# GPU-accelerated molecular dynamics and free energy methods in Amber18: performance enhancements and new features

**Tai-Sung Lee**[†], **David S. Cerutti**[†], **Dan Mermelstein**[‡], **Charles Lin**[‡], **Scott LeGrand**[¶], **Timothy J. Giese**[†], **Adrian Roitberg**[§], **David A. Case**[†], **Ross C. Walker**[*,‖], 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

[‡]Department of Chemistry & Biochemistry, University of California SanDiego, La Jolla, California 92093, USA

[¶]A9.com, Palo Alto, CA 94301

[§]Department of Chemistry, University of Florida, Gainesville, FL 32611, USA

[‖]GlaxoSmithKline PLC, 1250 S. Collegeville Rd, Collegeville, Pennsylvania 19426, USA

## Abstract

We report progress in GPU-accelerated molecular dynamics and free energy methods in Amber18. Of particular interest is the development of alchemical free energy algorithms, including free energy perturbation and thermodynamic integration methods with support for non-linear softcore potential and parameter interpolation transformation pathways. These methods can be used in conjunction with enhanced sampling techniques such as replica exchange, constant pH molecular dynamics and new 12-6-4 potentials for metal ions. Additional performance enhancements have been made that enable appreciable speed-up on GPUs relative to the previous software release.

## Graphical Abstract

[*]To whom correspondence should be addressed: ross@rosswalker.co.uk; Darrin.York@rutgers.edu.

Software Availability

Amber18 is available for download from the AMBER home page: http://ambermd.org/.

## Introduction

Molecular simulation provides an extremely powerful tool for the interpretation of experimental data, the understanding of biomolecular systems, and the prediction of properties important for molecular design. As the scope and scale of applications increase, so must the compute capability of molecular simulation software. The last decade has seen rapid advances motivated by the performance enhancements offered by the latest molecular dynamics software written for specialized hardware. Perhaps the most affordable and impactful of these are platforms using graphics processing units (GPUs).[1–7]

The present application note reports on advances made in the latest release of the AMBER molecular simulation software suite (Amber18),[8] and in particular, enhancements to the primary GPU-accelerated simulation engine (PMEMD). These advancements significantly improve the program's execution of molecular simulations, and offer new, integrated features for calculating alchemical free energies[9,10] including Thermodynamic Integration (TI),[11–15] Free Energy Perturbation FEP)[15–19] and Bennett's acceptance ratio and its variants (BAR/MBAR),[20–25] as well as enhanced sampling, constant pH simulation[26,27] and use of new 12-6-4 potentials.[28–30] Amber18 offers a broad solution for a wide range of free energy simulations, with expanded capability to compute forces in a hybrid environment of CPUs and GPUs, and establishes an add-on system for applying additional potential functions computed on the graphics processor without affecting optimizations of the main kernels. When run exclusively on GPU hardware, Amber18 shows consistent performance increases of 10% to 20% compared to Amber16 across Pascal (GTX-1080TI, Titan-XP, P100) and Volta architectures when running standard MD simulations, with more dependence on system size than architecture. Below we provide an overview of the software design, a description of new features, and performance benchmarks.

## Software Design

### Encapsulated free energy modules

The development of molecular simulation software designed for optimal performance on specialized hardware requires customization and careful redesign of the underlying

algorithmic implementation. In the case of the current GPU consumer market, single precision floating point performance outstrips that of double precision performance by a significant amount. In order to address this issue in AMBER, new precision models[6,31] have been developed that leverage fixed point integer arithmetic to replace slow double precision arithmetic in certain key steps when higher precision is required, such as the accumulation of components of the force. Free energy simulations, due to the mixing/interpolation of hybrid Hamiltonians and the averaging of TI and FEP quantities, present new challenges with respect to precision compared with conventional MD simulation. The GPU enhanced features in Amber18 were designed to address these different precision requirements to ensure statistical and thermodynamically derived properties are indistinguishable from the CPU version of the code[32] while maintaining or improving the level of performance of previous versions of AMBER for conventional MD simulations. To fulfill these goals, we utilized two architectural concepts of object-oriented programming: encapsulation and inheritance.[33] The original AMBER GPU data structures are encapsulated into base C++ classes, containing all coordinates, forces, energy terms, all simulation parameters and settings. New free energy classes are derived from the base classes that contain the original GPU functionality and data structures for MD simulations. New derived free energy classes inherit all the properties and methods of existing MD classes. Through encapsulation and inheritance, free energy capability can be implemented so that: 1. there is little or no need to modify the original MD GPU codes except they are wrapped into base classes now, since new add-ons can be implemented in the derived free energy classes; 2. the new speci c free energy algorithms and associated data structures are transparent to the base classes such that modifying or optimizing the base classes will have minimal effects on the derived classes; 3. derived free energy classes can utilize different algorithms, different precision models, and even different force fields.

Such encapsulation and inheritance approach, on the other hand, could introduce additional computational overhead, compared to direct modification of MD GPU kernels so that similar computational tasks are executed within the same kernels. Consequently the reported approach here will be ideal for new modules where only small portions of calculations needed to be altered, such as TI; while direct modification of MD GPU kernels will be suitable for situations where algorithms changes are global, e.g., incorporation of polarizable force fields.

### Extensions of new modules

The present software design can be easily extended to accommodate implementation of new methods or algorithms. For example, the 12-6-4 potential modules have been implemented using this framework by treating the extra $r^{-4}$ terms through add-on modules with minimal, if any, modification of MD CPU codes and GPU kernels.

