

Are Alchemical Free Energy Calculations Reproducible?

Hannes H. Loeffler,^{*,†} Stefano Bosisio,[‡] Guilherme Duarte Ramos Matos,[¶]
Donghyuk Suh,[§] Benoit Roux,[§] David L. Mobley,^{||} and Julien Michel[‡]

Science & Technology Facilities Council, Daresbury, Warrington, WA4 4AD, United Kingdom, EaStCHEM School of Chemistry, University of Edinburgh, David Brewster Road, Edinburgh EH9 3FJ, UK, Department of Chemistry, University of California, Irvine, University of Chicago, and Departments of Pharmaceutical Sciences and Chemistry, University of California, Irvine

E-mail: Hannes.Loeffler@stfc.ac.uk

Abstract

Alchemical free energy calculations are an increasingly important modern simulation technique. Contemporary Molecular Dynamics software such as AMBER, CHARMM, GROMACS and SOMD include support for the method. Implementation details vary among those codes but users expect reliability and reproducibility, i.e. a simulation must yield a comparable free energy within statistical bounds regardless of code used. *Relative* alchemical free energy (RAFE) simulation is increasingly used to support molecular design activities, yet the reproducibility of the methodology has been less well tested

*To whom correspondence should be addressed

[†]Scientific Computing Department, STFC

[‡]University of Edinburgh

[¶]University of California, Irvine

[§]University of Chicago

^{||}University of California, Irvine

than its absolute counterpart. Here we present the results for RAFF calculations for hydration free energies of a set of small organic molecules and show that free energies can be satisfactorily reproduced with aforementioned codes. However, this requires attention to detail and package-specific protocols. The benchmarks and protocols reported here should be useful for the community to validate new and future versions of free energy calculations software.

1 Introduction

The free energy is a fundamental function of thermodynamics as it explains how processes in nature evolve. The equilibrium balance of products and reactants in a hypothetical chemical reaction can be immediately determined from the knowledge of the free energy difference of reactants and products and their concentrations. The free energy landscape of a given system, however, can be very complicated and rugged with barriers which impose limits on how fast the process can take place. It is therefore of little surprise that the determination of free energy changes is of utmost importance in the natural sciences, e.g. for binding and molecular association, solvation and solubility, protein folding and stability, partition and transfer, and design and improvement of force fields.

The calculation of free energies via molecular simulations¹⁻⁵ has been particularly attractive as it promises to circumvent certain limitations of experimental approaches. Specifically, processes can be understood at the atomic level and there is the potential that computational techniques can be more cost and time effective, especially if they can predict the properties of new molecules before their synthesis. Thus, a multitude of methods have been devised to make reversible work estimates accessible through computation.¹⁻⁵ However, the reliability of estimates is still very much a matter of concern.^{2,6}

Here we are interested in *alchemical* free energy methods because they are firmly rooted in statistical thermodynamics and thus give asymptotically correct free energy estimates, i.e. they are correct for a given potential energy function in the limit of sufficient simulation

time.^{1,7-9} The method has been applied in various forms for several decades now since the early days of computer simulation.¹⁰⁻¹⁵ The method has gained renewed attention in recent years — concomitant with improvements in computer hardware design — within the traditional equilibrium framework¹⁶⁻¹⁸ and also increasingly in combination with non-equilibrium techniques.¹⁹⁻²¹ The name comes from the nonphysical intermediates that often need to be created to obtain reliable estimates of free energy differences between physical end states and because parts or all of a molecule may appear or disappear in a transformation. In the context of force field methods the transformation takes place in parameter space, i.e. the various force field parameters are varied by scaling. This can be a particularly efficient approach as it does not require sampling of diffusive motions, avoid crossing prohibitively large energy barriers if pathways are not well chosen and is easier to automate.

Alchemical free energy simulations rely on the concept of thermodynamic cycles.¹⁴ As the free energy is a state function, the sum of free energy changes computed around any closed cycle must be zero. This also implies that the reversible work can be computed along conveniently chosen legs of the cycle, even if the cycle is artificial. For example, in Fig. 1 the relative free energy of hydration can be computed along the vertical legs, that is, following the physical process of moving a molecule from the gas phase to the liquid phase, or along the horizontal legs in a non-physical but computationally more efficient alchemical calculation.

Absolute (standard) alchemical free energy calculation has been of particular interest for many years.^{16-19,21,23} *Absolute* here really means that the equilibrium constant of a physical reaction, e.g. binding and dissociation, can be calculated directly by completely decoupling or annihilating a whole molecule from its environment and the term is mostly being used to discriminate against techniques usually referred to as *relative* (see below). We emphasize that the “absolute” approach will still result in a *relative* free energy between the state where the solute fully interacts with its environment and the state where it does not. Decoupling means the scaling of the non-bonded *inter*-molecular interactions between the perturbed group (all atoms that differ in at least one force field parameter between the end states)



Figure 1: The thermodynamic cycle to compute the relative free energy of hydration $\Delta\Delta G_{\text{hydr}} = \Delta G_{\text{sol}} - \Delta G_{\text{vac}} = \Delta G'' - \Delta G'$. The example is for the ethanol \leftrightarrow methanol transformation. Alchemical simulations are performed along the non-physical horizontal legs while vertical legs illustrate the physical process of moving a molecule from the vacuum to the solution. The latter is also accessible through absolute alchemical free energy simulation, see e.g. Ref. 22.

and its environment. We distinguish this from “annihilation” which also scales the *intra*-molecular interactions in addition to the inter-molecular interactions. These schemes may require two simulations along the opposite edges of a quadrilateral thermodynamic cycle but approaches that produce the reversible work directly in one simulation have been proposed too.^{24,25}

Relative alchemical free energy (RAFE) calculations mutate one molecule into another. RAFEs have proven useful for instance to rank sets of related molecules according to their binding affinity for a given receptor. This approach has recently gained increased traction in the context of relative free binding energies between small molecules, e.g. drug or lead like molecules and biomolecules.^{26,27}

RAFEs are readily implemented^{28,29} by making use of the single topology method.^{15,28,30} Single topology means that there is only one representation of the molecule to be mutated, also implying a single set of coordinates. Thus, atom types are directly transformed into the new type, typically by linearly scaling the force field parameters. In typical implementations,

disappearing/appearing atoms need to be balanced with “dummy” atoms to ensure constant number of atoms in both end states. Dummy atoms have no non-bonded interactions in the end state but retain the bonded terms of the original atom to avoid complications with unbound atoms³⁰ (see also wandering ligand problem, discussed below). However this need not be. For instance, the AMBER implementation is a special case as it does not require the user to define dummy atoms explicitly, i.e. as atoms with coordinates and zero end state parameters in the topology, but only to mark the disappearing and appearing atoms in the control file of the MD engine. The contributions of bonded terms that involve at least one dummy atom will not be factored into the free energy as it is assumed that those contributions will perfectly cancel out in the thermodynamic cycle.^{31,32}

The single topology approach²⁸ requires topological and structural similarity. But also chemical similarity is also of importance; e.g. chirality and binding modes where the relative three dimensional arrangement of groups in space must be taken into account. Furthermore, ring breaking is technically challenging²⁷ but it has also been shown that this can be done in certain circumstances.^{31,32}

When the two molecules are structurally dissimilar or do not spatially overlap, the dual topology method^{28,30} can be applied to compute relative free energies. In this approach *all* atoms of the end states are duplicated and thus both sets are present at all times but do not interact with each other. Only non-bonded interactions need to be scaled such that the disappearing end state corresponds to an ideal gas molecule.³⁰ This, however, comes with additional complications as two independent molecules can drift apart and so suffer from the “wandering” ligand problem as in absolute transformations.^{16–18} Technically, a dual topology calculation is the same as two absolute calculations run simultaneously in opposite directions. It has been shown though that with the introduction of special restraints or constraints this can be a viable option.^{33–35} Restraints between corresponding atoms, thus emulating the single topology approach, can also be used without affecting the free energy.³⁵ A recent alternative⁶ considered molecules with a common core where all atom types are the same.

The charges that would be typically different in individual parameterization due to the local chemistry were made equal. This means that the core does not need to be duplicated and thus is not included in the mutation. But all the chemistry is now exclusively handled by, possibly, just a few non-core atoms such that this approach may only be of limited use.

A covalent link, e.g. as in side-chain mutation simulations, provides a natural restraint such that dual topology simulations can be applied without further problems. Modern MD software, e.g. AMBER,³⁶ CHARMM,³⁷ GROMACS,³⁸ GROMOS³⁹ and SOMD,^{40,41} often provide a hybrid single/dual topology approach i.e. the user can specify which part of a perturbed group should be handled by which method.³²

As alluded to above, reliability is a principal matter of concern. In particular, we need to ensure reproducibility of free energy results among computer codes. To the best of our knowledge this has not been systematically tested yet for a set of different MD packages. However, there have been some recent efforts to test *energy* reproducibility across packages⁴² — a necessary but not sufficient prerequisite. Another study went further and also compared densities across packages, revealing a variety of issues.⁴³ For free energies, given a predefined force field and run-time parameters we ought to be able to obtain comparable free energy results within statistical convergence limits, though this has not yet been tested.

In practice, we have the problem, however, that the methodologies used in one MD program are not always present in another package, or the same functionality is provided via different algorithms (e.g. algorithms for pressure and temperature scaling, integrators, etc). Nevertheless, it is critical that free energy changes computed with different simulation software should be reproducible within statistical error, as this otherwise limits the transferability of potential energy functions, and the relevance of properties computed from a molecular simulation with a given package. This is especially important as the community increasingly combines or swaps different simulation packages within workflows aimed at addressing challenging scientific problems.^{44–48}

In this work we present the results of relative hydration free energies of a set of small

organic molecules (see Fig. 2). Solvation free energies have a wide range of uses and various methods exist to compute them.⁴⁹ They are also needed for calculations of a variety of important physical properties, and to calculate binding free energies where the simulation in solution (see Fig. 1) is combined with a mutation of the molecule bound to a partner.⁴⁹ A large database of hydration free energies computed from alchemical free energy (AFE) simulations, FreeSolv, has been presented recently⁵⁰ and was just updated.²² Here, we focus on the reproducibility of RAFF with the simulation programs AMBER, CHARMM, GROMACS and SOMD. We will discuss the reversible work results obtained with these packages and make recommendations regarding simulation protocols, setup procedures and analysis techniques. We will also deliberate on what needs to be done to progress the field, both from a usability perspective as well as from the view point of code development.

