JCTC Journal of Chemical Theory and Computation

Letter

pubs.acs.org/JCTC

Toward Fast and Accurate Binding Affinity Prediction with pmemdGTI: An Efficient Implementation of GPU-Accelerated Thermodynamic Integration

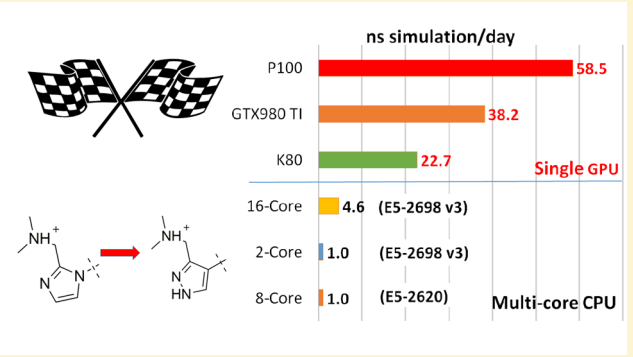
Tai-Sung Lee,*,† Yuan Hu,‡ Brad Sherborne,‡ Zhuyan Guo,‡ and Darrin M. York*,†

†Laboratory for Biomolecular Simulation Research, Center for Integrative Proteomics Research and Department of Chemistry and Chemical Biology, Rutgers University, Piscataway, New Jersey 08854, United States

‡Department of Chemistry, Modeling and Informatics, Merck Research Laboratories, 2000 Galloping Hill Road, Kenilworth, New Jersey 07033, United States

Supporting Information

ABSTRACT: We report the implementation of the thermodynamic integration method on the pmemd module of the AMBER 16 package on GPUs (pmemdGTI). The pmemdGTI code typically delivers over 2 orders of magnitude of speed-up relative to a single CPU core for the calculation of ligand−protein binding affinities with no statistically significant numerical differences and thus provides a powerful new tool for drug discovery applications.



Accurate prediction of the binding affinity between a drug candidate and the target protein is one of the biggest challenges for computer-aided drug design (CADD).[1−3] A broad range of methods exists for the prediction of protein−ligand binding affinities (henceforth referred to simply as "binding affinities") from molecular dynamics (MD) simulations, including fast empirical or semiempirical methods such as the linear interaction energy (LIE) model[4] and its variants,[5] the molecular mechanics (MM) combined with Poisson−Boltzmann or generalized Born solvation plus surface area correction (MM/PBSA and MM/GBSA),[6] and more rigorous and time-consuming alchemical free energy methods.[7−10] Currently, a wide range of alchemical free energy methods provides the most robust and accurate estimates of binding affinities from molecular dynamics simulations.[3,7,11−15] These methods include Thermodynamic Integration (TI)[11,16−18] and Free Energy Perturbation (FEP),[11,19−22] as well as analysis through Bennett's acceptance ratio and its variants (BAR/MBAR)[23−28] and are augmented with enhanced sampling such as replica exchange molecular dynamics,[29] metadynamics,[30] driven adiabatic free energy dynamics,[31] orthogonal space random walk,[32] adaptive integration,[33] and other methods.[34,35] Advanced alchemical free energy methods for binding affinity prediction[9,10,36] have evolved to the point that they approach quantitative predictive accuracy for ligand-binding affinities,[27,37,38] although much work remains, such as addressing sampling and convergence issues,[27,37] to improve precision so as to afford meaningful comparison with experimental uncertainties.[39]

Nevertheless, the calculation of binding affinities through alchemical free energy simulations is computationally intensive, and often limitations in computing resources and/or required turn-around time for calculations render these methods impractical, particularly for industrial applications.[40] A promising solution is to develop free energy simulation software that can leverage affordable graphical processing units (GPUs)[10,38,41] to increase throughput, in some cases by more than 2 orders of magnitude.

GPU-accelerated molecular dynamics has been implemented in software packages such as NAMD,[42] ACEMD,[43] AMBER,[44−46] OpenMM,[47] GROMACS,[48] and CHARMM[49] and has rapidly extended the time and spacial scales that MD simulation can reach.[10] On the other hand, GPU-accelerated alchemical free energy methods have only very recently emerged in a few simulation codes. Recent implementation of the FEP method into the Desmond/GPU MD engine,[50] together with replica exchange with solute tempering,[51] has demonstrated promising results for the accurate calculation of known binding affinities for a wide range of target classes and ligands.[38] The impact of these methods on making true predictions in blind tests remains to be fully determined.[52]

Received: January 31, 2017
Published: June 15, 2017

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

We report a GPU-accelerated TI implementation of the AMBER 16 pmemd program[41,45,46] (henceforth referred to as pmemdGTI). Object-oriented programming concepts of encapsulation and inheritance were utilized to create a highly efficient, maintainable, and expandable code. We demonstrate that pmemdGTI can be used to explore simulation times scales up to 100 times longer, opening the door to more meaningful assessment of errors in free energy estimates, and more practical, reliable predictions.

Alchemical free energy methods such as TI and FEP/BAR have different advantages and disadvantages relating to the phase-space overlap of the quantities being averaged that may vary widely in different applications, and while these issues remain a topic of considerable discussion,[11,27,53−55] both classes of methods are widely used. In principle, the calculated quantities required by these methods appear similar, but they are not exactly the same: for each time step, TI only needs the derivative of the Hamiltonian with respect to the current λ_i; while FEP+BAR/MBAR needs the Hamiltonian values at the current λ_i and at neighboring λ_{i+1} values as well as the exponential of the difference of the Hamiltonian values. As a result, their implementation on GPUs requires separate consideration in terms of storage and transfer of data, as well as the precision requirements in the averaging. For this reason, extension of the current implementation of the GPU-accelerated TI method to FEP/BAR is forthcoming.

