Robust, efficient and automated methods for accurate prediction of protein-ligand binding affinities in AMBER Drug Discovery Boost

Tai-Sung Lee, Hsu-Chun Tsai, Abir Ganguly, Timothy J. Giese, and Darrin M. York[*]

Laboratory for Biomolecular Simulation Research, Center for Integrative Proteomics Research and Department of Chemistry and Chemical Biology, Rutgers University, Piscataway, NJ 08854, USA

E-mail: Darrin.York@rutgers.edu

[*] To whom correspondence should be addressed

Abstract

Recent concurrent advances in methodology development, computer hardware and simulation software has transformed our ability to make practical, quantitative predictions of relative ligand binding affinities to guide rational drug design. In the past, these calculations have been hampered by the lack of affordable software with highly efficient implementations of state-of-the-art methods on specialized hardware such as graphical processing units, combined with the paucity of available workflows to streamline throughput for real-world industry applications. Herein we discuss recent methodology development, GPU-accelerated implementation, and workflow creation for alchemical free energy simulation methods in the AMBER Drug Discovery Boost (AMBER-DD Boost) package available as a patch to AMBER20. Among the methodological advances are 1) new methods for the treatment of softcore potentials that overcome long standing end-point catastrophe and softcore imbalance problems and enable single-step alchemical transformations between ligands, and 2) new adaptive enhanced sampling methods in the "alchemical" (or "λ") dimension to accelerate convergence and obtain high precision ligand binding affinity predictions, 3) robust network-wide analysis methods that include cycle closure and reference constraints and restraints, and 4) practical workflows that enable streamlined calculations on large datasets to be performed. Benchmark calculations on various systems demonstrate that these tools deliver an outstanding combination of accuracy and performance, resulting in reliable high-throughput binding affinity predictions at affordable cost.

1 Introduction

A major goal in drug discovery is to use fast computational methods to make predictions about the binding thermodynamics, and in some cases kinetics, of lead compounds to protein or nucleic acid drug targets in order to guide the process of lead refinement. Among the most rigorous approaches are so-called alchemical free energy simulations that enable the prediction of absolute and relative binding free energies (RBFEs).[1–6] Recent progress and improvements in computer hardware, simulation software, and free energy methods,[6–18] especially the development of highly efficient and cost-effective GPU-accelerated free energy calculations,[2,18–23] have significantly extended the accessible time scales of computer simulations and scope of applications. In addition to ongoing challenges of developing more accurate force fields and efficient sampling methods, there is need to improve our ability to optimally set up alchemical free energy calculations,[1,24–36] and perform robust analysis.[31,37–45]

Over the past several years we have made advances in addressing key barriers to progress in the field, and have implemented state-of-the-art methods for GPU-accelerated free energy simulations into AMBER20. Most recently, we have created the AMBER Drug Discovery Boost package that contains new features, methods, tools and workflows to greatly extend the capabilities and usability of the software at production scale. The most promising, validated innovations in the boost package will be integrated into a future AMBER official release (currently on a 2-year release cycle).

The remainder of the book chapter discusses these advances and presents illustrative examples, and it is organized as follows. The following section (section 2) provides a theoretical overview in implementation-level detail as it pertains to AMBER20 and the AMBER Drug Discovery Boost package. Emphasis is placed on strictly defining the terms and functional forms for the λ-dependent potentials, and how they relate to the user control flags in the software. These potentials form the foundation of the free energy simulation and enhanced sampling methods that follow. Section 3 describes the AMBER Drug Discovery Boost package, including a summary of its functionality in relation to AMBER20, its availability, as well as detailed subsections that describe new features, methods and tools, including:

- Pairwise smoothstep softcore potentials to construct stable and robust alchemical transformations
- AlChemical Enhanced Sampling (ACES) for efficient sampling in the λ-dimension
- BARnet and MBARnet methods for network-wide analysis of compound libraries with cycle closure and experimental reference constraints
- Streamlined workflows for performing and analyzing simulations of relative binding free energies

The chapter concludes with a perspective outlook for the future.

2 Theoretical Overview

2.1 Background

The change in free energy between two thermodynamic states can be computed from equilibrium simulations using a free energy perturbation (FEP)[46] or thermodynamic integration (TI)[47,48] formulation, or through non-equilibrium ensemble simulations using the Jarzynski equality and its equation variations.[49–56] In the current work, we focus on calculation of relative free energies from equilibrium simulations using TI and FEP formulations with Bennett Acceptance Ratio[57,58] (BAR) and its multistate generalization[39,59] (MBAR), that have recently been extended in their formulation to enable solution of network-wide analysis using a constrained variational approach (BARnet and MBARnet).[60]

In the thermodynamic integration (TI) free energy formulation,[47,48] a parametric pathway is introduced to connect thermodynamic states. In practice, use of BAR and MBAR methods also require use of a parametric pathway in order to ensure sufficient phase space overlap required to obtain reliable thermodynamic averages, and hence the discussion of thermodynamic pathways that follow is equally valid for these methods. Often the parameter in this path is designated by the variable λ and varies between 0 and 1. As the the end states are generally chemically distinct molecules with different compositions, the pathway is non-physical and sometimes referred to as an "alchemical" transformation. The free energy change, $\Delta A_{0\to 1}$, between states "0" and "1" can be achieved through integration of the thermodynamic derivative as:

$$\Delta A_{0\to 1} = \int_0^1 d\lambda \cdot \left(\frac{dA}{d\lambda} \right) = \int_0^1 d\lambda \cdot \left\langle \frac{\partial U(\mathbf{r}^N;\lambda)}{\partial \lambda} \right\rangle_\lambda \approx \sum_{k=1}^{M} w_k \cdot \left\langle \frac{\partial U(\mathbf{r}^N;\lambda)}{\partial \lambda} \right\rangle_{\lambda_k} \tag{1}$$

where the second integral involves the derivative of the potential energy U with respect to the parameter that smoothly connects the end states $\lambda = 0$ and $\lambda = 1$, and the sum indicates numerical integration over M quadrature points (λ_k, for $k = 1, \cdots M$) with associated weights w_k. While the free energy is a state function, and formally is invariant to the pathway connecting states, as will be demonstrated below, in practical simulations the thermodynamic averages in Eqn. 1 are extremely sensitive to the pathway.

Here, we present a generalized formulation of the λ-dependent transformation that occurs through both scaling of different potential energy components using λ-dependent weights, as well as introducing explicit non-linear λ-dependence into certain potential energy terms themselves. In order to describe the necessary details to facilitate clear, precise discussion, we introduce a flexible notation which pertains to classical additive force fields. These are summarized in Tables 1 and 2.

Table 1: Definition of potential energy terms and their abbreviations used as subscripts (lack of a term subscript implies all energy terms).

Abbreviation	Energy Term	Collective Terms	
rec	PME reciprocal space	Electrostatic(Ele) $U_{Ele} = U_{rec} + U_{dir} + U_{1-4Ele}$	
dir	PME direct/real space		Non-bonded (nb) $U_{nb} = U_{dir} + U_{1-4Ele} + U_{LJ} + U_{1-4LJ}$
1-4 Ele	1-4 Electrostatic		
LJ	van der Waals/Lennard-Jones		
1-4 LJ	1-4 Lennard-Jones		
bond	Bond stretch		Bonded (b) $U_b = U_{bond} + U_{ang} + U_{tor}$
ang	Angle bend		
tor	Torsion rotate (proper/improper)		

Herein we assume that no interaction energy term (e.g., U_{ang} or U_{tor}) would contain atoms that span all three TS, TC and I regions.

Table 2: Definition of regions (non-overlapping sets of atoms) and the notation used as superscripts to indicate internal potential energy within a region, and interaction energy between regions (lack of a superscript implies all regions, i.e., the entire system).

Notation	Region/Interactions	Description
TS	Transforming: Separable coordinate/softcore	Region that is transforming and is described by a dual topology with separate coordinates for each state. This region contains all atoms that will be transformed into "dummy" atoms using softcore potentials.
TC	Transforming: Constrained coordinate/common core	Region that is transforming and is described formally by a dual topology but with coordinates constrained to be the same. This region does not interact using softcore potentials except for interactions with the TS region.
I	Immutable	Region that is NOT transforming (immutable) and is described by a single topology and set of coordinates. This region does not interact using softcore potentials except for interactions with the TS region.
TS+TC+I	Combined TC and I region	Union of the sets of atoms in the TC and I regions.
$\mathbf{U}^{TS}$	Internal energy of TS region	Each of the contributing bonded or non-bonded internal energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that are contained within TS region; i.e., all atoms of the term belong to the TS region.
$\mathbf{U}^{TC}$	Internal energy of TC region	Each of the contributing bonded or non-bonded internal energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that are contained within TC region; i.e., all atoms of the term belong to the TC region.
$\mathbf{U}^{I}$	Internal energy of I region	Each of the contributing bonded or non-bonded internal energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that are contained within I region; i.e., all atoms of the term belong to the I region.
$\mathbf{U}^{TC+I}$	Internal energy of TC+I region	Each of the contributing bonded or non-bonded internal energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that are contained within the combined (TC+I) region; i.e., all atoms of the term belong to the TC+I region.
$\mathbf{U}^{TS/TC}$	Interaction energy between TS and TC regions	Each of the contributing bonded or non-bonded interaction energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that span the TS and TC regions; i.e., some belong to the TS region, while others in the same term belong to the TC region.
$\mathbf{U}^{TS/I}$	Interaction energy between TS and I regions	Each of the contributing bonded or non-bonded interaction energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that span the TS and I regions; i.e., some belong to the TS region, while others in the same term belong to the I region.
$\mathbf{U}^{TS/TC+I}$	Interaction energy between TS and (TC+I) regions	Each of the contributing bonded or non-bonded interaction energy terms (bond, angle, torsion, LJ, dir, etc) arises from a set of atoms that span the TS and combined (TC+I) regions; i.e., some belong to the TS region, while others in the same term belong to the (TC+I) region. Note: $\mathrm{U}^{TS/(TC+I)} = \mathrm{U}^{TS/TC} + \mathrm{U}^{TS/I}$

Specifically, we introduce a system whereby we use subscripts to indicate the state ("0" or "1") and (optionally) the specific term in the potential energy, and superscripts to indicate the specific atoms involved in the interaction. The state of the system and specific energy terms follow the general form $U^{X}_{\{0/1\},abbr}$ or $U^{X/Y}_{\{0/1\},abbr}$ where the state is indicated as either 0 or 1, the energy term is designated an appropriate abbreviation (abbr) as indicated in Table 1, and the superscript "X" indicated an internal potential energy for region "X" and "X/Y" indicated the interaction energy between regions "X" and "Y" as indicated in Table 2.

The two 1-4 terms in Table 1 involve how the Lennard-Jones (LJ) and electrostatic (Ele) terms are treated between the atoms in the 1 and 4 position of a torsion angle (where there is a chemical bond between 1-2, 2-3 and 3-4 atoms, and the torsion involves rotation about the 2-3 bond). These interactions are frequently scaled by a fixed constant, in which case the 1-4 LJ and 1-4 Ele are the compensating corrections that need to be made to the unscaled interactions in order to achieve the desired scaling. Hence in the present context, we will consider the 1-4 LJ and 1-4 Ele terms as separate from the total unscaled LJ and Ele terms. In the above, the total electrostatic energy for state 0 is the sum of the PME reciprocal and direct space terms ($U_{0,Ele} = U_{0,rec} + U_{0,dir} + U_{0,1-4Ele}$). In some cases, we will refer collectively to "bonded" (b) and "non-bonded" (nb) terms. The bonded terms are collectively the bond, angle, torsion($U_{0,b} = U_{0,bond} + U_{0,ang} + U_{0,tor}$). The non-bonded terms are collectively the PME direct space and LJ terms, including 1-4 Ele and 1-4 LJ ($U_{0,nb} = U_{0,LJ} + U_{0,dir} + U_{0,1-4LJ} + U_{0,1-4Ele}$). Note: the PME reciprocal space term is a separate term not part of the non-bonded terms. If the second energy term subscript is dropped, this implies summation over all relevant energy terms for given state (indicated by the first subscript).

In addition, we indicate the atoms involved in the evaluation of each of the energy terms as superscripts. We first introduce notation that distinguishes different regions of the system. The two main subdivisions: one region is alchemical transforming (T), whereas the rest of the surrounding environment is immutable (I), i.e., not transforming. In the AMBER20 implementation, a hybrid dual-topology[61] approach is utilized. The immutable part is represented by a single "topology" and set of coordinates. The transforming part of the system is represented by a formal dual topology and set of coordinates. The transforming region is defined by keywords "timask1" and "timask2". In order to facilitate phase space overlap between states during the alchemical transformation, it may be desirable that certain atoms of the transforming region are constrained to have the same coordinates (e.g., if two drug molecules share a common core of atoms, and differ only by certain attached substituents). Other atoms in the transforming region cannot be easily mapped between states and have separable coordinates that can adopt different conformations that do not directly interact with one another. Often this separable dual-coordinate approach requires the introduction of explicit non-linear λ-dependent terms to "soften" the interaction of these atoms with their surroundings. These are most often employed for non-bonded interactions such as LJ and Ele (or in the case of PME electrostatics, often just the dir term), but other forms have also been developed for bonds and other terms in the potential.[62–64] These modified λ-dependent softening potentials are referred to as "softcore potentials", and in AMBER20 the atoms involved with these potentials are defined by the keywords "scmask1" and "scmask2". In our notation, we will subdivide the transforming region (T) into the constrained coordinate/common core (TC) and the separable-coordinate/softcore (TS) regions.

Thus the system can be divided into regions I (immutable), TC (transforming constrained) and TS (transforming separable) regions. The I region has the same atomic coordinates, parameters and internal potential energy for both states 0 and 1. The TC region can have different parameters between states 0 and 1, but the coordinates of mapped atoms are constrained to be the same. The TS region also can have different parameters between states 0 and 1, but unlike the TC region each state has its own separable set of atomic coordinates.

For notational purposes, we designate combined regions using a "+" symbol, such as "TC+I" or "TS+TC", and with this convention the complete system is "TS+TC+I". In our notation, we refer to internal potential energy U^X that occur between atoms contained within the region "X" by using a superscript indicating the region (e.g., U^{TS} and U^{TC+I} represent the internal potential energy of the TS and $TC + I$ regions, respectively). We refer to potential energy interactions $U^{X/Y}$ that occur between regions X and Y using a superscript indicating the two regions separated by a "/" (e.g., $U^{TS/TC+I}$ would represent all potential interactions between the TS and $TC + I$ regions). It should be clarified that for the internal potential energy U^X of a region X involves all energy terms where all atoms involved in each term (e.g., the four atoms involved in a torsion angle potential energy) are contained within the region X. If atoms involved in a potential energy term occupy two regions X and Y, then this term will be contained in the interaction energy $U^{X/Y}$. We assume that there are no energy terms that span more than two regions (e.g., there are no 3-body angle bending or 4-body torsion angle rotation terms that involve atoms of the TS, TC and I regions within the same term). If the superscript is omitted, this implies the region considered is the entire system, i.e., $TS + TC + I$ (e.g., U_0 is the total system potential energy in state 0).