For most developers, the CPU code remains the most accessible means for prototyping new methods. In select cases, where complex potentials are applied to only a small number of atoms, the CPU may afford performance advantages for development code that have to be fully optimized. To serve these needs, AMBER18 has a new "atom shuttling" system, which extracts information on a pre-defined list of selected atoms and transfers it between the host

CPU memory and GPU device. In previous versions, all coordinates, forces, charges, and other data can be downloaded and uploaded between the host and device, at costs approaching 30% of the typical time step. When the majority of the system's atoms will not influence the CPU computations, this is wasteful. The shuttle system accepts a pre-defined list of atoms and collects them into a buffer for a lighter communication requirement between the host and device. The cost of organizing the atoms into the list is minor, but the extra kernel calls to manage the list and the latency of initiating the transfer are considerations. In the limit of transferring very small numbers of atoms, the cost of the shuttle can be less than 1% of the total simulation time, but methods that require transferring the majority of atom data may be more efficient porting entire arrays with the methods in Amber16.

## Features

The current Amber18 has a host of features available that work together to perform MD and free energy simulations (Table 1). A brief description of the most relevant features are provided below.

### Direct implementation of alchemical free energy methods

Alchemical free energy simulations[9,10] provide accurate and robust estimates of relative free energies from molecular dynamics simulations,[10,15,34–38] but are computationally intensive, and are often limited by the availability of computing resources and/or required turn-around time. The limitations can render these methods impractical, particularly for industrial applications.[39] GPU-accelerated alchemical free energy methods change this landscape, but have only recently emerged in a few simulation codes. The free energy methods implemented in the Amber18 GPU code build on the efficient AMBER GPU MD code base (pmemd.cuda), and include both thermodynamic integration and free energy perturbation classes.

Thermodynamic Integration (TI):[11–15] The free energy change from state 0 to state 1, $A_{0\rightarrow1}$, is approximately calculated by numerical integration of the derivative of the system potential energy $U$ with respect to the target parameter $\lambda$:

$$\Delta A_{0\rightarrow1} = \int_0^1 \left\langle \frac{dU(\lambda, \vec{q})}{d\lambda} \right\rangle_\lambda d\lambda. \quad (1)$$

Free Energy Perturbation (FEP):[15–19] The free energy change between state 0 and 1, $A_{0\rightarrow1}$ is calculated by averaging the exponential of the potential energy differences sampled on the potential surface of state 0 (the Zwanzig equation[16]):

$$\Delta A_{0\rightarrow1} = \beta^{-1}\ln\left\langle e^{-\beta\left(U_1(\vec{q}) - U_0(\vec{q})\right)} \right\rangle_0 = \beta^{-1}\ln\left\langle e^{-\beta\Delta U(\vec{q})} \right\rangle_0 \quad (2)$$

The quantities calculated from FEP can be output for post-analysis through Bennett's acceptance ratio and its variants (BAR/MBAR).[20–25]

Both TI and FEP methods can be used for linear alchemical transformations, non-linear "parameter-interpolated" pathways[40] and so-called "softcore"[41–43] schemes for both van der Waals and electrostatic interactions. All above are available in the current Amber18 GPU release by utilizing the same input formats as the CPU version.

The GPU free energy implementation has been demonstrated to deliver speed-ups are generally significantly more than one order of magnitude when comparing a single GPU to a comparably priced single (multi-core) microprocessor (see Performance section for detailed benchmarks). The GPU free energy implementation code performs TI with linear alchemical transformations roughly at the speed of 70 % of running an MD simulation with the fast SPFP precision mode,[31] similar to the ratios seen in the CPU counterpart.[32] Overall, the free energy simulation speed-up relative to the CPU code is very similar to that for conventional MD simulation. As will be discussed in the next section, in certain instances the overhead (relative to conventional MD) for the linear alchemical free energy simulations can be greatly reduced by use of non-linear parameter-interpolated TI.[40]

## Parameter-interpolated thermodynamic integration

Amber18 is also able to exploit the properties of a parameter-interpolated thermodynamic integration (PI-TI) method,[40] which has recently been extended to support PME electrostatics, to connect states by their molecular mechanical parameter values. This method has the practical advantage that no modification to the MD code is required to propagate the dynamics, and unlike with linear alchemical mixing, only one electrostatic evaluation is needed (e.g., single call to particle-mesh Ewald). In the case of Amber18, this enables all the performance benefits of GPU-acceleration to be realized, in addition to unlocking the full spectrum of features available within the MD software. The TI evaluation can be accomplished in a post-processing step by reanalyzing the statistically independent trajectory frames in parallel for high throughput. Additional tools to streamline the computational pipeline for free energy post-processing and analysis are forthcoming.

## Replica-Exchange Molecular Dynamics

During the past two decades, the replica exchange methods[44,45] have become popular in over‑coming the multiple-minima problem by exchanging non-interacting replicas of the system at different conditions. The original replica exchange methods were applied on systems at several temperatures,[44] and have been extended to various conditions, such as Hamiltonian,[46] pH,[47] and redox potentials. Amber18 is capable of performing temperature, Hamiltonian, and pH replica exchange simulations using GPU. Hamiltonian replica exchange can be con-figured in a flexible way, as long as the "force field" (or equivalently the prmtop file) is properly defined for each replica. Hence, the newly implemented free energy methods in Amber18 can be performed as Hamiltonian replica exchange so that different $\lambda$ windows can exchange their conformations. Other types of Hamiltonian replica exchange simulations, such as Hamiltonian tempering or umbrella sampling, can be easily setup as well.

Multiple dimension replica exchange simulations[48–53] where two or more conditions are simulated at the same time, are supported as well. By utilizing the multi-dimensional replica exchange capability, many practical combinations are possible to enhance sampling: such as TI simulation combined with temperature or pH replica exchange.

