Reproducing Relative Alchemical Free Energies of Hydration (draft)

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

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

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

Abstract

Alchemical free energy calculation is a very important branch of modern simulation techniques. Contemporary Molecular Dynamics and Monte Carlo software such as AMBER, CHARMM, GROMACS and Sire/SOMD include support for the method. Implementation details vary among those codes but users expect reliability and reproducibility i.e. a simulation must yield a comparable free energy within statistical bounds regardless of code used. *Relative* alchemical free energy simulation has been less well tested than its absolute counterpart in this context, however, relative transformations

[*]To whom correspondence should be addressed
[†]Scientific Computing Department, STFC
[‡]University of Edinburgh
[¶]University of California, Irvine
[§]University of California, Irvine
[‖]University of Chicago

promise to be computationally and statistically more efficient. Here we present the results for relative hydration free energies (RAFE) for a set of small organic molecules and show that free energies can be satisfactorily reproduced with aforementioned codes. We will also recommend simulation protocols, setup procedures and analysis techniques.

1. Objectives: Do the relative free energy results produced across different softwares agree with each other? What if they don't?

2. Objectives: Can we outline what should be a standard procedure to calculate relative free energies of hydration alchemically?

3. Objectives: We want to emphasise that we aim at improving codes/protocols/practices and not highlight "bad" codes

4. Be clear on what reproducibility means here e.g. no exact numerical

1 Introduction

The free energy is a fundamental function of thermodynamics and kinetics as it explains how processes in nature evolve. The direction that, say, a chemical reaction takes can be immediately determined from the knowledge of the free energy difference of reactant and product. The free energy landscape of a given system, however, can be very complicated and rugged such that barriers exist which impose limits on how fast the process can take place. It is therefore of little surprise that the determination of this reversible work is of utmost importance to all 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 through computers [1–5] has been particularly attractive as it promises to circumvent limitations of experimental approaches, processes can be understood at the molecular level and it offers the potential of being more cost and time effective. 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] Roughly speaking, fast methods tend to be less accurate while more accurate methods tend to be slow to compute. Nevertheless, rigorous methods are obligatory in obtaining accurate, precise and reliable results, while less accurate methods can be used as an input filter for more expensive approaches.

One such method is the so-called *alchemical* free energy approach whose theory is firmly rooted in statistical thermodynamics and is argued to be the most accurate method in quantitative prediction of free energies.[1,7–9] The method has been applied in various forms for many decades now since the early days of computer simulation.[10–15] The method has gained renewed attention in recent years — concomitant with improvements in computer hardware design — both within the traditional equilibrium framework[16–18] but also increasingly being used in non-equilibrium regimes.[19–21] The name comes from the nonphysical intermediates that often need to be created to smoothly interpolate between end states and because parts or all of a molecule may appear or "disappear" in a transformation. In the context of force field methods the transformation takes place in parameter space i.e. the force field's parameters determining strength and equilibrium of interactions are varied by scaling. This can be a particular efficient approach as it does not require translocation in configuration space. For instance, the dissociation of a ligand from a large biomolecule may involve many degrees of freedom while, at the same time, it is generally unclear along which coordinates a translocation simulation should take place.

Alchemical free energy simulations are constructed around the concept of thermodynamic cycles. As the free energy is a state function it must always add up to zero within a closed cycle. This also implies that the reversible work can be computed arbitrarily along conveniently chosen legs of the cycle. E.g. 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 an nonphysical alchemical calculation. As mentioned above, the latter may be computationally and statistically

more efficient.

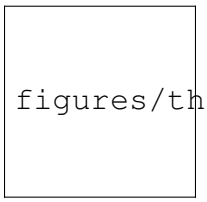

Figure 1: The thermodynamic cycle to compute the relative free energy of hydration $\Delta\Delta G_{\mathrm{hydr}} = \Delta G_{\mathrm{sol}} - \Delta G_{\mathrm{vac}} = \Delta G'' - \Delta G'$. The example is for the ethanol $\leftrightarrow$ methanol transformation. Alchemical simulations can be performed along the non–physical horizontal legs while vertical legs illustrate the physical process (which is directly accessible through absolute alchemical free energy simulation, see [FreeSolv]).

Absolute (standard) alchemical free energy calculation has been a particular focus of recent years.[16–19,21] *Absolute* here really means that the equilibrium constant of a physical reaction, e.g. binding and dissociation, can be calculated directly by completely decoupling a whole molecule from its environment and the term is mostly being used to discriminate against *relative* techniques (see below). These schemes may require two simulations along the parallel legs of a rectangular thermodynamic cycle but approaches that produce the reversible work directly in one simulation have been proposed too.[22,23]