Before moving on, we should make some important clarifications that will serve to set the stage for discussion of alchemical enhanced sampling methods later on. In the case of alchemical free energy simulations, the goal is to transform state 0 into state 1 while stably computing the free energy change in small steps, designated "λ windows", each of which requires a separate simulation. This requires reasonable phase space overlap between the end states of each window, and this overlap is very sensitive to the choice of the TC and TS regions, and the use (and form) of softcore potentials. The most common use case in free energy simulations is that phase space overlap is usually facilitated by judicious choice of the TC and TS regions, where it is generally assumed that the TS region having seperable coordinates for the two states will also be the one that will require use of λ-dependent softcore potentials in the transformation. In fact, AMBER20 has this assumption hardwired into the code: the "scmask1" and "scmask2" will flag the code to make separable coordinates for these sets of atoms, and also treat their non-bonded interactions with softcore potentials.

We are now ready to introduce the general equation for the λ-dependent potential energy. We first formally create a linear algebraic vector notation that uses a superindex, designated i in the summations below, to combine all possible elemental energy terms and internal/interaction regions. Specifically, summation over the superindex i implies summation over each individual type of energy term, indexed by the subscript abbreviation (abbr), and further subdivided into internal energy contributions of each region (TS, TC and I) and interactions between regions (TS/TC, TS/I and TC/I). Further, we introduce a λ-dependent weighting function $W_{0/1}(\lambda)$ that is (super)indexed in the same way as the potential energy. The λ-dependent total potential energy $U(\mathbf{r}^N;\lambda)$ can thus be written as

$$U(\mathbf{r}^N;\lambda) = \sum_i W_{0,i}(\lambda)\cdot U_{0,i}(\mathbf{r}^N;\lambda) + W_{1,i}(\lambda)\cdot U_{1,i}(\mathbf{r}^N;\lambda) \tag{2}$$

where the individual state energies, $U_0(\mathbf{r}^N;\lambda)$ and $U_1(\mathbf{r}^N;\lambda)$ are given in terms of their energy components as

$$\begin{aligned} U_0(\mathbf{r}^N;\lambda) &= \sum_i U_{0,i}(\mathbf{r}^N;\lambda) \\ &= U_{0,rec}(\mathbf{r}^N;\lambda) \\ &+ \sum_{abbr\neq rec} \Big\{ U^{TS}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TC}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{I}_{0,abbr}(\mathbf{r}^N;\lambda) \\ &\quad + U^{TS/TC}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TS/I}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TC/I}_{0,abbr}(\mathbf{r}^N;\lambda) \Big\} \end{aligned} \tag{3}$$

and similarly for $U_{1,i}(\mathbf{r}^N;\lambda)$. Note that the PME-reciprocal terms (rec) cannot be easily broken into contributions from different regions. The λ-dependence of the total potential energy $U(\mathbf{r}^N;\lambda)$ arises from two sources: the weights $W_{0,i}(\lambda)$ and $W_{1,i}(\lambda)$ that scale the individual state energy components $U_{0,i}(\mathbf{r}^N;\lambda)$ and $U_{1,i}(\mathbf{r}^N;\lambda)$, respectively, and the individual state energy components themselves. The general equations above indicate an explicit λ-dependence for each individual state energy component $U_{0,i}(\mathbf{r}^N;\lambda)$ and $U_{1,i}(\mathbf{r}^N;\lambda)$ that implies, for example, the use of a softcore potential. Other types of non-linear λ-dependence can also be introduced into the potentials, for example, through transformation of the force field parameters themselves. This type of approach is referred to as "parameter interpolation thermodynamic integration", or PI-TI,[21,65] and has distinct advantages for certain types of transformations. In practice, many or even most of the individual state energy components do *not* use softcore potentials or have other explicit dependence on λ. The specific λ-dependent energy components can vary depending on user selection and code implementation. For example, in the current implementation of AMBER20, only the non-bonded (LJ, dir, 1-4 Ele, and 1-4 LJ) terms that involve atoms in the TS region and their interactions with other regions will employ softcore potentials and thus have an explicit λ dependence. While in alchemical free energy simulations the TC region is also typically transforming, the transformation can be brought about by scaling by the weights $W_{0,i}(\lambda)$ and $W_{1,i}(\lambda)$ alone.

In the current AMBER implementation, the individual PME reciprocal terms are calculating with the charges of the corresponding end states and hence are not λ-dependent. Using this convention, one needs to perform the PME reciprocal calculations twice for each time step and the λ-dependency of total PME reciprocal contribution is solely due to the λ-dependent weights. Hence we will remove the indicated λ-dependence of the PME reciprocal terms in subsequent equations. An alternative approach is to apply the PI-TI method[21] and perform the PME reciprocal calculation only once with a set of charges defined by the λ-dependent combination of two end states, which will reduce the computation time but will require evaluating the derivative of the PME reciprocal term with respect to λ. The PI-TI type PME reciprocal calculation will be implemented in a future release of AMBER Drug Discovery Boost package.

2.2 Control (On/Off) of the weight functions

Eq. 3 can be rearranged as follows

$$\begin{aligned}
U_0(\mathbf{r}^N;\lambda) &= U_{0,rec}(\mathbf{r}^N) + \\
&\quad \sum_{abbr \neq rec} \Big\{ U^{TS}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TC}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{I}_{0,abbr}(\mathbf{r}^N;\lambda) \\
&\quad + U^{TS/TC}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TS/I}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TC/I}_{0,abbr}(\mathbf{r}^N;\lambda) \Big\} \\
&= U_{0,rec}(\mathbf{r}^N) + \sum_{abbr \neq rec} U^{TC+I}_{0,abbr}(\mathbf{r}^N;\lambda) \\
&\quad + \sum_{abbr \neq rec} \Big\{ U^{TS/(TC+I)}_{0,abbr}(\mathbf{r}^N;\lambda) + U^{TS}_{0,abbr}(\mathbf{r}^N;\lambda) \Big\}
\end{aligned} \tag{4}$$

The weight functions need to be defined before merging Eq. 4 to Eq. 2 to get the total potential energy. In this section, we describe flags that control the "On" or "Off" behavior of the weight functions that are associated with the TS region for which atoms will transform into a "dummy" state where their interactions with the (TC+I) regions are turned off, but there may remain flexibility to control the internal potential within the TS region. So long as the internal reference potential energy of the dummy state is treated consistently in different legs of the free energy cycle, the will, in principle, cancel. However, in practice, the specific energy terms that are left unscaled by the weight functions (i.e., remain turned "On" in the dummy state) can create kinetic traps that lead to differences in free energy due to incomplete sampling within only a single local free energy basin. Ideally, one should chose to keep internal potential energy terms that help to limit the conformational space needed to be sampled, but at the same time do not create multiple deep local minima that act as kinetic traps and are difficult to sample and sensitive to initial conditions. The AMBER Drug Discovery Boost package offers flexibility to control these terms through the gti_add_sc flag, the options for which are summarized in Table 3.

With the control flags in Table 3 set, the total λ-dependent potential energy can be written

$$\begin{aligned}
U(\mathbf{r}^N;\lambda) &= W_{0,rec}(\lambda)U_{0,rec}(\mathbf{r}^N) + W_{1,rec}(\lambda)U_{1,rec}(\mathbf{r}^N) \\
&+ \sum_{abbr\neq rec} W_{0,abbr}(\lambda)U^{TC+I}_{0,abbr}(\mathbf{r}^N;\lambda) + \sum_{abbr\neq rec} W_{1,abbr}(\lambda)U^{TC+I}_{1,abbr}(\mathbf{r}^N;\lambda) \\
&+ \sum_{abbr\neq rec} \left\{ W^{TS/(TC+I)}_{0,abbr}(\lambda)U^{TS/(TC+I)}_{0,abbr}(\mathbf{r}^N;\lambda) + W^{TS}_{0,abbr}(\lambda)U^{TS}_{0,abbr}(\mathbf{r}^N;\lambda) \right\} \\
&+ \sum_{abbr\neq rec} \left\{ W^{TS/(TC+I)}_{1,abbr}(\lambda)U^{TS/(TC+I)}_{1,abbr}(\mathbf{r}^N;\lambda) + W^{TS}_{1,abbr}(\lambda)U^{TS}_{1,abbr}(\mathbf{r}^N;\lambda) \right\} \quad (5)
\end{aligned}$$

where $W_{\{0/1\},abbr}(\lambda)$ is the general weight function for the *abbr* energy term and the weight functions associated with the TS regions are defined as follows. When considering the interactions between TS region and the surrounding TC+I region:

$$W^{TS/(TC+I)}_{\{0/1\},abbr}(\lambda) = \begin{cases} W_{\{0/1\},abbr}(\lambda), & \text{if scaled with } \lambda, \\ 1, & \text{if not scaled with } \lambda; \end{cases} \quad (6)$$

while considering the interactions within TS region

$$W^{TS}_{\{0/1\},abbr}(\lambda) = \begin{cases} W_{\{0/1\},abbr}(\lambda), & \text{if scaled with } \lambda, \\ 1, & \text{if not scaled with } \lambda. \end{cases} \quad (7)$$

These equations lead to the implementation of the "gti_add_sc" input flag, which controls the behaviors of $W^{TS/(TC+I)}_{\{0/1\},abbr}(\lambda)$ and $W^{TS}_{\{0/1\},abbr}(\lambda)$ for non-bounded terms as summarized in Table 3. Mentioned in the legend, but not explicitly indicated in Table 3 is that the bonded terms between the TS and (TC+I) regions must obey certain constraints and conditions in order that the ensembles generated in the state that contains “dummy atoms” reproduce the same potential of mean force on the real atoms as the real system without the dummy atoms.[61,66–69] Further, all non-bonded interactions, including 1-4 terms, between the

TS and (TC+I) regions should be scaled. The pre-AMBER20 behavior (gti_add_sc=0) is not theoretically correct[70] and has been fixed, although the option has been kept to enable testing and comparisons with earlier versions but should not be used for production work.

2.3 Functional forms of the weight functions

We now describe a general form for the weight functions $W(\lambda)$, where we only retain the 0 and 1 subscript to indicate the state in Eqn. 3. Toward this we introduce the family of smoothstep functions of orders P ($P = 0, 1, 2, \cdots$) defined as the polynomial functions (up to $P = 4$ shown):

$$\begin{aligned}
&\text{for } 0 \leq x \leq 1: \\
&\qquad S_0(x) = x, \\
&\qquad S_1(x) = -2x^3 + 3x^2, \\
&\qquad S_2(x) = 6x^5 - 15x^4 + 10x^3, \\
&\qquad S_3(x) = -20x^7 + 70x^6 - 84x^5 + 35x^4, \\
&\qquad S_4(x) = 70x^9 - 315x^8 + 540x^7 - 420x^6 + 126x^5, \\
&\qquad\quad \text{and} \\
&\qquad S_P(x \leq 0) = 0; S_P(x \geq 1) = 1, \forall\ P \in \mathbb{N}
\end{aligned} \tag{8}$$

The smoothstep functions are monotonically increasing functions that have desirable 0 and 1 endpoint values and vanishing endpoint derivative properties:

$$\left[\frac{d^k S_P(x)}{dx^k}\right]_{x=0} = \left[\frac{d^k S_P(x)}{dx^k}\right]_{x=1} = 0 \ \forall \ k \in \mathbb{N}, \ 0 < k \leq \mathrm{P} \tag{9}$$

In addition, the smoothstep functions obey the symmetry condition

$$S_P(1 - x) = 1 - S_p(x) \tag{10}$$

Table 3: The scaling behavior/λ-dependence of the weight functions in Eqns. 5, 6 and 7, controlled by the gti_add_sc flag, for different energy terms and regions/interactions in AMBER20.

Weight	**Energy Term**	**Region /**	**AMBER20 gti_add_sc flag**						
Symbol	**Abbreviation**	**Interaction**	0	1	2	3	4	5	6
$W^{TS/(TC+I)}_{\{0/1\},dir}$	dir	TS/(TC+I)	S	S	S	S	S	S	S
$W^{TS/(TC+I)}_{\{0/1\},1-4Ele}$	1-4 Ele	TS/(TC+I)	P	S	S	S	S	S	S
$W^{TS/(TC+I)}_{\{0/1\},LJ}$	LJ	TS/(TC+I)	S	S	S	S	S	S	S
$W^{TS/(TC+I)}_{\{0/1\},1-4LJ}$	1-4 LJ	TS/(TC+I)	P	S	S	S	S	S	S
$W^{TS}_{\{0/1\},dir}$	dir	TS	P	P	S	S	P	S	S
$W^{TS}_{\{0/1\},1-4Ele}$	1-4 Ele	TS	P	P	S	S	S	S	S
$W^{TS}_{\{0/1\},LJ}$	LJ	TS	P	P	P	S	P	P	S
$W^{TS}_{\{0/1\},1-4LJ}$	1-4 LJ	TS	P	P	P	S	S	S	S
$W^{TS}_{\{0/1\},dir}$	tor	TS	P	P	P	P	S	S	S
$W^{TC+I}_{\{0/1\}}$	all	TC+I	S	S	S	S	S	S	S
$W_{\{0/1\},rec}$	rec	all	S	S	S	S	S	S	S

Energy terms are defined in Table 1, and different regions/interactions are defined in Table 2.
Flags:

S: Scaled with λ (weight in Eq. 6 and 7 set to the λ-dependent weight function described in subsection 2.4) and corresponding energy term is NOT present in the dummy state.

P: Not scaled with λ (weight in Eq. 6 and 7 set to 1) and corresponding energy term IS present in the dummy state.

It should be noted that for theoretically correct treatment of the system where atoms of the TS region has been transformed to a "dummy" state, all non-bonded interactions (including 1-4 terms) between the TS and (TC+I) regions should be scaled. In addition, care should be taken that the bonded terms between the TS and (TC+I) regions obey certain constraints and conditions in order that the ensembles generated in the state that contains "dummy atoms" reproduce the same potential of mean force on the real atoms as the real system without the dummy atoms. A discussion of the energy term requirements that satisfy these conditions has been made by Boresch and Karplus[66,67] and Roux and co-workers.[61,69] These conditions are present in AMBER20, with the exception of the gti_add_sc flag value of 0 which (incorrectly) does not scale the 1-4 terms across the TS/(TC+I) boundary. This flag was created in order to have compatability with earlier versions of AMBER, and was corrected in a recent validation paper.[70]

A smoothstep function with a higher order will have a smoother function curve and smaller derivatives near 0 and 1 but a larger derivative in between. The zero-order ($P = 0$) smoothstep function is in fact simply linear with constant slope, including at the endpoints, which can lead to endpoint catastrophe problems. As illustrated in previous work,[71] the second order smoothstep function ($P = 2$) overall offers a good balance between smooth vanishing derivatives at the end points, and modest derivatives for intermediate values of λ. AMBER20 offers the flexibility to choose different smoothstep functions through the λ-scheduling mechanism described below.