1. IMPLEMENTATION

The software design and execution flow for the pmemdGTI code is illustrated in Figure 1, and addresses the following key implementation goals: 1) to meet the different precision requirements of the TI method relative to MD; 2) to maintain the same level of single-precision performance as in the current AMBER package; 3) to deliver the same level of accuracy for binding free energies as in the current AMBER CPU implementation of TI. To fulfill these goals, we utilized two architectural concepts of object-oriented programming: encapsulation and inheritance.[56]

Encapsulation. Encapsulation is the concept to hide the internal functionality and data of a programming object from the outside. In our implementation, the original AMBER GPU data structures are encapsulated into base C++ classes. Here we use a GPU terminology, "GPU context", to refer to a GPU program unit representing the executing instance on the GPU. Shown in the upper panel of Figure 1, the original AMBER GPU context, which contains all simulation data structures, including all coordinates, forces, energy terms, all simulation parameters and settings, is now encapsulated as a base class. The MD GPU simulation constant set, a collection of fast-access mathematical and physical constants and *the pointers* to the data structures of the GPU context, is packaged as a base class as well.

Inheritance. Inheritance enables newly derived classes to receive, or inherit, the properties and methods of existing classes. In our GPU-TI implementation, new TI classes are derived from the base classes that contain the original GPU functionality and data structures for MD simulations. Since the TI classes inherit the base classes, they can be used just as the original base classes, and further, new functionality and data structures can be added without modifying the original GPU functionality and data structures.

Benefits. Through encapsulation and inheritance (upper panel of Figure 1), TI capability can be implemented so that 1) there is little or no need to modify the original MD GPU code, since all new add-ons can be implemented in the derived TI classes; 2) the new specific 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 TI classes can utilize, for example, different precision models and even different force fields if necessary.

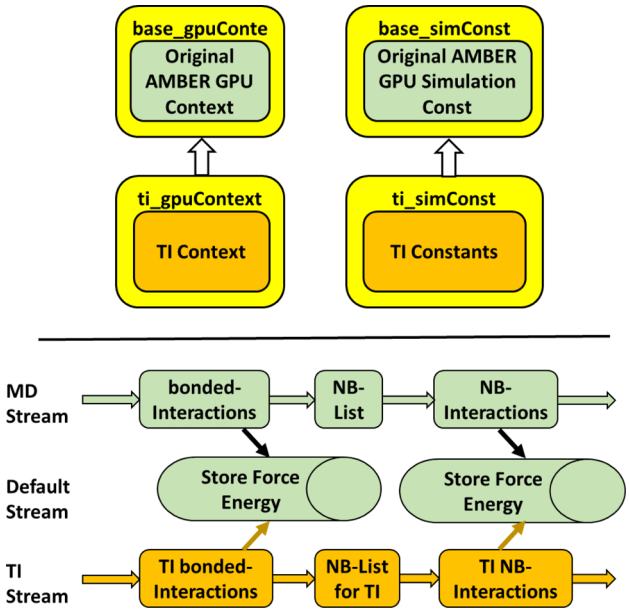
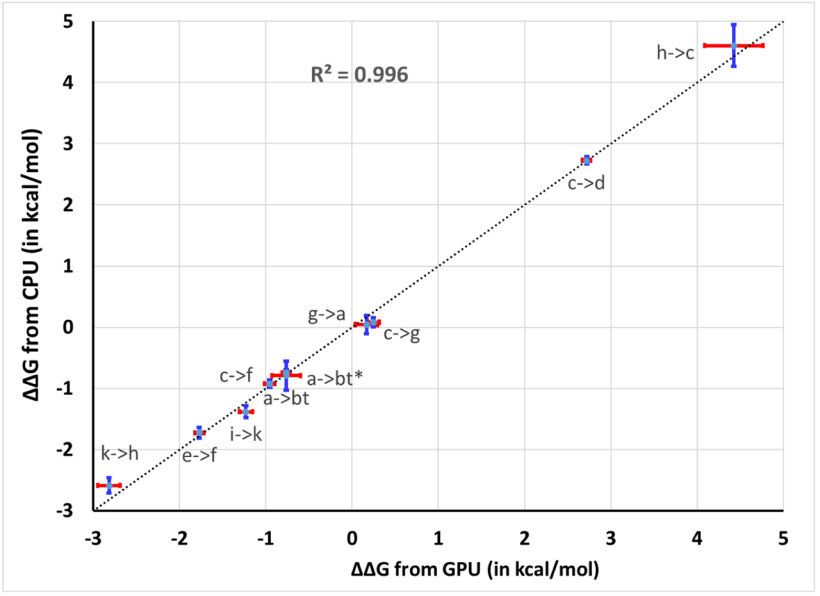


