical estimator may serve as a good indicator for the lack of convergence during the non-equilibrium transitions.
6. The current implementation of the `pmx` mutation libraries follow the single topology formalism and the bond lengths are allowed to change between the two end states. When bond length constraints are used during the simulations, the contribution of the constraints (upon changes in bond lengths) to $\delta H/\delta\lambda$ is taken into account by Gromacs. Therefore, TI-based approaches and the non-equilibrium free energy calculations in Gromacs properly account for the changes in the bond length. However, Gromacs currently does not incorporate this contribution into the data used by FEP approaches to estimate free energy differences. This means that while equilibrium FEP and non-equilibrium approaches should return the same ΔG values for the same mutations in theory, in practice this is not the case. Since the mutation libraries have been generated using the non-equilibrium free energy protocol, the tabulated values for the tripeptide mutations should be used only in combination with free energy calculations that make use of the $\delta H/\delta\lambda$ curve integration. For FEP-based approaches, the tripeptide mutations need to be calculated separately.

References

1. Gapsys V, Michielssens S, Seeliger D, de Groot BL (2016) Accurate and rigorous prediction of the changes in protein free energies in a large-scale mutation scan. Angew Chem Int Ed Engl 55(26):7364–7368
2. Griss R, Schena A, Reymond L, Patiny L, Werner D, Tinberg CE, Baker D, Johnsson K (2014) Bioluminescent sensor proteins for point-of-care therapeutic drug monitoring. Nat Chem Biol 10(7):598–603
3. Feng J, Jester BW, Tinberg CE, Mandell DJ, Antunes MS, Chari R, Morey KJ, Rios X, Medford JI, Church GM, Fields S, Baker D (2015) A general strategy to construct small

molecule biosensors in eukaryotes. eLife 4:323–329

4. Zhou L, Bosscher M, Zhang C, Özçubukçu S, Zhang L, Zhang W, Li CJ, Liu J, Jensen MP, Lai L, He C (2014) A protein engineered to bind uranyl selectively and with femtomolar affinity. Nat Chem 6(3):236–241
5. Correia BE, Bates JT, Loomis RJ, Baneyx G, Carrico C, Jardine JG, Rupert P, Correnti C, Kalyuzhniy O, Vittal V, Connell MJ, Stevens E, Schroeter A, Chen M, MacPherson S, Serra AM, Adachi Y, Holmes MA, Li Y, Klevit RE, Graham BS, Wyatt RT, Baker D, Strong RK, Crowe JE, Johnson PR, Schief WR (2014) Proof of principle for epitope-focused vaccine design. Nature 507(7491):201–206
6. Koday MT, Nelson J, Chevalier A, Koday M, Kalinoski H, Stewart L, Carter L, Nieusma T, Lee PS, Ward AB, Wilson IA, Dagley A, Smee DF, Baker D, Fuller DH (2016) A computationally designed hemagglutinin stem-binding protein provides in vivo protection from influenza independent of a host immune response. PLoS Pathog 12(2):e1005409
7. Clark AJ, Gindin T, Zhang B, Wang L, Abel R, Murret CS, Xu F, Bao A, Lu NJ, Zhou T, Kwong PD, Shapiro L, Honig B, Friesner RA (2017) Free energy perturbation calculation of relative binding free energy between broadly neutralizing antibodies and the gp120 glycoprotein of HIV-1. J Mol Biol 429(7):930–947
8. Fowler PW, Cole K, Gordon NC, Kearns AM, Llewelyn MJ, Peto TEA, Crook DW, Walker AS (2018) Robust prediction of resistance to trimethoprim in Staphylococcus aureus. Cell Chem Biol 25:339–349
9. Hauser K, Negron C, Albanese SK, Ray S, Steinbrecher T, Abel R, Chodera JD, Wang L (2018) Predicting resistance of clinical Abl mutations to targeted kinase inhibitors using alchemical free-energy calculations. Commun Biol 1:70
10. Tinberg CE, Khare SD, Dou J, Doyle L, Nelson JW, Schena A, Jankowski W, Kalodimos CG, Johnsson K, Stoddard BL, Baker D (2013) Computational design of ligand-binding proteins with high affinity and selectivity. Nature 501(7466):212
11. Yang W, Lai L (2017) Computational design of ligand-binding proteins. Curr Opin Struct Biol 45:67–73
12. Brender JR, Zhang Y (2015) Predicting the effect of mutations on protein-protein binding interactions through structure-based interface profiles. PLoS Comput Biol 11(10):e1004494
13. Pires DEV, Ascher DB, Blundell TL (2014) mCSM: predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics 30(3):335–342
14. Schymkowitz J, Borg J, Stricher F, Nys R, Rousseau F, Serrano L (2005) The FoldX web server: an online force field. Nucleic Acids Res 33(Suppl 2):W382–W388
15. Kortemme T, Baker D (2002) A simple physical model for binding energy hot spots in protein-protein complexes. Proc Natl Acad Sci USA 99(22):14116–14121
16. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, Kaufman K, Renfrew PD, Smith CA, Sheffler W, Davis IW, Cooper S, Treuille A, Mandell DJ, Richter F, Ban YEA, Fleishman SJ, Corn JE, Kim DE, Lyskov S, Berrondo M, Mentzer S, Popović Z, Havranek JJ, Karanicolas J, Das R, Meiler J, Kortemme T, Gray JJ, Kuhlman B, Baker D, Bradley P (2011) Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487 (C):545–574
17. Petukh M, Li M, Alexov E (2015) Predicting binding free energy change caused by point mutations with knowledge-modified MM/PBSA method. PLoS Comput Biol 11 (7):e1004276
18. Beard H, Cholleti A, Pearlman D, Sherman W, Loving KA (2013) Applying physics-based scoring to calculate free energies of binding for single amino acid mutations in protein-protein complexes. PLoS ONE 8(12):e82849
19. Moreira IS, Fernandes PA, Ramos MJ (2007) Computational alanine scanning mutagenesis - An improved methodological approach. J Comput Chem 28(3):644–654
20. Seeliger D, de Groot BL (2010) Protein thermostability calculations using alchemical free energy simulations. Biophys J 98 (10):2309–2316
21. Chipot C, Pohorille A (eds) (2007) Free energy calculations: theory and applications in chemistry and biology, vol 86. Springer, Berlin
22. Neidigh JW, Fesinmeyer RM, Andersen NH (2002) Designing a 20-residue protein. Nat Struct Mol Biol 9(6):425–430
23. 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 2:1–7
24. Gapsys V, Michielssens S, Seeliger D, de Groot BL (2015) pmx: automated protein structure and topology generation for alchemical perturbations. J Comput Chem 36(5):348–354