The con guration of GPU's in Amber18 replica exchange simulations is very flexible in order to fit various types of computational resources. Ideally for load balancing, the number of replicas should be an integer multiple (typically 1–6) of the number of available GPU's. One GPU can run one or multiple replicas given sufficient GPU memory is available, although one can expect some slow-down in the cases of multiple tasks running concurrently on a single GPU. Our experience shows that a 11GB GTX 1080TI GPU can handle six instances of typical kinase systems (around 30,000 to 50,000 atoms) without losing efficiency. One scenario is that to run free energy simulations with replica exchange on one multiple GPU node, e.g., executing 12 $\lambda$-windows on a 4-GPU or 6-GPU node with each GPU handling 3 or 2 $\lambda$-windows. Such scenarios take advantage of extremely fast intra-node communication and enable efficient performance optimization on modern large scale GPU clusters/supercomputers such Summit at Oak Ridge National Laboratory. In principle, a single replica can also be run in parallel on multiple GPUs but this is strongly discouraged as Amber18 is not optimized for this.

### Constant pH Molecular Dynamics

Conventional, all-atom molecular simulations consider ensembles constrained to have predetermined fixed protonation states that are not necessarily consistent with any pH value. Constant pH molecular dynamics (CpHMD) is a technique that enables sampling of different accessible protonation states (including different relevant tautomers) consistent with a bulk pH value.[26,27] These methods have been applied to a wide array of biological problems, including prediction of p$K_a$ shifts in proteins and nucleic acids, pH-dependent conformational changes, assembly and protein-ligand, protein-protein and protein-nucleic acid binding events.[54] These methods provide detailed information about the conditional probability of observing correlated protonation events that have biological implications. Very recently, a discrete protonation state constant pH molecular dynamics method has been implemented on GPUs, integrated with REMD methods (including along a pH dimension) and tested in AMBER.[55] The method has been applied for the first time to the interpretation of activity-pH pro les in a mechanistic computational enzymology study of the archetype enzyme RNase A.[56] The CpHMD method in Amber18 is compatible with enhanced sampling methods such as REMD, and is compatible with the new GPU-accelerated free energy framework.

The workflow of explicit Solvent CpHMD simulation has been described in detail elsewhere.[55] Briefly, the method follows the general approach of Baptista and co-workers[26,27] that involves sampling of discreet protonation states using a Monte Carlo sampling procedure. Simulations are performed in explicit solvent under periodic boundary conditions using PME to generate ensembles. The CpHMD method utilizes an extended force field that contains parameters (typically charge vectors) associated with changes in protonation state and reference chemical potentials for each titratable site calibrated for a

selected GB model to obtain correct p$K_a$ values in solution. In the Monte Carlo decision to accept or reject a trial protonation state, explicit solvent (including any nonstructural ions) is stripped and replaced using the selected GB model under non-periodic boundary conditions. Additional considerations are made for multi-site titration involving titratable residues that are considered "neighbors".[55] If any protonation state change attempts are accepted, the explicit solvent is replaced, the solute is frozen, and MD is used to relax the solvent degrees of freedom for a short period of time. After relaxation is complete, the velocities of the solute atoms are restored to their prior values and standard dynamics resumes. Full details can be found in reference.[55]

### 12-6-4 Potentials for Metal Ions

The GPU version of Amber18 (pmemd.cuda) is capable of utilizing 12-6-4 potentials, developed by Li et. al.[28] for metal ions in aqueous solution, and recently extended for $Mg^{2+}$, $Mn^{2+}$, $Zn^{2+}$ and $Cd^{2+}$ ions so as to have balanced interactions with nucleic acids.[30] The 12-6-4 potentials are derived from regular 12–6 potentials by adding $r^{-4}$ terms, which has been proposed by Roux and Karplus.[57,58]

The 12–6 potential[59] for non-bonded interactions is:

$$U_{ij}\left(r_{ij}\right) = \epsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{R_{ij}}{r_{ij}}\right)^{6}\right] \quad (3)$$

where the parameters $R_{ij}$ and $\epsilon_{ij}$ are the combined radius and well depth for the pairwise interaction and $r_{ij}$ is the distance between the particles. Equation (3) can be expressed equivalently as:

$$U_{ij}\left(r_{ij}\right) = \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \quad (4)$$

where $A_{ij} = \epsilon_{ij}R_{ij}^{12}$ and $B_{ij} = 2\epsilon_{ij}R_{ij}^{6}$.

The expanded 12-6-4 potential[60] is then:

$$U_{ij}\left(r_{ij}\right) = \epsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{R_{ij}}{r_{ij}}\right)^{6} - 2\kappa R_{ij}^{2}\left(\frac{R_{ij}}{r_{ij}}\right)^{4}\right] \quad (5)$$

$$= \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} - \frac{C_{ij}}{r_{ij}^{4}}$$

where $C_{ij}$ is equivalent to $B_{ij}\kappa$, and $\kappa$ is a scaling parameter with units of Å$^{-2}$. The additional attractive term, $\frac{C_{ij}}{r_{ij}^{4}}$, implicitly accounts for polarization effects by mimicking the

charge-induced dipole interaction. The 12-6-4 potentials have showed a marked improvement over the 12–6 LJ nonbonded model.[28,30]

### Future Plan

The Amber18 development roadmap will extend sampling capabilites for free energy simulations to facilitate advancement of drug discovery,[61] including: implementation of Gibbs sampling scheme[62] to improve the exchange rates in replica exchange simulations, the self-adjusted mixture sampling (SAMS)[63] to optimize the simulation lengths for different lambda windows, and replica exchange with solute scaling (REST2),[64] a more efficient version of replica exchange with solute tempering[65] to stably and efficiently perform "effective" solute tempering replica exchange simulations.