2 Methods

We will compare relative free energies calculated via simultaneous parameter change simulations — our “unified protocol” — and separated parameter change simulations — our “split protocol”— to relative free energy calculations obtained from two absolute hydration free energies — our “absolute protocol”. The unified protocol changes partial charges, van der Waals parameters, and bond parameters simultaneously along the alchemical path so that charge interactions and van der Waals interactions are varying simultaneously, while the split protocol separates the transformation of the van der Waals parameters from the charge transformation so that electrostatic interactions are modified separately from van der Waals interactions. The order in which this has to be done is detailed in section 2 of the SI. In the split protocol, the scaling of the bonded terms can be combined with either transformation.

2.1 Alchemical Free Energy Implementations

We begin by examining the differences in the alchemical free energy implementations of the four MD codes we consider — AMBER, CHARMM, GROMACS and SOMD. One key difference is in the softcore functions^{51,52} implemented in each code as summarised in section 1 of the SI. Softcore functions are used to avoid numerical and thus stability problems of the conventional van der Waals and Coulombic potentials⁵³ as they have singularities at zero distance (vertical asymptotes). Attempting to modify interactions by linearly scaling back these interaction potentials as a function of an interaction parameter, λ , causes these functions to increasingly behave like hard-sphere potentials as $\lambda \rightarrow 0$. This means that there is a discontinuous change between $\lambda = 0$ where particles can overlap, and $\lambda = \epsilon$ (where ϵ is very small) where particles still behave like hard though minuscule spheres. This can lead to strongly fluctuating forces/energies and to severe instabilities in the integrator, as well as numerical errors in post processing analyses even when simulations do terminate normally.^{51–53}

Another difference is how the codes scale force field parameters (“parameter scaling”) and/or the energy (“energy scaling”).³⁰ In the former case each parameter is scaled individually, e.g. in the case of a harmonic bond or angle term, the force constant and the equilibrium distance/angle are scaled. In the latter case, the total energy is scaled, all at once or, equivalently for each individual force field contribution. The two approaches are not mathematically equivalent and so the pathway through state space or alchemical space is different. It should be noted, however, that modification of Lennard-Jones interactions is handled through softcore functions in modern codes (as is, in some cases, that for electrostatic interactions) and thus parameter scaling^{28,54} is now only used for bonded terms and sometimes electrostatic interactions, at least in modern codes.

One more important issue is whether the code allows constraints to be applied to bonds which change bond lengths in a transformation e.g. C–H to C–C. Changes in bond length need to account for the associated change in the free energy. These and other details will be

outlined below.

AMBER. This code is strictly dual topology and all terms are energy scaled. The code allows, however, mapping of atoms in a single topology fashion and computes these non-softcore atoms by linearly scaling the energy and forces for each atom in the pair. The perturbed group must be entirely duplicated, i.e. for **sander** this means two topology files with one end state each, and for **pmemd** both end states in one topology file. The softcore potential applies to any atom chosen by the user which may include atoms that actually have an equivalent in the other state. Non-softcore atoms must still match 1:1 in both states. Explicit dummy atoms are not needed as the code will only compute bonded contributions for “real” atoms and thus ignores bonded energies involving dummy atoms. We will call this the “implicit dummy protocol”. The code cannot handle bond length changes involving a constraint. There is only one global λ for parameter transformation. Protocols that couple only some parameters (split protocols, see below) must be emulated through careful construction of topologies e.g. by keeping fixed the Lennard-Jones (LJ) and bonded terms at the initial state for a charge transformation. The setup for the two states must therefore use identical atom types for both end states while the charges are individually assigned.

CHARMM. The PERT module duplicates the topology similar to **sander** but mapped atoms are given in the topology only once. The module requires balancing with explicit dummy atoms. All terms are energy-scaled. The PSSP softcore potential is applied to *all* atoms in the perturbed group (see section 1 in the SI). The code can handle constraints of changing bond lengths in the perturbed group but this may cause incorrect results with PSSP softcores (Stefan Boresch, private communication). There is only one global λ for parameter transformation, however, the scripting facilities in CHARMM allow run time modification of topologies e.g. by setting charges or LJ parameters to arbitrary values.

GROMACS. This code uses a single topology description. Bonded terms are strictly parameter-scaled, which requires proper balancing of multi-term dihedrals i.e. each individual term in the Fourier series must have an equivalent in both end states. If the term does not exist it must be created with parameters zeroing its energy. The softcore potential applies to dummy atoms only determined from atoms having zero Lennard-Jones parameters in the end states. The code allows changing bond lengths involving constraints within the perturbed group but this can lead to instabilities and wrong results (Michael Shirts, private communication). There are separate λ s for LJ, Coulomb and bonded parameters (and some other) which allows easy implementation of split protocols.

SOMD. SOMD is a software built by linking Sire⁴⁰ and OpenMM⁴¹ molecular simulation libraries. This code uses a single topology description. The final state is constructed at run time from the initial state with a “patch” (list of force field parameters to be modified). All dummy atoms needed to describe the transformation must be present in the initial state. Bond and angle terms are parameter-scaled while the dihedral term is energy-scaled. The softcore potential applies to atoms that become dummy atoms in one end-state. Dummy atoms are specified by a keyword in the patch file. The code cannot handle constraints of changing bond lengths in the perturbed group. There is only one global λ for parameter scaling. Separated protocols (see below) must be emulated through careful construction of the patch file.

2.2 RAFE Setup

The setup for all relative free energy simulations has been carried out with the tool FESetup⁴⁷ (version 1.2). FESetup is a perturbed topology writer for AMBER, CHARMM, GROMACS, SOMD and also NAMD⁵⁵ (within the limits of the dual topology approach). The tool makes use of a maximum common substructure search algorithm to automatically compute atoms that can be mapped, i.e. atoms that have a direct relationship to an equivalent atom

in the other state – atoms undergoing atom type conversion or modification. The only current limit is that rings are required to be preserved.³¹ With this strategy a maximal single topology description is achieved: any atom that does not match will be made a dummy atom. FESetup allows equilibration of the solvated simulation systems and ensures that “forward” and “backward” simulations have the same number of total atoms. With SOMD the mass of each perturbed atom is taken as the mass of the heavier end-state atom (e.g. a hydrogen atom that is perturbed to a carbon atom has an atomic mass of 12 amu at all lambda values) The masses of perturbed atoms are set to the mass of the heavier atom description they are being perturbed to for SOMD. The other codes use the atom masses of the initial state (AMBER, CHARMM?) or allow the user to define how masses vary as a function of lambda (GROMACS). The tool creates all input files with control parameters, topologies and coordinates as required for RAFF simulations. Full details on FESetup can be found in Ref. 47.

Figure 2 shows all 18 transformation considered in this study including “forward” and “backward” mutations. In the limit of sufficient sampling, RAFF simulations do not have a directionality with respect to the coupling (order) parameter λ . But to test for possible discrepancies we have run simulations in both directions. As we shall discuss in the Results section we do see differences in some cases.

The ethane \rightarrow methanol transformation is traditionally regarded as a standard test for RAFF simulations.^{15,56} The other transformations are centered around mutations from and to methane, and are meant to mimic components of typical transformations that could be attempted in the context of e.g. protein-ligand binding calculations. The 2-cyclopentanylindole to 7-cyclopentanylindole (2-CPI to 7-CPI in our notation) transformation has been added to include both deletion as well as insertion of sub-parts of the perturbed group in one transformation, an aspect not tested by the other transformations. For neopentane \rightarrow methane we point out that there are two alternative mappings possible, see Figure 2. One mapping has methane matched to a terminal methyl (green) and the other one has the methane car-

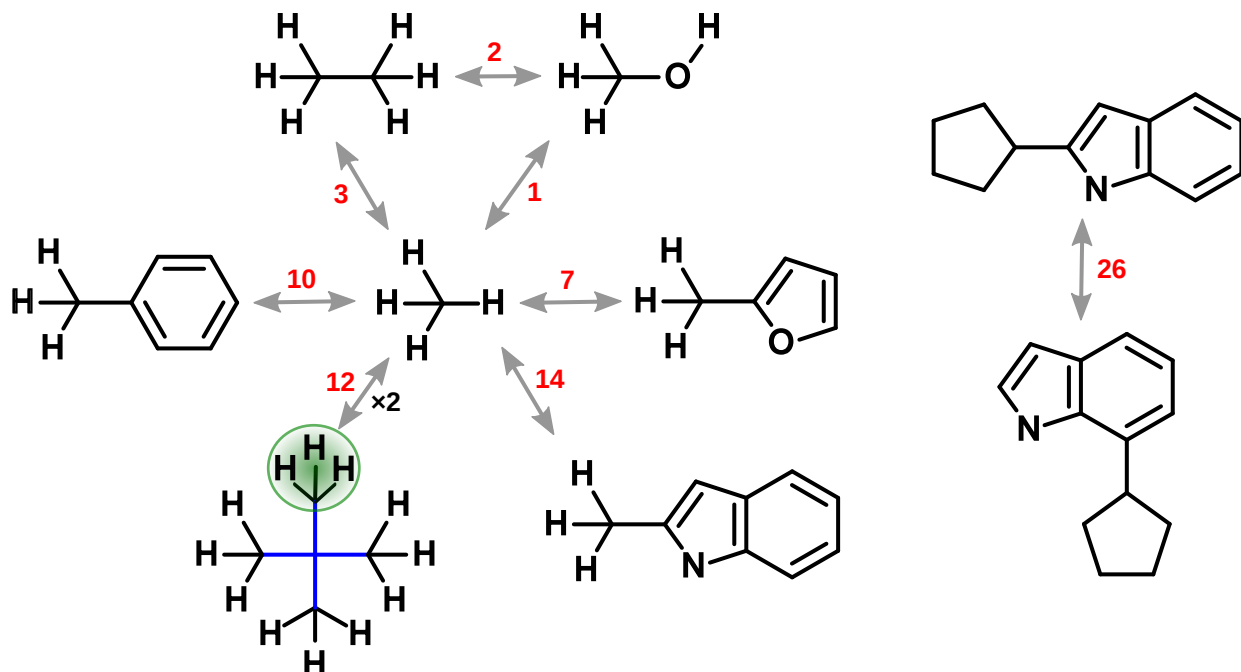


Figure 2: The thermodynamic cycles considered in this study. To compute the free energy of hydration, all pair-wise transformations have to be carried out once in solution and once in vacuum. Green and blue colours in neopentane show two alternative mappings for methane. The numbers in red denote the number of dummy atoms.

bon matched with the central carbon in neopentane (blue). The first approach will be called “terminally mapped” and the second one “centrally mapped”.

