Reproducibility of Free Energy Calculations Across Different Molecular Simulation Software

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, United Kingdom, 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@gmail.com

Abstract

Alchemical free energy calculations are an increasingly important modern simulation technique. Contemporary molecular simulation 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. for a given molecular model and set of forcefield parameters, comparable free energy should be obtained

[*]To whom correspondence should be addressed
[†]Scientific Computing Department, STFC
[‡]University of Edinburgh
[¶]University of California, Irvine
[§]University of Chicago
[‖]University of California, Irvine
[⊥]Current address: Eli Lilly and Company, Erl Wood Manor, Sunninghill Road, Windlesham GU20 6PH, United Kingdom

within statistical bounds regardless of the code used. *Relative* alchemical free energy (RAFE) simulation is increasingly used to support molecule discovery projects, yet the reproducibility of the methodology has been less well tested than its absolute counterpart. Here we present RAFE calculations of hydration free energies for a set of small organic molecules and demonstrate that free energies can be reproduced to within about 0.2 kcal/mol with aforementioned codes. Achieving this level of reproducibility requires considerable attention to detail and package–specific simulation protocols, and no universally applicable protocol emerges. The benchmarks and protocols reported here should be useful for the community to validate new and future versions of software for free energy calculations.

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 simulations[1–5] 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 should 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 is also increasingly referred as free energy perturbation (FEP) in the literature, even though different techniques may have actually been used to estimate free energy changes. The method has gained renewed attention in recent years — concomitant with improvements in computer hardware design — within the traditional equilibrium framework[16–18] and also increasingly in combination with non-equilibrium techniques.[19–21] The name "alchemical" 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 effectively 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 compared to methods involving physical transition pathways or order parameters, as it does not require sampling of diffusive motions, avoids crossing prohibitively large energy barriers if transition 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

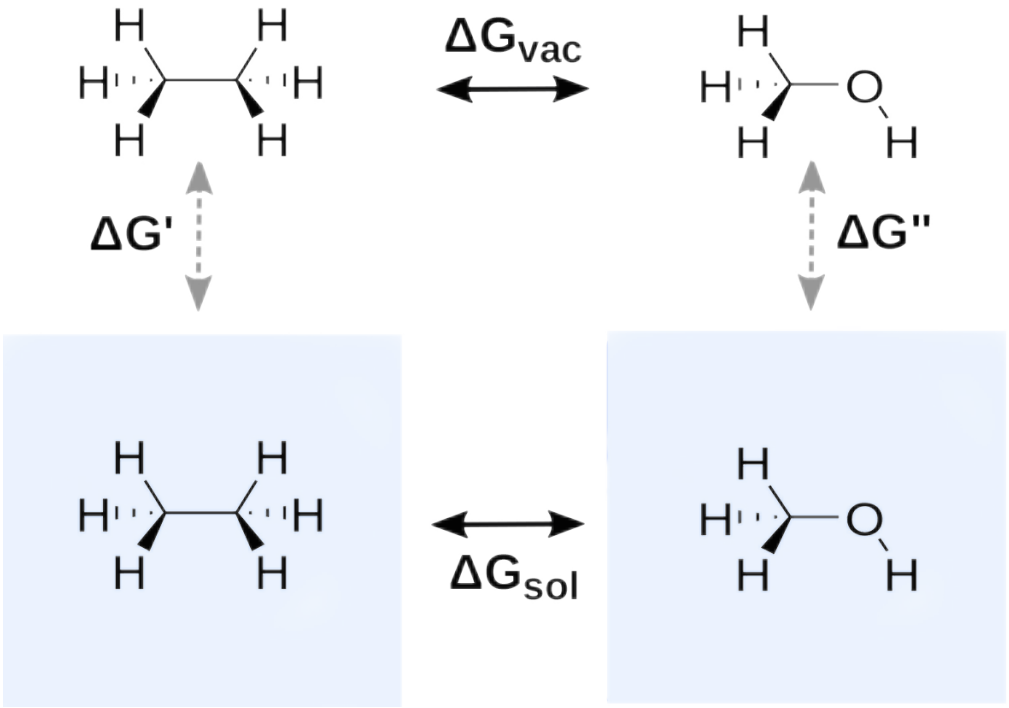

Figure 1: The thermodynamic cycle to compute the relative free energy of hydration $\Delta\Delta G_{\text{hydr}} = \Delta G_{\text{sol}} - \Delta G_{\text{vac}} = \Delta G'' - \Delta G'$. The example is for the ethanol $\leftrightarrow$ methanol transformation. A blue background indicates water and a white background indicates gas phase. 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.

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. This term is mostly used to distinguish it from techniques usually referred to as *relative* (see below). It should be emphasized that the "absolute" approach still results in a *relative* free energy between the state where the solute fully interacts with its environment and the state where it does not. The term *decoupling* here is taken as meaning 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) and its environment. We distinguish decoupling from *annihilation*, as the latter also includes a scaling of the *intra*–molecular non-bonded interactions in addition to the

inter–molecular interactions.[24]([1]) Torsional interactions may also be scaled in an annihilation protocol, but bond and angle terms are usually not scaled as this leads to poorly converging free energy changes.[25] 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 as well.[26,27]

Relative alchemical free energy (RAFE) calculations transform or mutate one molecule into another. An appealing aspect of RAFE calculations is the hope that they may be somewhat less demanding computationally or converge better than the more ambitious approaches that require a complete decoupling or annihilation of a ligand from its environment. 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.[28,29]

RAFEs can be calculated by making use of either the so–called single or dual topology method. Dual topology means that groups of atoms of the end states are duplicated and thus both sets are present at all times but do not interact with each other.[25,30] The atom types are not changed, and, in principal, the groups of both states would need to have the same total charge to avoid partially charged intermediates. In practice this could require, depending on force field, to duplicate all atoms of the end states. Only non–bonded interactions need to be scaled such that the disappearing end state is fully decoupled from its environment.[25] The dual topology method is the most straightforward approach to compute RAFEs when the two molecules are structurally dissimilar. In situations where all atoms in a perturbed molecule are duplicated a dual topology calculation is the technically same as two absolute calculations, executed simultaneously in opposite directions. This, however, comes with additional complications as the two independent molecules can drift apart and

[1]It is worth noting that the terms "double decoupling method" and "double annihilation method" also employ the words "decoupling" and "annihilation" but used in an entirely different sense in the context of standard binding free energy calculations.

sample completely different environments (e.g. binding site versus bulk solution). It has been shown though that with the introduction of special restraints or constraints this can be a viable option.[31–33] Restraints between corresponding atoms can also be used without affecting the free energy.[33] A recent alternative considered molecules with a common core where all atom types are the same.[34] 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.

Single topology means that there is only one connected representation of the molecule to be transformed into another molecule. Atoms of a given type are directly transformed, typically by linearly scaling the force field parameters, into atoms of a different type. The single topology method offers a straightforward route to implement RAFE calculations.[15,25,30,35] In typical implementations, a certain number of non-interacting "dummy" atoms must hold the place of disappearing/appearing atoms in order to balance the number of atoms in both end states. Dummy atoms have no non–bonded interactions in the end state but normally retain the bonded terms of the original atom to avoid complications with unbound atoms.[25] However, it is important to stress that a dummy atom should retain at most only one angle term (Atom1–Atom2–Dummy) and one dihedral term (Atom1–Atom2–Atom3–Dummy) with respect to non-dummy atoms to yield correct results.[25,36]

The single topology approach seeks to exploit the topological and structural similarity of the two end states.[30] 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. These considerations notwithstanding, the single topology approach is broadly applicable to a wide range of transformations. For example, ring breaking is technically challenging,[29] but it has been shown this can be done in certain circumstances.[36,37] Generally, modern MD software (e.g. AMBER,[38] CHARMM,[39] GROMACS,[40] GROMOS,[41] and SOMD.[42,43]) support a hybrid approach that combines aspects of single and dual topology.[36]

As alluded to above, consistency and reliability are the 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 packages[44] — a necessary but not sufficient prerequisite. Another study went further and also compared liquid densities across packages, revealing a variety of issues.[45] For free energies, given a predefined force field and run–time parameters we ought to be able to obtain comparable free energy results within the limits of statistical convergence. This comparison has not yet been carried out.