Relative alchemical free energy (RAFE) calculations "mutate" one molecule into another one. This is most efficiently achieved[24,25] by making use of the single topology method.[15,24,26] Single topology means that there is only one description of the molecule to be mutated. Thus, atom types are directly transformed into the new type, typically by linearly scaling force field parameters. Disappearing/appearing atoms are balanced with "dummy" atoms, i.e. atoms that have no non–bonded interactions in the end state but retain the bonded terms of the original atom. AMBER is a special case as it does not require the use of dummy atoms: the bonded terms of atoms that do not exist in the end state will not contribute to the free energy implying that contributions from dummy atoms perfectly cancel in the thermodynamic cycle. RAFEs are useful e.g. in ranking which one of a set of small molecules binds strongest to a chosen target. This approach has recently gained more traction in the context of relative free binding energies of biomolecules.[27]

The single topology approach[24] requires a certain similarity between the two mutated states. This means topological and structural similarity but also chemical similarity is of importance e.g. chirality and binding modes where the relative three dimensional arrangement in space must be taken care of. Furthermore, ring breaking is technically challenging[28] but it has also been shown that this should be done only in certain circumstances.[29,30]

When the two molecules are very dissimilar, the dual topology method[24,26] can be applied to compute relative free energies. In this approach all atoms in the end states are duplicated and thus both sets are present at all times but don't interact with each other. Only non–bonded interactions need to be scaled such that the disappearing end state corresponds to an ideal gas molecule.[26] This, however, comes with additional complications as two independent molecules can drift apart and so suffer from the "wandering" ligand problem as in absolute transformations.[16–18] Topolological similarity can only be exploited when the charges of the common core are explicitly made equivalent.[6] This approach, however, shifts all the chemical variability exclusively to the dummy atoms and is thus of only limited use.

Technically, a dual topology calculation is the same as two absolute calculations run simultaneously in opposite directions. It has been shown though that with the introduction of special restraints or constraints this can be a viable option.[31,32] A covalent link, e.g. as in side–chain mutation simulations, provides a natural restraint such that dual topology simulations can be applied without further problems. Modern MD software e.g. AMBER,[33] CHARMM,[34] GROMACS,[35] GROMOS[36] and Sire/SOMD[37,38] offer a hybrid single/dual topology approach i.e. the user can specify which part of a perturbed group should be handled by which method.[30] The perturbed group comprises of all atoms that differ in at least one force field parameter between the end states.

As alluded to above, reliability is a principal matter of concern. In particular, we need to ensure reproducibility of free energy results among computer codes. To the best of our knowledge this has not been systematically tested yet for a set of different MD packages. Given a predefined force field and run–time parameters we should be able to obtain com-

parable free energy results within statistical convergence limits. In practice, we have the problem, however, that the methods and algorithms used in MD programs are not always present in all packages or are the same, like pressure and temperature scaling, integrators, etc. Nevertheless, the reversible work computed with any simulation software should be expected to be reproducible. Modern MD packages support a wide range of force fields and methods such that these packages are replaceable with each other to an ever increasing extent and the choice of the right package for the user becomes less and less a matter of technical restrictions.

In this work we present the results of relative hydration free energies of a set of small organic molecules (see Fig. 2). Solvation free energies have a wide range of uses and various methods exist to compute them.[39] They are also needed to calculate binding free energies where the simulation in solution (see Fig. 1) is combined with a mutation of the molecule bound to a partner, and other important physical properties.[39] A large database of hydration free energies computed from AFE simulation, FreeSolve, has been presented recently (cite). Here, we are interested in the reproducibility of RAFE with the simulation programs AMBER, CHARMM, GROMACS and Sire/SOMD. We will discuss the reversible work results obtined with these packages as well as making recommendations regarding simulation protocols, setup procedures and analysis techniques.

2 Methods

2.1 Alchemical Free Energy Implementations

We begin by working out the differences in the alchemical free energy implementations of the four MD codes AMBER, CHARMM, GROMACS and Sire/SOMD. One key difference is the softcore functions[40,41] used in each code as summarised in section 1 of the SI. Softcore functions are used to avoid numerical and thus stability problems of the conventional van der Waals and Coulombic potentials[42] as they have singularities at zero distance (vertical

asymptotes). Direct scaling of these potentials causes the functions to increasingly behave like hard–sphere potentials as $\lambda \to 0$. This implies a higher probability of other atoms to penetrate into the highly repulsive short–range portion of the potential which can lead to strongly fluctuating forces/energies and to severe instabilities in the integrator.

Other differences are whether the code scales individual (force field) parameters and/or the total energy,[26] or if the code allows constraints for bonds with changing bond lengths. These and other details will be outlined below. The perturbed group comprises of all atoms that need to be transformed, i.e. any atom that differs in at least one force field parameter in the other end state.

AMBER This code is strictly dual topology and all terms are energy–scaled. The code allows, however, to map atoms in a single topology fashion and computes the forces appropriately (linearly) scaled for each atom in the pair. The perturbed group must be entirely duplicated i.e. for sander this means two topology files with one end state each, and for pmemd both end states in one topology file. The softcore potential applies to dummy atoms only which can be freely chosen by the user i.e. also for atoms that have an equivalent in the other state. Explicit dummy atoms are not needed as the code will only compute bonded contributions for "real" atoms thus ignores bonded energies involving dummy atoms (implicit dummy protocol). The code cannot handle constraints of changing bond lengths in the perturbed group. There is only on global λ for parameter transformation. Separated protocols (see below) must be emulated through careful construction of topologies by keeping force field parameters constant in the files.