2.3 RAFF Simulation Protocols

One of the major concerns of this reproducibility study is to ensure consistency in the applied protocols. This is complicated by the fact that a given MD software may employ a wide range of methods and algorithms that may not be available in other MD software. For example, pressure and temperature scaling, integrators and other algorithms can be very different. It is also unclear if and how implementation details can affect results, in particular see subsection 2.1 discussing the implementation details of alchemical free energy simulation in code.

In this study we consider a set of simple organic molecules (see Figure 2). As the focus here is on probing for reproducibility among various MD packages, we chose fairly small,

rigid and neutral molecules to minimise statistical sampling errors, and avoid difficulties with charged particles.^{57,58} The force field was chosen to be GAFF⁵⁹ (version 1.8), utilising AM1/BCC charges^{60,61} for the solute and TIP3P⁶² for the solvent. Charges were computed with the `antechamber` program and missing bonded and vdW terms were generated with the `parmchk2` program, both from the AmberTools16 distribution. The quality of free energies of various small molecule force fields has been discussed elsewhere, see e.g. Refs. 63,64.

While the MD packages principally allow a “one-step” transformation,⁶⁵ that is, both LJ and Coulombic softcore potentials vary simultaneously, it can be more efficient to carry out a split protocol.^{66,67} In such a protocol the charges are transformed linearly between the end states followed by a mutation of the van der Waals parameters using a softcore potential^{51,52} (see section 1 in the SI for details) on the LJ term only. It is important to note that in the split protocol charges have to be switched off before LJ parameters (and vice versa for the transformation in opposite direction) to avoid collapse of other atoms, e.g. solvents, onto a “naked” charge^{65,68,69} see section 2 in the SI.

All simulations were started from simulation boxes prepared by FESetup.⁴⁷ It should be noted, however, that in constructing the perturbed system steric overlaps between the solute and the solvent may happen. This is because each unperturbed solute is independently equilibrated but the final perturbed system combined from those, potentially differently sized solutes. To make the number of atoms the same for forward and backward setups the water coordinates of the larger of the two boxes are chosen. Thus, in transformation from a smaller to a larger solute, water molecules may be in close proximity to the solute. The production simulations were run at 298.15 K and 1.0 bar.

AMBER. We made use of AMBER16 for all free energy calculation. The starting coordinates were usually taken directly from the pre-equilibrated setup step but no further λ specific equilibration was carried out, i.e. RAFE MD simulations were started with new velocities appropriate for the final simulation temperature. In a very few cases it was nec-

essary to use coordinates from the end of the simulation at a nearby λ state because of simulation instabilities. This happened in transformations with a larger number of dummy atoms. Water hydrogens (TIP3P) were constrained with SHAKE. None of the atoms in the perturbed group were constraint and hence the time step set to 1 fs. We also tested an alternative protocol with SHAKE also on bonds that do not change during transformation and a time step of 2 fs (see SOMD protocol below). The temperature was controlled through a Langevin thermostat with a friction constant of 2.0 ps^{-1} and pressure rescaling through a Monte Carlo barostat with 100 steps between isotropic volume change attempts. Long-range electrostatics in solution was handled with PME and a cutoff of 8.0 \AA for both Coulomb and vdW interactions. In vacuum no cutoff was used.

CHARMM. The CHARMM c40b1 molecular dynamics package was used to carry out all relative hydration free energy simulations. 21 evenly spaced windows were used and all windows were run for 1.5 ns with a timestep of 1 fs. Most windows used the same pre-equilibrated configuration, but a few windows (simulations involving perturbing hydrogen to heavy atom or vice versa) at end-point lambdas were started with their own equilibration using 0.1 fs^{-1} to 0.5 fs^{-1} to avoid numerical instabilities. Constant temperature and pressure control was done with the Berendsen weak coupling method, with a compressibility of $4.63 \times 10^{-5} \text{ atm}^{-1}$ and temperature and pressure coupling constants of 5.0 ps^{-1} . A 12.0 \AA atom-based cutoff was used for non-bonded interactions and Particle Mesh Eward (PME) with order of 6 was used to calculate electrostatic forces along with periodic boundary condition. SHAKE was used to constrain water hydrogens only. The PERT module in CHARMM was used to handle alchemical transformations and PSSP softcore potential functions were used to avoid numerical instabilities. In vacuum, no cutoff was used.

The PERT module can currently not use force switching (option `VFSwitch`) for LJ potentials with softcores. Current CHARMM force fields, however, are parameterized to use force switching. In this work we have used potential switching only (option `VSwitch`).

GROMACS. We used GROMACS 4.6.7 to carry out simulations for the relative hydration free energies ($\Delta\Delta G_{\text{hydr}}$). Each transformation had its Gibbs free energy calculated: (i) in a single topology approach in which LJ energy terms were changed separately from the electrostatic and bonded components; (ii) in a single topology approach in which bonded, LJ, and electrostatic terms are changed together; and (iii) via the difference between two absolute calculations. In the first two cases, each alchemical transformation was described by 31 and 16 states, respectively, and simulated for 4.2 ns with time steps of 1.0 fs in water and vacuum. Atomic masses were not changed along the alchemical path; their change affect kinetic energy only and do not contribute to the free energy change on this case. We used the Langevin integrator implemented in GROMACS with a default friction coefficient of 1.0 ps/ m_{atom} , where m_{atom} is the mass of the atom. Absolute hydration free energies were calculated from 5 ns Langevin dynamics at 298.15 K. We used the 20-step alchemical protocol where charge coupling and LJ coupling were dealt with separately along the path.^{22,50} No bond constraints were used. A Parrinello–Rahman barostat with $\tau_p = 10$ ps and compressibility equal to $4.5 \times 10^{-5} \text{ bar}^{-1}$. We used two methods to calculate electrostatic interactions: Particle Mesh Ewald (PME) and charge group-based Reaction Field with a dielectric of 78.3, as implemented in the software. We set the electrostatic and non-bonded cutoffs to 10.0 Å; a switch was applied to the latter starting at 9.0 Å. Non-bonded interactions in the vacuum legs of the simulation were calculated to a cutoff of 50.0 Å. PME calculations were of order 6 and had a tolerance of 1.0×10^{-6} , with a grid spacing of 1.0 Å. All transformations required the use of softcore potentials to avoid numerical problems in the free energy calculation. We chose the 1–1–6 softcore potential for LJ terms ($\alpha=0.5$ and $\sigma=0.3$) for atoms whose parameters were being perturbed and used the default softcore Coulomb implementation in paths where charges, LJ, and bonded terms were modified together, but no soft core potentials were applied to Coulomb interactions when electrostatic interactions were modified separately.

SOMD. All simulations were carried out with SOMD from the Sire 2016.1 release^{40,41}. Each alchemical transformation was divided into 17 evenly spaced windows and simulated for 2 ns each both in water and in vacuum. A velocity-Verlet integrator was employed with a 2 fs time step, constraining only bonds involving hydrogens which are not alchemically transformed. This approach will be called the “unperturbed H bond constraint protocol”. Referring to eq. S1 the softcore parameters were set to default values for all the transformations, specifically $n = 0$ for Coulombic interactions and $\alpha = 2.0$ for the LJ potential.³³ A shifted atom-based Barker-Watts reaction field⁷⁰ with a dielectric constant of 78.3 was adopted for the solution phase simulations with a cutoff of 10 Å. A similar cutoff was used for Lennard-Jones interactions. Temperature control was achieved with the Andersen thermostat⁷¹ with a coupling constant of 10 ps⁻¹. A Monte Carlo barostat assured pressure control, with isotropic box edge scaling moves attempted every 25 time steps. The reaction field was not employed in the vacuum legs, where a Coulombic potential without cutoff was used. This leads to an inconsistent description of the intramolecular electrostatic interactions of the solute in the solvated and vacuum phases. To maintain a consistent description of intramolecular energetics across vacuum and water legs, a free energy correction term ΔG_c was evaluated as detailed in Ref. 72). The ΔG_c term was obtained via post-processing of the end state trajectories of each water phase simulation, using the Zwanzig relationship:⁷³

$$\Delta G_c = -\beta^{-1} \ln \langle \exp [-\beta (U_{\text{ic,nc}}(\mathbf{r}) - U_{\text{ic,sim}}(\mathbf{r}))] \rangle_{\text{sim}} \quad (1)$$

where $U_{\text{ic,nc}}(\mathbf{r})$ is the solute intramolecular electrostatic-no cutoff potential that depends on the coordinates \mathbf{r} of the solute and is given by Coulomb’s law computed without cutoffs. $U_{\text{ic,sim}}(\mathbf{r})$ is the intramolecular electrostatic potential term as computed in the simulation with the shifted atom-based Barker-Watts Reaction Field cutoff.

To compute absolute free energy of hydration a double annihilation technique^{16,23,72} was adopted. Initially the solute atoms’ partial charges are turned off both in water and in

vacuum (discharging step), giving a discharging free energy change $\Delta G_{\text{solv}}^{\text{elec}}$ and $\Delta G_{\text{vac}}^{\text{elec}}$ respectively. Secondly, the molecule is fully decoupled from the surrounding environment by switching off the Lennard-Jones terms (vanishing step), giving a $\Delta G_{\text{solv}}^{\text{LJ}}$ free energy change in water phase and $\Delta G_{\text{vac}}^{\text{LJ}}$ free energy change in vacuum. Next, the absolute free energy of hydration ΔG_{wat} is computed as:

$$\Delta G_{\text{wat}} = (\Delta G_{\text{solv}}^{\text{elec}} + \Delta G_{\text{solv}}^{\text{LJ}}) - (\Delta G_{\text{vac}}^{\text{elec}} + \Delta G_{\text{vac}}^{\text{LJ}}) + \Delta G_c \quad (2)$$

The same setup for relative free energy calculations was followed to compute absolute free energy of hydration. Thus, 17 evenly spaced λ windows were used and simulations were run for 2 ns employing a velocity-Verlet integrator with 2 fs time step, constraining only the hydrogen bonds which are not alchemically transformed.