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 to 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.[46–50]

In this work we compute the relative hydration free energies of a set of small organic molecules using several software and protocols (see Fig. **2**). Solvation free energies have a wide range of uses and various methods exist to compute them.[51] They are also needed for calculations of a variety of important physical properties, and to calculate binding free energies where the solution simulation (see Fig. **1**) is combined with a mutation of the molecule bound to a partner.[51] A large database of hydration free energies computed from alchemical free energy (AFE) simulations, FreeSolv, has been presented recently.[22,52] Here, we focus on the reproducibility of RAFE 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

One practical challenge is that the free energy methodologies used in one MD program are not always available in another package, or the same functionality is provided via different algorithms (e.g. algorithms for pressure and temperature scaling, integrators, etc). In addition there may be difference in the choice of physical constants used for evaluating potential energies. A previous study noted that variations in the hardcoded values of Coulomb's constant lead to detectable differences in single point energies calculated by CHARMM, AMBER or GROMACS.[44,53] To circumvent some of these practical problems, we will compare relative free energies calculated via three protocols. In the "unified protocol" we calculate relative free energies by scaling together all force field parameters i.e. partial charges, van der Waals parameters, and bonded parameters vary simultaneously along the alchemical path. In the "split protocol" we calculate relative free energies by scaling separately the van der Waals parameters and the partial charges parameters. The order in which this has to be done is detailed in section **??**x of the SI. The scaling of the bonded terms can be combined with either transformation. In the "absolute protocol" we calculate relative hydration free energies as the difference between two calculated absolute hydration free energies.

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 implemented in each code as summarized in section **??** of the SI. [54,55] Softcore functions are used to avoid the numerical stability problems of the conventional Lennard-Jones (LJ) and Coulombic inverse power law potentials,[56,57] as they display singularities at zero distance (vertical asymptotes). Attempting to modify interactions by linearly scaling back the LJ potential as a function of an interaction parameter, λ, causes the r^{-12} term to increasingly behave as a sharp repulsive singularity as $\lambda \to 0$.[56]

This means that there is an unbounded discontinuous change between $\lambda = 0$ where particles can overlap, and $\lambda = \delta$, even as $\delta \to 0$, where particles still behave like minuscule hard 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.[54,55,57]

Another difference is how the codes scale force field parameters ("parameter scaling") and/or the energy ("energy scaling").[25] 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 individually. In the latter case, the total energy is scaled, all at once, or, equivalently for each individual force field contribution. While free energy is a state function that depends only on the end points, the pathways taken by the two methods through state space or alchemical space are different.

One more important issue is whether the code allows holonomic 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 uses a hybrid dual/single topology approach. All terms are energy scaled. 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 code loads two separate input topologies that describe the end states of interest and allows users to map atoms between the two end–states that will share the same coordinates for the free energy calculation. Evaluation of the interactions involving these atoms as a function of the coupling parameter is done by default via linear scaling of the energy and forces of the end–states. Alternatively the user can request that a softcore potential be used. The non–bonded interactions of atoms that are not paired between the end–states are handled with a softcore potential. In addition, bonded terms involving different unpaired

atoms are ignored. This in effect amounts to defining unpaired atoms as dummy atoms in one of the end–states. We call this the "implicit dummy protocol" since the procedure is handled automatically by the software through analysis of the end–state topologies rather than via explicit definition of dummy atoms in an input topology.

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. For instance one can keep the LJ and bonded terms fixed at the initial state for a charge transformation. The setup for the two end–states must therefore use identical atom types with only the charges varying.

Alternatively it is possible for the user to construct an input topology of a single molecule that explicitly contains dummy atoms such that the desired end–states can be simulated. This is a similar approach to that employed by SOMD and GROMACS, and we call this the "explicit dummy protocol".

CHARMM. The PERT module duplicates the topology similarly to `sander` but mapped atoms are given in the topology only once. The module requires balancing with explicit dummy atoms. All energy terms are linearly scaled by the coupling parameter λ. The PSSP softcore potential is applied to *all* atoms in the perturbed group (see section **??**x 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 LJ 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 λs for LJ, Coulomb and bonded parameters (and some other possible terms in the potential) which allows easy implementation of split protocols.

SOMD. SOMD is a software built by linking Sire and OpenMM molecular simulation libraries.[42,43] This code uses a single topology description. The alchemical state is constructed at run time from an input topology together 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 FESetup (version 1.2).[49] FESetup is a perturbed topology writer for AMBER, CHARMM, GROMACS, SOMD and NAMD.[58] 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.[37] With this strategy, a single topology description is achieved: any atom that does not match is 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 RAFE simulations. Full details on FESetup can be found in Ref. 49.

Figure **2** shows all 18 transformations considered in the present study. In the limit of sufficient sampling, RAFE simulations should not depend on the "forward" and "backward" direction of change with respect to the coupling parameter λ. However 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 RAFE simulations.[15,59] 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 two alternative mappings have been considered, see Figure **2**. One mapping has methane matched to a terminal methyl (green) and the other one has the methane carbon matched with the central carbon in neopentane (blue). The first approach will be called "terminally mapped" and the second one "centrally mapped".

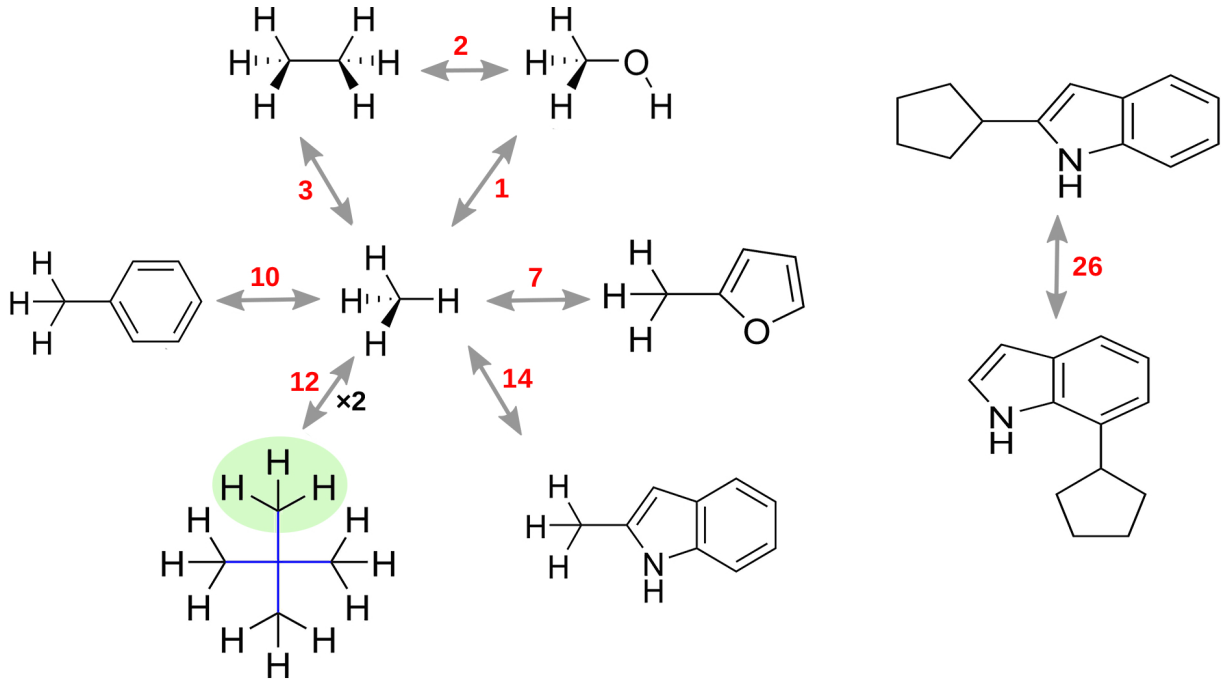

Figure 2: The thermodynamic cycles considered in this study. To compute the free energy of hydration, all pair–wise transformations have to be carried out once in solution and once in vacuum. Green and blue colours in neopentane show two alternative mappings for methane. The numbers in red denote the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