2.4 λ-dependent weight functions for scaling potential energy components

The weight functions are defined in term of the smoothstep functions as

$$W_0(\lambda) = 1 - S_P(\lambda) = S_P(1-\lambda) \tag{11}$$

$$W_1(\lambda) = S_P(\lambda) \tag{12}$$

Note: in the above general equation, we drop the explicit superscripts and subscripts that can be controlled by different flags available to the user in AMBER20. Previous work has illustrated that use of smoothstep functions of order greater than 0 (i.e., a weight function that goes beyond the simple linear λ-dependence and has vanishing drivatives at the endpoints), affords improvement of the the transformation pathway, particularly at the end-points where large variation in $< \partial U/\partial\lambda >_\lambda$ can occur.[71] Note: these weight functions both operate within the range $0 \leq \lambda \leq 1$ (they have constant endpoint values outside of this range), and satisfy the normalization condition:

$$W_0(\lambda) + W_1(\lambda) = 1 \tag{13}$$

and the symmetry condition:

$$W_0(1-\lambda) = W_1(\lambda) \tag{14}$$

λ-scheduling of weight functions. In some cases, it is desirable to have the flexibility to apply more complicated λ schedules that operate over a subinterval of λ values between 0 and 1. The generalized λ scheduling weight for W_0 can be defined so that it is changing only within the interval $\lambda_{min} \leq \lambda \leq \lambda_{max}$ as

$$\begin{aligned} W_0(\lambda) &= 1 - S_P(z(\lambda)) \\ z(\lambda) & \begin{cases} = 0, & \text{if } \lambda \leq \lambda_{min} \\ = \frac{\lambda-\lambda_{min}}{\lambda_{max}-\lambda_{min}}, & \text{if } \lambda_{min} \leq \lambda \leq \lambda_{max} \\ = 1, & \text{if } \lambda_{max} \leq \lambda \end{cases} \end{aligned} \tag{15}$$

where $0 \leq \lambda_{min} \leq \lambda_{max} \leq 1$. Most generally, the complementary weight function $W_1(\lambda)$ can be selected to either satisfy the normalization condition (Eqn.13), or the symmetry conditon (Eqn.14) above. Only if the interval $z(\lambda)$ is centered at $\lambda = 0.5$ are both the normalization and symmetry conditions simultaneously satisfied. AMBER20 allows flexible λ scheduling of this form for different energy components. The lambda-scheduling can be enabled by setting the input control gti_lam_sch=1 and the scheduling is defined in the lambda-scheduling control file (the file name has default value of “lambda.sch” and can be specified by the command line argument -lambda_sch), which contains control lines for each type of interactions in the following format:

LambdaType, FunctionType, Matchtype, parameter1, parameter2

Each line controls the lambda-scheduling of the corresponding type of interaction. When a type is not present, the default behavior will be utilized (smooth_step2, complementary, 0.0, 1.0). The entries of each line are

- *LambdaType:* possible values: TypeGen (for all general usage), TypeBAT (for bonded terms), TypeRestBA (for restraint bond/angle terms), TypeEleRec (for reciprocal space terms), TypeEleCC (for direct space terms of TC region atoms), TypeEleSC (for direct space terms of TS region atoms), TypeVDW (for vdw terms)

- *FunctionType:* possible values: linear, smooth_step0 (the same as linear), smooth_step1, smooth_step2, smooth_step3, smooth_step4

- *Matchtype:* possible values: symmetric: the TI region 2 will have exactly the same lambda scheduling as TI region 1 (in the reversed direction) complementary: the TI region 2 will have the lambda scheduling so that $W_0(\lambda) + W_1(\lambda) = 1$.

- *Parameter1 and parameter2* real numbers between 0.0 and 1.0. *Parameter1* is λ_{min} and *Parameter2* is λ_{max} in Eq. 15.

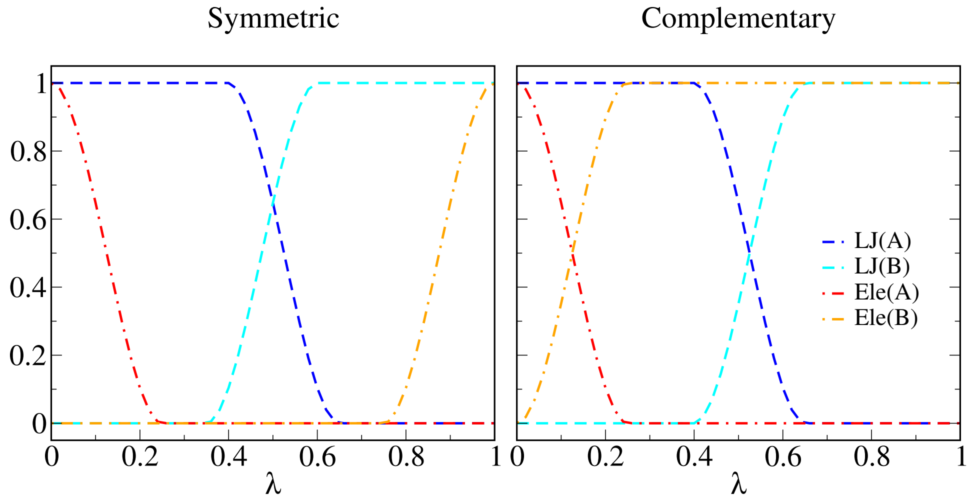

Figure 1: Illustrative plots of λ-scheduling: a transformation from State A to State B can be done with different λ-scheduling schemes.
Left panel: the weight of the Ele interaction of State A begins to decrease at $\lambda = 0$ and becomes zero when $\lambda \geq 0.25$ while the weight of the Ele interaction of State B begins to increase from 0 at $\lambda = 0.75$ and becomes 1 when $\lambda = 1$. The weight of the LJ iteration of State A begins to decrease at $\lambda = .35$ and becomes zero when $\lambda \geq 0.65$ while the weight of the LJ interaction of State B begins to increase from 0 at $\lambda = 0.35$ and becomes 1 when $\lambda \geq 0.65$. The Ele and LJ weights of State A are symmetric to State B about $\lambda = 0.5$.
Right panel: the weight of the Ele interaction of State A is the same as the one shown in the Left panel but the weight of the Ele interaction of State B begins to increase from 0 at $\lambda = 0.0$ and becomes 1 when $\lambda = 0.25$, i.e. the sum of the weights from State A and B is always 1.0 (complementary). The weights of the LJ interactions of State A and B are the same as shown in Left Panel.

Fig. 1 demonstrates a possible usage of λ-scheduling scheme to mimic stepwise protocols in the "symmetric" and "complementary" modes. Such flexibility of defining on/off timing of individual interactions might be useful in certain types of applications. Updated examples are available on the GitLab repository described below.

In addition, this general framework and the use of smoothstep functions will be extended below to include new softcore potentials for robust estimation of alchemical transformation pathways, in addition to alchemical enhanced sampling methods (ACES).

3 AMBER Drug Discovery Boost Package

During past years, we have devoted effort to develop our AMBER Drug Discovery Boost package consisting of newly-developed methods and extension of existing functionalities in the latest AMBER release. The main purpose of the boost package is to enable broad testing and validation of new methods for drug discovery to prepare for deployment in a future AMBER release. The package consists of new features and methods that are built on top of the latest AMBER20 as a code patch, in addition to a set of stand alone tools for data analysis and streamlined workflows. Table 4 summarizes the current status of major functionalities of the AMBER Drug Discovery Boost package.

Table 4: AMBER Drug Discovery Boost (AMBER-DD) new functionalities and bug fixes and their relation to the current AMBER20.

Category	description	input/enable	Status
REMD	Odd number REMD		A20
REMD	Targeted Volume	NPT=4	DD
Output	Detailed TI component output	gti_output	A20
Fix	Fix inconsistency of SC cutoff	gti_cut	A20
Fix	Charge neutralization	gti_chg_keep	A20
Fix	Fix cross NB problem	gti_add_sc (default 0 to 1)	A20
New Feature	User-control intra-SC NB terms	gti_add_sc	A20
New Feature	User-control cross NB terms	gti_add_sc	A20
New Feature	SSC(P)	different SC functional form	A20
New Feature	Lambda scheduling	different SC timing	A20
New Feature	cross bonded term correction	gti_bat_sc	A20
New Feature	User-control cross bonded terms	gti_bat_sc	A20
New Feature	12-6-4 potential MD/GPU		A20
New Feature	12-6-4 potential TI/GPU		A20
New Feature	Different m and n for SC		DD
New Feature	scaled NMR restraint for TS regions		A20
New Feature	ACES enhanced sampling	gti_add_sc/REMD	DD
New Feature	Self-djusted mixture sampling (SAMS)		DDx
New Feature	REST-type enhanced sampling		DDx
New Feature	BARnet		DDi
New Feature	Workflow		DDs

A20: already in AMBER20 pmemd.cuda
DD: only in AMBER-DD pmemd.cuda
DDx: only in AMBER-DD pmemd.cuda, under α-phase tests
DDi: only in AMBER-DD, as a set of separate programs
DDs: only in AMBER-DD, as a set of scripts

3.1 Availability of the AMBER-DD boost package

At the moment, the primary mode for accessing the AMBER Drug Discovery Boost package is through GitLab repository set up through the laboratory for Biomolecular Simulation Research (LBSR) at Rutgers. In order to checkout the GitLab repo, a user will need a GitLab account. If the user does not already have one, a new account can easily be created at www.GitLab.com. Then, the user will need to send the e-mail address/username associated with the GitLab account to: Abir Ganguly <abir.ganguly@rutgers.edu> or Darrin York

<Darrin.York@rutgers.edu> in order that we add the user to the GitLab projects. *Note*: if you have created a new GitLab account through a social media account such as Google or Facebook, you will need to manually set up your GitLab password in order for git clone to work. Once added, the user will receive three separate notification emails confirming that the user has been added to the following three projects:

Laboratory for Biomolecular Simulation Research / AMBER Drug Discovery Boost
Laboratory for Biomolecular Simulation Research / Alchemical_FE
Laboratory for Biomolecular Simulation Research / FE-ToolKit

Upon this confirmation the user will be able to check out the packages as follows:

With ssh-key setup in GitLab (recommended):

```
git clone git@gitlab.com:RutgersLBSR/amber-drug-discovery-boost.git
git clone git@gitlab.com:RutgersLBSR/Alchemical_fe.git
git clone git@gitlab.com:RutgersLBSR/FE-ToolKit.git
```

Without ssh-key setup in GitLab:

```
git clone https://gitlab.com/RutgersLBSR/amber-drug-discovery-boost.git
git clone https://gitlab.com/RutgersLBSR/Alchemical_fe.git
git clone https://gitlab.com/RutgersLBSR/FE-ToolKit.git
```

The Alchemical_fe folder contains the following sub-folders:

- *Documentation* - containing documentation specific to AMBER Drug Discovery Boost
- *Tutorials* - containing tutorials for setting up alchemical free energy simulations using AMBER Drug Discovery Boost
- *Examples* - containing test cases for relative binding free energy (RBFE) and relative solvation free energy (RSFE) calculations

- *bin* - containing scripts to help set up alchemical free energy simulations using Amber Drug Discovery Boost

We are also in the process of putting documentation up on the Wiki site that will be updated on a regular basis - https://gitlab.com/RutgersLBSR/alchemical_fe/-/wikis/Setup-AFE_AMBER_DD_BOOST

In the following sections, we will describe current major new development and additions of the AMBER Drug Discovery Boost package. Computational details for the examples illustrated are collected in an appendix at the end of the chapter.

3.2 New feature: pairwise smoothstep softcore potentials to construct stable and robust alchemical transformations

One of the key requirements for conducting stable alchemical free energy simulations is to construct a suitable pathway that connects thermodynamic states. This pathway is typically sampled in discreet windows (or in some cases continuously with methods such as λ-dynamics), and should have well-behaved thermodynamic derivatives that can be easily integrated with TI, or similarly, exhibit favorable phase space overlap to obtain reliable BAR or MBAR estimates. One of the mechanisms to facilitate these properties is to use so-called "softcore" potentials along with λ-dependent weighting functions to create a λ-dependent potential energy (Eqn. 2) and set of thermodynamic derivatives. We have recently made progress in developing new softcore potentials based on the smoothstep functions described above[22] in order to overcome three major problems that commonly occur in alchemical simulations, particularly for "concerted transformation" that involve simultaneous changes in both nonbonded Lennard-Jones (LJ) and Coulombic electrostatic (Coul) interactions. These are refereed as the "endpoint catastrophe", the "particle collapse", and the "large gradient-jump" problems.

Endpoint Catastrophe. One of the major obstacles in concerted transformations is the "endpoint catastrophe" due to poor phase space overlap. This problem often occurs with linear alchemical transformations where the total potential is defined as the linear combinations of the end state potentials and involved the system with introducing or removing van der Waals centers which can result in unphysical atom-atom overlap.[72,73] With linear alchemical transformations, the endpoint catastrophe is the sharp divergence of the free energy difference and is prone to occur at the thermodynamic endpoints (λ becomes close to 0 or 1). To avoid the endpoint catastrophe, the introduction of softcore potentials for LJ and Coul interactions are commonly used.[62,64]

Particle Collapse. While the use of softcore potentials can solve the endpoint catastrophe, they can lead to large amplitude fluctuations or phase transition behavior along the λ dimension and result in the particle collapse problem.[74] It involves the introduction of new artificial minima at intermediate λ states and results from an imbalance of Coulomb attraction and exchange repulsion.[64] The particle collapse problem can be overcome by ensuring that these terms are scaled in such a way that preserves overall repulsive behavior at short distances for all λ values.

Large Gradient-Jump. With softcore potentials, large gradient jump can result from sensitivity of the thermodynamic derivatives to certain softcore parameter values that adjust the exchange repulsion. This issue often happens when large β values are required to adjust the softcore parameter ratio to solve the Coulomb-exchange imbalance problem. To solve the above problems, the production of the smoothly varying alchemical transformation pathway is required. A family of smooth softcore potentials that use smoothstep weighting functions with favorable endpoint derivative properties and proper choices of softcore parameters can be utilized for efficient concerted transformations.[71]

3.2.1 Alchemical Transformation Pathway and Softcore Potentials

The LJ and Coul interactions for a set of interacting point particles i and j separated by a distance r_{ij} are given by

$$U_{\mathrm{LJ}}(r_{ij}) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \tag{16}$$

and

$$U_{\mathrm{Coul}}(r_{ij}) = \left(\frac{q_i q_j}{4\pi\epsilon_0}\right)\frac{1}{r_{ij}} \tag{17}$$

where σ_{ij} and ϵ_{ij} are the pairwise LJ contact distance and well depth, respectively, and q_i and q_j are the partial charges of particles i and j, respectively. In the current PME implementation in AMBER20, while the reciprocal part contribution takes care of the long range periodic part, the total short range direct space contribution consists of the Coulombic term $U_{\mathrm{Coul}}(r_{ij})$ and the PME correction term:

$$\begin{aligned} U_{dir}(r_{ij}) &= U_{\mathrm{Coul}}(r_{ij}) - \mathrm{erf}(\kappa\, r_{ij})\left(\frac{q_i q_j}{4\pi\epsilon_0}\right)\frac{1}{r_{ij}} \\ &= \mathrm{erfc}(\kappa\, r_{ij})\left(\frac{q_i q_j}{4\pi\epsilon_0}\right)\frac{1}{r_{ij}} \end{aligned} \tag{18}$$

where erf() and erfc() are the error function and the complementary error function, respectively, and κ is the Ewald coefficient.