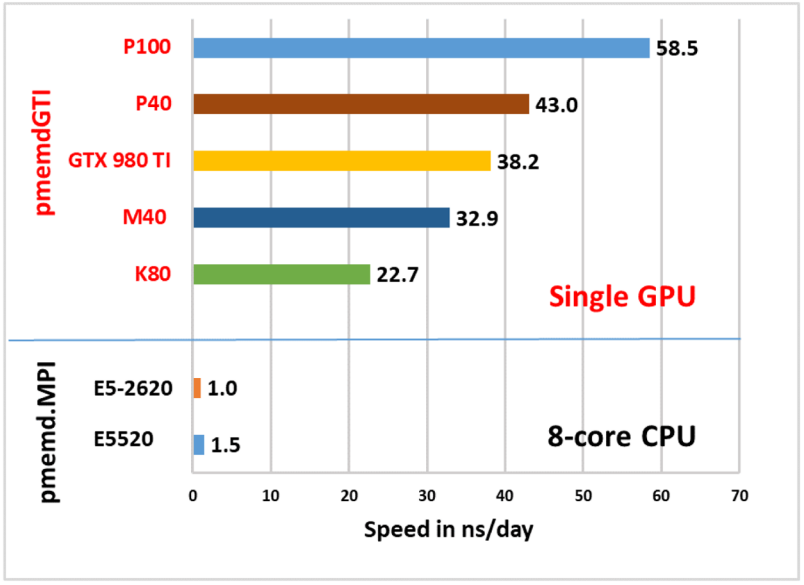
Figure 1. Schematic view of the pmemdGTI software design and execution flow (GPU context is terminology referring to a GPU program unit representing the executing instance on the GPU.) *Upper panel (software design)*: The original AMBER GPU context is packaged as a base class and contains all simulation data structures, including all coordinates, forces, energy terms, simulation parameters, and settings. The MD GPU simulation constant set is a collection of fast-access mathematical and physical constants and *the pointers* to the data structures of the GPU context and is packaged as a base class as well. *Lower panel (execution flow)*: The CUDA multiple-stream capability is utilized, and the original AMBER MD code runs on a separate stream from the newly added TI code. *Color codes*: light green: the original AMBER modules; yellow: the add-on to encapsulate AMBER modules; light orange: the new TI add-on modules.

Execution Flow. Shown in the lower panel of Figure 1, the execution of TI modules is implemented to execute on different CUDA streams (see the Supporting Information for CUDA multistream) and is independent from the execution of the original MD modules. The workflow of the particle mesh Ewald (PME[57]) modules is shown in the Supporting Information as an example to demonstrate our implementation and to evaluate the contribution from two end point states in a TI calculation.

Precision Model. Two precision models were implemented in pmemdGTI, the SPFP (single precision evaluation/fixed precision accumulation) and the DPFP (double precision evaluation/fixed precision accumulation); both already utilized the pmemd.cuda code of AMBER 16.[45,46] Similar to the MD code, the performance of pmemdGTI with DPFP is significantly slower than SPFP (see the Results section). Since it has been shown that the GPU MD code with SPFP is sufficient for MD simulations to reproduce the results from the CPU MD code with the high precision DPDP (double precision evaluation/double precision accumulation) model,[46] there is no strong incentive to perform MD simulations with

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX





mutation		GPU	CPU
a → bt		-0.76 (0.18)	-0.79 (0.24)
a → bt*		–0.76 (0.05)	-0.75 (0.05)
c → d		2.72 (0.05)	2.73 (0.06)
c → f		-0.96 (0.06)	-0.93 (0.06)
c → g		0.25 (0.07)	0.08 (0.07)
e → f		-1.77 (0.05)	-1.73 (0.09)
g → a		0.17 (0.13)	0.04 (0.15)
h → c		4.42 (0.33)	4.60 (0.34)
i → k		-1.23 (0.08)	-1.38 (0.10)
k → h		-2.82 (0.13)	-2.59 (0.12)

Figure 2. Binding affinities of various ligands with Factor Xa calculated from pmemdGTI (GPU) and pmemd (CPU). Ligands are from ref 59, and their structures are provided in the Supporting Information. Binding affinities were calculated with TI using 11 evenly spaced λ-windows ranging from 0.0 to 1.0 and integrated with Simpson's rule. Listed are average $\Delta\Delta G$'s values from 10 independent runs with 4 ns of data collection for each λ-window. The $\Delta\Delta G$ is the difference of $\Delta G_{\text{Complex}}$ and ΔG_{Soln} (see text). The softcore approach[63,64] was used for all TI calculations, except a → bt* where no softcore potential was used. All numbers are in kcal/mol. *Left Panel:* The plot of $\Delta\Delta G$ results from GPU (x-axis) and CPU (y-axis) with the associated standard errors as the error bars. The squared correlation coefficient (R^2) between these two sets of data is 0.996. The dotted line is the ideal $y = x$ regression line. *Right Panel:* The numerical values of $\Delta\Delta G$ and the associated standard errors (in parentheses), with the mutated chemical function groups shown with softcore atoms colored red. The lower left panel also shows selected benchmark results from Table 1.

DPFP for production, although it is valuable for validation purposes in some cases. In this work, we report benchmark performance timings for pmemdGTI with both SPFP and DPFP, as well as the SPFP binding free energy results compared to the CPU results with the high precision DPDP model.

Development Platforms. The reported work was developed with Microsoft Visual C++ 2015, Intel Visual Fortran, Nvidia CUDA 7.5/8.0, and Nvidia's Nsight on Windows 10 and with Eclipse IDE, GNU C++, GFortran, Nvidia CUDA 7.5/8.0, and Nvidia's Nsight on Linux systems. The current version was tested on Windows 10 and various x86-based Linux systems, as well as on an IBM Minsky system with PowerPC8 CPUs and Tesla P100 GPUs. The current GPU implementation extends the GPU version of the pmemd MD code with TI calculations, hence CPU-only functionalities in pmemd are not supported with the current implementation. The current pmemdGTI code only runs on a single GPU.