2.3 Free Energy Simulation Protocols

One of the major goals of the present study is to ensure consistency and reproducibility from the computational protocols. This is complicated by the fact that a given MD software may employ a range of methods and algorithms that one may not be able to duplicate exactly with other MD software. In particular, how the alchemical transformation is controlled via the coupling parameter may be very different. At the most basic level, even pressure and temperature scaling, integrators and other algorithms can also display important differences. It is unclear if and how any of these implementation details can affect results. The implementation details of alchemical free energy simulation in code are discussed in subsection 2.1.

In this study we consider at 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 minimize statistical sampling errors, and avoid difficulties with charged particles.[60,61] The force field was chosen to be GAFF (version 1.8),[62] utilizing

AM1/BCC charges for the solute,[63,64] and TIP3P for the solvent.[65] 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. 66,67.

While the MD packages principally allow a "one–step" transformation,[68] that is with both LJ and Coulombic parameters varied simultaneously (unified protocol), it has also been proposed that carrying out a split protocol may be more efficient.[69–71] 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 (see section **??**x in the SI for details) on the LJ term only.[54,55] 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,[68,72,73] see section **??**x in the SI.

All simulations were started from simulation boxes prepared by FESetup.[49] During construction of the perturbed systems, 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 transformations from a smaller to a larger solute, water molecules may be in close proximity to the solute. The production simulations were run at 298 K and 1.0 bar. Atomic masses were not changed along the alchemical transformations as this would affect only the kinetic energy, and would not contribute to the free energy change.

AMBER. The AMBER16 program was used for this set of free energy calculations. Typically 11 windows were used for charge mutations and 21 windows for VdW mutations. In some instances, steep variations in gradients were observed with this protocol and addi-

tional windows were added to obtain smoother integration profiles. 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 necessary 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. Absolute transformations were carried out using a one step protocol featuring 21 windows initially. For some perturbations additional windows were run in regions where the free energy gradients varied sharply. Each window was simulated for 2.5 ns, with the first 0.2 ns discarded prior analysis. Water hydrogens (TIP3P) were constrained with SHAKE. None of the atoms in the perturbed group where constrained and hence the time step was set to 1 fs. An alternative protocol with SHAKE on bonds that do not change during transformation and a time step of 2 fs was also tested (see SOMD protocol below). The temperature was controlled through a Langevin thermostat with a friction constant of $2.0\,\mathrm{ps}^{-1}$ and pressure rescaling through a Monte Carlo barostat with 100 steps between isotropic volume change attempts. Long–range electrostatics in solution was handled with Particle Mesh Eward (PME) and an atom–based cutoff of 8.0 Å for the real-space Coulomb and vdW interactions. No cutoff was used for the vacuum simulations. A Long Range Correction (LRC) term for truncated VdW interactions is applied during the MD simulations.

CHARMM. The version c40b1 was used for this set of free energy calculations. The PERT module was used to handle the alchemical transformations. Three different approaches were used to calculate the relative Gibbs free energy: (i) RAFE simulation where electrostatic and VdW interactions were changed separately (split-protocol) , (ii) RAFE simulation where electrostatic and VdW interactions were changed together (unified-protocol) , and (iii) difference between free energies from two AFE simulations where AFE simulations followed unified-protocol. In total, 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. A few windows at the end-points (involving hydrogen being transformed to heavy atom or vice versa) were unstable due to steric clashes with starting coordinates and were equilibrated using 0.1 fs to 0.5 fs. Only water hydrogens (TIP3P) were constrained with SHAKE. Conditions of constant temperature and pressure control were maintained using the Berendsen weak coupling method, with a compressibility of $4.63 \times 10^{-5}\,\mathrm{atm}^{-1}$ and temperature and pressure coupling constants of $5.0\,\mathrm{ps}^{-1}$. Long–range electrostatics in solution was handled with PME to order 6 with a cutoff of 12.0 Å for the real-space Coulomb and vdW interactions. No cutoff was used for the vacuum simulations. No LRC term was applied during the alchemical MD simulations but a solute-solvent LRC term was included in post-processing to calculate the final free energy. The PSSP softcore potential function was used for the perturbed atoms. The PERT module currently does not currently support the force switching (option `VFSwitch`) for LJ potentials with softcores. The CHARMM PARAM27 force fields, however, is parameterized to use force switching.[39] Accordingly, we used the potential switching only (option `VSwitch`) with an inner cutoff of 10 Å and outer cutoff of 12 Å.

GROMACS. GROMACS version 4.6.7 was used to carry out this set of free energy calculations. 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. We used a 20-step alchemical protocol where charge coupling and LJ coupling were dealt with separately along the path.[22,52] The free energies were calculated from 5 ns Langevin dynamics at 298 K. A friction coefficient of $1.0\,\mathrm{ps}/\mathrm{m}_{\mathrm{atom}}$ was used, where m_{atom} is the the mass of the atom. No holonomic bond or angle constraints were used. A Parrinello–Rahman baro-

stat with $\tau_p = 10\,\text{ps}$ and compressibility equal to $4.5 \times 10^{-5}\,\text{bar}^{-1}$ was used. Two methods were used 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. PME calculations were of order 6 and had a tolerance of 1.0×10^{-6}, with a grid spacing of 1.0 Å. We set the real-space electrostatic and VdW cutoffs to 10.0 Å; a switch was applied to the latter starting at 9.0 Å. A cutoff 50.0 Å was used for the vacuum simulations. A Long Range Correction (LRC) term for truncated VdW interactions was applied during the MD simulations. 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 (α=0.5 and σ=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. This set of free energy calculations was carried out with SOMD from the Sire 2016.1 release.[42,43] Each alchemical transformation was divided into 17 evenly spaced windows and simulated for 2 ns each both in water and in vacuum. The absolute hydration free energies were computed by annihilating non-bonded interactions of the solute in two steps. In the first step the free energy change for discharging the solute was computed. In the second step the free energy change for turning off the Lennard-Jones terms of the discharged solute was computed. Each step was carried out using 17 evenly spaced windows. The starting coordinates for each window were obtained by energy minimization of the same pre-equilibrated configuration generated by FESetup. A velocity-Verlet integrator was employed with a 2 fs time step. Only bonds involving hydrogens which are not alchemically transformed were constrained. This approach is referred as the "unperturbed H bond constraint protocol". Temperature control was achieved with the Andersen thermostat,[74] with a stochastic collision frequency of $10\,\text{ps}^{-1}$. A Monte Carlo barostat assured pressure control,

with isotropic box edge scaling moves attempted every 25 time steps. A shifted atom–based Barker–Watts reaction field,[75] 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 LJ interactions. The reaction field was not employed in the vacuum legs, where a Coulombic potential without cutoff was used. A protocol to account for the different treatment of intramolecular electrostatics in vacuum and solution is described in the supporting information. The softcore parameters (Eq. S??x) were set to default values for all the transformations, specifically $n = 0$ for Coulombic interactions and $\alpha = 2.0$ for the LJ potential.[31] Additionally, an end-point correction for truncated VdW potentials was applied by post-processing of end-state trajectories as described previously elsewhere.[76,77]

Table 1: Summary of the technical details for the relative hydration free energy calculations carried out with the various codes.