CHARMM The PERT module duplicates the topology similar to sander but requires explicit dummy atoms. All terms are energy–scaled. The PSSP softcore potential is applied to *all* atoms in the perturbed group (see section 1 in the SI). The code can handle constraints of changing bond lengths in the perturbed group but this may cause wrong results with PSSP softcores (Stefan Boresch, private communication). There is only one global λ for parameter

transformation, however, the scripting facilities allow run time modification of topologies e.g. by setting charges or vdW 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. Non–bonded terms are energy–scaled. The softcore potential applies to dummy atoms only which are determined if atoms have zero vdW parameters. The code allows constraints of changing bond lengths in the perturbed group but this can lead to instabilities and wrong results (Michael Shirts, private communication). There are separate λs for vdW, Coulomb and bonded parameters (and some other).

Sire/SOMD This code uses a single topology description. The final state is constructed at run time from the initial state with a "patch" (list of force field parameters to be modified). Bond and angle terms are parameter–scaled while all other terms are energy–scaled. The softcore potential applies to dummy atoms only. 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. Sire/SOMD is Sire[37] employing OpenMM[38] for MD simulation.

2.2 RAFE Setup

The setup for all relative free energy simulations has been carried out with the tool FESetup[43] in version 1.2. FESetup is a perturbed topology writer for AMBER, CHARMM, GROMACS, Sire/SOMD and also NAMD[44] (within the limits of the dual topology approach). The tool makes use of a maximum common substructure search algorithm to automatically compute atoms that can be mapped i.e. atoms that have a direct relationship to an equivalent atom in the other state. This means atom type to atom type conversion and the only current limit is that rings are required to be preserved.[29] In this way we achieve a maximal single topology description: any atom that does not match will be made a dummy atom. FESetup allows

equilibration of the solvated simulation systems and ensures that "forward" and "backward" simulations will have the same amount of total atoms. 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. 43.

Figure 2 shows all 18 transformation considered in this study including "forward" and "backward" mutations. RAFE simulations are principally fully symmetric i.e. they do not have a directionality with respect to the coupling (order) parameter λ. Performing simulations in either direction, however, is applied here as a test for possible discrepancies.

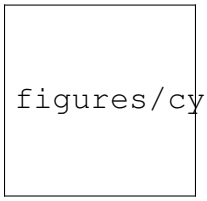

Figure 2: The thermodynamic cycles considered in this study. To compute the free energy of hydration all pair–wise transformations have to be carried out once in solution and once in vacuum. Green and blue colours in neopentane show two alternative mappings for methane. The numbers in red denote the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

The ethane $\rightarrow$ methanol transformation is traditionally regarded as a standard test for RAFE simulations.[15,45] The other transformations are centred around mutations from and to methane. The 2–cyclopentanylindole to 7–cyclopentanylindole transformation has been added to include both deletion as well as insertion of sub–parts of the perturbed group in one simulation. For neopentane $\rightarrow$ methane we point out that there are two alternative mappings possible, see Figure 2. One in which methane is mapped with a terminal methyl (green) and the other one where the methane carbon is mapped with the central carbon in neopentane (blue). Results from both mutations will be shown and discussed.

2.3 RAFE Simulation Protocols

One of the major concerns in a reproducibility study is to ensure consistency in the applied protocols. This is complicated by the fact that MD software employs a wide range of methods

and algorithms that may not be available in other MD software. For example, pressure and temperature scaling, integrators and other algorithms can be very different. It is also unclear if and how implementation details can affect results, in particular see discussion in subsection 2.1 of alchemical free energy implementations.

In this study we look 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 keep problems with sampling low, and avoid difficulties with charged particles.[46,47] The force field was chosen to be GAFF[48] in version 1.8 utilising AM1/BCC charges[49,50] for the solute and TIP3P[51] for the solvent. The quality of free energies of various small molecule force fields has been shown elsewhere, see e.g. Refs. 52,53.

While the MD packages principally allow a "one–step" transformation that is both van der Waals and Coulombic contributions can vary simultaneously, it can be(is? ref?) more efficient to carry out a separated protocol. 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[40,41] (see section 1 in the SI for details) on the vdW term only. It is important to note that in the separated protocol charges have to be switched off before vdW parameters (and vice versa for the transformation in opposite direction) to avoid collapse of other molecules e.g. solvents onto a "naked" charge, see section 2 in the SI. This has consequences for setting up simulations as discussed (somewhere else).

All simulations were started from the simulation box pre–equilibrated with FESetup.[43] It should be noted, however, that in constructing the system steric overlaps between the solute and the solvent can happen. This is because the number of atoms are always chosen to be the same for forward and backward setups by using the larger box of the two. Thus, in transformation from a smaller to a larger solute water molecules may be in close distance. The simulations were run at 298.15 K and 1.0 bar.