25. Chipot C (2014) Frontiers in free-energy calculations of biological systems. Wiley Interdiscip Rev Comput Mol Sci 4(1):71–89
26. Gapsys V, Michielssens S, Peters JH, de Groot BL, Leonov H (2015) Molecular modeling of proteins, vol 1215. Humana Press, New York
27. Pohorille A, Jarzynski C, Chipot C (2010) Good practices in free-energy calculations. J Phys Chem B 114(32):10235–10253
28. Hansen N, van Gunsteren WF (2014) Practical aspects of free-energy calculations: a review. J Chem Theory Comput 10(7):2632–2647
29. Goette M, Grubmüller H (2009) Accuracy and convergence of free energy differences calculated from nonequilibrium switching processes. J Comput Chem 30(3):447–456
30. Jarzynski C (1997) Nonequilibrium equality for free energy differences. Phys Rev Lett 78 (14):2690–2693
31. Jarzynski C (1997) Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. Phys Rev E 56:5018–5035
32. Crooks GE (1998) Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems. J Stat Phys 90(5/6):1481–1487
33. Crooks GE (1999) Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. Phys Rev E 60(3):2721–2726
34. Crooks GE (2000) Path-ensemble averages in systems driven far from equilibrium. Phys Rev E 61(3):2361–2366
35. Hummer G, Szabo A (2001) Free energy reconstruction from nonequilibrium single-molecule pulling experiments. Proc Natl Acad Sci USA 98(7):3658–3661
36. Hummer G (2001) Fast-growth thermodynamic integration: error and efficiency analysis. J Chem Phys 114(17):7330–7337
37. Hummer G, Szabo A (2005) Free energy surfaces from single-molecule force spectroscopy. Acc Chem Res 38(7):504–513
38. Zwanzig RW (1954) High-temperature equation of state by a perturbation method. I. nonpolar gases. J Chem Phys 22 (8):1420–1426
39. Kirkwood JG (1935) Statistical mechanics of fluid mixtures. J Chem Phys 3(5):300–313
40. Cuendet MA (2006) The Jarzynski identity derived from general Hamiltonian or non-Hamiltonian dynamics reproducing NVT or NPT ensembles. J Chem Phys 125 (14):144109
41. Wood RH, Mühlbauer WCF, Thompson PT (1991) Systematic errors in free energy perturbation calculations due to a finite sample of configuration space: sample-size hysteresis. J Phys Chem 95(17):6670–6675
42. Gore J, Ritort F, Bustamante C (2003) Bias and error in estimates of equilibrium free-energy differences from nonequilibrium measurements. Proc Natl Acad Sci USA 100 (22):12564–12569
43. Nanda H, Lu N, Woolf TB (2005) Using non-Gaussian density functional fits to improve relative free energy calculations. J Chem Phys 122(13):134110
44. Massey FJ Jr (1951) Kolmogorov-Smirnov test for goodness of fit. Test 46(253):68– 78
45. Efron B, Tibshirani RJ (1994) An introduction to the bootstrap, vol 5, 1st edn. Chapman and Hall/CRC, London/West Palm Beach
46. Bennett CH (1976) Efficient estimation of free energy differences from Monte Carlo data. J Comput Phys 22(2):245–268
47. 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
48. Nelder JA, Mead R (1964) A simplex method for function minimization. Comput J 7 (4):308–313
49. Hahn AM, Then H (2010) Measuring the convergence of Monte Carlo free-energy calculations. Phys Rev E Stat Nonlinear Soft Matter Phys 81(4):041117
50. Lindorff-Larsen K, Trbovic N, Maragakis P, Piana S, Shaw DE (2012) Structure and dynamics of an unfolded protein examined by molecular dynamics simulation. J Am Chem Soc 134(8):3787–3791
51. Rauscher S, Gapsys V, Gajda MJ, Zweckstetter M, de Groot BL, Grubmüller H (2015) Structural ensembles of intrinsically disordered proteins depend strongly on force field: a comparison to experiment. J Chem Theory Comput 11(11):5513–5524
52. Prevost M, Wodak SJ, Tidor B, Karplus M (1991) Contribution of the hydrophobic effect to protein stability: analysis based on simulations of the Ile-96 → Ala mutation in barnase. Proc Natl Acad Sci USA 88(23):10880–10884
53. Sneddon SF, Tobias DJ (1992) The role of packing interactions in stabilizing folded proteins. Biochemistry 31(10):2842–2846
54. Pitera JW, Kollman PA (2000) Exhaustive mutagenesis in silico: multicoordinate free