Test Case Setup. In order to test our implementation, we selected an important well-studied benchmark protein–ligand system: the blood coagulation factor Xa protein with a series of ligands (Figure 2, right panel),[58] from which we recently reported an extensive study on the effect of ligand protonation and tautomerization state in the prediction of relative binding

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX–XXX

affinities.[59] Standard AMBER TI calculations were performed in the NVT ensemble, the same as in recently reported GPU benchmarks,[45,60] and a 1 fs integration time step at 300 K with PME (detailed parameters are described in the Supporting Information). Binding affinities were calculated using the TI method with 11 evenly spaced λ-windows ranging from 0.0 to 1.0. It has been suggested that a 2 fs time step can be used with covalent bonds to hydrogen atoms constrained, or else a 1 fs time step is recommended for MD simulations.[61,62] As there are no detailed studies of the impact of the bond constraint SHAKE algorithm on the soft-core implementation of alchemical free energy calculations, we disabled SHAKE for all TI mutations and consequently used 1 fs time steps for all reported simulations. For comparison, sets of softcore and nonsoftcore simulations with SHAKE enabled and 2 fs time steps were performed (see the Supporting Information).

Independent 5 ns TI simulations were performed for each λ-window, the last 4 ns of which were used for free energy analysis.

The chemical formulas of all compounds and mutations are listed in Figure 2. The compound "bt" refers to the tautomer form of the compound b. The softcore approach[63,64] was utilized for both van der Waals and electrostatic interactions in all TI calculations, except a → bt* where no softcore potential was used ("*" denotes the simulation does not utilize softcore potential), which is to test the nonsoftcore implementation as this mutation is the only one that can be reasonably treated with nonsoftcore. The complex of Factor Xa and L51a ligand has total 41,563 atoms, including solvent molecules. The ligand L51a has 62 atoms, and other ligands have slightly different numbers of atoms as shown in their chemical formulas. In the TI simulations, the whole ligands were defined as the perturbed regions.

The free energy change of each mutation was calculated by numerical integration over all λ-windows using Simpson's rule.[53,65] For all mutations, both the free energy changes of ligands in solution (ΔG_{Soln}) and ligands in complexes ($\Delta G_{Complex}$) were calculated, and the relative binding affinities ($\Delta\Delta G$) were obtained from the differences of free energy changes of ligands in complexes and in solution. Detailed simulation protocols can be found in the Supporting Information.

Simulations were carried out on both CPU and GPU platforms. While we previously reported CPU results derived from a single simulation run,[59] to obtain benchmark quality statistical sampling and error estimates (see the section below), in the present work each mutation was repeated 10 times with different initial velocities for both CPU and GPU versions. An extra set of 20 ns (for each λ-window) simulations of the a → bt* mutation was performed on the GPU. Various types of GPUs (K40, K80, M40, and P100) of the Nvidia's testing cluster were utilized.

2. RESULTS AND DISCUSSION

2.1. Ensemble Average Results. There is an abundance of evidence that suggests an "ensemble average approach",[66−69] referring to performing an ensemble of independent simulations to calculate averaged free energies, is more effective than simply running a single long simulation for producing reliable, reproducible free energy results with meaningful error estimates.[70−72] Taking this approach, we have performed 10 independent TI simulation runs for each mutation (10 runs for each ligand mutation in solution, and 10 runs for each ligand mutation in the complex). The resulting error estimates and statistical analysis will allow assessment as to whether our TI implementation delivers the same results as the CPU version with proper statistical confidence.

The GPU Implementation Delivers Statistically Equivalent Results to That of the Current CPU Implementation. Figure 2 lists and plots 10-run averaged relative binding affinities for all mutations, along with the corresponding standard errors. We calculated the GPU results with the SPFP precision model and the CPU results with the default DPDP precision model. The 10-run average relative binding affinities from GPU and from CPU differ in the range between 0.01 and 0.2 kcal/mol. Based on the standard errors of these 10-run data (shown in the left panel of Figure 2), the unpaired t test p-values at the 95% confidence level (shown in the Supporting Information) of the null hypothesis "GPU and CPU results are from the same distribution" are in the range of 0.11 to 0.92. Hence we conclude that there is no statistically significant difference between the GPU and CPU results.

The Ensemble Average Approach Is an Effective Way To Obtain Reproducible and Reliable Free Energies. Additional simulations were performed (not shown), and we found that the ensemble average relative binding affinities ($\Delta\Delta G$) and their corresponding standard deviations are able to predict the results of new simulations, which confirmed the reproducibility of the calculated $\Delta\Delta G$ within the range of the estimated errors. We have performed 10-run 20 ns simulations of the a → bt mutation (detailed results in the Supporting Information). The standard deviation of binding affinity in the 10-run 20 ns simulations is similar to the 10-run 5 ns simulations, for both the ligand in solvent (0.010 vs 0.013 kcal/mol) and in the complex (0.162 vs 0.152 kcal/mol), consistent with recent TI studies.[39,72] This observation suggests that performing appropriate ensemble simulations may be more effective for convergence than a single long time TI simulation. In fact, "converged" long time TI results from a single simulation may be misleading in some cases, as our results demonstrate that the 20 ns simulations manifest almost exactly the same ensemble deviation as seen in 5 ns simulations.