## Performance

Amber18 runs efficiently on GPU platforms for both MD and free energy simulations. Performance benchmarks for equilibrium MD are shown in Figure 1 and listed for selected GPUs in Table 2, including comparisons with Amber16. The figure works in a particle-normalized metric, trillions of atom-time steps per day, which puts most systems on equal footing and shows performance improvements of up to 24%, without implying improper comparisons to other codes (the cutoffs used in these benchmarks are smaller than some other benchmarks, and other settings may not be comparable). Longer time steps, if safe, tend to improve overall throughput at a marginal increase in the cost of computing each step (requiring more frequent pair list updates). Small systems tend to perform less efficiently (small FFTs and pair list building kernels do not fully occupy the GPU). Virial computations are also costly, as seen for the Satellite Tobacco Mosaic Virus (STMV) system, the only one of this abbreviated list of benchmarks to include pressure regulation with a Berendsen barostat.

Using a GTX-1080Ti, still the most cost-effective GPU at the time of publication, the 23,558 atom Dihydrofolate Reductase (DHFR) benchmark (4 fs time step, constant energy) runs at 657 ns/day in Amber18 and 588 ns/day in Amber16. The same codes run the 90,906 atom blood clotting Factor IX system at 100 and 89 ns/day, respectively, with a 2 fs time step. Performance in thermodynamic integration free energy simulations for mutating ligands of the clotting Factor Xa system is shown in Figure 2. TI with linear alchemical mixing generally exacts a toll of 1/3 the speed that could be achieved in a conventional MD simulation. Additional pairwise computations between particles are present, but the secondary reciprocal space calculation is about 85% of the additional cost (this cost is eliminated in the PITI method[40]). The main performance improvements derive from: 1. innovative spline tabulation look-up and particle mapping kernels for faster PME direct and reciprocal space calculations, and 2. more efficient memory access for bonded and non-bonded terms.

### Faster PME direct and reciprocal space calculations.

Most CPU codes use a quadratic or cubic spline for the derivative of the complementary error function used in the PME direct space energy term. Rather than costly conversion of

floating point values into integer indexes for table look-ups, we take the IEEE-754 representation of the 32-bit floating point number for the squared distance and use its high fourteen bits, isolated by interpreting it as an unsigned integer and shifting right 18 bits, as an integer index into a logarithmically coarsening look-up table. This approach uses a minimum of the precious streaming multiprocessor (SMP) cache, collects a huge number of arithmetic operations into a single cubic spline evaluation, and typically leads to a 6–8% speed-up. The workflow of the non-bonded kernel was further improved by eliminating _shared_ memory storage and dealing with all particle comparisons within the same warp via _shfl instructions. This permitted us to engage not just 768 but 1280 threads on each SMP.

### PME reciprocal space.

We have made improvements to the kernel that maps particles onto the 3DFFT mesh by parallel computation of *B*-spline coefficients for all three dimensions (utilizing 90 out of 96 threads in the block rather than less than 1/3 of them) and re-tuning the stencil for writing data onto the mesh to make better coalesced atomic transactions. This improves the throughput of the mapping kernel by more than 40%, and typically leads to a few percent speed-up overall.

### More efficient memory access for bonded and non-bonded terms.

Rather than reach into global memory for the coordinates of each individual atom needed by any bonded term, we draw groups of topologically connected atoms at the start of the simulation and assign bond and angle terms to operate on the atoms of these groups. At each step of the simulation, the coordinates of each group are cached on the SMP and forces due to their bonded interactions are accumulated in _shared_ memory. Last, the results are dumped back to global via atomic transactions, reducing the global reads and writes due to bonded interactions more than ten-fold. The approach generalizes one described ten years ago for the Folding@Home client,[66] where bond and angle computations were computed by the threads that had already downloaded the atoms for a dihedral computation. Our approach makes much larger groups of atoms (up to 128) and does not compute redundant interactions. However, the block-wide synchronization after reading coordinates and prior to writing results may leave threads idle. The modular programming that creates our networks of interactions facilitates combining or partitioning the GPU kernels to optimize register usage and thread occupancy.

We have also gained a considerable amount of improvement by trimming the precision model where low significant bits are wasted. Rather than convert every non-bonded force to 64-bit integers immediately, we accumulate forces from 512 interactions (evaluated sequentially in sets of 16 by each of 32 threads in a warp) before converting the sums to integer and ultimately committing the result back to global memory. Because the tile scheme in our non-bonded kernel remains warp-synchronous, the sequence of floating-point operations that evaluates the force on each atom is identical regardless of the order the tile was called. In other words, each tile evaluation is immune from race conditions. Conversions to integer arithmetic always occur, as in Amber16, before combining the results of separate warps, as the coordination of different warps is not guaranteed. These optimizations

therefore maintain the numerical determinism of the code: a given GPU will produce the identical answers for a given system and input parameters.

### Minimal computational load on CPU and GPU/CPU intercommunication.

As it does for MD, Amber18 performs free energy computations almost entirely on the GPU and requires very little communication between the CPU and the GPU. This results in a tremendous practical advantage over other implementations in that independent or loosely coupled simulations (e.g., different lambda windows of a TI or FEP, possibly with REMD) can be run efficiently in parallel on cost-effective nodes that contain multiple GPUs with a single (possibly low-end) CPU managing them all without loss of performance. This is a critical design feature that distinguishes Amber18 free energy simulations from other packages that may require multiple high end CPU cores to support each GPU for standard dynamics and free energy calculations. The result is an implementation of TI/FEP that is not only one of the fastest available but also the most cost effective when hardware costs are factored in.