2.4 Free Energy Estimations

In this work we primarily focus on TI as this is supported by all the tested MD packages “out-of-the-box”, whereas BAR and MBAR are not. Equation 3 computes the free energy as

$$\Delta G = \int_{\lambda=0}^{\lambda=1} \left\langle \frac{\mathcal{H}(\mathbf{q}, \mathbf{p}; \lambda)}{\partial \lambda} \right\rangle_{\lambda} d\lambda \quad (3)$$

where $\mathcal{H}(\mathbf{q}, \mathbf{p}; \lambda)$ is the Hamiltonian as a function of the coordinate vectors \mathbf{q} and the momentum vectors \mathbf{p} , and parametric dependence on the coupling parameter λ . The angle brackets denote the ensemble average of the gradient of the Hamiltonian with respect to λ at a given λ value. An AFE simulation is typically carried out in a series of equilibrium simulations at discrete values of λ but the gradient can also be evaluated with a continuously varying coupling parameter as a function of the simulation time. The free energy is finally computed through a suitable numerical integration method.

Results from additional estimators will be given where available. We have used the alchemical analysis tool⁷⁴ for analysis of all free energies. This tool provides various estimators

such as TI, TI with cubic splines, BAR and MBAR. All data was sub-sampled to eliminate correlated data.

All RAFF simulations were run in triplicate in forward as well as backward direction for a total of 6 simulations per mutation. The final hydration free energy $\Delta\Delta G_{\text{hydr}}$ was computed as the average for each direction separately. For comparison we have also calculated the absolute (standard) hydration free energies for all molecules in Figure 2.

To estimate the reliability and convergence of the results, the standard error of the mean (SEM) has been calculated. The SEM is defined as

$$\text{err}(\Delta\Delta G_{\text{hydr}}) = \frac{\sigma}{\sqrt{n}} \quad (4)$$

where σ is the sample standard deviation and n is the size of the uncorrelated sample. The SEM for component free energies is combined as

$$\text{err}(\text{combined}) = \sqrt{\sum_i \sigma_i^2}. \quad (5)$$

which is appropriate if the property to be computed is a sum of contributions.

We also make use of the mean absolute error MAE (also called mean unsigned error, MUE) to compare data sets.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (6)$$

where n is the total number of samples, y_i and x_i are the i -th datum to be compared.

3 Results

We will use absolute hydration free energies here as our standard point of comparison because for the present dataset these are easily calculated with high precision²² and are considerably simpler to set up and implement than relative calculations. Prior work actually compared calculated absolute hydration free energies across different codes with significant success⁷⁵

(see Fig. 7 in the reference), further supporting this view.

Table 1 summarizes results for relative free energies of hydration obtained from absolute transformations which are given in more detail in Table S1. The table shows the data from simulations with the recommended protocol for each MD code, as discussed in detail in the following subsections. The $\Delta\Delta G_{\text{hydr}}$ obtained with the various MD packages in this way agree quite well, although some larger deviations are apparent as well. GROMACS predicts a smaller $\Delta\Delta G_{\text{hydr}}$ for the methanol transformations by about $0.2 \text{ kcal mol}^{-1}$. Similarly, there are some discrepancies in the toluene, 2-methylfuran and 2-methylindole cases where it is mostly CHARMM that produces somewhat smaller $\Delta\Delta G_{\text{hydr}}$ with $0.4 \text{ kcal mol}^{-1}$ difference between the smallest and highest values. The largest deviation can be found for the largest transformation (2-CPI to 7-CPI) with the AMBER result higher by $0.3\text{--}0.5 \text{ kcal mol}^{-1}$ than any other MD package.

Table 1: Comparing relative free energies of hydration for various MD packages as obtained from the absolute protocol.

transformation		AMBER ^a	CHARMM ^a	GROMACS ^b	SOMD ^a
ethane	methane	-0.02 ± 0.01	-0.08 ± 0.01	-0.04 ± 0.01	-0.05 ± 0.02
methanol	methane	6.20 ± 0.01	6.15 ± 0.02	5.95 ± 0.01	6.21 ± 0.06
ethane	methanol	-6.22 ± 0.01	-6.23 ± 0.02	-5.98 ± 0.01	-6.26 ± 0.05
toluene	methane	3.19 ± 0.01	3.00 ± 0.01	3.16 ± 0.01	3.07 ± 0.03
neopentane	methane	-0.13 ± 0.02	-0.18 ± 0.02	-0.14 ± 0.01	-0.19 ± 0.06
2-methylfuran	methane	2.96 ± 0.02	2.79 ± 0.01	2.95 ± 0.01	2.90 ± 0.03
2-methylindole	methane	8.72 ± 0.01	8.36 ± 0.02	8.79 ± 0.02	8.57 ± 0.03
2-CPI	7-CPI	0.39 ± 0.04	-0.10 ± 0.04	0.02 ± 0.05	0.08 ± 0.14

^aunified protocol.

^bsplit protocol with PME.

Table 2 shows the MAD between SOMD, GROMACS, AMBER and CHARMM. The SOMD code produces figures that agree the most with all other MD packages (MAD ca. $0.1 \text{ kcal mol}^{-1}$), and the MAD reaches just under $0.2 \text{ kcal mol}^{-1}$ for CHARMM/GROMACS, and CHARMM/AMBER.

Having established the predictive value from absolute transformations we now turn to

Table 2: Mean Absolute Deviations (MAD) (kcal mol^{-1}) between relative free energies obtained with the absolute protocol for the SOMD, GROMACS, AMBER and CHARMM packages.

Package	GROMACS	AMBER	CHARMM
SOMD	0.12	0.09	0.08
GROMACS		0.13	0.17
AMBER			0.16

computing $\Delta\Delta G_{\text{hydr}}$ from relative mutations. Table 3 summarizes the results for the four MD packages. Again the data is from the recommended protocol (see detailed discussions in the following subsections).

CHARMM tends to show smaller relative free energies for a number of transformations: toluene, neopentane, 2-methylfuran and 2-methylindole. SOMD displays smaller $\Delta\Delta G_{\text{hydr}}$ in the methanol and toluene transformations. The largest discrepancy, however, is in the neopentane transformation with central mapping where AMBER with implicit dummy atoms is about $0.5 \text{ kcal mol}^{-1}$ higher and CHARMM about $0.3 \text{ kcal mol}^{-1}$ lower than the other two codes. The terminal mapped neopentane case reveals AMBER to be in line with GROMACS and SOMD while CHARMM’s results are essentially the same as in the central mapped transformations. AMBER deviates also quite strongly from the other codes in the cyclopentanyll indole cases.

The MADs of the relative free energy simulations are presented in Table 4. They tend to be larger than the MADs from the absolute simulations (Table 2) and reach $0.3 \text{ kcal mol}^{-1}$ for CHARMM compared with AMBER and GROMACS.

We can also compute the cycle closure error from Table 5 for the closed cycle ethane \rightarrow methanol \rightarrow methane \rightarrow ethane (see Figure 2). For the implicit dummy simulation with AMBER we calculate a cycle error for $\Delta\Delta G_{\text{hydr}}$ of $(0.069 \pm 0.041) \text{ kcal mol}^{-1}$ and for the explicit dummy simulation the error is $(-0.016 \pm 0.047) \text{ kcal mol}^{-1}$.

Table 3: Comparison of relative free energies of hydration for various MD packages as obtained from relative transformations via unified or split protocols. Signs of the backward transformation have been reverted to correspond to the forward transformation.

		AMBER ^a		CHARMM ^b	GROMACS ^a	SOMD ^b
		implicit ^c	explicit ^c			
ethane	methane	0.02 ± 0.01	-0.13 ± 0.02	-0.11 ± 0.02	-0.04 ± 0.02	-0.01 ± 0.05
methane	ethane	0.00 ± 0.03	-0.19 ± 0.03	-0.06 ± 0.01	-0.02 ± 0.01	-0.04 ± 0.02
methanol	methane	6.19 ± 0.01	6.20 ± 0.02	6.17 ± 0.01	6.20 ± 0.01	5.99 ± 0.05
methane	methanol	6.20 ± 0.03	6.15 ± 0.01	6.20 ± 0.01	6.20 ± 0.01	5.96 ± 0.04
ethane	methanol	-6.20 ± 0.01	-6.27 ± 0.01	-6.26 ± 0.01	-6.19 ± 0.01	-6.09 ± 0.03
methanol	ethane	-6.20 ± 0.01	-6.25 ± 0.01	-6.29 ± 0.01	-6.19 ± 0.01	-6.09 ± 0.02
toluene	methane	3.24 ± 0.02	3.39 ± 0.02	2.93 ± 0.02	3.21 ± 0.01	2.89 ± 0.09
methane	toluene	3.42 ± 0.03	3.52 ± 0.03	2.96 ± 0.02	3.20 ± 0.01	3.06 ± 0.02
neopentane ^d	methane	0.32 ± 0.04	-0.03 ± 0.06	-0.44 ± 0.01	-0.15 ± 0.02	-0.20 ± 0.05
methane ^d	neopentane	0.25 ± 0.03	-0.07 ± 0.03	-0.33 ± 0.02	-0.16 ± 0.05	-0.13 ± 0.05
neopentane ^e	methane	-0.13 ± 0.01	-0.12 ± 0.02	-0.65 ± 0.02	-0.14 ± 0.01	-0.11 ± 0.01
methane ^e	neopentane	-0.13 ± 0.03	-0.12 ± 0.03	-0.49 ± 0.02	-0.18 ± 0.03	-0.10 ± 0.06
2-methylfuran	methane	3.09 ± 0.01	3.10 ± 0.01	2.76 ± 0.03	2.93 ± 0.05	2.92 ± 0.05
methane	2-methylfuran	3.10 ± 0.03	3.15 ± 0.03	2.76 ± 0.02	2.96 ± 0.01	2.83 ± 0.03
2-methylindole	methane	8.78 ± 0.03	8.78 ± 0.04	8.34 ± 0.01	8.73 ± 0.03	8.64 ± 0.06
methane	2-methylindole	9.14 ± 0.02	9.13 ± 0.03	8.42 ± 0.02	8.74 ± 0.01	8.67 ± 0.08
2-CPI ^f	7-CPI	0.36 ± 0.03	0.63 ± 0.06	-0.01 ± 0.01	-0.03 ± 0.03	-0.11 ± 0.07
7-CPI ^f	2-CPI	0.34 ± 0.05	0.50 ± 0.03	0.03 ± 0.01	-0.20 ± 0.04	-0.01 ± 0.08

^asplit protocol.