To soften these pairwise interactions particles, the introduction of a parametric form for scaling with an adjustable parameter is utilized to modify the interaction distance. A commonly used form of these modifications[62,64] is shown as

$$r_{ij}^{\mathrm{LJ}}(\lambda;\alpha) = \left[r_{ij}^{n} + \lambda\alpha\sigma_{ij}^{n}\right]^{1/n} \tag{19}$$

and

$$r_{ij}^{\mathrm{Coul}}(\lambda;\beta) = \left[r_{ij}^{m} + \lambda\beta\right]^{1/m} \tag{20}$$

where n and m are positive integers and α and β are adjustable positive semi-definite parameters for the LJ and Coul softcore interactions (note that α is unitless whereas β has units of distance raised to the power of m). The value of $n = 6$ and $m = 2$ are currently used as the default of AMBER.

Recently, the new smoothstep potential for nonbonded LJ and Coul interactions was implemented in AMBER20.[71] Since that time we have further improved the softcore potentials to be rigorously smooth with pairwise parameters such that the the softcore potential itself depends on the LJ contact distance σ_{ij}. This enabled us to overcome man of the problematic edge cases where the previous smoothstep softcore potential still had issues. Herein, we discuss three forms of the softcore potential: 1) the traditional softcore potential[64] that was the default in AMBER18, 2) the recently developed smooth softcore potential[71] that is the current default in AMBER20, and 3) the most recently developed smooth pairwise softcore potential that is the recommended methods in the AMBER Drug Discovery Boost package.

We introduce a new form of the interaction distance with separation-shifted scaling, given as

$$r_{ij}^{\mathrm{LJ}}(\lambda;\alpha^{\mathrm{LJ}}) = \left[r_{ij}^{n} + \alpha^{\mathrm{LJ}} W(r_{ij}) S_2(\lambda) \sigma_{ij}^{n}\right]^{1/n} \tag{21}$$

and

$$r_{ij}^{\mathrm{Coul}}(\lambda;\alpha^{\mathrm{Coul}}) = \left[r_{ij}^{m} + \alpha^{\mathrm{Coul}} W(r_{ij}) S_2(\lambda) \sigma_{ij}^{m}\right]^{1/m} \tag{22}$$

where α^{LJ} and α^{Coul} are the corresponding unitless parameter, and $W(r_{ij})$ is a switching function designed to smoothly return to the normal r_{ij} by the end of the cutoff

$$W(r_{ij}) \equiv 1 - S_2\left(\frac{r_{ij} - R_{cut,i}}{R_{cut,f} - R_{cut,j}}\right) \tag{23}$$

where $R_{cut,i}$ is the distance that the switching function begins switching and $R_{cut,f}$ is the final distance where the switching ends (returning the effective interaction distance to be r_{ij}).

A set of cases that illustrate the endpoint catastrophe, particle collapse, and large gradient-jump problems has been selected. The first test case involves absolute hydration free energy calculation of 3, 4-diphenyltoluene (denoted as the DPT/0), a hydrophobic system. The second test case is the absolute hydration free energy calculation of a Na^+ ion (denoted as the Na^+) , a small charged system that will introduce issues when it vanishes in solution. The third test case is the relative hydration free energy calculation of two Factor Xa ligands, L51c and L51h, which involve the transformation L51c → L51h in solution (denoted as the L51c/h) and migration of charge from one region of the ligand to another.[75]

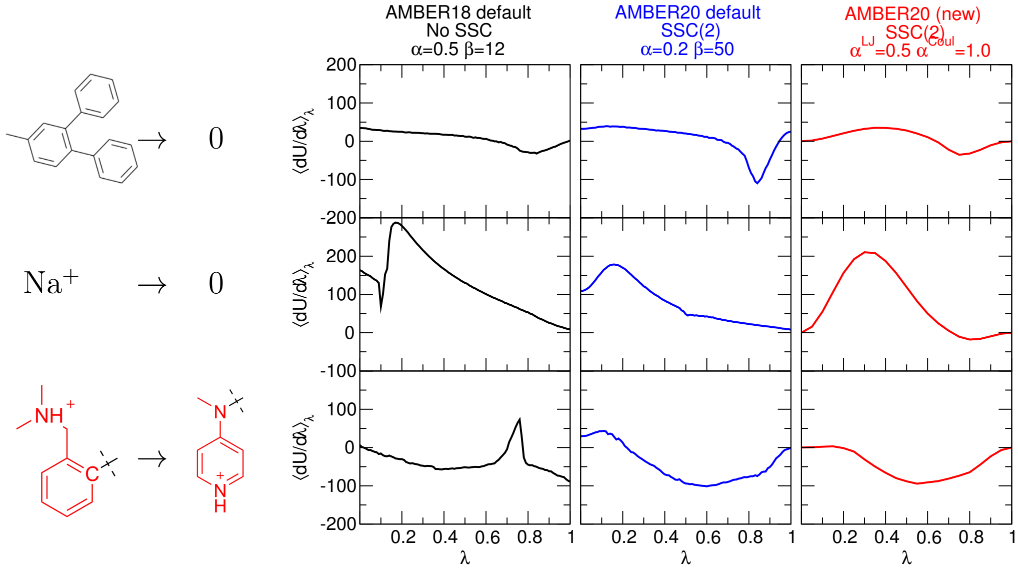

Figure 2: The $\langle \partial U/\partial \lambda \rangle_\lambda$ vs. λ plots for alchemical simulations of three molecular systems using the concerted scheme: the absolute hydration free energies for diphenyltoluene (upper rows) and the Na^+ ion (middle rows), and the relative hydration free energy simulations for the Factor Xa ligand L51c to L51h mutation (bottom rows). The L51c ligand has 65 atoms and L51h has 58 atoms. The red-colored atoms shown are the defined softcore regions. The atoms common to both ligands are not shown except the connecting carbon shown in black.
The first left column shows the results using the original AMBER softcore potentials. The middle column shows the results from the softcore potential with smoothstep function. The right column shows the results from the new unitless softcore potentials along with the switching function at cutoff.

Figure 2 shows the $\langle \partial U/\partial \lambda \rangle$ versus λ plots for alchemical free energy simulations of three test cases using the concerted scheme and different softcore potentials. The original AMBER softcore potential form of Eqs. 19 and 20 is utilized in the results shown in the first left column. The results from the softcore potential with smoothstep function, where λ is replaced with $S_2(\lambda)$, are shown in the middle column and the results with the unitless α^{LJ} and α^{Coul} scaling parameters and the switching function at cutoff (Eqs. 21, 22, and 23) are shown in the last right column. Note that there are the endpoint and the large gradient-jump problems with the original softcore potentials, which is the AMBER18 default, near $\lambda = 0$ and 1. With the smoothstep potential, which is the default AMBER20, those issues can be solved. The particle collapse problem can be observed in L51c/h around $\lambda = 0.7$ to 0.8 and in $\mathrm{Na}^{+}/0$ around $\lambda = 0.2$ with the original softcore potential, and disappear with smoothstep potential and the unitless softcore potential with the switching function at cutoff. Overall, we feel this is a considerable advance relative to the alternative softcore potentials we have been able to test.

3.3 New sampling tool: AlChemical Enhanced Sampling (ACES) for efficient sampling in the λ-dimension

If recalling Eq. 5 and focusing on the terms involving the TS regions, the alchemical transformation can be interpreted as a process that eliminates the interactions of the TS region with its surroundings (i.e., turn off the $U^{TS/(TC+I)}_{\{0/1\},abbr}$ terms), while at the same time, optionally eliminating certain internal potential energy terms within the TS region (i.e., certain $U^{TS}_{\{0/1\},abbr}$ terms). The specific terms in the latter are controlled by the gti_add_sc flag, as discussed above. In principle, as long as the internal potential energy within the TS region is treated consistently, and the proper constraints and conditions are imposed for the TS/(TC+I) interactions in the dummy state,[61,66–69] then theoretically, the specific choice of how to treat $U^{TS}_{\{0/1\}}$ in the dummy state is arbitrary. However, in practice, free energy estimates can be quite sensitive to this choice. The underlying reason is that one must be able to sufficiently sample the conformations involving the dummy atoms such that their contribution to the potential of mean force on the real atoms vanish, and further that their overall contribution to the free energy will be sufficiently converged such that it will cancel compensating edges of the thermodynamic graph. Ideally, one would want to create a dummy state that has a single local free energy basin with minimal phase space volume. Conversely, one strives to avoid a dummy state that have multiple free energy minima separated by high barriers. In a worse case scenario, sampling of the dummy state in one edge of a thermodynamic graph, for example a ligand transformation in aqueous solution, gets trapped in one local free energy basin, whereas the edge corresponding to the same ligand transformation in the protein complex is trapped in a different local free energy basin. This scenario is entirely possible if the conformation of the ligand differs when bound to the protein, and as a result would lead to skewed free energy predictions due to the systematic sampling error. Given sufficient sampling, ultimately both edges would converge, but this is often not practically feasible.

One way to reduce sampling barriers is to simply scale or eliminate interactions. However, taking this approach naively to an extreme, one easily realizes that this can lead to sampling of a much greater volume of phase space as degrees of freedom become less restrained. In practice, one must consider both of these factors: 1) elimination of barriers that prevent uniform sampling within the relevant phase space, and 2) reduction of the accessible phase space to a manageable sampling volume. It has been well-established that alchemical transformation can accelerate conformational sampling.[76–79] Further, methods that involve replica-exchange molecular dynamics (REMD) have been applied in many contexts for enhanced sampling.[80–84] One of the most popular in the drug discovery community has been variations of replica-exchange with solute tempering (REST2).[2,85] An excellent discussion of different strategies for overcoming orthogonal barriers in alchemical free energy calculations has been made recently by Hanh, König and Hüenberger.[86] It should be emphasized that, in practice, it is critical to be able to focus enhanced sampling only on the most relevant regions that are changing, and not to unduly increase the volume of phase space in the process. It is illustrative to consider two extreme examples as what "not to do". Consider the case where the binding of two different ligands requires the side chain of an amino acid in the binding pocket to adopt different rotamers. In the absence of the appropriate side chain re-arrangement during the alchemical transformation, the free energy estimates will differ substantially and lead to incorrect predictions of their relative binding affinities. In order to achieve the required conformational re-arrangement and converged free energy, the following would NOT be productive strategies to overcome the conformational side chain rotational barrier: 1) use an "unfocused" enhanced sampling method that attempted to unfold and refold the entire protein, or 2) use a "focused" enhanced sampling method to transformed the amino acid into an ideal gas of non-interacting atoms. The point of this illustration is that a more desirable method is one that is focuses enhanced sampling on the region that is undergoing conformational change, but that in doing so constructs an enhanced sampled "dummy state" by conservatively selecting only certain terms in the potential energy that facilitate the desired conformational transition while otherwise restricting the volume of phase space occupied by other degrees of freedom.

Herein we present a strategy for enhanced sampling that utilized Hamiltonian replica exchange (H-REMD) in the alchemical dimension, and leverages the use of new, robust smoothstep softcore potential framework together with scaling of select terms in the potential energy that enable a phase-space confined enhanced sampling "dummy state" for the specific region of interest. In the context of free energy simulations, the "specific region of interest" can encompass more than just the traditional TS region that is alchemically transforming: it can include regions of the ligand and/or protein itself that may be undergoing (possibly concerted) conformational re-arrangements along the alchemical dimension. These extended regions of interest are selected to be part of the separable-coordinate transforming regions in BOTH the forward and reverse directions (i.e., in both states 0 and 1) such that there is no net contribution to the free energy that arises in the transformation, but greater enhanced sampling is realized. In this way, the method fundamentally engineers a scenario where the conformational sampling dimension is NOT orthogonal to the alchemical dimension, and hence conformational barriers can be traversed in concert with the λ-dimension. In fact, the method can be used outside the context of free energy simulations to only consider sampling of a real end state by selecting the same TS region in both the forward and reverse directions. Several technical and feature advancements are required to make this strategy, which we designate as "AlChemical Enhanced Sampling" (ACES), work effectively, and these are discussed below and supported by a series of illustrative examples.

In order for ACES to perform efficient enhanced sampling, two requirements need to be achieved:

- Focused enhanced sampled “dummy state” needs to be created where the potential energy terms and interactions that give rise to the targeted conformational barriers are reduced or eliminated
- Enhanced sampled conformations in the “dummy state” need to be efficiently propagated to the real state endpoint

In the AMBER Drug Discovery Boost package, the first requirement can be achieved through proper selection of the TS region targeted for focused enhanced sampling and use of the "gti_add_sc" control flag to enable tuning of the potential energy terms in the dummy state (Table 3), along with the λ-scheduling implementation and control parameters that provide flexibility to set the corresponding weights and schedules. The second requirement can be achieved by using the Hamiltonian replica exchange[76,78,87–89] framework in AMBER, along with the new smoothstep softcore potentials and λ-scheduling features in the boost package to enable suitable transformation pathways that facilitate conduction in the λ-dimension of the conformational ensembles generated in the enhanced sampled “dummy state” to the real state endpoint. It should be re-emphasize that ACES is not limited to alchemical free energy simulations, but can be used for traditional molecular dynamics simulations of a single thermodynamic state of the system. In real drug discovery applications, ACES could be used to both predict the conformational ensembles of the real state endpoints (i.e., to sample and refine the structure of each ligand-protein complex) corresponding to nodes in a thermodynamic graph, and then be used again in the free energy simulations of each edge of the thermodynamic graph to obtain estimates of the RBFEs and ultimately a ranking. Below we illustrate the use of ACES in several examples that build up in complexity.

3.3.1 Energy barrier in the dummy state: acetic acid as an example:

It is well known that acetic acid has different favorable conformations in gas phase and in aqueous phase and the energy barriers between the *cis* and *trans* conformations (about the O=C-O-H torsion) are ~ 11.0 and ~ 6.5 kcal/mol in gas phase and in aqueous phase, respectively.[90] These high energy barriers lead to challenges for accurate calculation of the absolute hydration free energy of acetic acid. One might naively think that simply using a starting structure that had the proton in the correct conformation, should that conformation be known *a priori*, would solve the problem. In fact, this is not the case in general. The reason is that conformational barriers may persist even in the dummy state, depending on how the dummy state is defined. Recall, for an absolute hydration free energy, the dummy state arising from the gas phase and aqueous phase edges are formally identical. However, if there is a large energy barrier between conformations in the dummy state itself, the transformation from the gas phase (*cis*) will remain trapped in the *cis* conformation in the dummy state, whereas the transformation from the aqueous phase (*trans*) will remain trapped in the *trans* conformation in the dummy state, leading to inconsistent results. In order to remedy this potential problem, one must define the energy terms in the dummy state such that transitions can readily occur between *cis* and *trans*, and an enhanced sampling equilibrium can be achieved. This also ensures, that the conformers in the real state will be sampled with the correct occupations. As will be discussed below, a sufficient condition for this to occur for acetic acid is to have only the bond, angle and van der Waals terms contribution to the internal TS potential energy in the dummy state (no electrostatics, torsion angle or 1-4 terms).