## Conclusion

In this application note, we report new features and performance benchmarks for the Amber18 software official release. The code is able to perform GPU-accelerated alchemical free energy perturbation and thermodynamic integration highly efficiently on a wide range of GPU hardware. The free energy perturbation simulations output metadata that can be analyzed using conventional or multistate Bennett's acceptance ratio methods. Additionally, thermodynamic integration capability is enabled for linear alchemical transformations, and non-linear transformations including softcore potentials and parameter-interpolated TI methods recently extended for efficient use with particle mesh Ewald electrostatics. These free energy methods can be used in conjunction with a wide range of enhanced sampling methods, constant pH molecular dynamics, and new 12-6-4 potentials for metal ions. The Amber18 software package provides a rich set of high-performance GPU-accelerated features that enable a wide range of molecular simulation applications from computational molecular biophysics to drug discovery.

## Acknowledgement

## References

(1). Stone JE; Phillips JC; Freddolino PL; Hardy DJ; Trabuco LG; Schulten K Accelerating molecular modeling applications with graphics processors. J. Comput. Chem 2007, 28, 2618–2640. [PubMed: 17894371]

(2). Anderson JA; Lorenz CD; Travesset A General purpose molecular dynamics simulations fully implemented on graphics processing units. J. Comput. Phys 2008, 227, 5342–5359.

(3). Harvey MJ; Giupponi G; Fabritiis GD ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. J. Chem. Theory Comput 2009, 5, 1632–1639. [PubMed: 26609855]

(4). Stone JE; Hardy DJ; Ufimtsev IS; Schulten K GPU-accelerated molecular modeling coming of age. J. Mol. Graphics Model 2010, 29, 116–125.

(5). Xu D; Williamson MJ; Walker RC Chapter 1 - Advancements in Molecular Dynamics Simulations of Biomolecules on Graphical Processing Units. Annu. Rev. Comput. Chem 2010, 6, 2–19.

(6). Salomon-Ferrer R; Götz AW; Poole D; Le Grand S; Walker RC Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent Particle Mesh Ewald. J. Chem. Theory Comput 2013, 9, 3878–3888. [PubMed: 26592383]

(7). Eastman P; Swails J; Chodera JD; McGibbon RT; Zhao Y; Beauchamp KA; Wang L-P; Simmonett AC; Harrigan MP; Stern CD; Wiewiora RP; Brooks BR; Pande VS OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. PLoS Comput. Biol 2017, 13, 1005659–1005659.

(8). Case DA et al. AMBER 18. University of California, San Francisco: San Francisco, CA, 2018.

(9). Straatsma TP; McCammon JA Computational alchemy. Annu. Rev. Phys. Chem 1992, 43, 407–435.

(10). 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.

(11). Kirkwood JG Statistical mechanics of fluid mixtures. J. Chem. Phys 1935, 3, 300–313.

(12). Straatsma TP; Berendsen HJC; Postma JPM Free energy of hydrophobic hydration: A molecular dynamics study of noble gases in water. J. Chem. Phys 1986, 85, 6720–6727.

(13). Straatsma TP; Berendsen HJ Free energy of ionic hydration: Analysis of a thermodynamic integration technique to evaluate free energy differences by molecular dynamics simulations. J. Chem. Phys 1988, 89, 5876–5886.

(14). Straatsma TP; McCammon JA Multiconfiguration thermodynamic integration. J. Chem. Phys 1991, 95, 1175–1188.

(15). Shirts MR; Pande VS Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. J Chem Phys 2005, 122, 144107. [PubMed: 15847516]

(16). Zwanzig RW High-temperature equation of state by a perturbation method. I. Non-polar gases. J. Chem. Phys 1954, 22, 1420–1426.

(17). Torrie GM; Valleau JP Monte Carlo free energy estimates using non-Boltzmann sampling: Application to the sub-critical Lennard-Jones fluid. Chem. Phys. Lett 1974, 28, 578–581.

(18). Jarzynski C Equilibrium free-energy differences from nonequilibrium measurements: a master-equation approach. Phys. Rev. E 1997, 56, 5018–5035.

(19). Lu N; Kofke DA Accuracy of free-energy perturbation calculations in molecular simulation. II. Heuristics. J. Chem. Phys 2001, 115, 6866–6875.

(20). Bennett CH Efficient estimation of free energy differences from Monte Carlo data. J. Comput. Phys 1976, 22, 245–268.

(21). Shirts MR; Bair E; Hooker G; Pande VS Equilibrium free energies from nonequilibrium measurements using maximum-likelihood methods. Phys. Rev. Lett 2003, 91, 140601. [PubMed: 14611511]

(22). Shirts MR; Chodera JD Statistically optimal analysis of samples from multiple equilibrium states. J. Chem. Phys 2008, 129, 124105. [PubMed: 19045004]

(23). König G; Boresch S Non-Boltzmann sampling and Bennett's acceptance ratio method: how to profit from bending the rules. J. Comput. Chem 2011, 32, 1082–1090. [PubMed: 21387335]

(24). Mikulskis P; Genheden S; Ryde U A large-scale test of free-energy simulation estimates of protein-ligand binding affinities. J. Chem. Inf. Model 2014, 54, 2794–2806. [PubMed: 25264937]

(25). Klimovich PV; Shirts MR; Mobley DL Guidelines for the analysis of free energy calculations. J. Comput.-Aided Mol. Des 2015, 29, 397–411. [PubMed: 25808134]

(26). Baptista AM; Teixeira VH; Soares CM Constant-pH molecular dynamics using stochastic titration. J. Chem. Phys 2002, 117, 4184–4200.