	AMBER	CHARMM	GROMACS	SOMD
Version	AMBER16	c40b1	4.6.7	2016.1
Module	`pmemd, sander`	`PERT`	`gmx`	`somd-freenrg`
Protocol	Split protocol	Unified protocol	Split protocol	Unified protocol
Number of λ windows	11 (charge mutations) 21 (vdW mutations)	21 evenly spaced	31 (charge mutations) 31 (vdW mutations)	17 evenly spaced
Starting coordinates	FESetup pre-equilibration	FESetup pre-equilibration	FESetup pre-equilibration	FESetup pre-equilibration
Simulation length per window	2.5 ns	1.5 ns	4.2 ns	2 ns
Timestep	1 fs	1 fs	1fs	2fs
Electrostatic method	PME	PME	PME	atom-based RF
Solvated phase cutoff	8 Å	12 Å	10 Å	10 Å
Vacuum phase cutoff	no cutoff	no cutoff	50 Å	no cutoff
Constraint	none	none	none	H-bonds not perturbed
LRC corrections	during MD	post-processing	during MD	post-processing
Barostat	Monte Carlo	Berendsen	Parrinello-Rahman	Monte Carlo
Thermostat	Langevin	Berendsen	Langevin	Andersen
Soft core parameters	$r_{LJ} = (2\sigma_{ij}^6\lambda + r_{ij}^6)^{1/6}$ $r_{Coul} = (\beta\lambda + r_{ij}^p)^{1/p}$ $n = 1$	$r_{LJ} = (2\lambda + r_{ij}^2)^{1/2}$ $r_{Coul} = (\beta\lambda + r_{ij}^2)^{1/2}$ $n = 1$	$r_{LJ} = (2\sigma_{ij}^6\lambda + r_{ij}^6)^{1/6}$ $r_{coul} = r_{LJ}$ $n = 1$	$r_{LJ} = (2\sigma_{ij}\lambda + r_{ij}^2)^{1/2}$ $r_{Coul} = (\lambda + r_{ij}^2)^{1/2}$ $n = 1$

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". Equation 1 computes the free energy as

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

where $\mathscr{H}(\mathbf{q}, \mathbf{p}; \lambda)$ 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 is explicit. The angle brackets denote the ensemble average of the gradient of the Hamiltonian with respect to λ, at a given λ value. 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 tool`[78] for all analyses. 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 RAFE 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 $\Delta\Delta G_{\mathrm{hydr}}$ 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

$$\mathrm{err}(\Delta\Delta G_{\mathrm{hydr}}) = \frac{\sigma}{\sqrt{n}} \tag{2}$$

where σ is the sample standard deviation and n is the number of uncorrelated samples, as computed by the `pymbar` module. The SEM for component free energies is combined as:

$$\mathrm{err}(\mathrm{combined}) = \sqrt{\sum_i \sigma_i^2}. \tag{3}$$

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

$$\mathrm{MAE} = \frac{1}{n}\sum_{\mathrm{i}=1}^{n} |y_\mathrm{i} - x_\mathrm{i}| \tag{4}$$

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 they can be calculated with high precision,[22] and are simpler to set up and implement than relative calculations. Prior work has successfully compared calculated absolute hydration free energies across GROMACS and DESMOND codes.[79]

Table **2** summarizes results for the absolute hydration free energies. The table shows the data from simulations with the recommended protocol for each MD code, as discussed in detail in the following subsections. The precision of the calculated free energies is similar between AMBER, CHARMM and GROMACS, whereas the SOMD free energies are less precise. This may reflect differences in the lambda schedules and length of trajectories between the different codes. Nonetheless the standard errors are typically well under 0.1 kcal/mol, thus it becomes meaningful to investigate small differences of a few tenths of kcal/mol between codes.

The ΔG_hydr obtained with the various MD packages in this way agree quite well given statistical errors, although some larger deviations are apparent as well. GROMACS predicts a smaller ΔG_hydr for methanol by about 0.2 kcal mol^{-1}. Similarly, there are some small discrepancies in the toluene, 2–methylfuran and 2–methylindole cases, where CHARMM produces slightly smaller ΔG_hydr. These small discrepancies may be due to the differences in calculated densities between CHARMM and other codes (typically smaller by ca. 0.01 g/cm^3). The largest deviation can be found for one of the largest molecules (7–CPI) with the AMBER result being less negative than with the other MD packages by 0.4–0.8 kcal mol^{-1}. This par-

ticular discrepancy does not correlate with significant variations in density between AMBER and other codes.

As an additional check we computed densities in the fully decoupled states and compared the results to reported densities for a pure TIP3P water box. The average densities across all simulations are $(0.980 \pm 0.002)\,\mathrm{g/cm^3}$, $(0.973 \pm 0.002)\,\mathrm{g/cm^3}$, $(0.979 \pm 0.002)\,\mathrm{g/cm^3}$, $(0.976 \pm 0.003)\,\mathrm{g/cm^3}$ for AMBER, CHARMM, GROMACS and SOMD respectively. AMBER and GROMACS show higher densities presumably because a LRC term was applied during the MD simulations, whereas LRC terms for SOMD and CHARMM are only applied via post-processing of trajectories. For reference, a recent study from Lee–Ping et al. reports a TIP3P water density of $0.980\,\mathrm{g/cm^3}$.[80]

Table 2: Absolute hydration free energies (in kcal/mol) and end-state densities (in g/cm3) as obtained from AFE calculations. Uncertainties on the last decimal are given in parenthesis.

Solute	AMBER		CHARMM		GROMACS		SOMD	
	Free energy (kcal/mol)	Density (g/cm^3)	Free energy (kcal/mol)	Density (g/cm^3)	Free energy (kcal/mol)	Density (g/cm^3)	Free energy (kcal/mol)	Density (g/cm^3)
methane	2.47(1)	0.986(1)	2.48(1)	0.977(1)	2.44(1)	0.987(1)	2.52(2)	0.982(1)
methanol	-3.73(1)	0.988(1)	-3.72(1)	0.980(1)	-3.51(1)	0.988(1)	-3.70(5)	0.987(1)
ethane	2.50(1)	0.988(1)	2.50(1)	0.979(1)	2.48(1)	0.988(1)	2.56(1)	0.984(1)
toluene	-0.72(1)	0.991(1)	-0.64(1)	0.983(1)	-0.72(1)	0.991(1)	-0.55(2)	0.989(1)
neopentane	2.61(1)	0.990(1)	2.58(2)	0.981(1)	2.58(1)	0.990(1)	2.71(6)	0.987(1)
2-methylfuran	-0.49(2)	0.991(1)	-0.42(1)	0.983(1)	-0.51(1)	0.991(1)	-0.39(2)	0.989(1)
2-methylindole	-6.24(1)	0.993(1)	-6.06(1)	0.984(1)	-6.35(1)	0.993(1)	-6.06(4)	0.990(1)
2-CPI	-6.05(2)	0.995(1)	-6.18(4)	0.992(1)	-6.54(1)	0.994(1)	-6.14(9)	0.991(1)
7-CPI	-5.66(3)	0.995(1)	-6.28(3)	0.982(1)	-6.52(2)	0.995(1)	-6.1(1)	0.992(1)

Table **3** shows the MAD between SOMD, GROMACS, AMBER and CHARMM. CHARMM produces figures that agree the most with other MD packages. The largest difference reaches $0.2\,\mathrm{kcal\,mol^{-1}}$ for SOMD and GROMACS. Variabilities between the codes may be partly explained by differences in densities due to different treatments of long range electrostatics and vdW interactions.