AMBER The starting coordinates were usually taken from the starting state as taken from the setup step. In some cases it was necessary though to use coordinates from a nearby λ state. Water hydrogens (TIP3P) were constraint with SHAKE. None of the atoms in the perturbed group were constraint and so the time step was set to 1 fs (see Sire protocol below for an alternative approach). Temperature was controlled through a Langevin thermostat and pressure through a Monte Carlo barostat.

Sire/SOMD All simulations were carried out with Sire/OpenMM 6.3 (revision 16.1). Each alchemical transformation was divided into 17 evenly spaced windows and simulated for 2 ns both in water an vacuum phase. A velocity-Verlet integrator was employed with 2 fs timestep, constraining all the hydrogen bonds not alchemically transformed (unperturbed H protocol). An atom–based Barker–Watts reaction field[54] with a dielectric constant of 82.0 was employed for water phase simulations. A cutoff of 12 Å was set for non-bonded interactions and periodic boundary conditions were imposed. Temperature control was achieved with the Andersen thermostat[55] with a coupling constant of $10\,\mathrm{ps}^{-1}$. A Monte Carlo barostat assured pressure control, attempting isotropic box edge scaling every 25 time steps. At each λ window all bond, angle and dihedral energy terms were scaled, differently (how?) from Gromacs and Amber. For each vacuum simulation a free energy correction term G_{FUNC} was evaluated to treat intramolecular Coulombic interactions consistently between water and vacuum legs of the thermodynamic cycle, as explained in[56] (some details here?).

GROMACS We used GROMACS 4.6.7 to carry out simulations for the relative hydration free energies ($\Delta\Delta G_{\mathrm{hydr}}$). Each transformation had its Gibbs free energy calculated: (i) in a single topology approach on which van der Waals energy terms were changed separately from the electrostatic and bonded components; (ii) in a single topology approach that changed bonded, van der Waals, and electrostatic terms simultaneously; (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 fs

in water and a vacuum. We used the Langevin integrator implemented in GROMACS with default friction coefficient of 1.0 ps/m_{atom}. Absolute hydration free energies were calculated using a protocol used in previous works [refs]A Parrinello-Rahman barostat with $\tau_p = 10$ ps and compressibility equal to 4.5 $\cdot 10^{-5}$ bar^{-1}. We used two methods to calculate electrostatic interactions: Particle Mesh Ewald (PME) and Reaction Field with a dielectric of 78.3, as implemented in the software. We set the non-bonded cutoff to 10 angstroms, with a switch starting at 9 angstroms. PME calculations were of order 6 and had a tolerance of 1.0 $\cdot 10^{-6}$, with a grid spacing of 1 angstrom. Ewald sums were performed in three dimensions. All transformations required the use of soft-core potentials to avoid numerical problems in the free energy calculation. We chose the 1-1-6 soft-core potential for Lennard-Jones terms and used the default soft-core Coulomb implementation in paths where charges, van der Waals, and bonded terms were modified together.

2.4 Analysis

Various estimators have been proposed to obtain the free energy. Early work by Kirkwood[57] expresses the free energy as an ensemble average of the derivative of the Hamiltonian with respect to the coupling parameter λ. The method is now known as Thermodynamic Integration (TI). Zwanzig devised the exponential formula[58] (EXP), also known as Free Energy Perturbation (FEP) or thermodynamic perturbation (TP), which calculates the free energy from the exponential average of the energy difference of the end states. The energy difference is computed with the configuration of one end state being used for both end states assuming that this configuration is representative for either state. As the phase–space overlap needs to be sufficiently large[59,60] the EXP approach typically needs intermediates, controlled by λ. However, it has been shown that EXP has an asymmetric bias depending on the directionality of λ[61] and that the Bennett Acceptance Ratio (BAR) method[11] is considerably more effective in obtaining an accurate result.[62] BAR is a generalisation of EXP by making explicit use of the "forward" ($\lambda_i \rightarrow \lambda_{i+1}$) and "backward" ($\lambda_{i+1} \rightarrow \lambda_i$) estimates. Later is has

been demonstrated that this can be effectively extended to incorporating more than just the immediate λ neighbour and, in fact, all other λs. This approach has been called multi–state BAR (MBAR)[63] method. MBAR has been shown to have the lowest variance of any known estimator.[64]

In this work we primarily focus on TI as this is supported by all 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. The angle brackets denote the ensemble average of the gradient of the Hamiltonian with respect to λ at a given λ value. An AFE simulation is typically carried out in a series of equilibrium simulations at discrete values of λ but the gradient can also be evaluated with a continuously varying coupling parameter as a function of the simulation time. The free energy is finally computed through a suitable numerical integration method.

Results from additional estimators will be given where available. We have used the alchemical analysis tool[65] for all free energies. This tools provides various estimators such as TI, TI with cubic splines, BAR and MBAR. All data can be sub–sampled to eliminate correlated data.