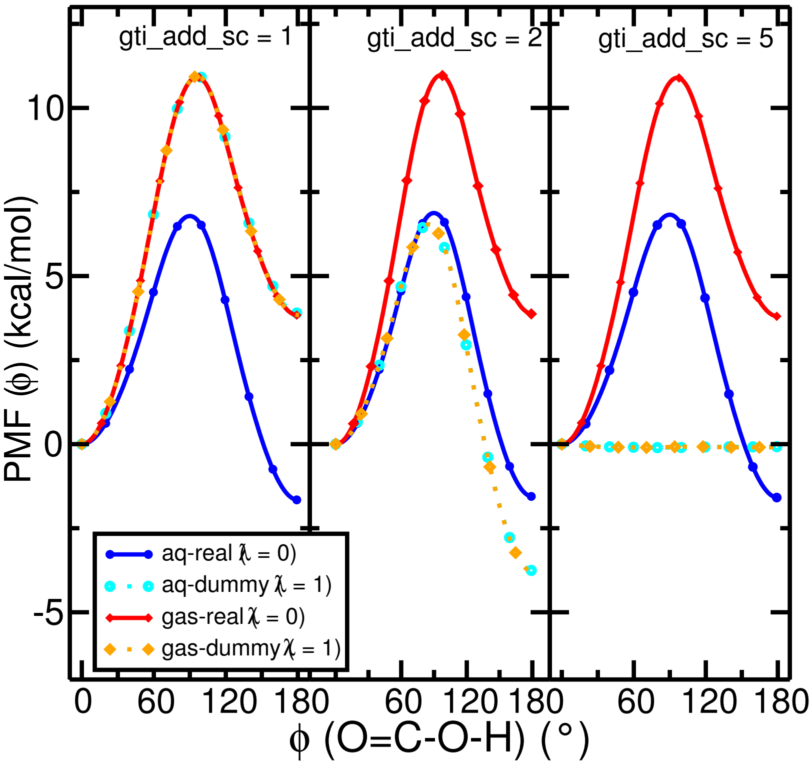

Figure 3: The PMFs along the O=C-O-H torsion angle of acetic acid in the aqueous and gas phases, created through umbrella sampling with 11 windows for the real state (λ = 0, the acetic acid has full interactions with the environment) and the dummy state (λ = 1, the acetic acid is fully decoupled from the environment)) and with three gti_add_sc switches (see Table 3).

In order to establish an independent benchmark for the conformational free energy profile for acetic acid, we first performed umbrella sampling simulations scanning the relevant O=C-O-H torsion angle for the real state where the acetic acid molecule has full interactions with its environment, and also in the dummy state where the acetic acid molecule has no interactions with its environment and different internal potential energy controlled by the gti_add_sc flag values. With gti_add_sc=1, all internal interactions within the TS region (defined as the whole acetic acid molecule) are not scaled hence kept in the dummy states, the dummy states of the acetic acid both in gas phase and in aqueous phase is exactly the same as the real state in gas phase. This is confirmed and illustrated in the leftmost panel of Figure 3. The forward and reverse barriers of the dummy state with gti_add_sc=1 are 10.9 and 6.5 kcal/mol, respectively. With gti_add_sc=2, the internal electrostatic interactions within the TS region are scaled to zero in the dummy states, and this leads to a dummy states that prefer the *trans* conformation and energy barriers are similar in magnitude but the order is reversed (forward and reverse barriers are 7 and 11, respectively), showed in the middle panel of Figure 3. With gti_add_sc=5, when only the internal vDW interactions (excluding 1-4 LJ), bond terms, and bond angles terms are kept in the dummy states (Table 3), the PMFs become essentially flat in the dummy states (forward and reverse barriers less than 0.1 kcal/mol, shown in the rightmost panel of Figure 3). As a result, simulations of latter dummy state will not suffer from the high energy barriers between the *cis* and *trans* conformations, and enhanced conformational sampling and free energy convergence can be fairly easily achieved.

In order to achieve the ACES requirement of conformation propagation between different λ-windows, the Hamiltonian replica exchange (H-REMD) framework of AMBER20 is utilized. Herein, we performed the absolute free energy calculation of acetic acid with the gti_add_sc=5 flag but with different starting conformations. Since the Hamiltonian replica exchange (H-REMD) framework propagates the conformational ensembles through the different λ states, the real state end point can sample conformations originating from the enhanced sampled dummy state, and will do so in the correct Boltzmann populations. As a result, the computed absolute hydration free energies without the H-REMD are 5.33 ± 0.23 kcal/mol and 6.83 ± 0.28 kcal/mol started from *cis* and *trans*, respectively. The free energy differences derived from different conformational starting points here reflects the degree to which sampling of the conformations is incomplete (larger differences result from less complete sampling). With the H-REMD and gti_add_sc=5, the absolute hydration free energies are 6.06 ± 0.08 kcal and 5.95 ± 0.10 kcal/mol from *cis* and *trans* starting conformations, respectively, which are not statistically distinguishable. A previous study of absolute hydration free energy of acetic acid employing multiple real and dummy state conformations connected with rigorous umbrella sampling PMFs[70] produced 5.96 ± 0.10 kcal/mol, exactly the same as the ACES (gti_add_sc=5 with H-REMD) result here. The agreement suggests that ACES successfully overcomes the conformational challenges in calculating the absolute free hydration free energy of acetic acid.

Table 5: The forward and reverse energy barriers for acetic acid and the absolute hydration free energy with different git_add_sc flags.

			gti_add_sc=1	**gti_add_sc=2**	**gti_add_sc=5**
PMF profile	**aq**	$\Delta G^{\ddagger}$ (forward)	10.95	6.54	-0.10
		$\Delta G^{\ddagger}$ (reverse)	7.05	10.29	-0.03
		ΔG	3.90	-3.75	-0.07
	gas	$\Delta G^{\ddagger}$ (forward)	10.94	6.54	-0.10
		$\Delta G^{\ddagger}$ (reverse)	7.10	10.32	-0.01
		ΔG	3.84	-3.78	-0.09
TI with HREMD	$\Delta\Delta G$ (*cis* starting)		4.57	4.41	6.06*
	$\Delta\Delta G$ (*trans* starting)		9.60	9.64	5.95*

All entries are in kcal/mol. The data for PMF profiles are from the dummy state.
gti_add_sc=1 : $U^{TS} = U_b + U_b$; gti_add_sc=2 : $U^{TS} = U_b + U_{LJ} + U_{1-4LJ}$; gti_add_sc=5 : $U^{TS} = U_{bond} + U_{ang} + U_{LJ}$.
*ACES procedure.

3.3.2 Ligand internal rotation problem–CDK2 as an example:

We further applied the ACES approach on a well-known protein-ligand binding problem: the 1h1r ligand bound to CDK2.[91,92] The torsion angle of the phenyl ring of the 1h1r ligand has two distinguished states, *cis* (torsion angle = -8.81 ^{o}) and *trans* (torsion angle = 150.75 ^{o}) , in the crystal structure (PDBID: 1OIY). A binding free energy study using the REST2 enhanced sampling approach[44] has demonstrated that simulations without enhanced sampling cannot access both *cis* and *trans* conformational states and utilizing the REST2 approach will overcome the problem. We applied ACES on this system with different starting conformational states and studied the distributions of both *cis* and *trans* conformational states in the CDK2-1h1r to CDK2-1h1q alchemical relative binding simulations. Fig. 4 shows the time series and distributions of the relevant torsion of the real state of 1h1r ($\lambda = 0$) with and without ACES and clearly demonstrates that ACES is able to sample both states regardless of the starting conformation.

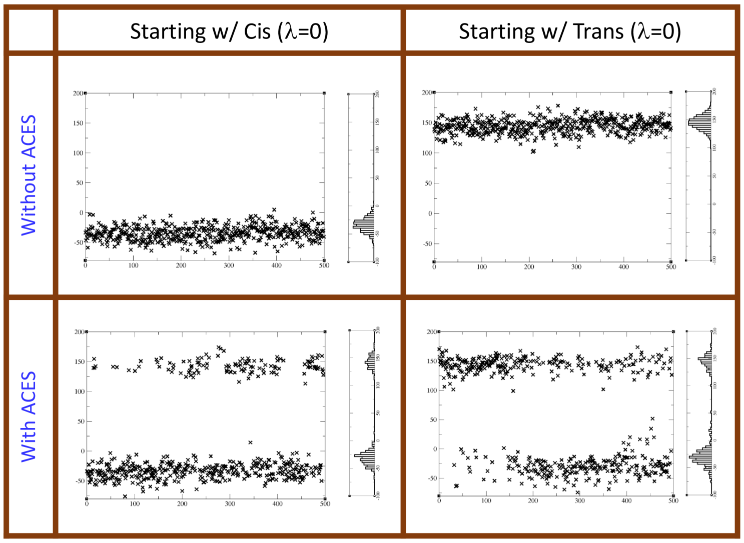

Figure 4: Time series (5 ns) of the relevant torsion with and without ACES from CDK2-1h1r to CDK2-1h1q alchemical relative binding simulations. The distributions are for the the real state of 1h1r ($\lambda = 0$). Without ACES (gti_add_sc=2 and no REMD), the ligand torsion will stay at the initial conformation, either *cis* or *trans*. Without ACES (gti_add_sc=5 and REMD enabled), the ligand torsion will jump between *cis* and *trans* regardless of the initial conformation. The distribution figures are created for the last 3 ns data and show that ACES delivers similar distributions for simulations starting from different initial conformations.

3.4 New analysis tool: BARnet and MBARnet methods for network-wide analysis of compound libraries with cycle closure and experimental reference constraints

Free energy simulations have a number of applications, including the calculation of relative ligand binding free energies (RBFEs).[61,93–97] In this type of application, a drug (ligand) has been found to bind to a protein target, and one seeks to find similar drugs that are potentially more efficacious. A set of ligands is proposed that differ from the lead ligand by adding or removing functional groups, and the computational chemist is tasked with predicting their RBFEs.[98–105] The ligands ranked according to their RBFEs relative to the lead ligand, and the ligands that strongly bind to the protein are presumed to have improved efficacy. The RBFE between any two ligands are performed using as a series of alchemical free energy simulations, and transformations are chosen that can connect any ligand to any other ligands, either directly or through a free energy pathway that includes other ligands. In other words, the set of ligands form a network, or "graph", where the "edges" are the transformations performed using free energy simulations. Redundant pathways may be considered, such that the network contains free energy cycles. The free energy along a closed path is formally zero; however, systematic or random errors in the sampling often lead to nonzero values (cycle closure errors). Traditionally, the free energy of each edge is analyzed independently from the other edges, preventing the enforcement of cycle closure constraints on the analysis.[43,44] There are several methods frequently used for analyzing alchemical transformations, including thermodynamic integration (TI),[47] Bennett acceptance ratio method (BAR),[57] multistate BAR (MBAR),[39] and the unbinned WHAM (UWHAM) method.[38] The MBAR and UWHAM methods are formally equivalent, and the solution to the transformation free energies can be obtained by nonlinear optimization of a convex objective function.[38]

We have recently introduced an analysis method called MBARnet, that simultaneously solves for the free energies of all free energies in the network, rather than a single edge.[60] The simultaneous solution is obtained by optimizing a global objective function (see Eqn. 24). The global objective function is a weighted sum of the MBAR/UWHAM objectives for each edge. One can then constrain the optimization to enforce cycle closure conditions, or enforce the solution to reproduce experimental or highly-converged reference RBFEs for a select subset of transformations, if available. The BARnet and MBARnet methods, along with new methods for analysis of high-dimensional free energy surfaces,[106] have been implemented into the `FE-ToolKit` software package.

$$\begin{gathered}\min F(\mathbf{G}^*) = \min \left\{ \sum_{e}^{N_{\text{edges}}} w_e f(\mathbf{G}_e^*) \right\} \\ \text{subject to } h_c(\mathbf{G}_e^*) = 0 \;\; \text{for } c = 1, \cdots, N_{\text{con.}}\end{gathered} \tag{24}$$

$$h_c(\mathbf{G}_e^*) = \sum_{e}^{N_{\text{edges}}} \sum_{i}^{M_e} C_{\text{con.},(c,ie)} G_{ie}^* - \Delta G_{\text{con.},c}^* \tag{25}$$

The MBAR/UWHAM objective function is shown in Eqn. 26.

$$\begin{aligned} f(G_{1e}^*, \cdots, G_{M_e e}^*) &= f(\mathbf{G}_e^*) \\ &= \frac{1}{N_e} \sum_{j=1}^{M_e} \sum_{k=1}^{N_{je}} \ln \left(\sum_{l=1}^{M_e} \exp(-[u_{le}(\mathbf{r}_{je}^k) + b_{le}]) \right) + \sum_{i=1}^{M_e} \frac{N_{ie}}{N_e} b_{ie} \end{aligned} \tag{26}$$

The b_{ie} values (Eqn. 27) are used for notational compactness.

$$b_{ie} = -\ln \frac{N_{ie}}{N_e} - G_{ie}^* \tag{27}$$

In our notation, N_{edges} is the number of edges in the free energy network. G_{ie}^* is the free energy of alchemical state i within edge e. M_e is the number of alchemical states within edge e. Molecular dynamics simulations of the M_e states are performed, generating N_{ie} frames of coordinates $\mathbf{r}_{ie}^k$. where k indexes the frame within the trajectory. u_{ie} is the reduced potential energy using the Hamiltonian defined by alchemical state i within edge e, which needs to be evaluated for each of the collection of $N_e = \sum_{i=1}^{M_e} N_{ie}$ frames sampled from all simulations within the edge. w_e is a weight assigned to each edge. In the present work, the edges are uniformly weighted; the optimal choice of the edge weights is an active area of research.

The nonlinear optimization of Eqn. 24 with respect to the G^*_{ie} values is subjected to linear constraints (Eqn. 25). $N_{\text{con.}}$ is the number of constraints and $\Delta G^*_{\text{con.},c}$ is the target value of constraint c. The constraint is a linear combination of free energy values, where $C_{\text{con.},(c,ie)}$ is the contribution from λ-state i within edge e to constraint c. The values of $C_{\text{con.},(c,ie)}$ are zero, except for the $\lambda = 0$ and $\lambda = 1$ states of the alchemical transformation; that is, $\Delta G = G(\lambda = 1) - G(\lambda = 0)$. If N_{trial} independent trials of the the alchemical transformation is performed, then there are multiple G^*_{ie} values corresponding to the same physical process. A constraint could be applied to each trial; however, our preference is to apply constraints to the trial average free energy, $\langle \Delta G \rangle$. In this case, the nonzero $C_{\text{con.},(c,ie)}$ coefficients for the $\lambda = 0$ and $\lambda = 1$ states of the physical process are $-N^{-1}_{\text{trial}}$ and N^{-1}_{trial}, respectively. The nonzero elements of a cycle closure constraint involve the trial average free energy for each edge along the path.