(27). Chen J; Brooks III CL; Khandogin J Recent advances in implicit solvent-based methods for biomolecular simulations. Curr. Opin. Struct. Biol 2008, 18, 140–148. [PubMed: 18304802]

(28). Li P; Roberts BP; Chakravorty DK; Merz KM, Jr. Rational design of Particle Mesh Ewald compatible Lennard-Jones parameters for +2 metal cations in explicit solvent. J. Chem. Theory Comput 2013, 9, 2733–2748. [PubMed: 23914143]

(29). Panteva MT; Giamba u GM; York DM Comparison of structural, thermodynamic, kinetic and mass transport properties of $Mg^{2+}$ ion models commonly used in biomolecular simulations. J. Comput. Chem 2015, 36, 970–982. [PubMed: 25736394]

(30). Panteva MT; Giambasu GM; York DM Force field for $Mg^{2+}$, $Mn^{2+}$, $Zn^{2+}$, and $Cd^{2+}$ ions that have balanced interactions with nucleic acids. J. Phys. Chem. B 2015, 119, 15460–15470. [PubMed: 26583536]

(31). Le Grand S; Göetz AW; Walker RC SPFP: Speed without compromise–A mixed precision model for GPU accelerated molecular dynamics simulations. Comput. Phys. Commun 2013, 184, 374–380.

(32). Lee T-S; Hu Y; Sherborne B; Guo Z; York DM Toward Fast and Accurate Binding Affinity Prediction with pmemdGTI: An Efficient Implementation of GPU-Accelerated Thermodynamic Integration. J. Chem. Theory Comput 2017, 13, 3077–3084. [PubMed: 28618232]

(33). Scott ML Programming Language Pragmatics; Morgan Kaufmann Publishers Inc: San Francisco, CA, USA, 2000.

(34). Chodera J; Mobley D; Shirts M; Dixon R; Branson K; Pande V Alchemical free energy methods for drug discovery: progress and challenges. Curr. Opin. Struct. Biol 2011, 21, 150–160. [PubMed: 21349700]

(35). Gallicchio E; Levy RM Advances in all atom sampling methods for modeling protein-ligand binding affinities. Curr. Opin. Struct. Biol 2011, 21, 161–166. [PubMed: 21339062]

(36). Bruckner S; Boresch S Efficiency of alchemical free energy simulations. I. A practical comparison of the exponential formula, thermodynamic integration, and Bennett's acceptance ratio method. J. Comput. Chem 2011, 32, 1303–1319. [PubMed: 21425288]

(37). Hansen N; van Gunsteren WF Practical Aspects of Free-Energy Calculations: A Review. J. Chem. Theory Comput 2014, 10, 2632–2647. [PubMed: 26586503]

(38). Homeyer N; Stoll F; Hillisch A; Gohlke H Binding Free Energy Calculations for Lead Optimization: Assessment of Their Accuracy in an Industrial Drug Design Context. J. Chem. Theory Comput 2014, 10, 3331–3344. [PubMed: 26588302]

(39). Chipot C; Rozanska X; Dixit SB Can free energy calculations be fast and accurate at the same time? Binding of low-affinity, non-peptide inhibitors to the SH2 domain of the src protein. J. Comput.-Aided Mol. Des 2005, 19, 765–770. [PubMed: 16365699]

(40). Giese TJ; York DM A GPU-Accelerated Parameter Interpolation Thermodynamic Integration Free Energy Method. J. Chem. Theory Comput 2018, 14, 1564–1582. [PubMed: 29357243]

(41). Steinbrecher T; Mobley DL; Case DA Nonlinear scaling schemes for Lennard-Jones interactions in free energy calculations. J. Chem. Phys 2007, 127, 214108. [PubMed: 18067350]

(42). Beutler TC; Mark AE; van Schaik René C. and Gerber Paul R. and van Gunsteren Wilfred F., Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. Chem. Phys. Lett 1994, 222, 529–539.

(43). Steinbrecher T; Joung I; Case DA Soft-Core Potentials in Thermodynamic Integration: Comparing One- and Two-Step Transformations. J. Comput. Chem 2011, 32, 3253–3263. [PubMed: 21953558]

(44). Sugita Y; Okamoto Y Replica-exchange molecular dynamics method for protein folding. Chem. Phys. Lett 1999, 314, 141–151.

(45). Earl DJ; Deem MW Parallel tempering: theory, applications, and new perspectives. Phys. Chem. Chem. Phys 2005, 7, 3910–3916. [PubMed: 19810318]

(46). Arrar M; de Oliveira CAF; Fajer M; Sinko W; McCammon JA w-REXAMD: A Hamiltonian replica exchange approach to improve free energy calculations for systems with kinetically trapped conformations. J. Chem. Theory Comput 2013, 9, 18–23. [PubMed: 23316122]

(47). Meng Y; Roitberg AE Constant pH Replica Exchange Molecular Dynamics in Biomolecules Using a Discrete Protonation Model. J. Chem. Theory Comput 2010, 6, 1401–1412. [PubMed: 20514364]

(48). Sugita Y; Okamoto Y Replica-exchange multicanonical algorithm and multicanonical replica-exchange method for simulating systems with rough energy landscape. Chem. Phys. Lett 2000, 329, 261–270.