2.2. Timing Benchmarks. Timing benchmarks are shown in Table 1 on the simulation performance rates reported in terms of ns per day and the ratios between MD and TI simulation rates. The simulation system is for the "a → bt" mutation in Factor Xa. MD results are for L51a in Factor Xa (Figure 2, right panel), while TI results are for the whole ligand (62 atoms) defined as the TI region. Benchmarks were measured based on a 100 ps simulation period in the NVT ensemble using a 1 fs time step.

The relative speeds of different GPUs and CPUs vary significantly (Table 1). In SPFP mode, the GPU performance ranges from 19.8 (K40) to 58.5 (P100) ns per day, whereas the fastest CPU (Xeon E5-2698) performance ranges from 0.78 (2-cores) to 5.95 (32-cores) ns per day. The speed-up of fastest single GPU (P100) to single CPU (Xeon E5-2698) core is 150X, whereas the speed-up for the slowest GPU (K40) to single CPU (Xeon E5-2620) core is 160X. Overall, pmemdGTI typically delivers over 2 orders of magnitude of speed-up relative to a single CPU core. Note that usually multiple CPU cores are utilized for simulations, thus the single CPU core timing estimates are merely used as a reference to establish normalized performance ratio comparisons.

For standard linear thermodynamic coupling, TI formally requires the evaluation of the Hamiltonians at both end points

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Table 1. Timing Benchmarks (ns/day) for pmemdGTI Running on Various Nvidia GPUs[a]

	MD	TI	TI/MD
GPU: pmemdGTI: SPFP			
GTX 980 TI	53.31	38.20	0.72
GTX Titan X Pascal	66.10	43.28	0.65
Tesla K40	28.62	19.82	0.69
Tesla K80	31.22	22.74	0.73
Tesla M40	43.83	32.85	0.75
Tesla P40	58.99	43.04	0.73
Tesla P100	84.13	58.49	0.70
GPU: pmemdGTI: DPFP			
GTX 980 TI	4.08	3.77	0.92
GTX Titan X Pascal	6.21	5.91	0.95
Tesla K40	9.74	7.91	0.81
Tesla K80	9.03	7.89	0.87
Tesla M40	3.42	3.19	0.93
Tesla P40	5.98	5.51	0.92
Tesla P100	37.61	30.73	0.82
CPU: pmemd.MPI			
Xeon E5520 (8 cores)	2.07	1.50	0.72
Xeon E5-2620 (8 cores)	1.36	0.99	0.73
Xeon E5-2698 v3 (2 cores)	1.02	0.78	0.76
Xeon E5-2698 v3 (16 cores)	4.64	3.79	0.82
Xeon E5-2698 v3 (2 × 16 cores)	7.61	5.95	0.78

[a]Performance rates are shown in units of ns per day for standard MD simulation (MD) and TI simulation (TI), as well as the ratio (TI/MD). Benchmarks were measured based on 100 ps of simulation in the NVT ensemble using a 1 fs time step and PME electrostatics, as discussed in the text. SPFP (single precision/fixed precision) and DPFP (double precision/fixed precision) refer to the different precision models used in AMBER. The CPU version always runs with the default DPDP model (double precision/double precision). MD results are for L51a in Factor Xa; TI results (total 41563 atoms, including solvent) are for the L51a (62 atoms) to L51bt (62 atoms) mutation with the whole ligand defined as the TI region. See the Implementation section for further details. The $dH/d\lambda$ sampling frequency is set to 10 fs in order to ensure benchmark-level statistics. Preliminary tests indicate that the overhead associated with the $dH/d\lambda$ evaluation required for TI sampling, relative to the force-only requirement for each MD step, is about 30%, and increasing the stride for TI sampling to 1 ps only improves performance by around 3%.

of the perturbed region. Avoiding explicit computation of the redundant energy and force terms, the CPU and GPU version (SPFP) both only need ∼30% overhead for TI compared to plain MD. This 30% overhead derives mainly from the need to repeat the long-range PME calculations, even in the case that only a relatively small region is changing charges. Whereas the interpolation of the changed charges to the PME B-spline grid can be optimized, the cost of the fast Fourier transforms and interpolation of the potentials and forces back to all atoms results in the bulk of the computational overhead. The TI calculations of the nonelectrostatic nonbonded terms only account for less than 5% of time when SPFP is utilized. Hence a smaller TI region does not make a tremendous difference in reducing this overhead. When running with the DPFP precision model, where the GPU is much slower in calculating nonbonding terms and the time needed for repeating the long-range PME part becomes relatively smaller, the TI/MD ratio becomes higher as a result.

In short, our benchmark timing results demonstrate that currently available GPUs, when performing the reported TI free energy simulations, are able to deliver at least 2 orders of magnitude speed-up compared to a single CPU core for typical applications. The pmemdGTI code performs TI roughly at the speed of 70% of running an MD simulation with the fast SDFP precision mode, similar to the ratios seen in CPU versions.

It should be noted that the current implementation of pmemdGTI is not parallelized for multiple GPUs, whereas the CPU codes do run in parallel, and one can expect that significantly greater throughput could be obtained from use of a sufficiently high number of CPU cores. Nonetheless, typical free energy applications such as those presented here require, for each calculation, multiple windows (11 in the current work) to be examined and, for each window, multiple independent simulations (10 in the current work) to obtain converged averages and error estimates.[70−72] Further, for most pharmaceutical applications, predictions are typically required for an entire library of ligand compounds.[59] This leads to the need to perform many independent simulations (1,100 in the current work) which can be performed in parallel. Given that the parallel scaling performance of GPU MD codes is typically much worse than that for parallel CPU codes, it is usually overwhelmingly the case that the most practical and efficient strategy for high-throughput free energy predictions is to run multiple single-GPU TI simulations at the same time in parallel.