^bunified protocol.

^cusing either the implicit or the explicit dummy atom approach.

^dcentral mapping.

^eterminal mapping.

^fpartial re/discharge i.e. only the charges of the appearing and the disappearing 5-rings are switched.

Table 4: MAD (in kcalmol⁻¹) comparing relative free energies from relative simulations between SOMD, GROMACS, AMBER and CHARMM.

Package	GROMACS	AMBER	CHARMM
SOMD	0.18	0.23	0.19
GROMACS		0.14	0.31
AMBER			0.32

3.1 AMBER

Using AMBER for RAFE simulations has revealed several problems with the implementation. Some bugs were identified and the developers have fixed those for AMBER16, e.g. energy minimization in **sander** led to diverged coordinates for mapped atoms. For a single topology description, however, it is necessary to have the same coordinates. Other issues are that vacuum simulations can only be carried out with the **sander** program because **pmemd** cannot handle AFE simulations in vacuum at the moment. This will, however, be rectified in future versions.⁷⁶ A disadvantage of **sander** is that it cannot be used to simulate the λ end points⁷⁷ such that the TI gradients need to be extrapolated (minimum and maximum allowed λ s are 0.005 and 0.995).

Also, **sander** considers the whole system as the perturbed region while **pmemd** restricts this to a user chosen atom selection. This has obvious implications for performance.⁷⁷

We also found that, in contrast to the other three codes, AMBER does not yield correct relative free energies with the unified protocol, i.e. when all force field parameters are scaled simultaneously (see Table S2). This appears to be a problem when more than a few dummy atoms are involved, while the unified protocol works for the smaller transformations (refer to Figure 2). The split RAFE protocol and absolute free energies, however, are very close to the other MD packages as demonstrated in Table 5 below.

End point geometries appear to be another issue with AMBER simulations in both solution and vacuum. This is most obvious in the neopentane \rightarrow methane test case with central mapping (see RAFE Setup and Figure 1). As shown in Figure S3, the methane end state exhibits incorrect distances between the carbon and the four attached hydrogens of approximately 1.23 Å. This value is about 1.12 Å for the terminal dummy atoms in the other test cases but still higher than the expected 1.09 Å on average. Figure S3 demonstrates how this depends on the number of dummy atoms immediately surrounding the central atom.

We also compare free energies obtained from the implicit dummy approach in AMBER with results from explicit dummy atom simulations and results from absolute transforma-

tions, see Tables 1 and 3. The relative simulations have been carried out with the split protocol while the absolute simulations used a unified protocol throughout. SHAKE was explicitly deactivated for all bonds in the perturbed region in these protocols. Table 5 shows selected results for transformations with SHAKE enabled for all bonds to hydrogens except those bonds that change bond length during transformation.

Table 5: Comparing AMBER results for simulations with various split protocols. The emphasis is here on the data with SHAKE enabled and a time step of 2 fs (last column). Implicit, explicit and absolute protocols had SHAKE disabled and a time step of 1 fs. Signs of the backward transformation have been reverted to correspond to the forward transformation.

transformation		implicit $\Delta\Delta G$	explicit $\Delta\Delta G$	absolute ΔG	SHAKE ^a $\Delta\Delta G$
ethane	methanol	-6.20 ± 0.01	-6.27 ± 0.01	-6.22 ± 0.01	-6.18 ± 0.01
methanol	ethane	-6.20 ± 0.01	-6.25 ± 0.01		
toluene	methane	3.24 ± 0.02	3.39 ± 0.02	3.19 ± 0.01	3.27 ± 0.03
methane	toluene	3.42 ± 0.03	3.52 ± 0.03		
neopentane ^b	methane	0.32 ± 0.04	-0.03 ± 0.06	-0.13 ± 0.02	0.35 ± 0.02
methane ^b	neopentane	0.25 ± 0.03	-0.07 ± 0.03		
neopentane ^c	methane	-0.13 ± 0.01	-0.12 ± 0.02		
methane ^c	neopentane	-0.13 ± 0.03	-0.12 ± 0.03		

^aimplicit dummy atom protocol with $\delta t = 2$ fs and SHAKE on all H-bonds except perturbed bonds.

^bcentral mapping.

^cterminal mapping.

The time step has been increased from 1 fs as used in the other three protocols to 2 fs. As the results are essentially the same as the non-SHAKE simulations, this SHAKE protocol appears to be a viable solution to increase the performance of RAFE simulations. We have repeated this protocol with AMBER in response to the good results obtained with SOMD using this implementation. From a practical point of view, AMBER uses an *atom* based mask for bond SHAKEs such that the mask must be set for the hydrogens in question while the same is not possible for their non-H counter-part in the other state because *all* bonds emanating from this atom would be affected.

In general, the free energies computed with each approach are in good agreement with each other and with the results of the other MD packages (Tables 1 and 3). There are,

however, a few notable deviations. Neopentane \rightarrow methane with central mapping differs from the result with terminal mapping by about $0.4 \text{ kcal mol}^{-1}$. The terminal mapping and the free energies from the explicit dummy simulations are, however, consistent with the absolute transformations (Table 1). We also observe a systematic deviation between forward and backward vacuum transformations in the 2-methylindole simulation (see Table S3. The gradient is consistently shifted by $0.2\text{--}0.4 \text{ kcal mol}^{-1}$ for each λ step of the vdW plus bonded transformation with both implicit and explicit dummy atoms.

3.2 CHARMM

CHARMM for alchemical free energy calculation (AFE) has been widely used with PERT module, but few bugs not previously reported in CHARMM c40b1 were found and careful AFE setup is needed to get robust and accurate result. Bugs regarding TI gradient accumulation on parallel version were identified and fixed by Dr. Stefan Boresch.

The PERT module does not allow a hydrogen bond constraint (SHAKE) to be applied on the perturbed region, and this requires end point lambdas to be equilibrated carefully. These windows at end-point lambda were started with their own equilibration using 0.1 fs^{-1} 0.5 fs^{-1} before the run. VSwitch option which is a switching function applied for potential is used, since the PERT module is not compatible with VFSwitch option that is a switching function applied to force.

PSSP softcore potential function cannot handle Long-Range Correction (LRC) correctly. This effect is not clearly shown when the initial and final states are comparable in size, but the deviation becomes larger for large changes or for absolute alchemical free energy calculations. It is necessary to disable the LRC to obtain consistent free energies from relative and absolute alchemical free energy calculation protocols.

Figure 3 shows the effect of turning LRC on/off as function of the cut-off values on the absolute solvation free energy of methane. The switching function starts 2 \AA before the cut-off value. The figure shows that cutoff values greater than 10 \AA give consistent

results, if the LRC has been disabled. The bottom graph shows the relative solvation free energy of 2-methylindole to methane from direct RAFF and from AAFEs of methane and 2-methylindole with LRC on/off. This shows that the protocols will be inconsistent by ca. $0.8 \text{ kcal.mol}^{-1}$ if the LRC is not disabled.

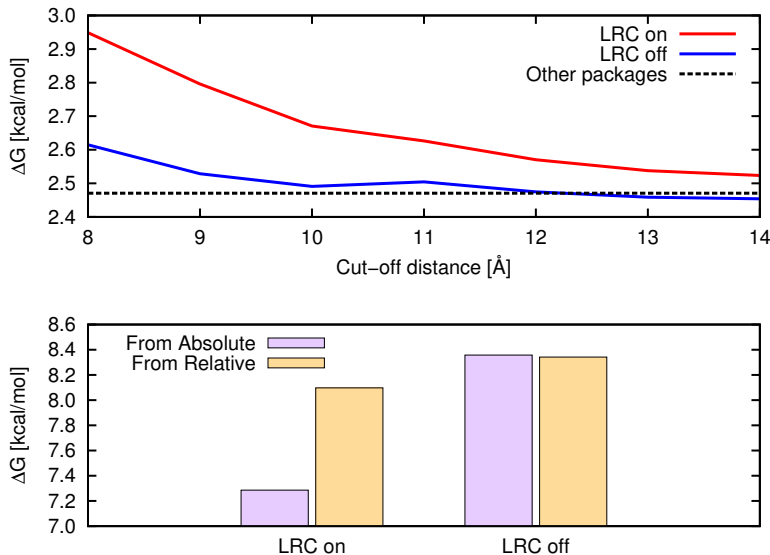


Figure 3: (top) Absolute solvation free energy of methane as a function of different cut-off values and presence or absence of a Long Range Correction term. (bottom) Relative solvation free energy between 2-methylindole and methane from relative or absolute alchemical free energy protocols and with/out a LRC term.

3.3 GROMACS

GROMACS has some run input options which can simplify the procedure for setting up free energy calculations. Specifically, `couple-moltype` implicitly defines the initial and final states by giving a special tag to a molecule and controls whether intramolecular interactions of the tagged molecule are retained or not along the alchemical path. It should be used in absolute free energy calculations to tag the molecule which will be decoupled from the rest of the system. Using this in relative calculations is possible, but will result in unintended behavior and errors. `couple-lambda0` and `couple-lambda1` control the interactions of the

molecule specified by `couple-moltype` with its surroundings. The entries `vdw-lambdas` and `fep-lambdas` define the lambda schedule. The former indicates the value of the λ vector component that modifies van der Waals interactions for each state, while the latter changes all λ vector components that are not specified in the `.mdp` file. For instance, in split protocol simulations, these entries are sets such that the components of the energy are modified in different stages. If the transformation involves particle deletion (“forward process”), `fep-lambdas` is set to change charges and bonds before `vdw-lambdas` changes van de Waals components. If the process involves particle insertion (“backward process”) we reverse the roles. In this work, `mass-lambdas` were all set to zero to not included mass changes in the free energy. Unified protocols set all λ vectors the same.