(49). Bergonzo C; Henriksen NM; Roe DR; Swails JM; Roitberg AE; Cheatham TE, III Multidimensional replica exchange molecular dynamics yields a converged ensemble of an RNA tetranucleotide. J. Chem. Theory Comput 2014, 10, 492–499. [PubMed: 24453949]

(50). Mitsutake A; Okamoto Y Multidimensional generalized-ensemble algorithms for complex systems. J. Chem. Phys 2009, 130, 214105. [PubMed: 19508054]

(51). Gallicchio E; Levy RM; Parashar M Asynchronous replica exchange for molecular simulations. J. Comput. Chem 2008, 29, 788–794. [PubMed: 17876761]

(52). Radak BK; Romanus M; Gallicchio E; Lee T-S; Weidner O; Deng N-J; He P; Dai W; York DM; Levy RM; 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.

(53). Radak BK; Romanus M; Lee T-S; Chen H; Huang M; Treikalis A; Balasubramanian V; Jha S; York DM Characterization of the Three-Dimensional Free Energy Manifold for the Uracil Ribonucleoside from Asynchronous Replica Exchange Simulations. J. Chem. Theory Comput 2015, 11, 373–377. [PubMed: 26580900]

(54). Chen W; Morrow BH; Shi C; Shen JK Recent development and application of constant pH molecular dynamics. Mol. Simulat 2014, 40, 830–838.

(55). Swails JM; York DM; Roitberg AE Constant pH replica exchange molecular dynamics in explicit solvent using discrete protonation states: Implementation, testing, and validation. J. Chem. Theory Comput 2014, 10, 1341–1352. [PubMed: 24803862]

(56). Dissanayake T; Swails JM; Harris ME; Roitberg AE; York DM Interpretation of pH-Activity Pro les for Acid-Base Catalysis from Molecular Simulations. Biochemistry 2015, 54, 1307–1313. [PubMed: 25615525]

(57). Roux B; Karplus M Ion transport in a model gramicidin channel. Structure and thermodynamics. Biophys. J 1991, 59, 961–981. [PubMed: 1714305]

(58). Roux B; Karplus M Potential energy function for cation–peptide interactions: An ab initio study. J. Comput. Chem 1995, 16, 690–704.

(59). Jones JE On the Determination of Molecular Fields. II. From the Equation of State of a Gas. Proc. R. Soc. Lond. A 1924, 106, 463–477.

(60). Li P; Merz KM, Jr. Taking into account the ion-induced dipole interaction in the nonbonded model of ions. J. Chem. Theory Comput 2014, 10, 289–297. [PubMed: 24659926]

(61). Abel R; Wang L; Harder ED; Berne BJ; Friesner RA Advancing Drug Discovery through Enhanced Free Energy Calculations. Acc. Chem. Res 2017, 50, 1625–1632. [PubMed: 28677954]

(62). Chodera JD; Shirts MR Replica exchange and expanded ensemble simulations as Gibbs sampling: Simple improvements for enhanced mixing. J. Chem. Phys 2011, 135, 194110. [PubMed: 22112069]

(63). Tan Z Optimally Adjusted Mixture Sampling and Locally Weighted Histogram Analysis. J. Comput. Graph. Stat 2017, 26, 54–65.

(64). Wang L; Friesner RA; Berne B Replica Exchange with Solute Scaling: A More Efficient Version of Replica Exchange with Solute Tempering (REST2). J. Phys. Chem. B 2011, 115, 9431–9438. [PubMed: 21714551]

(65). Liu P; Kim B; Friesner RA; Berne BJ Replica exchange with solute tempering: A method for sampling biological systems in explicit water. Proc. Natl. Acad. Sci. USA 2005, 102, 13749–13754. [PubMed: 16172406]

(66). Friedrichs MS; Eastman P; Vaidyanathan V; Houston M; Legrand S; Beberg AL; Ensign DL; Bruns CM; Pande VS Accelerating molecular dynamic simulation on graphics processing units. J. Comput. Chem 2009, 30, 864–872. [PubMed: 19191337]
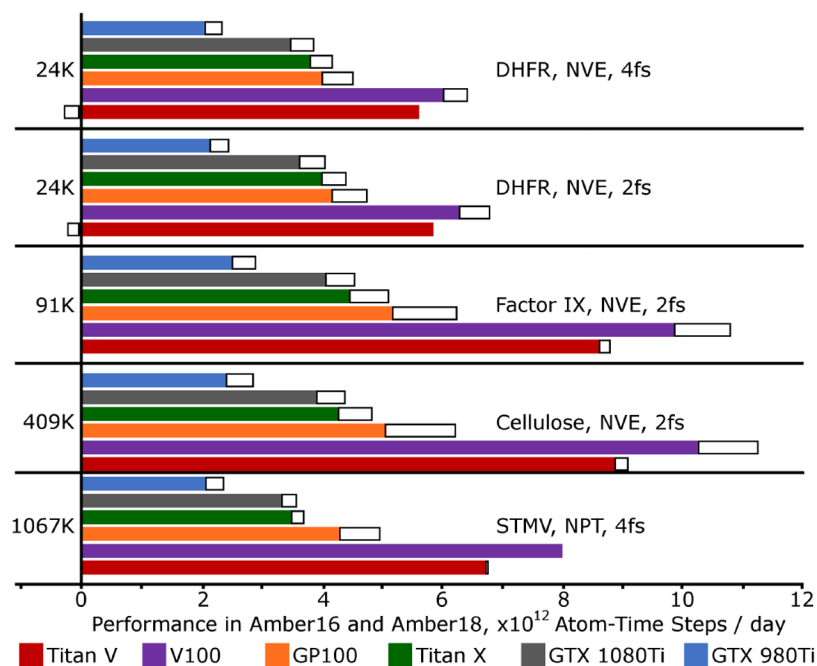
**Figure 1:**
Performance of Amber18 relative to Amber16 seen on multiple GPU architectures. Performance is given in a particle-normalized metric which emphasizes the number of inter actions that each card is able to compute in a given time. Performance in Amber16 is shown in solid color bars, and improvements with Amber18 in black outlined extensions. In a few cases, performance in Amber18 is lower than in Amber16, indicated by the extensions falling to the left of the *y*-axis. (Beta tests of an upcoming patch make Amber18 even faster, and consistently superior to Amber16.) System, ensemble, and time step are displayed on the right, while the system size (thousands of atoms) is given on the left. All systems were run with an 8 Å cutoff for real-space interactions and other default Amber parameters.
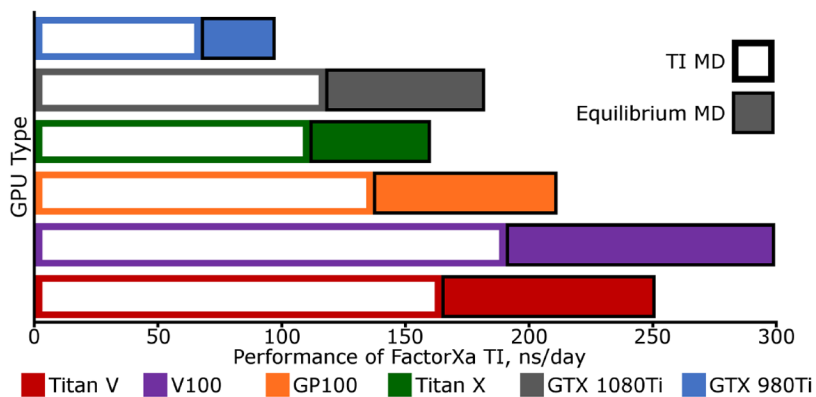
**Figure 2:**