energy calculations on proteins and peptides. Proteins Struct Funct Bioinf 41(3):385–397

55. Pearlman DA, Kollman PA (1991) The overlooked bond-stretching contribution in free energy perturbation calculations. J Chem Phys 94(6):4532
56. Pearlman DA (1994) A comparison of alternative approaches to free energy calculations. J Phys Chem 98(5):1487–1493
57. Boresch S, Karplus M (1999) The role of bonded terms in free energy simulations: 1. Theoretical analysis. J Phys Chem A 103 (1):103–118
58. Boresch S, Karplus M (1996) The Jacobian factor in free energy simulations. J Chem Phys 105(12):5145–5154
59. Boresch S, Karplus M (1999) The role of bonded terms in free energy simulations. 2. Calculation of their influence on free energy differences of solvation. J Phys Chem A 103 (1):119–136
60. 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(6):529–539
61. Zacharias M, Straatsma TP, McCammon JA (1994) Separation-shifted scaling, a new scaling method for Lennard-Jones interactions in thermodynamic integration. J Chem Phys 100:9025–9031
62. 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
63. 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
64. Buelens FP, Grubmüller H (2012) Linear-scaling soft-core scheme for alchemical free energy calculations. J Comput Chem 33 (1):25–33
65. Gapsys V, de Groot BL (2017) pmx Webserver: a user friendly interface for alchemistry. J Chem Inf Model 57(2):109–114
66. Šali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 234(3):779–815
67. Schrödinger, LLC (2015) The PyMOL molecular graphics system, version 1.8, November 2015
68. Vriend G (1990) WHAT IF: a molecular modeling and drug design program. J Mol Graph 8 (1):52–56
69. 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 Struct Funct Bioinf 65 (3):712–725
70. Best RB, Hummer G (2009) Optimized molecular dynamics force fields applied to the helix-coil transition of polypeptides. J Phys Chem B 113(26):9004–9015
71. Lindorff-Larsen K, Piana S, Palmo K, Maragakis P, Klepeis JL, Dror RO, Shaw DE (2010) Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins Struct Funct Bioinf 78(8):1950–1958
72. Lindahl E (2015) Molecular dynamics simulations. In: Molecular modeling of proteins. Springer, Berlin, pp 3–26
73. Barua B, Andersen NH (2001) Determinants of miniprotein stability: can anything replace a buried H-bonded Trp sidechain? Lett Pept Sci 8(3–5):221–226
74. Barua B, Lin JC, Williams VD, Kummler P, Neidigh JW, Andersen NH (2008) The Trp-cage: optimizing the stability of a globular miniprotein. Protein Eng Des Sel 21 (3):171–185
75. Darden T, York D, Pedersen L (1993) Particle mesh Ewald: an Nlog(N) method for Ewald sums in large systems. J Chem Phys 98 (12):10089–10092
76. Essmann U, Perera L, Berkowitz ML, Darden T, Lee H, Pedersen LG (1995) A smooth particle mesh Ewald method. J Chem Phys 103 (19):8577–8593
77. 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(18):184103
78. 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
79. 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