As an example application, Fig. 5 compares the ligand RBFEs bound to the P38 protein (PDBID: 3FLY) between experiment[107] and MBARnet analysis with and without cycle closure conditions. The free energy network consists of 33 ligands connected by 54 edges. The edge connectivity forms 42 minimum length cycle closures; that is, cycles that do not contains encompass interior cycles.

When cycle closure constraints are not enforced, the 42 cycles closure conditions have a mean unsigned error (MUE) of 0.83 kcal/mol. When cycle closure conditions are enforced, the unsigned errors are zero. Accurate RBFEs necessarily enforce cycle closures; however, enforcement of cycle closure constraints does not guarantee that the computed RBFEs will necessarily improve agreement with experiment. The calculated RBFE MUE of the 54 edges in the P38 system improves only 0.04 kcal/mol, relative to experiment, when cycle closure constraints are enforced. Similarly, the mean signed errors (MSE) improve by only 0.02 kcal/mol. Given sufficiently long sampling, the simulations should produce RBFEs that naturally satisfy cycle closures conditions without the need for constraints, but differences with experiment will persist due the inaccuracies of the molecular mechanical force field. Further developments of the method would be to optimally choose the edge weights, w_e in the global objective function.

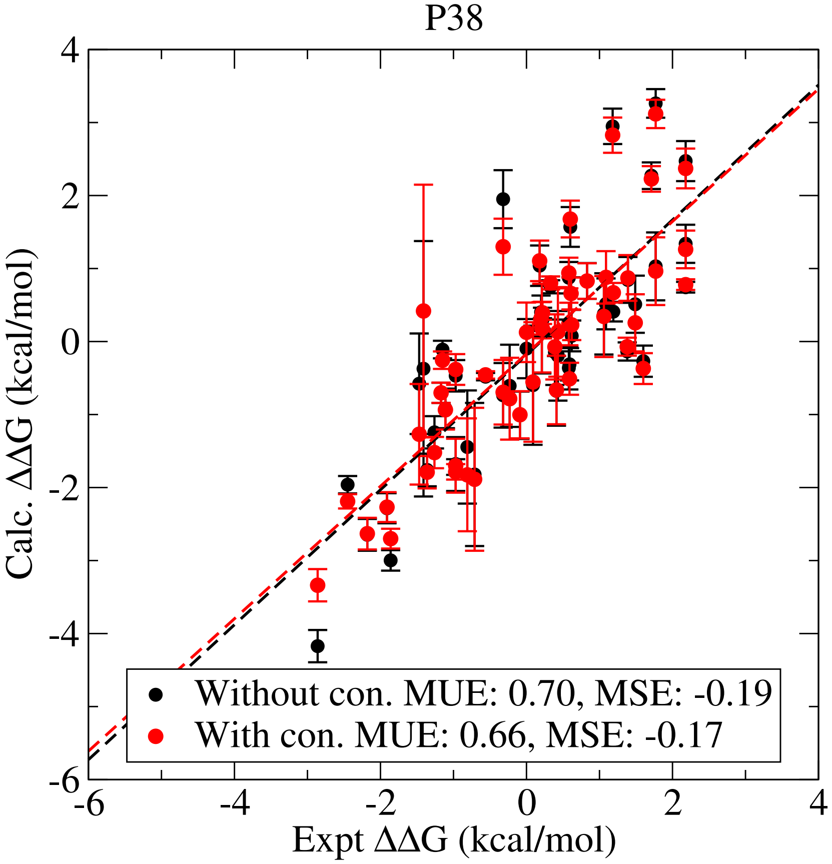

Figure 5: Comparison of ligand RBFEs bound to the P38 protein using MBARnet with and without cycle closure constraints. The experimental values were computed from reported IC50 values (Ref. 107). The red and black dashed lines are linear regressions of the calculated results with (slope: 0.91, intercept: -0.17 kcal/mol, Pearson correlation coefficient: 0.82) and without cycle closure constraints (slope: 0.92, intercept: -0.18 kcal/mol, Pearson correlation coefficient: 0.81), respectively, to the experimental values.

3.5 New workflow tool: Streamlined workflows for performing and analyzing simulations of relative binding free energies with AMBER Drug Discovery Boost

Workflows are essential tools in large-scale active drug discovery projects.[108] Workflows help to ensure best practices[109] and enable high throughput with reduced human effort and error.[102,110–113] We have developed a robust workflow (Figure 6) that optimizes and automates various steps that are involved in set up, equilibration and production/data collection, and analysis of relative binding free energy simulations for a network of ligand transformations (thermodynamic graph for a set of compounds).

Key highlights of our workflow tool

- *Automated setup of file infrastructure* - The workflow, while giving top-level control to the user, automates the various laborious and time-consuming intermediate steps involved in setting up binding free energy calculations. For a given network of transformations, the workflow facilitates:
  - Generation of “single-topology” parameter and coordinate files starting from crystal structures
  - Generation of TC and TS regions for individual transformations using multiple algorithms
  - Generation of necessary AMBER input files and job submission scripts
- *Enhancing conformational sampling in simulations* - The workflow brings together several of our recent methodological advances in enhanced sampling techniques to accelerate convergence in simulations, and improve precision of predicted ligand binding free energies. Specifically, the workflow enables the use of:
  - *ACES* as a tool to increase sampling along the coordinates that are most relevant to a given transformation

  - *2-state* simulation setup in conjunction with *H-REMD* to increase overall conformational sampling
  - robust equilibration and production protocol to alleviate initial conformational bias

- *Seamless analysis of simulations with BARnet and MBARnet* - The workflow interfaces the simulation output files with *BARnet* and *MBARnet* thus enabling network-wide analysis of binding free energies with or without the use of experimental constraints.

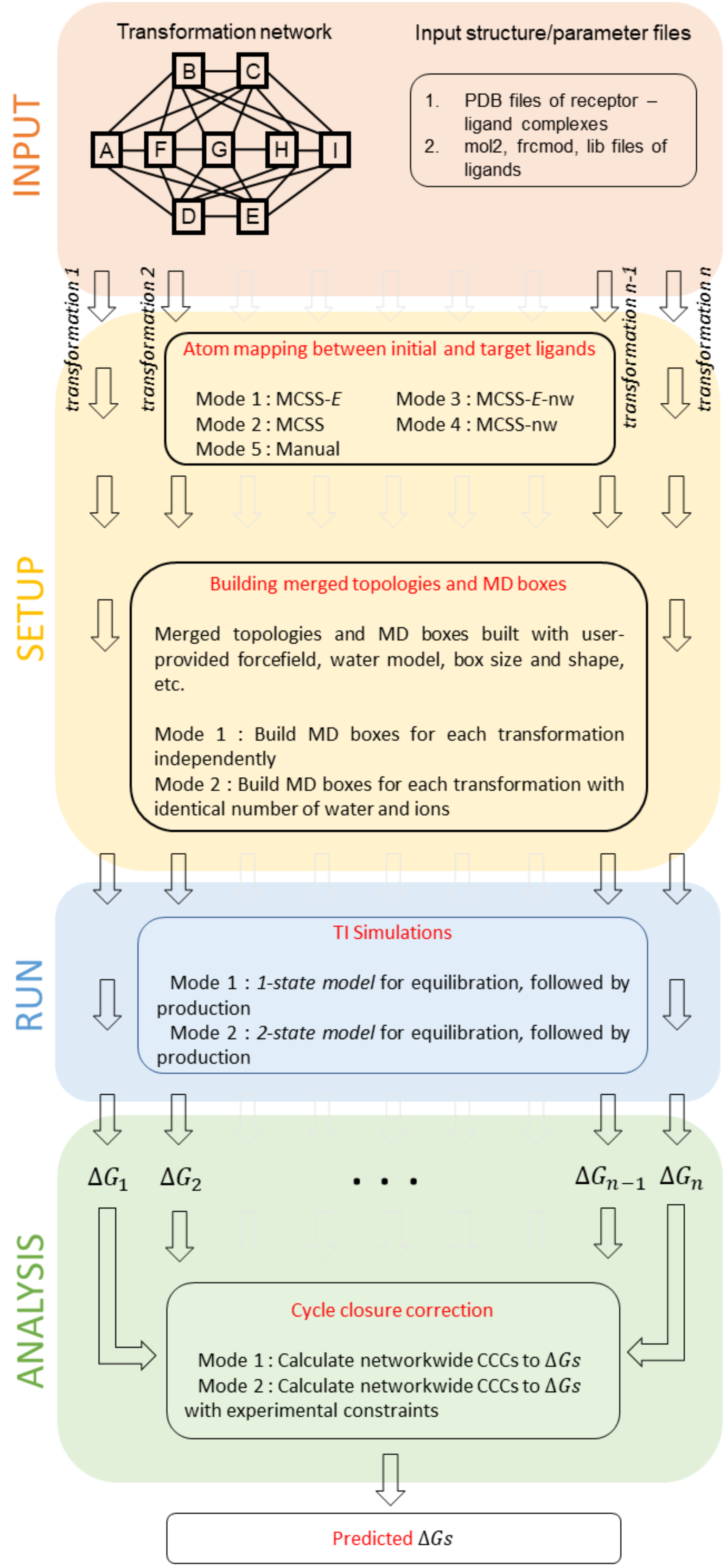
INPUT
Transformation network
B
C
A
F
G
H
I
D
E
Input structure/parameter files
1. PDB files of receptor – ligand complexes
2. mol2, frcmod, lib files of ligands
transformation 1
transformation 2
transformation n-1
transformation n
SETUP
Atom mapping between initial and target ligands
Mode 1 : MCSS-E
Mode 2 : MCSS
Mode 5 : Manual
Mode 3 : MCSS-E-nw
Mode 4 : MCSS-nw
Building merged topologies and MD boxes
Merged topologies and MD boxes built with user-provided forcefield, water model, box size and shape, etc.
Mode 1 : Build MD boxes for each transformation independently
Mode 2 : Build MD boxes for each transformation with identical number of water and ions
RUN
TI Simulations
Mode 1 : 1-state model for equilibration, followed by production
Mode 2 : 2-state model for equilibration, followed by production
ANALYSIS
ΔG_1
ΔG_2
• • •
ΔG_{n-1}
ΔG_n
Cycle closure correction
Mode 1 : Calculate networkwide CCCs to ΔGs
Mode 2 : Calculate networkwide CCCs to ΔGs with experimental constraints
Predicted ΔGs

Figure 6: Workflow for performing relative binding free energy simulations. Different options in each layer are indicated as numbered "Modes".

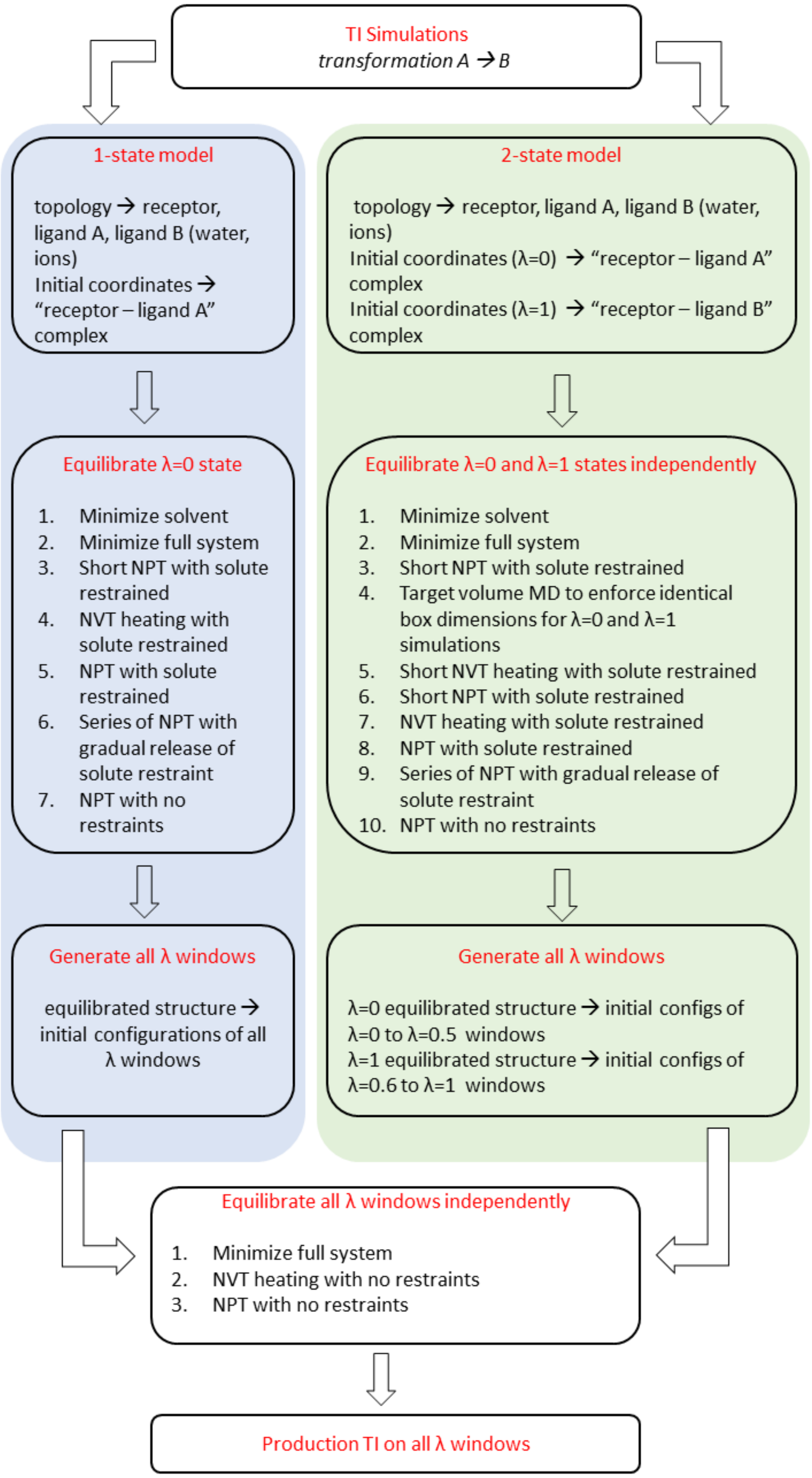
TI Simulations
transformation A → B
1-state model
topology → receptor, ligand A, ligand B (water, ions)
Initial coordinates → "receptor – ligand A" complex
2-state model
topology → receptor, ligand A, ligand B (water, ions)
Initial coordinates (λ=0) → "receptor – ligand A" complex
Initial coordinates (λ=1) → "receptor – ligand B" complex
Equilibrate λ=0 state
1. Minimize solvent
2. Minimize full system
3. Short NPT with solute restrained
4. NVT heating with solute restrained
5. NPT with solute restrained
6. Series of NPT with gradual release of solute restraint
7. NPT with no restraints
Equilibrate λ=0 and λ=1 states independently
1. Minimize solvent
2. Minimize full system
3. Short NPT with solute restrained
4. Target volume MD to enforce identical box dimensions for λ=0 and λ=1 simulations
5. Short NVT heating with solute restrained
6. Short NPT with solute restrained
7. NVT heating with solute restrained
8. NPT with solute restrained
9. Series of NPT with gradual release of solute restraint
10. NPT with no restraints
Generate all λ windows
equilibrated structure → initial configurations of all λ windows
Generate all λ windows
λ=0 equilibrated structure → initial configs of λ=0 to λ=0.5 windows
λ=1 equilibrated structure → initial configs of λ=0.6 to λ=1 windows
Equilibrate all λ windows independently
1. Minimize full system
2. NVT heating with no restraints
3. NPT with no restraints
Production TI on all λ windows

Figure 7: Protocol for running TI simulations.