Table 6 lists the relative free energies obtained from GROMACS simulations. Relative free energies are in good agreement with each other and with $\Delta\Delta G_{\text{hydr}}$ obtained from the other software used in this study (cmp. Tables 1 and 3). A noteworthy exception is the difference between the unified and split results of methane \rightarrow methanol and its reverse process, which was investigated with additional split protocol simulations using Coulomb softcore potentials (Table 7).

We noticed a difference of approximately $1.5 \text{ kcal mol}^{-1}$ between the split protocol without Coulomb softcore potentials and both protocols that use it. Our data suggests that softcore use in electrostatic interactions requires adjustments in the λ -distance between states in the rapidly varying part of the $\partial\mathcal{H}/\partial\lambda$, see Figure S7 A test with combining the bonded terms with the vdW transformation did not change this result. Thus, we find that the split protocol without Coulomb softcore potentials is the most effective way to calculate relative free energy calculations. We also find that there is no significant difference in calculated free energies depending on the choice of Particle Mesh Ewald or Reaction Field for calculation of long-range electrostatic interactions.

One peculiarity of the software worth mentioning is that relative free energy simulations will crash if a hydrogen alchemically becomes a heavy atom if the simulation employs hydro-

Table 6: Relative hydration free energies obtained from GROMACS simulations in $kcal \cdot mol^{-1}$. Signs of the backward transformation have been reverted to correspond to the forward transformation.

transformation	split ^a		unified ^b		absolute ^c	
	RF $\Delta\Delta G$	PME $\Delta\Delta G$	RF $\Delta\Delta G$	PME $\Delta\Delta G$	RF $\Delta\Delta G$	PME $\Delta\Delta G$
ethane	-0.025 \pm 0.005	-0.035 \pm 0.020	-0.017 \pm 0.003	-0.030 \pm 0.001	-0.06 \pm 0.01	-0.04 \pm 0.01
methane	-0.01 \pm 0.02	-0.02 \pm 0.01	0.046 \pm 0.020 ^d	0.01 \pm 0.02		
methanol	6.163 \pm 0.006	6.197 \pm 0.004	7.30 \pm 0.02	7.380 \pm 0.007	5.77 \pm 0.01	5.95 \pm 0.01
methane	6.168 \pm 0.005	6.199 \pm 0.008	7.09 \pm 0.02	7.17 \pm 0.02		
ethane	-6.123 \pm 0.007	-6.185 \pm 0.006	-7.117 \pm 0.005	-7.21 \pm 0.02	-5.83 \pm 0.01	-5.98 \pm 0.01
methanol	-6.124 \pm 0.005	-6.193 \pm 0.004	-7.338 \pm 0.004	-7.404 \pm 0.004		
toluene	3.22 \pm 0.01	3.211 \pm 0.006	3.229 \pm 0.008	3.22 \pm 0.01	2.97 \pm 0.01	3.16 \pm 0.01
methane	3.25 \pm 0.01	3.20 \pm 0.01	3.22 \pm 0.01	3.211 \pm 0.001		
neopentane ^e	-0.103 \pm 0.008	-0.15 \pm 0.02	-0.08 \pm 0.02	-0.18 \pm 0.03	-0.18 \pm 0.01	-0.14 \pm 0.01
methane ^e	-0.11 \pm 0.02	-0.16 \pm 0.05	0.00 \pm 0.03	-0.18 \pm 0.03		
neopentane ^f	-0.116 \pm 0.007	-0.13 \pm 0.01	-0.14 \pm 0.01	-0.14 \pm 0.01		
methane ^f	-0.10 \pm 0.03	-0.18 \pm 0.03	-0.089 \pm 0.007	-0.15 \pm 0.02		
2-methylfuran	2.986 \pm 0.006	2.930 \pm 0.050	3.05 \pm 0.01	3.00 \pm 0.01	2.87 \pm 0.01	2.95 \pm 0.01
methane	3.007 \pm 0.004	2.96 \pm 0.01	3.056 \pm 0.006	3.01 \pm 0.01		
2-methylindole	8.71 \pm 0.02	8.73 \pm 0.03	8.73 \pm 0.01	8.80 \pm 0.03	8.44 \pm 0.02	8.79 \pm 0.02
methane	8.73 \pm 0.03	8.74 \pm 0.01	8.30 \pm 0.02	8.77 \pm 0.04		
2-CPI	-0.07 \pm 0.02	-0.03 \pm 0.03	-0.10 \pm 0.05	-0.2 \pm 0.1	-0.02 \pm 0.05	0.02 \pm 0.02
7-CPI	-0.12 \pm 0.06	-0.20 \pm 0.04	-0.04 \pm 0.06	-0.14 \pm 0.09		

^aresults obtained from alchemical transformations with electrostatic and bonded scaling separate from vdW parameter change.

^bresults obtained from alchemical transformation with all parameters scaling together.

^cresults obtained from absolute free energy calculations.

^dinverted sign

^ecentral mapping

^fterminal mapping

Table 7: Relative hydration free energies of methanol \rightarrow methane and methane \rightarrow methanol transformations without and with the use of Coulomb softcore potentials from GROMACS. Signs of the backward transformation have been reverted to correspond to the forward transformation. The complete version of this table is in the SI.

		split		split+sc		absolute	
transformation		RF $\Delta\Delta G$	PME $\Delta\Delta G$	RF $\Delta\Delta G$	PME $\Delta\Delta G$	RF $\Delta\Delta G$	PME $\Delta\Delta G$
methanol	methane	6.163 ± 0.006	6.197 ± 0.004	7.32 ± 0.03	7.42 ± 0.04	5.77 ± 0.01	5.95 ± 0.01
methane	methanol	6.168 ± 0.005	6.199 ± 0.008	7.14 ± 0.03	7.21 ± 0.03		

gen bond constraining algorithms such as SHAKE or LINCS. Successful simulations require turning off the constraint and decreasing the time step. Alternative protocols that require some scripting and changes in the topology file could be pursued in the future. For instance 2-fs constraints protocols similar to those used in SOMD or AMBER in this study could be implemented via the definition of a new atom type for the hydrogen atoms that will be alchemically changed, or the use of `constraint = none` and of explicit constraints in the topology file.

3.4 SOMD

Fig. S4 compares relative free energy of hydration $\Delta\Delta G$ according to the protocol with unperturbed H bond constraints, with relative $\Delta\Delta G$ obtained from two absolute free energy calculations. Table 3 summarizes all the computed relative free energy of hydration for the dataset in Fig. 2. A very good agreement is observed between both methodologies ($R^2=0.99 \pm 0.01$ and $\text{MAD} = (0.10 \pm 0.03) \text{ kcal mol}^{-1}$), highlighting internal consistency within SOMD.

To achieve this level of reproducibility within SOMD it was crucial to pay close attention to constraints. Specifically, bonds that involve unperturbed hydrogen atoms are constrained. Bonds involving hydrogen atoms that are perturbed to a heavy element are unconstrained. Additionally the atomic mass of the perturbed hydrogen atom is set to the mass of the heavy atom it is perturbed to. Bonds involving hydrogen atoms that are perturbed to another

hydrogen atom type are constrained.

This protocol suppresses high frequency vibrations in flexible bonds involving hydrogen atoms, thus enabling a time step of 2 fs, whilst giving essentially negligible errors due to the use of constraints for perturbed bonds. This is apparent from the comparison with the absolute hydration free energy calculations. Additionally, the protocol yields relative hydration free energy very similar ($\text{MAE} = 0.09 \text{ kcal mol}^{-1}$) to those computed from simulations where no constraints are applied on the solutes and a timestep of 1 fs is used (See Figure S6).

By contrast, a protocol that constrains all bonds in a solute leads to significant differences with the absolute hydration free energies. For instance neopentane \rightarrow methane (centrally mapped) gives a RAFF $\Delta\Delta G = (2.04 \pm 0.01) \text{ kcal mol}^{-1}$ whereas the absolute hydration free energy calculations give $\Delta\Delta G = (-0.19 \pm 0.06) \text{ kcal mol}^{-1}$ as shown in tab. S6 and fig. S6.

This discrepancy occurs because in the SOMD implementation, the energies of constrained bonds are not evaluated, but the calculation of the energies of the solute at perturbed λ values is carried out using the coordinates of the reference λ trajectory. This leads to a neglect of contributions of the bonded term (and associated coupled terms) to the free energy change. The effect is more pronounced for perturbations that feature a large change in equilibrium bond lengths, such as those where a hydrogen atom is perturbed to/from a heavy atom. This problem has already been discussed by Pearlman⁷⁸ and Boresch,^{30,56} as a missing bond-stretching term, which introduce a free energy offset estimated as:

$$\Delta G = RT \ln \left(\frac{r_i}{r_f} \right) \quad (7)$$

where r_i and r_f are the initial and final bond lengths, respectively.

The reaction fields implemented in SOMD and GROMACS differ somewhat (atom-based shifted Barker Watts⁷⁰ vs group based switched Barker Watts), but nevertheless SOMD and GROMACS RF produce comparable results with a MAD of $0.18 \text{ kcal mol}^{-1}$.

Overall, the SOMD free energy estimations are in good agreement with the other MD

packages, as the MAE suggests (see Table 4). Reaction field and PME results are in good agreement. In particular, the unperturbed hydrogen bonds constraint protocol allows a timestep of 2 fs. All SOMD RAFF simulations were carried out with simultaneous transformation of Lennard-Jones, charges, and bonded terms. This suggests that the failure of the GROMACS “unified protocol” in some instances may be due to differences in the softcore Coulomb implementations.