Having established the predictive value from absolute transformations we now turn to computing $\Delta\Delta G_{\mathrm{hydr}}$ from relative mutations. Table **4** summarizes the results for the four

Table 3: Mean Absolute Deviations (MAD) ($\mathrm{kcal\,mol^{-1}}$) between relative free energies obtained with the absolute protocol for the SOMD, GROMACS, AMBER and CHARMM packages.

Package	GROMACS	AMBER	CHARMM
SOMD	0.20 ± 0.03	0.13 ± 0.04	0.08 ± 0.02
GROMACS		0.19 ± 0.01	0.15 ± 0.01
AMBER			0.12 ± 0.01

MD packages. Again the data is from the recommended protocol for each package (see detailed discussions in the following subsections).

We reviewed firstly internal consistency of the different codes with the computed absolute hydration free energies. For each implementation we counted the number of times a calculated relative free energy deviates from the difference in reference absolute hydration free energies by more than 0.1 kcal/mol. This is significantly above the estimated uncertainties in calculated free energies in most instances. According to this criterion, the AMBER explicit implementation is the least consistent (10 deviations), followed by AMBER implicit (6 deviations), SOMD (6 deviations), CHARMM (5 deviations), GROMACS (5 deviations). The perturbations that give a discrepancy are not the same across codes, for instance methane->toluene with AMBER explicit deviates from the reference absolute hydration free energies by 0.33 kcal/mol, but at most 0.04 kcal/mol with other codes. SOMD and GROMACS show deviations of ca. 0.25 kcal/mol for methanol->methane but this is not the case for AMBER (implicit or explicit) or CHARMM.

We next reviewed consistency between forwards and backwards relative hydration free energies. Again counting the number of deviations that exceed 0.1 kcal/mol indicates that AMBER explicit is the least consistent (3 deviations), followed by AMBER implicit (2 deviations), CHARMM (2 deviations), GROMACS (1 deviation), SOMD (1 deviation). The largest deviation is observed with AMBER implicit for 2-methylindole <-> methane (0.36 kcal/mol).

Next we compared relative free energies across packages. CHARMM tends to show

Table 4: Comparison of relative free energies of hydration for various MD packages as obtained from absolute (AFE) and relative (RAFE) transformations via unified or split protocols. The values deduced from AFE transformations (given in the first row) were taken from Table 1. Signs of the backward transformation have been reverted to correspond to the forward transformation.

Transformation[a]		AMBER[b]		CHARMM[c]	GROMACS[b]	SOMD[c]
		implicit[d]	explicit[d]			
ethane	methane		-0.02 ± 0.01	-0.03 ± 0.01	-0.04 ± 0.01	-0.05 ± 0.02
ethane	methane	0.02 ± 0.01	-0.13 ± 0.02	-0.09 ± 0.02	-0.04 ± 0.02	0.05 ± 0.02
methane	ethane	0.00 ± 0.03	-0.19 ± 0.03	-0.04 ± 0.01	-0.02 ± 0.01	0.01 ± 0.06
methanol	methane		6.20 ± 0.01	6.20 ± 0.02	5.95 ± 0.01	6.21 ± 0.06
methanol	methane	6.19 ± 0.01	6.20 ± 0.02	6.18 ± 0.01	6.20 ± 0.01	5.99 ± 0.05
methane	methanol	6.20 ± 0.03	6.15 ± 0.01	6.21 ± 0.01	6.20 ± 0.01	5.97 ± 0.04
ethane	methanol		-6.22 ± 0.01	-6.22 ± 0.02	-5.98 ± 0.01	-6.26 ± 0.05
ethane	methanol	-6.20 ± 0.01	-6.27 ± 0.01	-6.25 ± 0.01	-6.19 ± 0.01	-6.09 ± 0.03
methanol	ethane	-6.20 ± 0.01	-6.25 ± 0.01	-6.28 ± 0.01	-6.19 ± 0.01	-6.09 ± 0.02
toluene	methane		3.19 ± 0.01	3.12 ± 0.01	3.16 ± 0.01	3.07 ± 0.03
toluene	methane	3.24 ± 0.02	3.39 ± 0.02	3.04 ± 0.02	3.21 ± 0.01	2.89 ± 0.09
methane	toluene	3.42 ± 0.03	3.52 ± 0.03	3.09 ± 0.02	3.20 ± 0.01	3.06 ± 0.02
neopentane	methane		-0.13 ± 0.02	-0.11 ± 0.02	-0.14 ± 0.01	-0.19 ± 0.06
neopentane[e]	methane	0.32 ± 0.04	-0.03 ± 0.06	-0.35 ± 0.01	-0.15 ± 0.02	-0.20 ± 0.05
methane[e]	neopentane	0.25 ± 0.03	-0.07 ± 0.03	-0.24 ± 0.02	-0.16 ± 0.05	-0.13 ± 0.05
neopentane[f]	methane	-0.13 ± 0.01	-0.12 ± 0.02	-0.56 ± 0.02	-0.14 ± 0.01	-0.11 ± 0.01
methane[f]	neopentane	-0.13 ± 0.03	-0.12 ± 0.03	-0.40 ± 0.02	-0.18 ± 0.03	-0.10 ± 0.06
2–methylfuran	methane		2.96 ± 0.02	2.90 ± 0.01	2.95 ± 0.01	2.90 ± 0.03
2–methylfuran	methane	3.09 ± 0.01	3.10 ± 0.01	2.84 ± 0.03	2.93 ± 0.05	2.92 ± 0.05
methane	2-methyfuran	3.10 ± 0.03	3.15 ± 0.03	2.84 ± 0.02	2.96 ± 0.01	2.83 ± 0.03
2–methylindole	methane		8.72 ± 0.01	8.53 ± 0.02	8.79 ± 0.02	8.57 ± 0.03
2–methylindole	methane	8.78 ± 0.03	8.78 ± 0.04	8.49 ± 0.01	8.73 ± 0.03	8.64 ± 0.06
methane	2-methylindole	9.14 ± 0.02	9.13 ± 0.03	8.56 ± 0.02	8.74 ± 0.01	8.67 ± 0.08
2–CPI	7–CPI		0.39 ± 0.04	-0.11 ± 0.04	0.02 ± 0.05	0.08 ± 0.14
2–CPI[g]	7–CPI	0.36 ± 0.03	0.63 ± 0.06	-0.01 ± 0.01	-0.01 ± 0.03	-0.11 ± 0.07
7–CPI[g]	2–CPI	0.34 ± 0.05	0.50 ± 0.03	0.04 ± 0.01	-0.20 ± 0.04	-0.01 ± 0.08

[a] The values deduced from the AFE absolute of Table 1 are given first.
[b] split protocol.
[c] unified protocol.
[d] using either the implicit or the explicit dummy atom approach.
[e] central mapping.
[f] terminal mapping.
[g] partial re/discharge i.e. only the charges of the appearing and the disappearing 5–rings are switched.

relative free energies with smaller values for a number of transformations: neopentane, 2–methylfuran and 2–methylindole. SOMD displays smaller values $\Delta\Delta G_{\mathrm{hydr}}$ for 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 kcal mol^{-1} higher and CHARMM about 0.2 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 deviate further. AMBER deviates also quite strongly from the other codes in the cyclopentanyl indole cases.

The MADs of the relative free energy simulations are presented in Table **5**. They are on average slightly larger than the MADs from the absolute simulations (Table **3**) and reach 0.26 kcal mol^{-1} for AMBER compared with CHARMM.

Table 5: MAD (in kcal mol^{-1}) comparing relative free energies from relative simulations between SOMD, GROMACS, AMBER and CHARMM.