3.5.1 Automated generation of TC and TS regions

A critical step in the setup of a RBFE simulation is the one-to-one mapping of equivalent atoms in the reference and target ligand molecules that defines the TC and TS regions. Defining the TC and TS regions manually is a simple task when performing a small number of RBFE calculations between similar ligands but becomes increasingly tedious as the transformation network increases in size and complexity, and can become very time consuming and lead to human error. Our workflow enables the automated generation of the TC and TS regions associated with the various desired transformations using four different algorithms, referred to as MCSS, MCSS-E, MCSS_{nw}, and $\text{MCSS-E}_{\text{nw}}$. MCSS corresponds to the use of the *Maximum Common Substructure Search* algorithm[114] as implemented in the Cheminformatics software RDKit.[115] MCSS uses a similarity criterion to decide if an atom or bond match between two structures and aims to identify their maximum overlap. MCSS, while widely used in context of automated alchemical free energy simulations, in its original form may not always be suitable, particularly in cases where atom mapping based on "maximum overlap" is not desired, and may lead to unstable TI simulations or cycle closure issues. MCSS-E (or "extended" MCSS) is an atom-mapping algorithm we developed that builds on the original MCSS algorithm and excludes from the "maximum overlap" region that is identified purely from structural similarity, atoms that differ either in chemical identity or hybridization. This extension leads to more stable TI simulations. MCSS_{nw} and $\text{MCSS-E}_{\text{nw}}$ correspond to variants of MCSS and MCSS-E, respectively, which ensures that the TC and TS regions of each unique ligand molecule are identical in all transformations in which the ligand participates within the given network (nw). Such a definition, along with setting up each system with identical number of solvent particles, would enable new network-wide enhanced sampling methods to be used where H-REMD could be performed to exchange between simulations along different edges of the thermodynamic graph. This technology is forthcoming, but not yet fully mature.

3.5.2 Automated generation of topology and starting configuration files for RBFE simulations

RBFE simulations on a network of transformations require the MD simulation boxes for the various TI calculations to be prepared in a consistent fashion. Our workflow can in an automated way generate all the necessary topology and configuration files using user-defined force field, water and ion models, box size and shape, ion concentration, and if specified, containing identical number of water molecules and ions. Moreover, the workflow has the flexibility to generate the topologies with and without hydrogen mass repartitioning (HMR).

The initial configuration files for a given RBFE calculation can be generated in two different ways. In the conventional approach, referred to here as the *1-state model*, only the receptor-reference ligand structure is considered and corresponds to the λ=0 state, while in the *2-state model*, both the receptor-reference ligand structure and receptor-target ligand structures are considered and correspond to the λ=0 state and λ=1 state, respectively. The latter is particularly useful if the conformation of the receptor is different in the receptor-reference and receptor-target complexes.

3.5.3 Automated generation of AMBER DD BOOST input files for a robust equilibration protocol

Sufficient equilibration of starting structures is essential for accurate and precise RBFE predictions. Our workflow utilizes an exhaustive and carefully chosen equilibration protocol illustrated in Figure 7 and generates the input file infrastructure necessary for running equilibration and production simulations. Equilibration simulations are divided into two phases; the first phase consists of rigorous equilibration of only the λ=0 state in case of the *1-state model* and both λ=0 and λ=1 states for the *2-state model.* This is followed by the second phase in which all λ states are generated and equilibrated independently. In case of the *1-state model*, all λ states are generated from the equilibrated λ=0 state, while in case of the *2-state model*, first half of the λ windows are generated from the equilibrated λ=0 state and the other half of the λ windows are generated from the equilibrated λ=1 state. Note: the *1-state model* can produce hysteresis when the reference and target ligands are switched, as this will change the starting conditions for the equilibration. For the *2-state model*, initial conditions consider both end states symmetrically that greatly reduce or eliminate hysteresis.

Production simulations are initiated from the structures obtained at the end of the equilibration. The workflow allows top-level control on the production simulation parameters, such as simulation length, time step, use of replica exchange and flags that are specific to AMBER DD BOOST.

3.5.4 Automated analysis of RBFE simulations

Once the production TI simulations are performed, all simulations can then be collectively analyzed using BARnet and MBARnet methods[60] implemented into `FE-ToolKit`.

4 Outlook for the Future

Free energy simulations have been around for more than two decades, and for the most part have not yet lived up to their promise of delivering robust predictive capability for drug discovery. We are now at the stage where new methods, low-cost high-performance computing hardware, and GPU-accelerated software implementations[22] have come together to make accessible free energy simulations that, for certain types of transformations, can be made with meaningful precision.[6] This landmark achievement has positioned the field to begin to meaningfully explore factors that affect the accuracy of free energy predictions, perhaps the most important of which are the force field models themselves. As ligand binding involves the transfer of a ligand from often starkly different electrostatic environments (e.g., from aqueous solution to a protein binding pocket), it is likely that inclusion of explicit many-body polarization response will be important in the potential. Further, the short ranged interactions between the ligand and protein are likely too complex to be reliably modeled by traditional point-charge electrostatic and Lennard-Jones potentials. One strategy is to use a quantum mechanical force field (QMFF) framework[116–118] for the ligand using a fast, approximate QM model with enhanced QM-QM and QM/MM interaction potentials from deep learning.[119,120] While evaluation of these potentials are significantly more computationally intensive than a traditional MM force field, one can apply so-called book-ending corrections to only the end states with relatively modest sampling.[121]

Nonetheless, challenges still remain for certain types of complex ligand transformations or binding events, including those that involve: 1) significant coupled conformational re-arrangement of the ligand and protein binding site, 2) charge-changing perturbations between ligands, 3) scaffold/core hopping, 4) changes in tautomer and/or protonation state of the ligand and/or target, 5) binding to metal centers, 6) covalent inhibition. The field will evolve to address these challenges and advance the state-of-the-art. Further development of enhanced sampling methods that focus on the most relevant regions of phase space will continue to be vitally important for making robust predictions for more complex transformations. The full integration and testing of generalized ensemble methods that enable dynamic sampling of tautomer and protonation states will also be important. The use of an accurate QMFF framework (including augmentation with deep learning potentials) will be important for modeling ligand compounds in order to address the tautomer/protonation state problem, as well as challenges modeling binding to metal ions and mechanisms of covalent inhibition where new covalent bonds between the ligand and target are formed. Finally, the further development and application of methods for network-wide analysis that include integration of experimental data to enhance predictive capability will be extremely valuable for the development of new drugs in addition to helping to inform precision medicine therapies.

A Computational Details

A.1 Simulation setup for the PMF profile of acetic acid

The PMF calculations of acetic acid were done with ff14SB[122] and GAFF force field[22,123] with TIP3P[124] waters. There are a total of 61 umbrella simulations involving equally-spaced displacements along the O=C-O-H torsion angle coordinate between 0 and 180 degrees. Each window is minimized and followed by 5 ps equilibration. Each simulation is performed for 2.7 ns (the first 200 ps was discarded and the remaining 2.5 ns was used for data for analysis), and restrained harmonically using a force constant of 200 kcal/mol/rad^2.

A.2 General setup for the relative binding free energy simulations of CDK2

The relative binding free energy simulations of the 1h1r to 1h1q were performed with the pmemd.cuda module of AMBER20.[19,21,22] The ligand was modeled using the GAFF2 force field,[125] and the condensed phase environment was explicitly modeled with TIP4P Ewald[126] waters. The whole ligands are defined as the transforming regions (TC+TS) while the phenyl ring is defined as the TS region. The transformations were performed in the modified SSC(2) softcore potentials with (m=n=2, α=0.5; $\beta = 1$) and with one-step concerted softcore protocol using 21 alchemical evenly-spaced states between $\lambda = 0.0$ and $\lambda = 1.0$ with spacing of 0.05. Each simulation was run in the isothermal-isobaric ensemble for 5 ns using a 1 fs time step. The Berendsen barostat[127] and Langevin thermostat[128] were used to maintain a temperature of 298 K and 1 atm pressure. The long-range electrostatics were evaluated with the particle mesh Ewald method using a 1 Å^3 grid spacing.[129,130] The H-REMD exchange interval is 20 fs. Only ligands complexed with CDK2 were simulated to demonstrate the torsion distribution in the protein environment hence no $\Delta\Delta G$ of relative binding free energy values are reported here.

A.3 General setup for the relative binding free energy simulations of P38

The free energy simulations were performed with the pmemd.cuda module of AMBER20.[19,21,22] The ligand was modeled using the GAFF2 force field,[125] and the condensed phase environment was explicitly modeled with TIP3P[124] waters. The transformations were performed in 3 stages: decharge, softcore Lennard-Jones,[64] and recharge. The decharge and recharge stages were each performed using 5 evenly spaced λ values. The softcore stage was performed using 12 alchemical states: λ = 0.0, 0.0479, 0.1151, 0.2063, 0.3161, 0.4374, 0.5626, 0.6839, 0.7937, 0.885, 0.9521, and 1.0. Ten independent trials of each simulation were run using different random number seeds to adjust the initial conditions. Each simulation was run in the isothermal-isobaric ensemble for 2 ns using a 4 fs time step and hydrogen mass repartitioning. The Berendsen barostat[127] and Langevin thermostat[128] were used to maintain a temperature of 298 K and 1 atm pressure. The long-range electrostatics were evaluated with the particle mesh Ewald method using a 1 Å^3 grid spacing.[129,130]

Acknowledgement

The authors are grateful for financial support provided by the National Institutes of Health (No. GM107485 to DMY). Computational resources were provided by the Office of Advanced Research Computing (OARC) at Rutgers, The State University of New Jersey, the National Institutes of Health under Grant No. S10OD012346, the Blue Waters sustained-petascale computing project (NSF OCI 07-25070, PRAC OCI-1515572), and by the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation (No. ACI-1548562 and No. OCI-1053575). We gratefully acknowledge the support of the nVidia Corporation with the donation of several Pascal and Volta GPUs and the GPU-time of a GPU-cluster where the reported

benchmark results were performed.

References

(1) Jorgensen, W. L. *Acc. Chem. Res.* **2009**, *42*, 724–733.

(2) Abel, R.; Wang, L.; Harder, E. D.; Berne, B. J.; Friesner, R. A. *Acc. Chem. Res.* **2017**, *50*, 1625–1632.

(3) Armacost, K. A.; Riniker, S.; Cournia, Z. *J. Chem. Inf. Model.* **2020**, *60*, 1–5.

(4) Cournia, Z.; Allen, B. K.; Beuming, T.; Pearlman, D. A.; Radak, B. K.; Sherman, W. *J. Chem. Inf. Model.* **2020**, *60*, 4153–4169.

(5) Song, L. F.; Merz, K. M. *J. Chem. Inf. Model.* **2020**, *60*, 5308–5318.

(6) Lee, T.-S.; Allen, B. K.; Giese, T. J.; Guo, Z.; Li, P.; Lin, C.; Jr., T. D. M.; Pearlman, D. A.; Radak, B. K.; Tao, Y.; Tsai, H.-C.; Xu, H.; Sherman, W.; York, D. M. *J. Chem. Inf. Model.* **2020**, *60*, 5595–5623.

(7) Stone, J. E.; Phillips, J. C.; Freddolino, P. L.; Hardy, D. J.; Trabuco, L. G.; Schulten, K. *J. Comput. Chem.* **2007**, *28*, 2618–2640.

(8) Anderson, J. A.; Lorenz, C. D.; Travesset, A. *J. Comput. Phys.* **2008**, *227*, 5342–5359.

(9) Hardy, D. J.; Stone, J. E.; Schulten, K. *Parallel Computing* **2009**, *35*, 164–177.

(10) Harvey, M. J.; Giupponi, G.; Fabritiis, G. D. *J. Chem. Theory Comput.* **2009**, *5*, 1632–1639.

(11) Harvey, M. J.; De Fabritiis, G. *J. Chem. Theory Comput.* **2009**, *5*, 2371–2377.

(12) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. *J. Mol. Graphics Model.* **2010**, *29*, 116–125.

(13) Farber, R. M. *J. Mol. Graphics Model.* **2011**, *30*, 82–89.

(14) Göetz, A.; Williamson, M. J.; Xu, D.; Poole, D.; Le Grand, S.; Walker, R. C. *J. Chem. Theory Comput.* **2012**, *8*, 1542.

(15) Eastman, P. et al. *J. Chem. Theory Comput.* **2013**, *9*, 461–469.

(16) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. *J. Chem. Theory Comput.* **2013**, *9*, 3878–3888.

(17) Chipot, C. *WIREs Comput. Mol. Sci.* **2014**, *4*, 71–89.

(18) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. *PLoS Comput. Biol.* **2017**, *13*, 1005659–1005659.

(19) Lee, T.-S.; Hu, Y.; Sherborne, B.; Guo, Z.; York, D. M. *J. Chem. Theory Comput.* **2017**, *13*, 3077–3084.

(20) Mermelstein, D. J.; Lin, C.; Nelson, G.; Kretsch, R.; McCammon, J. A.; Walker, R. C. *J. Comput. Chem.* **2018**, *39*, 1354–1358.

(21) Giese, T. J.; York, D. M. *J. Chem. Theory Comput.* **2018**, *14*, 1564–1582.

(22) Lee, T.-S.; Cerutti, D. S.; Mermelstein, D.; Lin, C.; LeGrand, S.; Giese, T. J.; Roitberg, A.; Case, D. A.; Walker, R. C.; York, D. M. *J. Chem. Inf. Model.* **2018**, *58*, 2043–2050.

(23) Song, L. F.; Lee, T.-S.; Zhu, C.; York, D. M.; Merz, Jr., K. M. *J. Chem. Inf. Model.* **2019**, *59*, 3128–3135.

(24) Chipot, C., Pohorille, A., Eds. *Free Energy Calculations: Theory and Applications in Chemistry and Biology*; Springer Series in Chemical Physics; Springer: New York, 2007; Vol. 86.

(25) Mobley, D. L.; Dill, K. A. *Structure* **2009**, *17*, 489–498, 0969-2126.

(26) Christ, C.; Mark, A.; van Gunsteren, W. *J. Comput. Chem.* **2010**, *31*, 1569–1582.

(27) Foloppe, N.; Hubbard, R. *Curr. Med. Chem.* **2006**, *13*, 3583–3608.

(28) Pohorille, A.; Jarzynski, C.; Chipot, C. *J. Phys. Chem. B* **2010**, *114*, 10235–10253.

(29) Michel, J.; Essex, J. W. *J. Comput.-Aided Mol. Des.* **2010**, *24*, 639–658.

(30) Gallicchio, E.; Levy, R. M. *Curr. Opin. Struct. Biol.* **2011**, *21*, 161–166.