All RAFE simulations where run in triplicate in forward as well as backward direction so in total 6 simulations per mutation pairs. The final hydration free energy $\Delta\Delta G_{\mathrm{hydr}}$ was computed as average of each direction. 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 size of the sample. The SEM for component free energies is combined as

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

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

3 Results

5. what protocols did not work

6. problems: e.g. methane 2-methylindole: needed to use restart file from l=0.95 to start simulation at l=1.0.

3.1 AMBER

Using AMBER for RAFE simulations has revealed several problems with the implementation. Some bugs were idendified and have been fixed by the developers e.g. energy minimisation in sander led to diverged coordinates for mapped atoms which is, however, a necessary condition for a single topology description. Other issues are that vacuum simulation have to be carried out separately with the sander program because pmemd cannot handle AFE simulations in vacuum at the moment. This will, however, be rectified in future versions (Ross Walker, private communication). A disadvantage of sander is that it cannot be used to simulate the λ end points[66] 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.[66]

We also found that, in contrast to the other three codes, AMBER cannot correctly reproduce relative free energies in a 1–step protocol i.e. when all force field parameters are scaled simultaneously (see Table S1). This appears to be a problem when more than a few dummy atoms are involved while the 1–step protocol works for the smaller transformations (refer to Figure 2). The separated RAFE protocol and absolute free energies, however, are very close to the other MD packages as demonstrated in Table 1.

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 (central mapping). As shown in Figure S3 the methane end state exhibits too long 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, on average, 1.09 Å. Figure S3 demonstrates how this depends on the number of dummy atoms immediately surrounding the central atom.

We also compare free energies obtained from the implicit dummy approach in AMBER with results form explicit dummy atom simulations and results from absolute transformation. Table 1 lists the free energies for these three approaches together with the SEM. SHAKE was explicitly deactivated for all bonds in the perturbed region in these protocols. In addition, the table also shows results of transformations with SHAKE enabled for all bonds to hydrogens except those bonds that change bond length during transformation. These free energies are computed from a single run as we generally observe very small SEM and in practice one simulation is sufficient to obtain converged free energy averages for all the systems studied here. 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, the SHAKE protocol appears to be a viable solution to increase the performance of RAFE simulations. 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 counter–part in the other state because *all* bonds emanating from this atom would be

Table 1: Comparing AMBER results for simulations with implicit and explicit dummy atoms, and results from absolute transformation. A few select cases with SHAKE enabled and a time step of 2 fs are shown in addition. Simulations were carried out with the separated protocol.

		implicit		explicit		absolute		SHAKE[a]
transformation		$\Delta\Delta G$	SEM	$\Delta\Delta G$	SEM	ΔG^b	SEM	$\Delta\Delta G$
ethane	methane	0.021	0.014	-0.127	0.016	-0.022	0.012	
methane	ethane	0.001	0.025	0.187	0.028			
methanol	methane	6.189	0.014	6.200	0.017	6.201	0.012	
methane	methanol	-6.195	0.029	-6.145	0.013			
ethane	methanol	-6.200	0.012	-6.268	0.009	-6.223	0.014	-6.196
methanol	ethane	6.196	0.014	6.252	0.011			
toluene	methane	3.240	0.020	3.386	0.021	3.193	0.014	3.292
methane	toluene	-3.422	0.025	-3.523	0.033			
neopentane[c]	methane	0.315	0.038	-0.027	0.057			0.367
methane[d]	neopentane	-0.250	0.032	0.071	0.026	-0.132	0.016	
neopentane[d]	methane	-0.132	0.012	-0.118	0.015			
methane[e]	neopentane	0.125	0.031	0.123	0.034			
2-methylfuran	methane	3.089	0.014	3.102	0.009	2.964	0.023	
methane	2-methyfuran	-3.100	0.032	-3.147	0.033			
2-methylindole	methane	8.778	0.025	8.777	0.041	8.717	0.009	
methane	2-methylindole	-9.138	0.022	-9.130	0.034			
2–cyclopentanylindole[e]	7–cyclopentanylindole	0.361	0.032	0.628	0.059	0.388	0.036	
7–cyclopentanylindole[e]	2–cyclopentanylindole	-0.335	0.045	-0.497	0.032			

[a] implicit dummy atom protocol with $\delta t = 2$ fs and SHAKE on all H–bonds except perturbed bonds.
[b] sign for the forward transformation.
[c] central mapping.
[d] terminal mapping.
[e] partial re/discharge i.e. only the charges of the appearing and the disappering 5–rings are switched.

affected.

We can also compute the cycle closure error from Table 1 for the closed cycle ethane→ methanol → methane → ethane (see Figure 2). The free energy difference within a closed thermodynamic cycle must necessarily be zero. For the implict dummy simulation we calculate a cycle error for $\Delta\Delta G_{\mathrm{hydr}}$ of $(0.069 \pm 0.041)\,\mathrm{kcal\,mol^{-1}}$ and for the the explicit dummy simulation the error is $(-0.016 \pm 0.047)\,\mathrm{kcal\,mol^{-1}}$.