Package	GROMACS	AMBER	CHARMM
SOMD	0.11 ± 0.01	0.23 ± 0.01	0.15 ± 0.01
GROMACS		0.16 ± 0.01	0.13 ± 0.01
AMBER			0.26 ± 0.01

We also computed cycle closure errors from Table **6** for the closed cycle ethane→ methanol → methane → ethane (see Figure **2**). The results are shown in Table **6**. Uncertainties were estimated by propagating uncertainties from the individual perturbations. AMBER explicit dummy and CHARMM are the only protocol consistent within uncertainty estimates, but the deviations observed with the other protocols are small. The largest discrepancy is observed with the GROMACS unified PME protocol, with the error just under 0.2 kcal/mol.

Finally we also examined whether the codes reproduced consistent changes in mean box volumes between forward and backward transformations. We find that the codes are generally consistent with GROMACS giving the most precise volume changes, whereas SOMD gives the least precise volume changes (See Table S1 in the SI). This indicates that the barostats used by the different simulation packages relax volume fluctuations with different

Table 6: Cycle closure errors (in kcal mol^{-1}) for ethane→ methanol → methane → ethane

Package and Protocol	Closure Error
AMBER implicit	0.07 ± 0.04
AMBER explicit	0.02 ± 0.05
GROMACS split reaction field	0.05 ± 0.02
GROMACS unified reaction field	0.13 ± 0.03
GROMACS split PME	0.04 ± 0.01
GROMACS unified PME	0.18 ± 0.03
CHARMM	0.01 ± 0.03
SOMD	-0.11 ± 0.08

efficiency, or that they sample different volume fluctuations.

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 as of this writing. This will, however, be rectified in future versions.[81] A disadvantage of `sander` is that it cannot be used to simulate the λ end points,[82] such that the TI gradients need to be extrapolated (minimum and maximum allowed λs 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.[82]

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 **??**x). The issue becomes apparent 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 **7** below.

End point geometries appear to be another issue with AMBER simulations in both solution and vacuum. This is most obvious in the neopentane → methane test case with central mapping (see RAFE Setup and Figure **1**). As shown in Figure **??**x, 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 **??**x 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 transformations described in Tables **2** and **4**. 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 **7** shows selected results for transformations with SHAKE enabled for all bonds to hydrogens except those bonds that change bond length during transformation.

Table 7: 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		0.35 ± 0.02
methane[b]	neopentane	0.25 ± 0.03	-0.07 ± 0.03	-0.13 ± 0.02	
neopentane[c]	methane	-0.13 ± 0.01	-0.12 ± 0.02		
methane[c]	neopentane	-0.13 ± 0.03	-0.12 ± 0.03		

[a] implicit dummy atom protocol with $\delta t = 2$ fs and SHAKE on all H–bonds except perturbed bonds.
[b] central mapping.
[c] terminal 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 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 **2** and **4**). There are, however, a few notable deviations. Neopentane → methane with central mapping differs from the result with terminal mapping by about 0.4 kcal mol^{-1}. The terminal mapping and the free energies from the explicit dummy simulations are, however, consistent with the absolute transformations (Table **2**). We also observe a systematic deviation between forward and backward vacuum transformations in the 2–methylindole simulation (see Table **??**x). The gradient is consistently shifted by 0.2–0.4 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 produce robust and accurate results. Bugs regarding TI gradient accumulation in the 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 timesteps of 0.1 fs to 0.5 fs before the production run. The VSwitch option was used to apply a switching

function to the potential since that option is cannot be applied to forces for calculations run with the PERT module.

The 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 perturbations that involve large changes in solute size, 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 (see SI for details).

Table **8** shows the relative free energies obtained from CHARMM simulations. While

Table 8: Comparing CHARMM results for simulations with various split protocols. Signs of the backward transformation have been reverted to correspond to the forward transformation.

transformation		split $\Delta\Delta G$	unified $\Delta\Delta G$	absolute(unified) $\Delta\Delta G$
ethane	methane	-0.09 ± 0.01	-0.09 ± 0.02	-0.03 ± 0.01
methane	ethane	-0.04 ± 0.01	-0.04 ± 0.01	
methanol	methane	6.20 ± 0.01	6.18 ± 0.01	6.20 ± 0.01
methane	methanol	6.30 ± 0.01	6.21 ± 0.01	
ethane	methanol	-6.21 ± 0.01	-6.25 ± 0.01	-6.22 ± 0.02
methanol	ethane	-6.25 ± 0.01	-6.28 ± 0.01	
toluene	methane	3.22 ± 0.01	3.04 ± 0.02	3.12 ± 0.01
methane	toluene	3.28 ± 0.01	3.09 ± 0.02	
neopentane[a]	methane	-0.29 ± 0.01	-0.35 ± 0.01	-0.11 ± 0.02
methane[a]	neopentane	-0.15 ± 0.01	-0.24 ± 0.02	
neopentane[b]	methane	-0.42 ± 0.01	-0.56 ± 0.02	
methane[b]	neopentane	-0.31 ± 0.01	-0.40 ± 0.02	
2-methylfuran	methane	2.87 ± 0.01	2.84 ± 0.03	2.90 ± 0.01
methane	2-methylfuran	2.93 ± 0.01	2.84 ± 0.02	
2-methylindole	methane	8.88 ± 0.01	8.49 ± 0.01	8.53 ± 0.02
methane	2-methylindole	8.81 ± 0.01	8.56 ± 0.02	
2-CPI	7-CPI	-0.02 ± 0.01	-0.01 ± 0.01	-0.11 ± 0.04
7-CPI	2-CPI	-0.01 ± 0.01	0.04 ± 0.01	

[a] central mapping.
[b] terminal mapping.

results from all three protocols (split, unified, absolute) seem to be in good agreement with

each other, the split-protocol results are more precise due to the additional amount of data generated. It is notable that the split-protocol results are more similar to the ones obtained by other MD packages (i.e. neopentane and toluene), but the relative-unified results are more consistent with the CHARMM absolute simulations (e.g. 2-methylindole). Overall, the relative free energies obtained by these three different protocols are in good agreement with those reported for the other MD packages (Tables 1 and 3).

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. The keywords `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 avoid mass changes during the the free energy calculations. Unified protocols set all λ vectors the same.

Table **9** 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

Table 9: 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	methane	-0.025 ± 0.005	-0.035 ± 0.020	-0.017 ± 0.003	-0.030 ± 0.001	-0.06 ± 0.01	-0.04 ± 0.01
methane	ethane	-0.01 ± 0.02	-0.02 ± 0.01	0.046 ± 0.020[d]	0.01 ± 0.02		
methanol	methane	6.163 ± 0.006	6.197 ± 0.004	7.30 ± 0.02	7.380 ± 0.007	5.77 ± 0.01	5.95 ± 0.01
methane	methanol	6.168 ± 0.005	6.199 ± 0.008	7.09 ± 0.02	7.17 ± 0.02		
ethane	methanol	-6.123 ± 0.007	-6.185 ± 0.006	-7.117 ± 0.005	-7.21 ± 0.02	-5.83 ± 0.01	-5.98 ± 0.01
methanol	ethane	-6.124 ± 0.005	-6.193 ± 0.004	-7.338 ± 0.004	-7.404 ± 0.004		
toluene	methane	3.22 ± 0.01	3.211 ± 0.006	3.229 ± 0.008	3.22 ± 0.01	2.97 ± 0.01	3.16 ± 0.01
methane	toluene	3.25 ± 0.01	3.20 ± 0.01	3.22 ± 0.01	3.211 ± 0.001		
neopentane[e]	methane	-0.103 ± 0.008	-0.15 ± 0.02	-0.08 ± 0.02	-0.18 ± 0.03	-0.18 ± 0.01	-0.14 ± 0.01
methane[e]	neopentane	-0.11 ± 0.02	-0.16 ± 0.05	0.00 ± 0.03	-0.18 ± 0.03		
neopentane[f]	methane	-0.116 ± 0.007	-0.13 ± 0.01	-0.14 ± 0.01	-0.14 ± 0.01		
methane[f]	neopentane	-0.10 ± 0.03	-0.18 ± 0.03	-0.089 ± 0.007	-0.15 ± 0.02		
2-methylfuran	methane	2.986 ± 0.006	2.930 ± 0.050	3.05 ± 0.01	3.00 ± 0.01	2.87 ± 0.01	2.95 ± 0.01
methane	2-methylfuran	3.007 ± 0.004	2.96 ± 0.01	3.056 ± 0.006	3.01 ± 0.01		
2-methylindole	methane	8.71 ± 0.02	8.73 ± 0.03	8.73 ± 0.01	8.80 ± 0.03	8.44 ± 0.02	8.79 ± 0.02
methane	2-methylindole	8.73 ± 0.03	8.74 ± 0.01	8.30 ± 0.02	8.77 ± 0.04		
2-CPI	7-CPI	-0.07 ± 0.02	-0.03 ± 0.03	-0.10 ± 0.05	-0.2 ± 0.1	-0.02 ± 0.05	0.02 ± 0.02
7-CPI	2-CPI	-0.12 ± 0.06	-0.20 ± 0.04	-0.04 ± 0.06	-0.14 ± 0.09		