Performance of Amber18 Thermodynamic Integration with linear alchemical mix ing on multiple GPU architectures relative to conventional MD. The color scheme for each GPU type is consistent with Figure 1, but the performance of TI is given by an open bar, while performance of the equivalent, "plain" MD system is given by a black-bordered solid extension. The test system is for the Factor Xa protein with the ligand mutation from L51a (62 atoms) to L51b (62 atoms). The system has total 41563 atoms and the whole ligand defined as the TI region.

**Table 1:**

Comparison of free energy (FEP/TI) compatible features in Amber16 and Amber18 on CPUs and GPUs.

| Free energy compatible feature | | Amber16 | | Amber18 | |
|---|---|---|---|---|---|
| **Category** | **Functionality** | **CPU** | **GPU** | **CPU** | **GPU** |
| Ensemble | NVE | ✓ | ✗ | ✓ | ✓ |
| | NVT | ✓ | ✗ | ✓ | ✓ |
| | NPT | ✓ | ✗ | ✓ | ✓ |
| | semi-iso P | ✗ | ✗ | ✗ | ✗ |
| | CpHMD | ✓ | ✗ | ✓ | ✓ |
| Free Energy Analysis | TI | ✓ | ✗ | ✓ | ✓ |
| | MBAR | ✓ | ✗ | ✓ | ✓ |
| Enhanced Sampling | H-REMD | ✓ | ✗ | ✓ | ✓ |
| | AMD | ✓ | ✗ | ✓ | ✓ |
| | SGLD | ✓ | ✗ | ✓ | ✗ |
| | GAMD | ✓ | ✗ | ✓ | ✓ |
| Potentials | 12-6-4 | ✓ | ✗ | ✓ | ✓ |

List of features that have compatibility with free energy (FEP/TI) methods in Amber16 and Amber18 on CPUs and GPUs. Red color indicates a feature not compatible with free energy methods (although it may be compatible with conventional MD). Green color indicates new free energy compatible feature in Amber18.

**Table 2:**

Comparison of MD simulation rates in Amber16 and Amber18 on CPUs and GPUs.

| | | Simulation Rate, ns/day | | | | | |
|---|---|---|---|---|---|---|---|
| | | GTX-980 Ti | | GTX-1080 Ti | | Titan-X | |
| System | Atom Count | Amber16 | Amber18 | Amber16 | Amber18 | Amber16 | Amber18 |
| DHFR, NVE, 4fs | 24K | 347 | 382 | 588 | 657 | 643 | 710 |
| DHFR, NVE, 2fs | 24K | 181 | 209 | 306 | 345 | 338 | 374 |
| Factor IX, NVE, 2fs | 91K | 52 | 64 | 85 | 100 | 93 | 113 |
| Cellulose, NVE, 2fs | 409K | 12 | 14 | 19 | 21 | 21 | 23 |
| STMV, NVE, 4fs | 1067K | 8 | 9 | 12 | 13 | 13 | 14 |
| | | Simulation Rate, ns/day | | | | | |
| | | GP100 | | V100 (Volta) | | Titan-V | |
| System | Atom Count | Amber16 | Amber18 | Amber16 | Amber18 | Amber16 | Amber18 |
| DHFR, NVE, 4fs | 24K | 677 | 768 | 1020 | 1091 | 954 | 904 |
| DHFR, NVE, 2fs | 24K | 353 | 404 | 532 | 577 | 497 | 477 |
| Factor IX, NVE, 2fs | 91K | 114 | 137 | 217 | 238 | 189 | 194 |
| Cellulose, NVE, 2fs | 409K | 25 | 29 | 50 | 49 | 43 | 41 |
| STMV, NVE, 4fs | 1067K | 16 | 19 | 30 | 30 | 25 | 25 |

Timings for selected systems in Amber18 versus Amber16. The ensemble and time step are given in the left hand column. Other Amber default parameters included an 8 Å cutoff and  1 Å PME (3D-FFT) grid spacing.