For the methane \rightarrow neopentane transformations SOMD yields consistent results between central and terminal mappings, as shown in Table S5 while GROMACS RF displays a $\Delta\Delta G$ of (-0.20 ± 0.01) kcal mol $^{-1}$ for the centrally mapped transformation and $\Delta\Delta G = (-0.05 \pm 0.02)$ kcal mol $^{-1}$ for the terminally mapped transformation. The SOMD RF calculations show a MAD = 0.18 kcal mol $^{-1}$ and MAD = 0.23 kcal mol $^{-1}$ with GROMACS PME and AMBER respectively, similar to the MAD obtained with GROMACS RF.

4 Discussion

In this study we have addressed the question if the contemporary MD packages AMBER, CHARMM, GROMACS and SOMD are all able to reproduce relative alchemical free energies of hydration for a set of neutral small organic molecules given a pre-defined force field. We have found that arriving at a simulation protocol that does this, has been cumbersome due to technical difficulties encountered with every code. The MD codes have a wide range of options and setup features which makes it difficult for the unexperienced user to decide on the most appropriate ones.

The free energies we have computed appear to be in reasonable good agreement with each other (see Tables 1 and 3). The MAE is 0.2 kcal/mol for absolute free energies and 0.3 kcal/mol for relative free energies. This is what we call the “limit of reproducibility”. We have found viable protocols for each MD code to achieve this level of reproducibility. There is some doubt, however, over the AMBER results because that software cannot reproduce

the correct end-point geometries. This is particularly evident in the neopentane to methane case with central mapping where also the relative free energies are clearly different from the other packages. We suspect these issues reflect a bug in the AMBER package but have been unable to isolate it; we have reported the issue to the AMBER developers.

However, we have not arrived at a *single* protocol that could be recommended for use with all four codes. Unified protocols do not appear to work universally for AMBER and GROMACS while SOMD and CHARMM(?) had no problem in this regard. We cannot rule out that the problem may lie e.g. only with the vacuum leg of the thermodynamic cycle. In the case of AMBER the vacuum simulation has currently been done with the separately developed sander module. The problem may be a consequence of the different softcore functions (see Eq. S1) used in these MD packages but further investigations are needed to resolve this issue.

The unperturbed H bond protocol is an interesting alternative which applies constraints to all non-transforming bonds and thus allowed us to increase the time step to 2 fs. The split protocol was found to work well for all codes (also CHARMM?). It appears to be the most efficient approach for GROMACS as shown with the methanol to methane case because the unified protocol produces a less smooth function.⁷⁹ A complete separation of lambdas may not be necessary though as a certain degree of overlap between vdW and Coulomb λ may be a viable solution⁸⁰ for equilibrium AFEs.

Comparison between codes is hampered by several factors. Firstly, the codes use different simulation algorithms. e.g. electrostatics are handled differently in vacuum i.e. infinite cutoff vs. reaction field. Temperature and pressure control, time step integrators, etc. are other examples. But the data here suggest that, if there are any systematic errors introduced through these algorithms, then they are very small. It is reassuring that AFEs for the systems tested here show only a small dependence on MD protocol decisions (provided a correct implementation).

As part of this work we make our input data and protocols available (see SI). We recom-

mend to use this dataset to test and benchmark future RAFF implementations to validate reproducibility against other simulation packages.

Where possible, we recommend comparing results from both absolute and relative transformations to verify internal consistency. The relative transformation should be run in both forward and backward directions, even if the free energy estimator is agnostic to this decision, as other implementation details (e.g. bugs in parameters definition, atomic masses) may lead to inconsistent results.

More specifically, various issues with current code bases have been revealed through this work. We have found that constraints in connection with varying bond length can cause errors with GROMACS, just as masses must not be allowed to vary in RAFF simulations, both to avoid crashes and wrong results in the software. CHARMM has issues with constraints and the PSSP softcores, and the PERT module cannot make use of the force switch as it is now standard for CHARMM force fields. Care must be taken when using the LRC long range correction keyword to avoid producing inconsistent results. AMBER’s problem with end point geometries and unified protocols has been pointed out above.

Another question is the degree of usability of the softwares.

Lambda paths as implemented in GROMACS would also be beneficial for other codes as they make the setup of split protocols much easier. The current solution to this is to mimic this protocol through careful constructions of topologies through scripting. The setup of absolute free energy is nowadays as simple as instructing the MD software to mutate to “nothing”. For split protocols, however, extra care must be undertaken to ensure inclusion of all terms. E.g. in the case of AMBER this means to set all charges of the end state to zero for the charge transformation step but this also means that intramolecular charge–charge contributions will all be zero, thus requiring an additional simulation in vacuo to correct for this.

The primary focus of this work was to achieve low statistical errors to establish if codes are able to reproduce free energies. We have not investigated rate of convergences as this

would require further, complex investigations.

Beyond careful protocol validation, further automation of alchemical free energy studies will also decrease user errors, and thus increases reproducibility. Various attempts in this direction are currently underway for both absolute and relative setups.^{20,46,47,81–84}

To conclude, we hope this study will stimulate the field to strengthen the transferability of alchemical free energy calculation protocols across software. Reproducibility is crucial to enable robust use of alchemical free energy methods in molecular design.

5 Conclusions

Acknowledgement

HHL is supported through an EPSRC provided SLA, funding the core support of CCP-BioSim. CCPBioSim is the Collaborative Computational Project for Biomolecular Simulation funded by EPSRC grants EP/J010588/1 and EP/M022609/1. JM is supported by a Royal Society University Research Fellowship. The research leading to these results has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP7/2007–2013)/ERC Grant agreement No. 336289. GDRM appreciates the support from the Brazilian agency CAPES - Science without Borders program (BEX 3932-13-3). DLM appreciates support from the National Science Foundation (CHE 1352608), and computing support from the UCI GreenPlanet cluster, supported in part by NSF Grant CHE-0840513.

We thank Prof. Stefan Boresch for valuable discussions and making code modifications to CHARMM. We thank Dr. Ross Walker and Daniel Mermelstein for valuable discussions and making code modifications to AMBER. We thank Prof. Michael Shirts for valuable discussions about GROMACS.

We acknowledge use of Hartree Centre resources and the use of the SCARF HPC cluster in this work.

References

- (1) Hansen, N.; Van Gunsteren, W. F. *Journal of Chemical Theory and Computation* **2014**, *10*, 2632–2647.
- (2) Pohorille, A.; Jarzynski, C.; Chipot, C. *The Journal of Physical Chemistry B* **2010**, *114*, 10235–10253, PMID: 20701361.
- (3) Gallicchio, E.; Levy, R. M. In *Computational chemistry methods in structural biology*; Christov, C., Ed.; Advances in Protein Chemistry and Structural Biology; Academic Press, 2011; Vol. 85; pp 27 – 80.
- (4) Lu, X.; Fang, D.; Ito, S.; Okamoto, Y.; Ovchinnikov, V.; Cui, Q. *Molecular Simulation* **2016**, *42*, 1056–1078, PMID: 27563170.
- (5) Rickman, J. M.; LeSar, R. *Annual Review of Materials Research* **2002**, *32*, 195–217.
- (6) Wan, S.; Knapp, B.; Wright, D. W.; Deane, C. M.; Coveney, P. V. *Journal of Chemical Theory and Computation* **2015**, *11*, 3346–3356, PMID: 26575768.
- (7) Beveridge, D. L.; Dicapua, F. M. *Annual Review of Biophysics and Biophysical Chemistry* **1989**, *18*, 431–492.
- (8) Straatsma, T. P.; Mccammon, J. A. *Annual Review of Physical Chemistry* **1992**, *43*, 407–435.
- (9) Kollman, P. *Chemical Reviews* **1993**, *93*, 2395–2417.
- (10) Squire, D. R.; Hoover, W. G. *The Journal of Chemical Physics* **1969**, *50*, 701–706.
- (11) Bennett, C. H. *Journal of Computational Physics* **1976**, *22*, 245–268.
- (12) Mruzik, M. R.; Abraham, F. F.; Schreiber, D. E.; Pound, G. M. *The Journal of Chemical Physics* **1976**, *64*, 481–491.

- (13) Postma, J. P. M.; Berendsen, H. J. C.; Haak, J. R. *Faraday Symp. Chem. Soc.* **1982**, *17*, 55–67.
- (14) Tembe, B. L.; McCammon, J. *Computers & Chemistry* **1984**, *8*, 281 – 283.
- (15) Jorgensen, W. L.; Ravimohan, C. *The Journal of Chemical Physics* **1985**, *83*, 3050–3054.
- (16) Gilson, M.; Given, J.; Bush, B.; McCammon, J. *Biophysical Journal* **1997**, *72*, 1047 – 1069.
- (17) Boresch, S.; Tettinger, F.; Leitgeb, M.; Karplus, M. *The Journal of Physical Chemistry B* **2003**, *107*, 9535–9551.
- (18) Deng, Y.; Roux, B. *The Journal of Physical Chemistry B* **2009**, *113*, 2234–2246.
- (19) Ytreberg, F. M.; Swendsen, R. H.; Zuckerman, D. M. *Journal of Chemical Physics* **2006**, *125*, 184114.
- (20) Gapsys, V.; Michielssens, S.; Seeliger, D.; de Groot, B. L. *Journal of Computational Chemistry* **2015**, *36*, 348–354.
- (21) Sandberg, R. B.; Banchelli, M.; Guardiani, C.; Menichetti, S.; Caminati, G.; Procacci, P. *Journal of Chemical Theory and Computation* **2015**, *11*, 423–435, PMID: 26580905.
- (22) Duarte Ramos Matos, G.; Kyu, D. Y.; Loeffler, H. H.; Chodera, J. D.; Shirts, M. R.; Mobley, D. L. *Journal of Chemical & Engineering Data* **0**, *0*, null.
- (23) Jorgensen, W. L.; Buckner, J. K.; Boudon, S.; Tirado-Rives, J. *The Journal of chemical physics* **1988**, *89*, 3742–3746.
- (24) Woods, C. J.; Malaisree, M.; Hannongbua, S.; Mulholland, A. J. *The Journal of Chemical Physics* **2011**, *134*, 054114.