[a] results obtained from alchemical transformations with electrostatic and bonded scaling separate from vdW parameter change.
[b] results obtained from alchemical transformation with all parameters scaling together.
[c] results obtained from absolute free energy calculations.
[d] inverted sign
[e] central mapping
[f] terminal mapping

other software used in this study (Tables **2** and **4**). A noteworthy exception is the difference between the unified and split results of methane → methanol and its reverse process. This was investigated further with additional split protocol simulations using Coulomb softcore potentials (Table **10**).

Table 10: Relative hydration free energies of methanol → methane and methane → 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	
		RF	PME	RF	PME	RF	PME
transformation		$\Delta\Delta G$	$\Delta\Delta G$	$\Delta\Delta G$	$\Delta\Delta G$	$\Delta\Delta G$	$\Delta\Delta G$
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		

We noticed a difference of approximately 1.5 kcal mol^{-1} between the split protocol without Coulomb softcore potentials and both protocols that use it. The data shown in Figure **??**x suggests that softening of the electrostatic interactions requires adjustments in the λ-distance between states in the rapidly varying part of the $\partial\mathcal{H}/\partial\lambda$. A variant that combined 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 energies with the current GROMACS implementation.

Additionally it is worth mentioning is that relative free energy simulations that feature alchemical transformations of a hydrogen atom into a heavy atom will crash if the bond involving the hydrogen atom is constrained with algorithms such as SHAKE or LINCS. Successful simulations require turning off the bond constraint and decreasing the time step to 1 fs. 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 alchemically perturbed hydrogen atoms.

3.4 SOMD

Fig. ??x 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 **4** 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 ± 0.01 and MAD = (0.10 ± 0.03) 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. We stress that it is acceptable to artificially increase the atomic mass of hydrogen atoms because the calculated excess free energy changes do not depend on atomic masses.

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 (MAE = 0.09 kcal mol^{-1}) to those computed from simulations where noconstraints are applied on the solutes and a timestep of 1 fs is used (See Figure ??x).

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 RAFE $\Delta\Delta G$=(2.04 ± 0.01) kcal mol^{-1} whereas the absolute hydration free energy calculations give $\Delta\Delta G$=(-0.19 ± 0.06) kcal mol^{-1} as shown in tab. ??x and fig. ??x.

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 per-

turbed 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.

The reaction fields implemented in SOMD and GROMACS differ somewhat (atom-based shifted Barker Watts,[75] vs group based switched Barker Watts), but nevertheless SOMD and GROMACS RF produce comparable results with a MAD of $0.18\,\mathrm{kcal\,mol^{-1}}$. Overall, the SOMD free energy estimations are in good agreement with the other MD packages, as the MAD suggests (see Table **5**). For the methane $\rightarrow$ neopentane transformations SOMD yields consistent results between central and terminal mappings, as shown in Table **??**x. Reaction field and PME results are in good agreement. All SOMD RAFE 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.

4 Discussion and Conclusions

This study addressed whether contemporary MD packages such as AMBER, CHARMM, GROMACS and SOMD are 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 establishing a simulation protocol that leads to consistent results across codes 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 reasonably good agreement with each other (see Tables **2** and **4**). The average MAE between all codes 0.14 kcal/mol for

absolute free energies and 0.17 kcal/mol for relative free energies. This can be interpreted as the current "limit of reproducibility" for the field. 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 the particular version of the software we tested 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.

We were unable to define a *universal* protocol that could be recommended for use with all four codes. Unified protocols do not appear to work with 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. S**??**x) 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. 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.[83] A complete separation of lambdas may not be necessary though as a certain degree of overlap between vdW and Coulumb λ may be a viable solution[84] 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 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. We recommend using this dataset to test and benchmark future RAFE 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, 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 RAFE simulations, both to avoid crashes and incorrect results from 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 ease of use of the different software. For example, when a mutation entails both appearing and disappearing parts in split protocols there is the problem of intermediates having a non–integral total charge on the molecule. An alternative would be to totally discharge and then recharge the whole molecule which would have the advantage of eliminating one additional evaluation of the reciprocal sum in PME.[82] However this is not attractive as this could significantly increase the sampling needed to obtain converged free energy changes.

Another practical issue is the complex setup associated with the split protocol. For instance in GROMACS it is necessary to carry out two separate simulations per lambda because discharging and recharging groups cannot be selected separately. Lambda paths as

implemented in GROMACS could also be beneficial for other codes as they make the setup of split protocols easier. The alternative we have used in codes lacking this feature is to mimick this protocol through careful constructions of topologies via scripting.

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 in detail the efficiency of the respective protocols as this would require further, complex investigations. For absolute calculations the most demanding protocol and most precise protocol is GROMACS (200 million aggregate time–steps per solute, average SEM 0.011 kcal/mol), the least demanding protocol is CHARMM (31.5 million time–steps per solute, average SEM 0.015 kcal/mol). SOMD's aggregate time–steps is comparable to CHARMM (34 million time–steps) but the free energies are less precise (average SEM 0.045 kcal/mol). For relative calculations, the least demanding protocol is SOMD (17 million time–steps), and this is also the least precise (average SEM 0.048 kcal/mol). Remarkably the most demanding protocol (GROMACS 197.4 million time–steps, average SEM 0.020 kcal/mol) is less precise than CHARMM that used fewer time–steps (31.5 million time-steps, average SEM 0.015 kcal/mol). Further work should be pursued to understand what algorithmic details in the various implementations are important for the efficiency of the free energy calculations.

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.[20,48,49,85–88] To conclude, we hope this study will stimulate the field to improve 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.

Acknowledgement

HHL is supported through an EPSRC provided SLA, funding the core support of CCP-BioSim. CCPBioSim is the Collaborative Computational Project for Biomolecular Simu-

lation 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. DS and BR appreciate support from the National Science Foundation (NSF) through grant MCB-1517221 and additional computational resources were provided by the University of Chicago Research Computing Center.

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 thank Prof. Johan Åqvist and Brian Radak for valuable discussions on single vs. dual topology.

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

Supporting Information Available

Additional tables and details on the different free energy implementations are listed in the supporting information. All input files, setup, simulations and analysis of the dataset with the above mentioned packages are also freely available at https://github.com/halx/relative-solvation-inputs .

This material is available free of charge via the Internet at `http://pubs.acs.org/`.

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) Shirts, M. R.; Mobley, D. L. *Methods in Molecular Biology* **2013**, *22*, 271–311.

(25) Boresch, S.; Karplus, M. *The Journal of Physical Chemistry A* **1999**, *103*, 103–118.