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. There are, however, a few notable deviations. Neopentane → methane with central mapping differs from the result with terminal mapping by about $0.4\,\mathrm{kcal\,mol^{-1}}$ (cf. Table 1). The terminal mapping and the free energies from the explicit dummy simulations are, however, consistent with the absolute transformations. We also observe a systematic deviation between forward and backward vacuum transformations in particular in the 2–methylindole simulation (see Table S2). A discrepancy of consistently 0.2–$0.4\,\mathrm{kcal\,mol^{-1}}$ is evident from every λ step of the vdW plus bonded transformation with both implicit and explicit dummy atoms).

3.2 CHARMM

7. bug in TI gradient accumulation in parallel runs (does not affect serial?, does not affect EXP)

8. cannot handle LRC: test with larger cutoffs and/or LRC correction with arbitrary, single structure; check http://pubs.acs.org/doi/abs/10.1021/jp0735987

3.3 GROMACS

9. investigate why methane ethane and ethane methane differ so much from the other packages

10. make a compact $\partial \mathrm{H}/\partial\lambda$ plot

11. make figures XX1 and XX2 (I think they should probably go in the SI)

GROMACS has some tools that help setting up a relative free energy calculation. Analysis of the results depends on understanding how these utilities are used because they can

be a source of errors if misused. `couple-moltype` implicitly defines the initial and final states. It should be used in absolute free energy calculations to tag the molecule that will be decoupled from the rest of the system. `couple-moltype` should not be used in relative transformations in GROMACS. `couple-lambda0` and `couple-lambda1` define which of the **tagged** molecule's non-bonded parameters are present in the initial and in the final state, respectively. In absolute free energy calculations, one of them is usually set to turn on both van der Waals and electrostatic interactions, while the other, to turn off any interactions with the surroundings. In relative free energy calculations, however, they should be avoided. `couple-intramol` specifies if you want to modify intramolecular interactions along the alchemical path and should be set to `yes` in relative transformations.

Absolute hydration free energies are easily calculated with a good degree of precision, [refs] and it is reasonable to use it as a standard. With that in mind, we compared relative free energies calculated via simultaneous parameter change simulations – unified protocol – and separated parameter change simulations – split protocol– to relative free energy calculations obtained from two absolute hydration free energies – absolute protocol. The unified protocol changes partial charges, van der Waals parameters, and bond parameters simultaneously along the alchemical path, while the split protocol stages the transformation in a van der Waals parameter change followed by a simultaneous bonded terms and charges modifications, or vice-versa. In the framework of the split routine, windows without a reasonably strong counterbalancing Lennard-Jones component are subject to very large electrostatic forces that compromise the simulation. In this context, particle deletion processes require electrostatic terms to be turned off before van de Waals and bonded parameters. Insertion processes, in their turn, demand bonded and van der Waals parameters to be changed before charges.

Transformations having simultaneous particle insertion or deletion, such as 2-cyclopentanylindole $\rightarrow$ 7-cyclopentanylindole and its reverse, require an adapted split protocol. The simulation is divided into two stages: a deletion and an insertion stage. For the sake of optimizing computer resources use, the deletion step transforms a molecular branch into a dummy branch

Table 2: Relative hydration free energies obtained from GROMACS simulations in $kcal \cdot mol^{-1}$.

		Absolute [a]				Split [b]				Unified [c]			
		GROMACS (RF)		GROMACS (PME)		GROMACS (RF)		GROMACS (PME)		GROMACS (RF)		GROMACS (PME)	
Transformations		$\Delta\Delta G$	SEM	$\Delta\Delta G$	SEM	$\Delta\Delta G$	SEM	$\Delta\Delta G$	SEM	$\Delta\Delta G$	SEM	$\Delta\Delta G$	SEM
methane	ethane	0.059	0.012	0.036	0.009	-0.199	0.015	-0.183	0.007	-0.187	0.018	-0.165	0.011
ethane	methane	-0.059	0.012	-0.036	0.009	0.188	0.018	0.178	0.019	0.248	0.003	0.236	0.002
ethane	methanol	-5.829	0.012	-5.983	0.008	-6.046	0.008	-6.107	0.006	-7.038	0.005	-7.134	0.018
methanol	ethane	5.829	0.012	5.983	0.008	6.047	0.003	6.116	0.002	7.252	0.008	7.324	0.007
methane	methanol	-5.77	0.01	-5.947	0.009	-6.212	0.047	-6.243	0.047	-7.205	0.019	-7.289	0.019
methanol	methane	5.77	0.01	5.947	0.009	6.388	0.007	6.422	0.005	7.530	0.018	7.611	0.010
methane	2-methylfuran	-2.867	0.01	-2.95	0.01	-3.125	0.008	-3.080	0.013	-3.143	0.006	-3.102	0.011
2-methylfuran	methane	2.867	0.01	2.95	0.01	3.084	0.008	3.028	0.008	3.144	0.014	3.087	0.013
methane	toluene	-2.969	0.011	-3.159	0.011	-3.246	0.011	-3.204	0.012	-3.328	0.013	-3.310	0.010
toluene	methane	2.969	0.011	3.159	0.011	3.268	0.026	3.259	0.023	3.303	0.009	3.292	0.010
methane	2-methylindole	-8.438	0.017	-8.793	0.015	-8.789	0.031	-8.797	0.010	-8.670	0.027	-8.732	0.043
2-methylindole	methane	8.438	0.017	8.793	0.015	8.744	0.020	8.764	0.030	8.756	0.014	8.834	0.026
methane	neopentane[d]	0.181	0.012	0.135	0.009	-0.207	0.018	-0.154	0.047	-0.239	0.014	-0.076	0.011
neopentane	methane[d]	-0.181	0.012	-0.135	0.009	0.359	0.021	0.309	0.026	0.294	0.021	0.218	0.034
methane	neopentane[e]	0.181	0.012	0.135	0.009	-0.053	0.024	0.036	0.017	-0.025	0.010	0.036	0.017
neopentane2	methane[e]	-0.181	0.012	-0.135	0.009	-0.027	0.007	-0.043	0.012	-0.049	0.011	-0.069	0.012
2-cyclopentanylindole	7-cyclopentanylindole	-0.018	0.05	0.019	0.023	-0.067	0.022	-0.031	0.028	-0.094	0.052	-0.135	0.103
7-cyclopentanylindole	2-cyclopentanylindole	0.018	0.05	-0.019	0.023	0.119	0.057	0.204	0.043	0.117	0.077	0.220	0.102