3. CONCLUSION

We report the implementation of the thermodynamic integration method based on the pmemd module of the AMBER 16 package on GPU platforms. The pmemdGTI code utilizes object-oriented programming principles so that the new add-on codes are inherited from, not a direct extension of, the original pmemd GPU codes, resulting in a software implementation which is easy to maintain, test, and extend. Benchmarks with selected protein−ligand systems demonstrate that pmemdGTI is capable of delivering ligand−protein binding affinities with 2 orders of magnitude speed-up relative to a single CPU core. The pmemdGTI code enables routine TI simulations to be performed efficiently on affordable GPU computing platforms and makes practical robust estimation of ligand binding affinities with reliable precision. This enabling technology advances the state of the art and provides a powerful tool for new and emerging drug discovery applications.

Software Availability. We plan to package pmemdGTI with the next official AMBER release (AMBER 18). The current version of pmemdGTI source code can be installed as a patch of AMBER16 for Unix platforms, as well as precompiled pmemdGTI binary for 64-bit Windows 7/8/10.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.7b00102.

Ligand structures and experimental binding affinities, simulation protocols and setup, binding free energy results for all runs, and software execution flow (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

*E-mail: taisung@rutgers.edu.
*E-mail: Darrin.York@rutgers.edu.

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

ORCID

Yuan Hu: 0000-0002-1014-6594
Brad Sherborne: 0000-0002-0037-3427
Darrin M. York: 0000-0002-9193-7055

Funding

The authors are grateful for financial support provided by the National Institutes of Health (GM107485 to D.Y.). This work was funded in part by Merck Research Laboratories, including financial support (to Y.H.) from the Postdoctoral Research Fellows Program, and the technical support from the High Performance Computing (HPC) group at Merck & Co., Inc. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation grant number OCI-1053575, with project number TG-MCB110101 (D.Y.). We gratefully acknowledge the support of the Nvidia Corporation with the donation of a GTX Titan X (Pascal) GPU and the GPU-time of a GPU-cluster where the reported benchmark results were performed.

Notes

The authors declare no competing financial interest.

DEDICATION

The authors would like to dedicate this work to the memory of Frank Brown, whose valuable insight, support, and encouragement were instrumental to its success.

REFERENCES

(1) Jorgensen, W. L. Efficient drug lead discovery and optimization. *Acc. Chem. Res.* **2009**, *42*, 724−733.
(2) Jorgensen, W. L. The many roles of computation in drug discovery. *Science* **2004**, *303*, 1813−1818.
(3) 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.
(4) Aqvist, J.; Marelius, J. The linear interaction energy method for predicting ligand binding free energies. *Comb. Chem. High Throughput Screening* **2001**, *4*, 613−626.
(5) Miranda, W. E.; Noskov, S. Y.; Valiente, P. A. Improving the LIE Method for Binding Free Energy Calculations of Protein-Ligand Complexes. *J. Chem. Inf. Model.* **2015**, *55*, 1867−1877.
(6) Genheden, S.; Ryde, U. The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities. *Expert Opin. Drug Discovery* **2015**, *10*, 449−461.
(7) Hansen, N.; van Gunsteren, W. F. Practical Aspects of Free-Energy Calculations: A Review. *J. Chem. Theory Comput.* **2014**, *10*, 2632−2647.
(8) Gapsys, V.; Michielssens, S.; Peters, J. H.; de Groot, B. L.; Leonov, H. Calculation of binding free energies. *Methods Mol. Biol.* **2015**, *1215*, 173−209.
(9) De Vivo, M.; Masetti, M.; Bottegoni, G.; Cavalli, A. Role of Molecular Dynamics and Related Methods in Drug Discovery. *J. Med. Chem.* **2016**, *59*, 4035−4061.
(10) Perez, A.; Morrone, J. A.; Simmerling, C.; Dill, K. A. Advances in free-energy-based simulations of protein folding and ligand binding. *Curr. Opin. Struct. Biol.* **2016**, *36*, 25−31.
(11) Shirts, M. R.; Pande, V. S. 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.
(12) *Free Energy Calculations: Theory and Applications in Chemistry and Biology*; Springer Series in Chemical Physics; Chipot, C., Pohorille, A., Eds.; Springer: New York, 2007; Vol. *86*, DOI: 10.1007/978-3-540-38448-9.
(13) 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.
(14) Gallicchio, E.; Levy, R. M. Advances in all atom sampling methods for modeling protein-ligand binding affinities. *Curr. Opin. Struct. Biol.* **2011**, *21*, 161−166.
(15) 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.
(16) Straatsma, T. P.; Berendsen, H. J. C.; Postma, J. P. M. Free energy of hydrophobic hydration: A molecular dynamics study of noble gases in water. *J. Chem. Phys.* **1986**, *85*, 6720−6727.
(17) Straatsma, T. P.; Berendsen, H. J. 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.
(18) Straatsma, T. P.; McCammon, J. A. Multiconfiguration thermodynamic integration. *J. Chem. Phys.* **1991**, *95*, 1175−1188.
(19) Zwanzig, R. W. High-temperature equation of state by a perturbation method. I. Nonpolar gases. *J. Chem. Phys.* **1954**, *22*, 1420−1426.
(20) Torrie, G. M.; Valleau, J. P. Monte Carlo free energy estimates using non-Boltzmann sampling: Application to the sub-critical Lennard-Jones fluid. *Chem. Phys. Lett.* **1974**, *28*, 578−581.
(21) Jarzynski, C. Equilibrium free-energy differences from non-equilibrium measurements: a master-equation approach. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **1997**, *56*, 5018−5035.
(22) Lu, N.; Kofke, D. A. Accuracy of free-energy perturbation calculations in molecular simulation. II. Heuristics. *J. Chem. Phys.* **2001**, *115*, 6866−6875.
(23) Bennett, C. H. Efficient estimation of free energy differences from Monte Carlo data. *J. Comput. Phys.* **1976**, *22*, 245−268.
(24) Shirts, M. R.; Bair, E.; Hooker, G.; Pande, V. S. Equilibrium free energies from nonequilibrium measurements using maximum-likelihood methods. *Phys. Rev. Lett.* **2003**, *91*, 140601.
(25) Shirts, M. R.; Chodera, J. D. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.* **2008**, *129*, 124105.
(26) 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.
(27) 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.
(28) Klimovich, P. V.; Shirts, M. R.; Mobley, D. L. Guidelines for the analysis of free energy calculations. *J. Comput.-Aided Mol. Des.* **2015**, *29*, 397−411.
(29) Meng, Y.; Sabri Dashti, D.; Roitberg, A. E. Computing alchemical free energy differences with Hamiltonian replica exchange molecular dynamics (H-REMD) simulations. *J. Chem. Theory Comput.* **2011**, *7*, 2721−2727.
(30) Hošek, P.; Toulcová, D.; Bortolato, A.; Spiwok, V. Altruistic Metadynamics: Multisystem Biased Simulation. *J. Phys. Chem. B* **2016**, *120*, 2209−2215.
(31) Cuendet, M. A.; Tuckerman, M. E. Alchemical Free Energy Differences in Flexible Molecules from Thermodynamic Integration or Free Energy Perturbation Combined with Driven Adiabatic Dynamics. *J. Chem. Theory Comput.* **2012**, *8*, 3504−3512.
(32) Abella, J. R.; Cheng, S. Y.; Wang, Q.; Yang, W.; Ren, P. Hydration Free Energy from Orthogonal Space Random Walk and Polarizable Force Field. *J. Chem. Theory Comput.* **2014**, *10*, 2792−2801.
(33) Kaus, J. W.; McCammon, J. A. Enhanced ligand sampling for relative protein-ligand binding free energy calculations. *J. Phys. Chem. B* **2015**, *119*, 6190−6197.