- (25) Woods, C. J.; Malaisree, M.; Michel, J.; Long, B.; McIntosh-Smith, S.; Mulholland, A. J. *Faraday Discuss.* **2014**, *169*, 477–499.
- (26) Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, M. K.; Greenwood, J.; Romero, D. L.; Masse, C.; Knight, J. L.; Steinbrecher, T.; Beuming, T.; Damm, W.; Harder, E.; Sherman, W.; Brewer, M.; Wester, R.; Murcko, M.; Frye, L.; Farid, R.; Lin, T.; Mobley, D. L.; Jorgensen, W. L.; Berne, B. J.; Friesner, R. A.; Abel, R. *Journal of the American Chemical Society* **2015**, *137*, 2695–2703, PMID: 25625324.
- (27) Wang, L.; Deng, Y.; Wu, Y.; Kim, B.; LeBard, D. N.; Wandschneider, D.; Beachy, M.; Friesner, R. A.; Abel, R. *Journal of Chemical Theory and Computation* **2017**, *13*, 42–54, PMID: 27933808.
- (28) Pearlman, D. A. *The Journal of Physical Chemistry* **1994**, *98*, 1487–1493.
- (29) Michel, J.; Essex, J. W. *Journal of Computer-Aided Molecular Design* **2010**, *24*, 639–658.
- (30) Boresch, S.; Karplus, M. *The Journal of Physical Chemistry A* **1999**, *103*, 103–118.
- (31) Liu, S.; Wang, L.; Mobley, D. L. *Journal of Chemical Information and Modeling* **2015**, *55*, 727–735, PMID: 25835054.
- (32) Shobana, S.; Roux, B.; Andersen, O. S. *The Journal of Physical Chemistry B* **2000**, *104*, 5179–5190.
- (33) Michel, J.; Verdonk, M. L.; Essex, J. W. *Journal of Chemical Theory and Computation* **2007**, *3*, 1645–1655, PMID: 26627610.
- (34) Rocklin, G. J.; Mobley, D. L.; Dill, K. A. *The Journal of Chemical Physics* **2013**, *138*, 085104.

- (35) Axelsen, P. H.; Li, D. *Journal of Computational Chemistry* **1998**, *19*, 1278–1283.
- (36) Case, D. A.; Cheatham III, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz Jr., K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. *Journal of Computational Chemistry* **2005**, *26*, 1668–1688.
- (37) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caffisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. *Journal of Computational Chemistry* **2009**, *30*, 1545–1614.
- (38) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. *SoftwareX* **2015**, *1–2*, 19–25.
- (39) Scott, W. R. P.; Hünenberger, P. H.; Tironi, I. G.; Mark, A. E.; Billeter, S. R.; Fennel, J.; Torda, A. E.; Huber, T.; Krüger, P.; van Gunsteren, W. F. *The Journal of Physical Chemistry A* **1999**, *103*, 3596–3607.
- (40) Woods, C.; Mey, A. S.; Calabro, G.; Michel, J. Sire molecular simulations framework. 2016.
- (41) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; Tye, T.; Houston, M.; Stich, T.; Klein, C.; Shirts, M. R.; Pande, V. S. *Journal of Chemical Theory and Computation* **2013**, *9*, 461–469, PMID: 23316124.
- (42) Shirts, M. R.; Klein, C.; Swails, J. M.; Yin, J.; Gilson, M. K.; Mobley, D. L.; Case, D. A.; Zhong, E. D. *Journal of Computer-Aided Molecular Design* **2017**, *31*, 147–161.

- (43) Schappals, M.; Mecklenfeld, A.; Kröger, L.; Botan, V.; Köster, A.; Stephan, S.; García, E. J.; Rutkai, G.; Raabe, G.; Klein, P.; Leonhard, K.; Glass, C. W.; Lenhard, J.; Vrabec, J.; Hasse, H. *Journal of Chemical Theory and Computation* **0**, 0, null, PMID: 28738147.
- (44) Pronk, S.; Larsson, P.; Pouya, I.; Bowman, G. R.; Haque, I. S.; Beauchamp, K.; Hess, B.; Pande, V. S.; Kasson, P. M.; Lindahl, E. Copernicus: A New Paradigm for Parallel Adaptive Molecular Dynamics. Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. New York, NY, USA, 2011; pp 60:1–60:10.
- (45) Sadiq, S. K.; Wright, D.; Watson, S. J.; Zasada, S. J.; Stoica, I.; Coveney, P. V. *Journal of Chemical Information and Modeling* **2008**, 48, 1909–1919, PMID: 18710212.
- (46) Lundborg, M.; Lindahl, E. *The Journal of Physical Chemistry B* **2015**, 119, 810–823, PMID: 25343332.
- (47) Loeffler, H. H.; Michel, J.; Woods, C. *Journal of Chemical Information and Modeling* **2015**, 55, 2485–2490.
- (48) Balasubramanian, V.; Bethune, I.; Shkurti, A.; Breitmoser, E.; Hruska, E.; Clementi, C.; Laughton, C. A.; Jha, S. *CoRR* **2016**, *abs/1606.00093*.
- (49) Skyner, R. E.; McDonagh, J. L.; Groom, C. R.; van Mourik, T.; Mitchell, J. B. O. *Phys. Chem. Chem. Phys.* **2015**, 17, 6174–6191.
- (50) Mobley, D. L.; Guthrie, J. P. *Journal of Computer-Aided Molecular Design* **2014**, 28, 711–720.
- (51) Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. *Chemical Physics Letters* **1994**, 222, 529–539.

- (52) Zacharias, M.; Straatsma, T. P.; McCammon, J. A. *The Journal of Chemical Physics* **1994**, *100*, 9025–9031.
- (53) Steinbrecher, T.; Mobley, D. L.; Case, D. A. *The Journal of Chemical Physics* **2007**, *127*, 214108.
- (54) Cross, A. J. *Chemical Physics Letters* **1986**, *128*, 198 – 202.
- (55) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. *Journal of Computational Chemistry* **2005**, *26*, 1781–1802.
- (56) Boresch, S.; Karplus, M. *The Journal of Physical Chemistry A* **1999**, *103*, 119–136.
- (57) Rocklin, G. J.; Mobley, D. L.; Dill, K.; Hünenberger, P. H. *The Journal of chemical physics* **2013**, *139*, 184103.
- (58) Loeffler, H. H.; Sottriffer, C. A.; Winger, R. H.; Liedl, K. R.; Rode, B. M. *Journal of Computational Chemistry* **2001**, *22*, 846–860.
- (59) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *Journal of Computational Chemistry* **2004**, *25*, 1157–1174.
- (60) Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. *Journal of Computational Chemistry* **2000**, *21*, 132–146.
- (61) Jakalian, A.; Jack, D. B.; Bayly, C. I. *Journal of Computational Chemistry* **2002**, *23*, 1623–1641.
- (62) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *The Journal of Chemical Physics* **1983**, *79*, 926–935.
- (63) Shivakumar, D.; Harder, E.; Damm, W.; Friesner, R. A.; Sherman, W. *Journal of Chemical Theory and Computation* **2012**, *8*, 2553–2558, PMID: 26592101.

- (64) Hu, Y.; Sherborne, B.; Lee, T.-S.; Case, D. A.; York, D. M.; Guo, Z. *Journal of Computer-Aided Molecular Design* **2016**, *30*, 533–539.
- (65) Steinbrecher, T.; Joung, I.; Case, D. A. *Journal of Computational Chemistry* **2011**, *32*, 3253–3263.
- (66) Naden, L. N.; Pham, T. T.; Shirts, M. R. *Journal of Chemical Theory and Computation* **2014**, *10*, 1128–1149.
- (67) Naden, L. N.; Shirts, M. R. *Journal of Chemical Theory and Computation* **2015**, *11*, 2536–2549.
- (68) Pitera, J. W.; Gunsteren, W. F. v. *Mol. Simul.* **2002**, *28*, 45–65.
- (69) Anwar, J.; Heyes, D. M. *J. Chem. Phys.* **2005**, *122*, 224117.
- (70) Barker, J.; Watts, R. *Molecular Physics* **1973**, *26*, 789–792.
- (71) Andersen, H. C. *The Journal of Chemical Physics* **1980**, *72*, 2384–2393.
- (72) Bosisio, S.; Mey, A. S. J. S.; Michel, J. *Journal of Computer-Aided Molecular Design* **2016**, *30*, 1101–1114.
- (73) Zwanzig, R. *The Journal of Chemical Physics* **1954**, *22*, 1420–1426.
- (74) Klimovich, P. V.; Shirts, M. R.; Mobley, D. L. *Journal of Computer-Aided Molecular Design* **2015**, *29*, 397–411.
- (75) Klimovich, P. V.; Mobley, D. L. *Journal of Computer-Aided Molecular Design* **2010**, *24*, 307–316.
- (76) Lee, T.-S.; Hu, Y.; Sherborne, B.; Guo, Z.; York, D. M. *Journal of Chemical Theory and Computation* **0**, *0*, null, PMID: 28618232.

- (77) Kaus, J. W.; Pierce, L. T.; Walker, R. C.; McCammon, J. A. *Journal of Chemical Theory and Computation* **2013**, *9*, 4131–4139.
- (78) Pearlman, D. A.; Kollman, P. A. *The Journal of chemical physics* **1991**, *94*, 4532–4545.
- (79) Shirts, M. R.; Mobley, D. L.; Chodera, J. D. *Annual Reports in Computational Chemistry* **2007**, *3*, 41–59.
- (80) Procacci, P.; Cardelli, C. *Journal of Chemical Theory and Computation* **2014**, *10*, 2813–2823, PMID: 26586508.
- (81) Christ, C.; Fox, T. *Journal of chemical information and modeling* **2013**,
- (82) Liu, S.; Wu, Y.; Lin, T.; Abel, R.; Redmann, J. P.; Summa, C. M.; Jaber, V. R.; Lim, N. M.; Mobley, D. L. *Journal of Computer-Aided Molecular Design* **2013**, *27*, 755–770.
- (83) Ramadoss, V.; Dehez, F.; Chipot, C. *Journal of Chemical Information and Modeling* **2016**, *56*, 1122–1126, PMID: 27214306.
- (84) Bhati, A. P.; Wan, S.; Wright, D. W.; Coveney, P. V. *Journal of Chemical Theory and Computation* **2017**, *13*, 210–222, PMID: 27997169.

Graphical TOC Entry