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

while turns on the Lennard- Jones parameters of the new molecular piece in 31 windows. This modification allows a smaller insertion step of 16 windows.

Table 2 lists the relative free energies obtained from GROMACS simulations. Relative free energies are in good agreement with each other and with $\Delta\Delta G_{\mathrm{hydr}}$ obtained from the other software used in this study. A few noteworthy exceptions are the differences between the unified and split results of methane $\rightarrow$ methanol, ethane $\rightarrow$ methanol, and their reverse processes. In the former, $\langle \partial \mathrm{H}/\partial\lambda \rangle$ plots show a sharp decrease near $\lambda = 1.0$, while similar behavior is not observed in the latter (Figure XX1). Investigation of the phase space overlap plots (Figure XX2) obtained from `alchemical_analysis.py` show in the unified case a poor overlap between the last state and its neighbors, which can explain the difference in free energy.

Ethane $\rightarrow$ methane and methane $\rightarrow$ ethane split and unified results are quite different from their corresponding absolute results, and from other MD packages. Neopentane $\rightarrow$ methane transformations, and their reverse processes also disagree with the absolute protocol, but similar differences can be seen in AMBER.

There is no significant difference between choosing Particle Mesh Ewald or Reaction Field to calculate electrostatic interactions. One particularity of the software worth mentioning is that relative free energy simulations will blow up if a hydrogen alchemically becomes a heavy atom if the simulation employs hydrogen bond constraining algorithms such as SHAKE or LINCS. Successful simulations might involve turning off the constraint and decreasing the time step.

3.4 Sire/SOMD

Results for Sire relative free energy calculations are shown in (FIG OR TABLE TO BE DECIDED. Plot for MUD for each package?) good/bad agreement, final MUD = ?)

To achieve reproducibility the role of constraints was extremely influential. Fig. X shows the comparison between relative hydration free energies computed by using different con-

straint schemes during the simulation. When all bonds of the ligand are constraint a systematic offset with respect to the other constraints with a MUD = 0.38 kcal mol^{-1} was observed with respect to the no constraint case and MUD = 0.36 kcal mol^{-1} compared to the unperturbed H protocol (see section 2.3). Furthermore, analysing these simulations with Gromacs and Amber all bond constraints bring a MUD = 1.2 kcal mol^{-1}. The main problem, for all bond constraints in RAFE calculations in Sire, is the missing overlap between the potential energy functions between each λ–step. Further discrepancies appear when different mappings and all bonds are constraint as Fig. X shows.

Referring to Fig. X, transforming neopentane to methane (centrally mapped) shows a $\Delta\Delta G_{\text{hyd}}$ of (2.04 ± 0.01) kcal mol^{-1}, while for the backward transformation $\Delta\Delta G_{\text{hyd}} = (-2.10 \pm 0.01)$ kcal mol^{-1}, which highly overestimates the free energy compared to Gromacs and Amber (REFER TO TAB). Unexpectedly, simulation of neopentane to methane (terminally mapped) shows a $\Delta\Delta G_{\text{hyd}} = (0.59 \pm 0.01)$ kcal mol^{-1} and $\Delta\Delta G_{\text{hyd}} = (-0.48 \pm 0.01)$ kcal mol^{-1} for the reverse transformation. As the free energy must be independent of the chosen path the discrepancy highlights a problem with using constraints on transformed atoms. The solution adopted to achieve reproducibility in Sire is to rely on the unperturbed H protocol. These constraints allow a timestep of 2 fs with a MUD of (0.30 ± 0.01) kcal mol^{-1} with respect to Gromacs and Amber.