(31) Chodera, J.; Mobley, D.; Shirts, M.; Dixon, R.; Branson, K.; Pande, V. *Curr. Opin. Struct. Biol.* **2011**, *21*, 150–160.

(32) Mobley, D. L.; Klimovich, P. V. *J. Chem. Phys.* **2012**, *137*, 230901.

(33) Gumbart, J. C.; Roux, B.; Chipot, C. *J. Chem. Theory Comput.* **2013**, *9*, 794–802.

(34) Hansen, N.; van Gunsteren, W. F. *J. Chem. Theory Comput.* **2014**, *10*, 2632–2647.

(35) Cournia, Z.; Allen, B.; Sherman, W. *J. Chem. Inf. Model.* **2017**, *57*, 2911–2937.

(36) de Ruiter, A.; Oostenbrink, C. *Curr. Opin. Struct. Biol.* **2020**, *61*, 207–212.

(37) Klimovich, P. V.; Shirts, M. R.; Mobley, D. L. *J. Comput.-Aided Mol. Des.* **2015**, *29*, 397–411.

(38) Tan, Z.; Gallicchio, E.; Lapelosa, M.; Levy, R. M. *J. Chem. Phys.* **2012**, *136*, 144102.

(39) Shirts, M. R.; Chodera, J. D. *J. Chem. Phys.* **2008**, *129*, 124105.

(40) Ding, X.; Vilseck, J. Z.; Hayes, R. L.; Brooks, C. L. *J. Chem. Theory Comput.* **2017**, *13*, 2501–2510.

(41) Ding, X.; Vilseck, J. Z.; ; Brooks III, C. L. *J. Chem. Theory Comput.* **2019**, *15*, 799–802.

(42) Zhang, B. W.; Xia, J.; Tan, Z.; Levy, R. M. *J. Phys. Chem. Lett.* **2015**, *6*, 3834–3840.

(43) Cui, D.; Zhang, B. W.; Tan, Z.; Levy, R. M. *J. Chem. Theory Comput.* **2020**, *16*, 67–79.

(44) Wang, L.; Deng, Y.; Knight, J. L.; Wu, Y.; Kim, B.; Sherman, W.; Shelley, J. C.; Lin, T.; Abel, R. *J. Chem. Theory Comput.* **2013**, *9*, 1282–1293.

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

(46) Zwanzig, R. W. *J. Chem. Phys.* **1954**, *22*, 1420–1426.

(47) Kirkwood, J. G. *J. Chem. Phys.* **1935**, *3*, 300–313.

(48) Straatsma, T. P.; Berendsen, H. J. *J. Chem. Phys.* **1988**, *89*, 5876–5886.

(49) Jarzynski, C. *Phys. Rev. Lett.* **1997**, *78*, 2690–2693.

(50) Crooks, G. E. *Phys. Rev. E* **2000**, *61*, 2361–2366.

(51) Boresch, S.; Woodcock, H. L. *Mol. Phys.* **2017**, *115*, 1200–1213.

(52) Gapsys, V.; Michielssens, S.; Peters, J. H.; de Groot, B. L.; Leonov, H. *Methods Mol. Biol.* **2015**, *1215*, 173–209.

(53) Jeong, D.; Andricioaei, I. *J. Chem. Phys.* **2013**, *138*, 114110.

(54) Wei, D.; Song, Y.; Wang, F. *J. Chem. Phys.* **2011**, *134*, 184704.

(55) Kearns, F. L.; Hudson, P. S.; Woodcock, H. L.; Boresch, S. *J. Comput. Chem.* **2017**, *38*, 1376–1388.

(56) Suh, D.; Radak, B. K.; Chipot, C.; Roux, B. *J. Chem. Phys.* **2018**, *148*, 14101.

(57) Bennett, C. H. *J. Comput. Phys.* **1976**, *22*, 245–268.

(58) Torrie, G. M.; Valleau, J. P. *J. Comput. Phys.* **1977**, *23*, 187–199.

(59) Shirts, M. R.; Pande, V. S. *J Chem Phys* **2005**, *122*, 144107.

(60) Giese, T. J.; York, D. M. *J. Chem. Theory Comput.* **2021**, *17*, 1326–1336.

(61) Jiang, W.; Chipot, C.; Roux, B. *J. Chem. Inf. Model.* **2019**, *59*, 3794–3802.

(62) Beutler, T. C.; Mark, A. E.; René C. van Schaik and Paul R. Gerber and Wilfred F. van Gunsteren, *Chem. Phys. Lett.* **1994**, *222*, 529–539.

(63) Steinbrecher, T.; Mobley, D. L.; Case, D. A. *J. Chem. Phys.* **2007**, *127*, 214108.

(64) Steinbrecher, T.; Joung, I.; Case, D. A. *J. Comput. Chem.* **2011**, *32*, 3253–3263.

(65) Jorgensen, W. L.; Ravimohan, C. *J. Chem. Phys.* **1985**, *83*, 3050–3054.

(66) Boresch, S.; Karplus, M. *J. Phys. Chem. A* **1999**, *103*, 119–136.

(67) Boresch, S.; Karplus, M. *J. Phys. Chem. A* **1999**, *103*, 103–118.

(68) Boresch, S. *Mol. Simul.* **2002**, *28*, 13–37.

(69) Shobana, S.; Roux, B.; Andersen, O. S. *J. Phys. Chem. B* **2000**, *104*, 5179–5190.

(70) Tsai, H.-C.; Tao, Y.; Lee, T.-S.; Merz, K. M.; York, D. M. *J. Chem. Inf. Model.* **2020**, *60*, 5296–5300.

(71) Lee, T.-S.; Lin, Z.; Allen, B. K.; Lin, C.; Radak, B. K.; Tao, Y.; Tsai, H.-C.; Sherman, W.; York, D. M. *J. Chem. Theory Comput.* **2020**, *16*, 5512–5525.

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

(73) Gapsys, V.; Khabiri, M.; de Groot, B. L.; Freddolino, P. L. *J. Phys. Chem. B* **2020**, *124*, 1115–1123, PMID: 29741907.

(74) Pal, R. K.; Gallicchio, E. *J. Chem. Phys.* **2019**, *151*, 124116.

(75) Lee, Y.-K.; Parks, D. J.; Lu, T.; Thieu, T. V.; Markotan, T.; Pan, W.; McComsey, D. F.; Milkiewicz, K. L.; Crysler, C. S.; Ninan, N.; Abad, M. C.; Giardino, E. C.; Maryanoff, B. E.; Damiano, B. P.; Player, M. R. *J. Med. Chem.* **2008**, *51*, 282–297.

(76) Hritz, J.; Oostenbrink, C. *J. Chem. Phys.* **2008**, *128*, 144121.

(77) Roe, D. R.; Bergonzo, C.; Cheatham, T. E. *J. Phys. Chem. B* **2014**, *118*, 3543–3552.

(78) Yang, M.; Huang, J.; MacKerell, Jr., A. D. *J. Chem. Theory Comput.* **2015**, *11*, 2855–2867.

(79) He, P.; Zhang, B. W.; Arasteh, S.; Wang, L.; Abel, R.; Levy, R. M. *J. Phys. Chem. Lett.* **2018**, *9*, 4428–4435.

(80) Hansmann, U. H.; Okamoto, Y. *Curr. Opin. Struct. Biol.* **1999**, *9*, 177–183.

(81) Sugita, Y.; Okamoto, Y. *Chem. Phys. Lett.* **1999**, *314*, 141–151.

(82) Radak, B. K.; Romanus, M.; Gallicchio, E.; Lee, T.-S.; Weidner, O.; Deng, N.-J.; He, P.; Dai, W.; York, D. M.; Levy, R. M.; Jha, S. *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*; XSEDE '13; ACM: New York, NY, USA, 2013; Vol. 26; pp 26–26.

(83) Gallicchio, E.; Xia, J.; Flynn, W. F.; Zhang, B.; Samlalsingh, S.; Mentes, A.; Levy, R. M. *Comput. Phys. Commun.* **2015**, *0*, 236–246.

(84) Xia, J.; Flynn, W. F.; Gallicchio, E.; Zhang, B. W.; He, P.; Tan, Z.; Levy, R. M. *J. Comput. Chem.* **2015**, *36*, 1772–1785.

(85) Jo, S.; Jiang, W. *Comput. Phys. Commun.* **2015**, *197*, 304–311.

(86) Hahn, D. F.; König, G.; Hünenberger, P. H. *J. Chem. Theory Comput.* **2020**, *16*, 1630–1645.

(87) Jiang, W.; Roux, B. *J. Chem. Theory Comput.* **2010**, *6*, 2559–2565.

(88) Arrar, M.; de Oliveira, C. A. F.; Fajer, M.; Sinko, W.; McCammon, J. A. *J. Chem. Theory Comput.* **2013**, *9*, 18–23.

(89) Itoh, S. G.; Okumura, H. *J. Comput. Chem.* **2013**, *34*, 2493–2497.

(90) Lim, V. T.; Bayly, C. I.; Fusti-Molnar, L.; Mobley, D. L. *J. Chem. Inf. Model* **2019**, *59*, 1957–1964.

(91) Hardcastle, I. R. et al. *J. Med. Chem.* **2004**, *47*, 3710–3722.

(92) Wang, L. et al. *J. Am. Chem. Soc.* **2015**, *137*, 2695–2703.

(93) Shivakumar, D.; Williams, J.; Wu, Y.; Damm, W.; Shelley, J.; Sherman, W. *J. Chem. Theory Comput.* **2010**, *6*, 1509–1519.

(94) Wang, M.; Mei, Y.; Ryde, U. *J. Chem. Theory Comput.* **2019**, *15*, 2659–2671.

(95) Abel, R.; Wang, L.; Mobley, D. L.; Friesner, R. A. *Curr. Top. Med. Chem.* **2017**, *17*, 2577–2585.

(96) de Ruiter, A.; Boresch, S.; Oostenbrink, C. *J. Comput. Chem.* **2013**, *34*, 1024–1034.

(97) Pickard, F. C.; König, G.; Simmonett, A. C.; Shao, Y.; Brooks, B. R. *Bioorg. Med. Chem.* **2016**, *24*, 4988–4997.

(98) Yang, Q.; Burchett, W.; Steeno, G. S.; Liu, S.; Yang, M.; Mobley, D. L.; Hou, X. *J. Comput. Chem.* **2020**, *41*, 247–257.

(99) König, G.; Brooks, B. R.; Thiel, W.; York, D. M. *Mol. Simul.* **2018**, *44*, 1062–1081.

(100) Li, Y.; Nam, K. *J. Chem. Theory Comput.* **2020**, *16*, 4776–4789.

(101) Liu, S.; Wu, Y.; Lin, T.; Abel, R.; Redmann, J. P.; Summa, C. M.; Jaber, V. R.; Lim, N. M.; Mobley, D. L. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 755–770.

(102) Loeffler, H. H.; Michel, J.; Woods, C. *J. Chem. Inf. Model.* **2015**, *55*, 2485–2490.

(103) Klimovich, P. V.; Mobley, D. L. *J. Comput.-Aided Mol. Des.* **2015**, *29*, 1007–1014.

(104) Bruckner, S.; Boresch, S. *J. Comput. Chem.* **2011**, *32*, 1303–1319.

(105) Gapsys, V.; Michielssens, S.; Seeliger, D.; de Groot, B. L. *J. Comput. Chem.* **2015**, *36*, 348–354.

(106) Giese, T. J.; Şölen Ekesan,; York, D. M. *J. Phys. Chem. A, in press* **2021**,

(107) Goldstein, D. M. et al. *J. Med. Chem.* **2011**, *54*, 2255–2265.

(108) Schindler, C. E. M. et al. *J. Chem. Inf. Model.* **2020**, *60*, 5457–5474.

(109) Mey, A. S. J. S.; Allen, B. K.; Macdonald, H. E. B.; Chodera, J. D.; Hahn, D. F.; Kuhn, M.; Michel, J.; Mobley, D. L.; Naden, L. N.; Prasad, S.; Rizzi, A.; Scheen, J.; Shirts, M. R.; Tresadern, G.; Xu, H. *Living Journal of Computational Molecular Science* **2020**, *2*, 18378–18429.

(110) Homeyer, N.; Gohlke, H. *J. Comput. Chem.* **2013**, *34*, 965–973.

(111) Homeyer, N.; Gohlke, H. *Biochim. Biophys. Acta* **2015**, *1850*, 972–982.

(112) Gathiaka, S.; Liu, S.; Chiu, M.; Yang, H.; Stuckey, J. A.; Kang, Y. N.; Delproposto, J.; Kubish, G.; Dunbar, J. B.; Carlson, H. A.; Burley, S. K.;

Walters, W. P.; Amaro, R. E.; Feher, V. A.; Gilson, M. K. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 651–668.

(113) Kuhn, M.; Firth-Clark, S.; Tosco, P.; Mey, A. S. J. S.; Mackey, M.; Michel, J. *J. Chem. Inf. Model.* **2020**, *60*, 3120–3130.

(114) Raymond, J. W.; Gardiner, E. J.; Willett, P. *Comput J* **2002**, *45*, 631–644.

(115) http://www.rdkit.org, RDKit: Open-source cheminformatics.

(116) Giese, T. J.; Chen, H.; Huang, M.; York, D. M. *J. Chem. Theory Comput.* **2014**, *10*, 1086–1098.

(117) Giese, T. J.; Huang, M.; Chen, H.; York, D. M. *Acc. Chem. Res.* **2014**, *47*, 2812–20.

(118) Giese, T. J.; York, D. M. *J. Phys. Condens. Matter* **2017**, *29*, 383002.

(119) Zhang, L.; Lin, D.-Y.; Wang, H.; Car, R.; E, W. *Phys. Rev. Materials* **2019**, *3*, 23804.

(120) Zhang, Y.; Wang, H.; Chen, W.; Zeng, J.; Zhang, L.; Han, W.; E, W. *Comput. Phys. Commun.* **2020**, *253*, 107206.

(121) Giese, T. J.; York, D. M. *J. Chem. Theory Comput.* **2019**, *15*, 5543–5562.

(122) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. *J. Chem. Theory Comput.* **2015**, *11*, 3696–3713.

(123) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. *J. Comput. Chem.* **2004**, *25*, 1157–1174.

(124) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926–935.

(125) He, X.; Man, V. H.; Yang, W.; Lee, T.-S.; Wang, J. *J. Chem. Phys.* **2020**, *153*, 114502.

(126) Horn, H. W.; Swope, W. C.; Pitera, J. W.; Madura, J. D.; Dick, T. J.; Hura, G. L.; Head-Gordon, T. *J. Chem. Phys.* **2004**, *120*, 9665–9678.

(127) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Dinola, A.; Haak, J. R. *J. Chem. Phys.* **1984**, *81*, 3684–3690.

(128) Loncharich, R. J.; Brooks, B. R.; Pastor, R. W. *Biopolymers* **1992**, *32*, 523–535.

(129) Darden, T.; York, D.; Pedersen, L. *J. Chem. Phys.* **1993**, *98*, 10089–10092.

(130) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Hsing, L.; Pedersen, L. G. *J. Chem. Phys.* **1995**, *103*, 8577–8593.