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

(34) Li, H.; Min, D.; Liu, Y.; Yang, W. Essential energy space random walk via energy space metadynamics method to accelerate molecular dynamics simulations. *J. Chem. Phys.* **2007**, *127*, 094101−094101.
(35) Mobley, D. L.; Klimovich, P. V. Perspective: Alchemical free energy calculations for drug discovery. *J. Chem. Phys.* **2012**, *137*, 230901.
(36) Ashida, T.; Kikuchi, T. Overview of binding free energy calculation techniques for elucidation of biological processes and for drug discovery. *Med. Chem.* **2015**, *11*, 248−253.
(37) Christ, C. D.; Fox, T. Accuracy assessment and automation of free energy calculations for drug design. *J. Chem. Inf. Model.* **2014**, *54*, 108−120.
(38) Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, M. K.; Greenwood, J.; Romero, D. L.; Masse, C.; Knight, J. L.; Steinbrecher, T.; Beuming, T.; Damm, W.; Harder, E.; Sherman, W.; Brewer, M.; Wester, R.; Murcko, M.; Frye, L.; Farid, R.; Lin, T.; Mobley, D. L.; Jorgensen, W. L.; Berne, B. J.; Friesner, R. A.; Abel, R. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *J. Am. Chem. Soc.* **2015**, *137*, 2695−2703.
(39) Bhati, A. P.; Wan, S.; Wright, D. W.; Coveney, P. V. Rapid, accurate, precise and reliable relative free energy prediction using ensemble based thermodynamic integration. *J. Chem. Theory Comput.* **2017**, *13*, 210−222.
(40) Chipot, C.; Rozanska, X.; Dixit, S. B. 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.
(41) Anderson, J. A.; Lorenz, C. D.; Travesset, A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *J. Comput. Phys.* **2008**, *227*, 5342−5359.
(42) Stone, J. E.; Phillips, J. C.; Freddolino, P. L.; Hardy, D. J.; Trabuco, L. G.; Schulten, K. Accelerating molecular modeling applications with graphics processors. *J. Comput. Chem.* **2007**, *28*, 2618−2640.
(43) Harvey, M. J.; Giupponi, G.; Fabritiis, G. D. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *J. Chem. Theory Comput.* **2009**, *5*, 1632−1639.
(44) Götz, A.; Williamson, M. J.; Xu, D.; Poole, D.; Le Grand, S.; Walker, R. C. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 1. Generalized Born. *J. Chem. Theory Comput.* **2012**, *8*, 1542.
(45) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent Particle Mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9*, 3878−3888.
(46) Le Grand, S.; Götz, A. W.; Walker, R. C. SPFP: Speed without compromise-A mixed precision model for GPU accelerated molecular dynamics simulations. *Comput. Phys. Commun.* **2013**, *184*, 374−380.
(47) Eastman, P.; Pande, V. S. Efficient nonbonded interactions for molecular dynamics on a graphics processing unit. *J. Comput. Chem.* **2010**, *31*, 1268−1272.
(48) Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; Hess, B.; Lindahl, E. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* **2013**, *29*, 845−854.
(49) Arthur, E. J.; Brooks, C. L. Efficient implementation of constant pH molecular dynamics on modern graphics processors. *J. Comput. Chem.* **2016**, *37*, 2171−2180.
(50) Bergdorf, M.; Kim, E. T.; Rendleman, C. A.; Shaw, D. E. *Desmond GPU Performance as of November 2014*; Technical Report, DESRESTR-2014-01; 2014.
(51) Liu, P.; Kim, B.; Friesner, R. A.; Berne, B. J. Replica exchange with solute tempering: A method for sampling biological systems in explicit water. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 13749−13754.
(52) 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. D3R grand challenge 2015: Evaluation of protein-ligand pose and affinity predictions. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 651−668.
(53) Bruckner, S.; Boresch, S. Efficiency of alchemical free energy simulations. II. Improvements for thermodynamic integration. *J. Comput. Chem.* **2011**, *32*, 1320−1333.
(54) Ytreberg, F. M.; Swendsen, R. H.; Zuckerman, D. M. Comparison of free energy methods for molecular systems. *J. Chem. Phys.* **2006**, *125*, 184114.