To test the quality of relative free energies of hydration, $\Delta\Delta G$ were estimated also with absolute free energy calculations. By computing the difference between absolute hydration of methane and all the other molecule, the relative free energy of hydration was computed. Fig. X demonstrates the reliability of the unperturbed H protocol through consistency between relative and absolute simulations. Furthermore, all the calculations can be performed in one step, differently (how?) from Amber and Gromacs. (EXPLAIN EXACTLY WHY)

4 Discussion

12. recommended protocols

13. protocols to avoid

14. lessons learned

15. 2-cyclopentanylindole to 7-cyclopentanylindole: better to go through intermediates?

16. what do we need to progress the field e.g. automation to make things easy but also consistent and thus more reproducible (FESetup also for reproducibility); GPU: Gromacs, SOMD but not AMBER (yet) and CHARMM; alternative softcore functions?; sampling?; force field improvements?; analysis?

17. developer notes: constraints, both appearing/disappearing; lambda paths for AMBER (relative), absolute: crgmask requires vacuum corr if separated protocol

18. further investigation required: binding RAFEs?

*ToDo

P.

Acknowledgement

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

1352608), and computing support from the UCI GreenPlanet cluster, supported in part by NSF Grant CHE-0840513.

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

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

References

(1) Hansen, N.; Van Gunsteren, W. F. *Journal of Chemical Theory and Computation* **2014**, *10*, 2632–2647.

(2) Pohorille, A.; Jarzynski, C.; Chipot, C. *The Journal of Physical Chemistry B* **2010**, *114*, 10235–10253, PMID: 20701361.

(3) Gallicchio, E.; Levy, R. M. In *Computational chemistry methods in structural biology*; Christov, C., Ed.; Advances in Protein Chemistry and Structural Biology; Academic Press, 2011; Vol. 85; pp 27 – 80.

(4) Lu, X.; Fang, D.; Ito, S.; Okamoto, Y.; Ovchinnikov, V.; Cui, Q. *Molecular Simulation* **2016**, *42*, 1056–1078, PMID: 27563170.

(5) Rickman, J. M.; LeSar, R. *Annual Review of Materials Research* **2002**, *32*, 195–217.

(6) Wan, S.; Knapp, B.; Wright, D. W.; Deane, C. M.; Coveney, P. V. *Journal of Chemical Theory and Computation* **2015**, *11*, 3346–3356, PMID: 26575768.

(7) Beveridge, D. L.; Dicapua, F. M. *Annual Review of Biophysics and Biophysical Chemistry* **1989**, *18*, 431–492.

(8) Straatsma, T. P.; Mccammon, J. A. *Annual Review of Physical Chemistry* **1992**, *43*, 407–435.

(9) Kollman, P. *Chemical Reviews* **1993**, *93*, 2395–2417.

(10) Squire, D. R.; Hoover, W. G. *The Journal of Chemical Physics* **1969**, *50*, 701–706.

(11) Bennett, C. H. *Journal of Computational Physics* **1976**, *22*, 245–268.

(12) Mruzik, M. R.; Abraham, F. F.; Schreiber, D. E.; Pound, G. M. *The Journal of Chemical Physics* **1976**, *64*, 481–491.

(13) Postma, J. P. M.; Berendsen, H. J. C.; Haak, J. R. *Faraday Symp. Chem. Soc.* **1982**, *17*, 55–67.

(14) Tembre, B. L.; Cammon, J. M. *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) Woods, C. J.; Malaisree, M.; Hannongbua, S.; Mulholland, A. J. *The Journal of Chemical Physics* **2011**, *134*, 054114.

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

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

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

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

(27) 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.

(28) 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.

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

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

(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) 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.

(34) 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.

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

(36) 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.

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

(38) 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.

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

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

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

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

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

(44) 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.

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

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

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

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

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

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

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

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

(53) 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.

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

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

(56) Bosisio, S.; Mey, A. S. J. S.; Michel, J. *Journal of Computer-Aided Molecular Design* **2016**, *30*, 1101–1114.

(57) Kirkwood, J. G. *Journal Of Chemical Physics* **1935**, *3*, 300–313.

(58) Zwanzig, R. *The Journal of Chemical Physics* **1954**, *22*, 1420–1426.

(59) Wu, D.; Kofke, D. A. *Journal of Chemical Physics* **2005**, *123*.

(60) Wu, D.; Kofke, D. A. *Journal of Chemical Physics* **2005**, *123*.

(61) Wu, D.; Kofke, D. A. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics* **2004**, *70*, 066702.

(62) Lu, N.; Singh, J. K.; Kofke, D. A. *The Journal of Chemical Physics* **2003**, *118*, 2977–2984.

(63) Shirts, M. R.; Chodera, J. D. *The Journal of Chemical Physics* **2008**, *129*.

(64) Shirts, M. R.; Chodera, J. D. *The Journal of Chemical Physics* **2008**, *129*, 124105.

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

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

Graphical TOC Entry