(26) Woods, C. J.; Malaisree, M.; Hannongbua, S.; Mulholland, A. J. *The Journal of Chemical Physics* **2011**, *134*, 054114.

(27) Woods, C. J.; Malaisree, M.; Michel, J.; Long, B.; McIntosh-Smith, S.; Mulholland, A. J. *Faraday Discuss.* **2014**, *169*, 477–499.

(28) 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.

(29) 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.

(30) Pearlman, D. A. *The Journal of Physical Chemistry* **1994**, *98*, 1487–1493.

(31) Michel, J.; Verdonk, M. L.; Essex, J. W. *Journal of Chemical Theory and Computation* **2007**, *3*, 1645–1655, PMID: 26627610.

(32) Rocklin, G. J.; Mobley, D. L.; Dill, K. A. *The Journal of Chemical Physics* **2013**, *138*, 085104.

(33) Axelsen, P. H.; Li, D. *Journal of Computational Chemistry* **1998**, *19*, 1278–1283.

(34) Wan, S.; Bhati, A. P.; Zasada, S. J.; Wall, I.; Green, D.; Bamborough, P.; Coveney, P. V. *Journal of Chemical Theory and Computation* **2017**, *13*, 784–795, PMID: 28005370.

(35) Michel, J.; Essex, J. W. *Journal of Computer-Aided Molecular Design* **2010**, *24*, 639–658.

(36) Shobana, S.; Roux, B.; Andersen, O. S. *The Journal of Physical Chemistry B* **2000**, *104*, 5179–5190.

(37) Liu, S.; Wang, L.; Mobley, D. L. *Journal of Chemical Information and Modeling* **2015**, *55*, 727–735, PMID: 25835054.

(38) 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.

(39) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, 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.

(40) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. *SoftwareX* **2015**, *1–2*, 19–25.

(41) Scott, W. R. P.; Hünenberger, P. H.; Tironi, I. G.; Mark, A. E.; Billeter, S. R.; Fennen, J.; Torda, A. E.; Huber, T.; Krüger, P.; van Gunsteren, W. F. *The Journal of Physical Chemistry A* **1999**, *103*, 3596–3607.

(42) Woods, C.; Mey, A. S.; Calabro, G.; Michel, J. Sire molecular simulations framework. 2016.

(43) 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.

(44) 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.

(45) 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.

(46) 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.

(47) 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.

(48) Lundborg, M.; Lindahl, E. *The Journal of Physical Chemistry B* **2015**, *119*, 810–823, PMID: 25343332.

(49) Loeffler, H. H.; Michel, J.; Woods, C. *Journal of Chemical Information and Modeling* **2015**, *55*, 2485–2490.

(50) Balasubramanian, V.; Bethune, I.; Shkurti, A.; Breitmoser, E.; Hruska, E.; Clementi, C.; Laughton, C. A.; Jha, S. *CoRR* **2016**, *abs/1606.00093*.

(51) Skyner, R. E.; McDonagh, J. L.; Groom, C. R.; van Mourik, T.; Mitchell, J. B. O. *Phys. Chem. Chem. Phys.* **2015**, *17*, 6174–6191.

(52) Mobley, D. L.; Guthrie, J. P. *Journal of Computer-Aided Molecular Design* **2014**, *28*, 711–720.

(53) To complete the data compiled in ref[44] we note that in SOMD (Rev 2016.1) Coulomb's constant is 332.0637090025476 kcal mol^{-1} angstrom e^{-2} .

(54) Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. *Chemical Physics Letters* **1994**, *222*, 529–539.

(55) Zacharias, M.; Straatsma, T. P.; McCammon, J. A. *The Journal of Chemical Physics* **1994**, *100*, 9025–9031.

(56) Simonson, T. *Mol. Phys.* **1993**, *80*, 441–447.

(57) Steinbrecher, T.; Mobley, D. L.; Case, D. A. *The Journal of Chemical Physics* **2007**, *127*, 214108.

(58) 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.

(59) Boresch, S.; Karplus, M. *The Journal of Physical Chemistry A* **1999**, *103*, 119–136.

(60) Rocklin, G. J.; Mobley, D. L.; Dill, K.; Hünenberger, P. H. *The Journal of chemical physics* **2013**, *139*, 184103.

(61) Loeffler, H. H.; Sotriffer, C. A.; Winger, R. H.; Liedl, K. R.; Rode, B. M. *Journal of Computational Chemistry* **2001**, *22*, 846–860.

(62) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *Journal of Computational Chemistry* **2004**, *25*, 1157–1174.

(63) Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I. *Journal of Computational Chemistry* **2000**, *21*, 132–146.

(64) Jakalian, A.; Jack, D. B.; Bayly, C. I. *Journal of Computational Chemistry* **2002**, *23*, 1623–1641.

(65) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *The Journal of Chemical Physics* **1983**, *79*, 926–935.

(66) Shivakumar, D.; Harder, E.; Damm, W.; Friesner, R. A.; Sherman, W. *Journal of Chemical Theory and Computation* **2012**, *8*, 2553–2558, PMID: 26592101.

(67) 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.

(68) Steinbrecher, T.; Joung, I.; Case, D. A. *Journal of Computational Chemistry* **2011**, *32*, 3253–3263.

(69) Deng, Y.; Roux, B. *J. Phys. Chem.* **2004**, *108*, 16567–16576.

(70) Naden, L. N.; Pham, T. T.; Shirts, M. R. *Journal of Chemical Theory and Computation* **2014**, *10*, 1128–1149.

(71) Naden, L. N.; Shirts, M. R. *Journal of Chemical Theory and Computation* **2015**, *11*, 2536–2549.

(72) Pitera, J. W.; Gunsteren, W. F. v. *Mol. Simul.* **2002**, *28*, 45–65.

(73) Anwar, J.; Heyes, D. M. *J. Chem. Phys.* **2005**, *122*, 224117.

(74) Andersen, H. C. *The Journal of Chemical Physics* **1980**, *72*, 2384–2393.

(75) Barker, J.; Watts, R. *Molecular Physics* **1973**, *26*, 789–792.

(76) Shirts, M. R.; Mobley, D. L.; Chodera, J. D.; Pande, V. J. *Journal of Physical Chemistry B* **2007**, *45*, 13052–13063.

(77) Bosisio, S.; Mey, A.; J., M. *Journal of Computer-Aided Molecular Design* **2017**, 61.

(78) Klimovich, P. V.; Shirts, M. R.; Mobley, D. L. *Journal of Computer-Aided Molecular Design* **2015**, *29*, 397–411.

(79) Klimovich, P. V.; Mobley, D. L. *Journal of Computer-Aided Molecular Design* **2010**, *24*, 307–316.

(80) Wang, L.-P.; Martinez, T. J.; Pande, V. S. *The Journal of Physical Chemistry Letters* **2014**, *5*, 1885–1891, PMID: 26273869.

(81) Lee, T.-S.; Hu, Y.; Sherborne, B.; Guo, Z.; York, D. M. *Journal of Chemical Theory and Computation* **0**, *0*, null, PMID: 28618232.

(82) Kaus, J. W.; Pierce, L. T.; Walker, R. C.; McCammon, J. A. *Journal of Chemical Theory and Computation* **2013**, *9*, 4131–4139.

(83) Shirts, M. R.; Mobley, D. L.; Chodera, J. D. *Annual Reports in Computational Chemistry* **2007**, *3*, 41–59.

(84) Procacci, P.; Cardelli, C. *Journal of Chemical Theory and Computation* **2014**, *10*, 2813–2823, PMID: 26586508.

(85) Christ, C.; Fox, T. *Journal of chemical information and modeling* **2013**,

(86) 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.

(87) Ramadoss, V.; Dehez, F.; Chipot, C. *Journal of Chemical Information and Modeling* **2016**, *56*, 1122–1126, PMID: 27214306.

(88) 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