(55) Paliwal, H.; Shirts, M. R. A benchmark test set for alchemical free energy transformations and its use to quantify error in common free energy methods. *J. Chem. Theory Comput.* **2011**, *7*, 4115−4134.
(56) Scott, M. L. *Programming Language Pragmatics*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000.
(57) Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: An N log(N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089−10092.
(58) 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. 7-fluoroindazoles as potent and selective inhibitors of factor Xa. *J. Med. Chem.* **2008**, *51*, 282−297.
(59) Hu, Y.; Sherborne, B.; Lee, T.-S.; Case, D. A.; York, D. M.; Guo, Z. The importance of protonation and tautomerization in relative binding affinity prediction: a comparison of AMBER TI and Schrödinger FEP. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 533−539.
(60) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; Tye, T.; Houston, M.; Stich, T.; Klein, C.; Shirts, M. R.; Pande, V. S. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **2013**, *9*, 461−469.
(61) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kaleé, L.; Schulten, K. Scalable Molecular Dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781−1802.
(62) Case, D.; Babin, V.; Berryman, J.; Betz, R.; Cai, Q.; Cerutti, D.; Cheatham, T., III; Darden, T.; Duke, R.; Gohlke, H.; Goetz, A.; Gusarov, S.; Homeyer, N.; Janowski, P.; Kaus, J.; Kolossváry, I.; Kovalenko, A.; Lee, T.; LeGrand, S.; Luchko, T.; Luo, R.; Madej, B.; Merz, K.; Paesani, F.; Roe, D.; Roitberg, A.; Sagui, C.; Salomon-Ferrer, R.; Seabra, G.; Simmerling, C.; Smith, W.; Swails, J.; Walker, R.; Wang, J.; Wolf, R.; Wu, X.; Kollman, P. *AMBER 14*; University of California, San Francisco: San Francisco, CA, 2014.
(63) Steinbrecher, T.; Joung, I.; Case, D. A. Soft-Core Potentials in Thermodynamic Integration: Comparing One- and Two-Step Transformations. *J. Comput. Chem.* **2011**, *32*, 3253−3263.
(64) Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. *Chem. Phys. Lett.* **1994**, *222*, 529−539.
(65) Jorge, M.; Garrido, N. M.; Queimada, A. J.; Economou, I. G.; Macedo, E. A. Effect of the Integration Method on the Accuracy and Computational Efficiency of Free Energy Calculations Using Thermodynamic Integration. *J. Chem. Theory Comput.* **2010**, *6*, 1018−1027.
(66) Caves, L. S.; Evanseck, J. D.; Karplus, M. Locally accessible conformations of proteins: multiple molecular dynamics simulations of crambin. *Protein Sci.* **1998**, *7*, 649−666.
(67) Loccisano, A. E.; Acevedo, O.; DeChancie, J.; Schulze, B. G.; Evanseck, J. D. Enhanced sampling by multiple molecular dynamics trajectories: carbonmonoxy myoglobin 10 micros A0−>A(1−3) transition from ten 400 ps simulations. *J. Mol. Graphics Modell.* **2004**, *22*, 369−376.
(68) Likic, V. A.; Gooley, P. R.; Speed, T. P.; Strehler, E. E. A statistical approach to the interpretation of molecular dynamics simulations of calmodulin equilibrium dynamics. *Protein Sci.* **2005**, *14*, 2955−2963.

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

(69) Elofsson, A.; Nilsson, L. How consistent are molecular dynamics simulations? Comparing structure and dynamics in reduced and oxidized Escherichia coli thioredoxin. *J. Mol. Biol.* **1993**, *233*, 766−780.
(70) Genheden, S.; Ryde, U. How to obtain statistically converged MM/GBSA results. *J. Comput. Chem.* **2010**, *31*, 837−846.
(71) Wan, S.; Knapp, B.; Wright, D. W.; Deane, C. M.; Coveney, P. V. Rapid, Precise, and Reproducible Prediction of Peptide-MHC Binding Affinities from Molecular Dynamics That Correlate Well with Experiment. *J. Chem. Theory Comput.* **2015**, *11*, 3346−3356.
(72) Coveney, P. V.; Wan, S. On the calculation of equilibrium thermodynamic properties from molecular dynamics. *Phys. Chem. Chem. Phys.* **2016**, *18*, 30236−30240.

DOI: 10.1021/acs.jctc.7b00102
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX