A peer-reviewed version of this preprint was published in PeerJ on 16 March 2020.

View the peer-reviewed version (peerj.com/articles/cs-264), which is the preferred citable publication unless you specifically need to cite this preprint.

Mirabzadeh CA, Ytreberg FM. 2020. Implementation of adaptive integration method for free energy calculations in molecular systems. PeerJ Computer Science 6:e264 https://doi.org/10.7717/peerj-cs.264

Implementation of adaptive integration method for free energy calculations in molecular systems

Christopher A Mirabzadeh [1], **F. Marty Ytreberg** [Corresp. 1, 2, 3]

[1] Department of Physics, University of Idaho, Moscow, Idaho, United States

[2] Center for Modeling Complex Interactions, University of Idaho, Moscow, Idaho, United States

[3] Institute for Bioinformatics and Evolutionary Studies, University of Idaho, Moscow, Idaho, United States

Corresponding Author: F. Marty Ytreberg
Email address: ytreberg@uidaho.edu

Estimating free energy differences by computer simulation is useful for a wide variety of applications such as virtual screening for drug design and for understanding how amino acid mutations modify protein interactions. However, calculating free energy differences remains challenging and often requires extensive trial and error and very long simulation times in order to achieve converged results. Here, we present an implementation of the adaptive integration method (AIM). We tested our implementation on two molecular systems and compared results from AIM to those from a suite of standard methods. The model systems tested here include calculating the solvation free energy of methane, and the free energy of mutating the peptide GAG to GVG. We show that AIM is more efficient than standard methods for these test cases, that is, AIM results converge to a higher level of accuracy and precision for a given simulation time.

Implementation of Adaptive Integration Method for Free Energy Calculations in Molecular Systems

CA Mirabzadeh[1] **and FM Ytreberg**[2]

[1,2]**Department of Physics, University of Idaho, Moscow ID**

Corresponding author:
FM Ytreberg[2]

Email address: ytreberg@uidaho.edu

ABSTRACT

Estimating free energy differences by computer simulation is useful for a wide variety of applications such as virtual screening for drug design and for understanding how amino acid mutations modify protein interactions. However, calculating free energy differences remains challenging and often requires extensive trial and error and very long simulation times in order to achieve converged results. Here, we present an implementation of the adaptive integration method (AIM). We tested our implementation on two molecular systems and compared results from AIM to those from a suite of standard methods. The model systems tested here include calculating the solvation free energy of methane, and the free energy of mutating the peptide GAG to GVG. We show that AIM is more efficient than standard methods for these test cases, that is, AIM results converge to a higher level of accuracy and precision for a given simulation time.

INTRODUCTION

Measuring free energy differences using computer simulations can be computationally expensive, yet is useful for many different applications (see e.g., Steinbrecher and Labahn (2010); Chodera et al. (2011); Mobley et al. (2012); Zhan et al. (2013); Miller et al. (2014); Petukh et al. (2015); Zhan and Ytreberg (2015); Wichman et al. (2016); Cournia et al. (2017); Hossain et al. (2019); Aminpour et al. (2019)). Specific examples include determining protein conformational preferences, virtual screening for drug design or drug discovery (Steinbrecher and Labahn, 2010; Chodera et al., 2011; Zhan and Ytreberg, 2015; Śledź and Caflisch, 2018; Aminpour et al., 2019; Zhang et al., 2019). Of specific relevance to the current study is that free energy calculations allow prediction of how amino acid mutations may modify protein-protein binding (Zhan et al., 2013; Miller et al., 2014; Petukh et al., 2015; Wichman et al., 2016; Geng et al., 2019). We are particularly interested in developing and implementing efficient methods for calculating free energy differences and using them to understand how amino acid mutations modify protein-protein and protein-substrate interactions.

For this study, we have implemented the adaptive integration method (AIM) introduced by Fasnacht et al. (2004) for use in the GROMACS (Berendsen et al., 1995) molecular dynamics simulation package. Though there are many free energy methods for molecular systems (see e.g., Lyubartsev et al. (1996); Gonçalves and Stassen (2004); Kofke (2005); Shirts et al. (2007); Chodera and Shirts (2011); Klimovich et al. (2015)), in previous studies AIM has shown promise to provide high quality, precise and efficient estimates of binding free energies(Ytreberg et al., 2006; Kaus et al., 2014; Kaus and Mccammon, 2015). AIM is an adaptive sampling method that continuously improves the estimate for free energy during the simulation by using Metropolis Monte Carlo to sample λ space (λ defines progress along the reaction pathway). The algorithm automatically increases sampling in regions along the reaction pathway, wherever there is a need.

In order to compare to other free energy methods we used the Python tool, alchemical-analysis.py (Klimovich et al., 2015), part of the Pymbar package (Shirts and Chodera, 2008). The alchemical-analysis tool takes the output from molecular dynamics simulations and estimates the free energy using

some standard methods, including the Bennett acceptance ratio, multistate Bennett acceptance ratio, thermodynamic integration and exponential averaging. The most substantial difference between these methods and AIM is that they all expect equilibrium sampling of configurations for each value of λ. This is achieved via fixed λ standard molecular dynamics simulations, in contrast to the Monte Carlo λ moves used in AIM.

For the current study we chose two molecular systems that have well-documented results and are important starting points for biomolecular free energy studies. First, we calculated the solvation free energy of methane. Simulations were performed and the free energies were calculated using the standard methods provided by alchemical-analysis. Simulations were also performed using AIM and results compared to standard simulations. Using the lessons learned from the methane system, we then calculated the free energy of mutating GAG to GVG in water. For both systems, we found that AIM produces free energy estimates that are within statistical uncertainty of standard methods but with greater efficiency (i.e., more accurate for a given simulation time).

METHODS

For this study we performed alchemical free energy simulations where the system is changed from a reference state to an end state by constructing a reaction pathway that modifies, adds or removes atoms. Such alchemical simulations are non-physical, i.e., the simulation does not represent what could occur naturally. Since the free energy is a state variable, it is independent of the path taken, and we may provide any path we wish. To perform these simulations the reaction pathway is divided into many separate, non-physical, λ states between a reference state and an end state. The λ states represent the progress along the reaction pathway as the reference state transforms into the end state.

Like most methods used to calculate free energies we start from the free energy identity,

$$F = U - TS, \tag{1}$$

where U is the potential energy, T is the temperature and S is the entropy of the system. For free energy differences we generalize the formulation of the change in free energy by separating calculations into two, non-overlapping, thermodynamic end states, A and B, at constant system temperature T,

$$\Delta F \equiv \Delta F_{A\to B} = F_B - F_A = \Delta U - T\Delta S. \tag{2}$$

ΔF is the change in free energy, ΔU is the change in potential energy and ΔS is the change in entropy of the system. According to statistical mechanics, the free energy difference between the two end states, A and B, of the system is the log of the ratio of the partition functions,

$$\Delta F = -k_B T \ln \frac{Z[U_B(\vec{x})]}{Z[U_A(\vec{x})]}. \tag{3}$$

Here, k_B is the Boltzmann constant and $Z[U(\vec{x})]$ is the partition function for the energy states $U_A(\vec{x})$ and $U_B(\vec{x})$, where $\vec{x}$ is the vector of configuration coordinates. The partition function is given by

$$Z[U(\vec{x})] = \int \exp\left(-\beta U(\vec{x})\right) dx, \tag{4}$$

and $\beta = \frac{1}{k_B T}$.

Computationally, we calculate free energy differences between end states by performing molecular dynamics simulations along a reaction pathway of intermediate states, defined by λ, such that,

$$0 \leq \lambda \leq 1. \tag{5}$$

This pathway connects the two end states of the system. In the case of poor overlap, where the end states may be separated by a high energy barrier, $|U_B - U_A| \gg k_B T$, this pathway mitigates the otherwise very slow convergence of free energy estimates (Shirts et al., 2007). Care should be taken when choosing intermediate states such that there is adequate overlap in the conformation space between the end states (Shirts et al., 2007; Klimovich et al., 2015). For our simulations the number of λ values and time per λ were chosen through extensive trial and error (more on this below).

The method of exponential averaging (Zwanzig, 1954) starts from Eq. (3) above and then adding and subtracting $\exp(-\beta U(\vec{x}))$ from the integral in the partition function of the numerator we end up with the final relationship,

$$\Delta F_{ij} = -k_B T \ln\langle \exp(-\beta \Delta U_{ij}(\vec{x}))\rangle_{\lambda_i}. \tag{6}$$

where ΔF_{ij} is the free energy between λ_i and λ_j and $\langle\cdot\rangle_{\lambda_i}$ represents an average of the equilibrium configuration for λ_i. Unlike some other methods, exponential averaging has an exact solution since it is only used to evaluate the difference between two states. However, it is the least efficient method and should not be used if difference in potential energies are much larger than $k_B T$ Shirts and Pande (2005). In addition, exponential averaging can be noisy, biased and dependent on the tails of the distribution of λ states (Bruckner and Boresch, 2011; Shirts and Pande, 2005).

For thermodynamic integration (TI) we estimate the free energy by first looking at the derivative of Eq. (1) with respect to λ,

$$\frac{\partial F}{\partial \lambda} = \left\langle \frac{\partial U}{\partial \lambda} \right\rangle_{\lambda}. \tag{7}$$

This differential equation, Eq. (7), can then be integrated to give,

$$\Delta F = \int_{\lambda=0}^{1} \left\langle \frac{\partial U_{\lambda}(\vec{x})}{\partial \lambda} \right\rangle_{\lambda} d\lambda \tag{8}$$

where the $\langle\cdot\rangle_{\lambda}$ notation represents the ensemble average at a given intermediate state, λ. The free energy is estimated by numerically integrating Eq. (8) after running equilibrium simulations at each intermediate λ state. Since numerical integration is required, TI can be biased by the chosen method of integration. Some of that bias can be removed by using cubic-spline interpolation or more complex integration estimators(Shirts and Pande, 2005; Shyu and Ytreberg, 2009).

The Bennett (Bennett, 1976) and multistate (Shirts and Chodera, 2008) Bennett acceptance ratio (BAR and MBAR) methods are far more efficient than exponential averaging and are commonly used to avoid the shortcomings of other methods (Shirts and Pande, 2005; Ytreberg et al., 2006). BAR and MBAR typically achieve the same statistical precision as TI with fewer λ states unless the integrand for TI is very smooth (Shirts and Mobley, 2013; Ytreberg et al., 2006). The complete derivation can be found in Bennett's paper (Bennett, 1976) but the premise is; for sufficiently large samples n_i of U_i and n_j of U_j,

$$\Delta F(i \to j) = k_B T \ln \frac{\langle f(\Delta U_{ij} + C)\rangle_j}{\langle f(\Delta U_{ji} - C)\rangle_i} + C. \tag{9}$$

C is a shift constant,

$$C = k_B T \ln \frac{n_j}{n_i}, \tag{10}$$

and $f(x)$ is the Fermi function,

$$f(x) = \frac{1}{1+\exp(\beta x)}. \tag{11}$$

Equation (9) is the ratio of canonical averages of two different potentials U_i and U_j acting on the same configuration space meaning it requires information from two neighboring states. However, this limitation is not too much of a concern with a trivial coordinate transformation or when using dummy coordinates in alchemical simulations. MBAR, an extension of BAR, differs in that it takes data from more than two states hence the name "multistate".

AIM is similar to TI in that numerical integration of Eq. (8) is performed; the key difference is how the averages $\langle \partial U/\partial\lambda\rangle_{\lambda}$ are obtained. AIM uses Metropolis Monte Carlo to move in λ space and ordinary running averages are calculated at each λ value. In AIM, a random move from λ_{old} to λ_{new} is accepted with probability

$$\min\{1, \exp(-\beta(U_{new} - U_{old}) + \beta(F_{new} - F_{old})\} \tag{12}$$

where $U_{new} - U_{old}$ is the difference in the potential energy for the old and new λ values. $F_{new} - F_{old}$ is the estimated free energy difference based on the current running averages of $\partial U/\partial\lambda$.

Implementation

AIM was implemented in GROMACS as an expanded ensemble calculation. That is, the Hamiltonian must be calculated along with its derivative, and an expanded ensemble step must be performed for every dynamics step. In GROMACS, `nstexpanded` is the number of integration steps between attempted λ moves changing the system Hamiltonian in expanded ensemble simulations. This value must be a multiple of `nstcalcenergy`, the number of steps before calculating the system energy, but can be greater or less than `nstdhdl`, the number of steps before calculating $\partial U/\partial\lambda$ (referred to as dHdλ in GROMACS documentation). For a detailed explanation of all technical terms see reference Abraham et al. (2016). The GROMACS package was further altered to print out the $\partial U/\partial\lambda$ averages computed by AIM to the log file when AIM is used as the `lmc-mover`.

AIM requires the $\partial U/\partial\lambda$ value from every dynamics step to be stored regardless of whether a move in λ space is attempted. Since $\partial U/\partial\lambda$ is only calculated at each step where free energies are calculated, every `nstdhdl` step, we set `nstexpanded` = `nstdhdl` = `nstcalcenergy` = 1 for AIM simulations. This further implies that `lmc-stats` functions were not used during AIM simulations because those functions modify the Hamiltonian which is not needed for AIM.

For the implementation of AIM with GROMACS we follow the outline given in our previous study Ytreberg et al. (2006).

1. Start the simulation from an equilibrated configuration at λ=0 and perform one molecular dynamics step.
2. Randomly choose a trial move in λ space. For example, if our λ spacing is 0.05, a move from λ=0.35 to 0.4 or 0.3 may be attempted but not to 0.45.
3. Calculate the difference in potential energy between the trial and current λ values.
4. Estimate the free energy difference between the trial and current λ values using the running averages of $\partial U/\partial\lambda$ and the trapezoidal rule.
5. Accept λ trial with probability given in Eq. (12).
6. If the move is accepted then λ is updated to the trial value, otherwise the simulations stays at the current λ.
7. The running average of $\partial U/\partial\lambda$ is updated.

Simulation Details

All simulations described in this paper were performed using the molecular dynamics package GROMACS 5.1.4. The simulations were carried out at 300 K and solvated in a dodecahedron box with TIP3P waters. The molecule was parameterized using the OPLS (Optimized Potential for Liquid Simulations) force field (Jorgensen et al., 1996). The OPLS force field was chosen for this study because it is known to perform well on small molecules (Shirts et al., 2003). In future studies, we anticipate using AIM on protein systems where other force fields are more appropriate such as AMBER (Salomon-Ferrer et al., 2013) and CHARMM (Mackerell et al., 2001). Since all molecular dynamics force fields have similar form and number of parameters, it is expected that the performance of AIM would not depend on the force field chosen.

For the GAG to GVG mutations, Na+ and Cl- ions were added to keep the simulation box neutral and reach a physiologically relevant 150 mM salt concentration. Energy minimization was performed using steepest descent for 1000 steps. The system was then equilibrated using simulated annealing for 1000 ps to heat the system from 100K to 300K. For production simulations, electrostatic interactions were handled by Reaction field with a cut-off of 0.9 nm, Potential-shift-Verlet modifier and Verlet cutoff scheme. Van der Waals interactions were handled by twin range cutoffs with neighbor list cutoff of 1.15 nm and van der Waals cutoff of 0.9 nm. The hydrogen bonds were constrained with the Shake algorithm, allowing for a 2 fs time step. Long range dispersion corrections for energy and pressure were applied.

For the standard methods we ran fixed λ alchemical simulations. That is, an equal amount of simulation time was spent at each λ value. For AIM we ran expanded ensemble simulations. During these simulations we first take a molecular dynamics step, then make a trial move in λ space. That is, for AIM the amount of time spent at each λ value is determined by the algorithm.

In order to determine the best distribution of intermediate λ states we followed a simple strategy: (i) Conduct short simulations with a small set of intermediates. (ii) Generate a plot comparing slope values between AIM and fixed λ (iii) Determine the locations of curvature in the estimate of the free energy. (iv) Increase the density of intermediate states in locations of high curvature. (v) Repeat until all areas of high curvature have been well explored.

Methane and GAG to GVG Solvation Free Energy

The first system used here, methane in water, is detailed in systematic studies of force fields and the free energies of hydration of amino acid side chain analogs Sun et al. (1992); Lyubartsev et al. (1996); Chodera and Shirts (2011); Paliwal and Shirts (2011).

For the GAG to GVG mutation the PMX (Gapsys et al., 2015) software package was used to construct the tri-peptide mutation. Using PMX, we generated the hybrid protein structure and topology for simulations of the chosen mutation, alanine to valine.

For both of these systems, we calculated the free energy for decoupling the Lennard-Jones interactions between the atomic sites of the molecule of interest in water. For fixed λ simulations, separate equilibrium simulations of equal length were run in order to represent each of the intermediate λ states. The same values for λ were used in both fixed λ and AIM expanded ensemble simulations. For fixed λ simulations the free energy was estimated using an external tool, alchemical-analysis.py. AIM estimates were calculated using both the trapezoidal rule and cubic-spline. All methods and code are available upon request.

RESULTS

Methane

After conducting short simulations, generating plots to determine locations of high curvature and increasing λ density in those regions, we averaged eight trial simulations of 100 ps per λ for separate λ distributions (see Fig. 1). We found, by progressively increasing the λ density between $\lambda = 0.5$ and $\lambda = 1.0$, that a distribution of 31 λ values gave us a dense enough distribution to properly compare AIM to fixed λ methods for the methane simulations.

Fig. 2 is a violin plot used to visualize the distribution and probability densities over the eight trials for each method as a function of simulation time per value of λ. A violin plot combines a box plot and a density plot to show the shape of the distribution around the mean. The thick black bar in the center represents the interquartile range, the white dot is the median and the thin black line going vertically through the middle represents the upper and lower adjacent values. Reading a violin plot is similar to reading a density plot. The thicker parts represent high frequency values and the thinner parts represent low frequency values. The advantage of a violin plot over a box plot is that we are able to view the underlying distribution of the data.

In Fig. 2, for 31 λ values at 100 ps per λ, the slower convergence of MBAR leads to two separate distributions of converging points. At 500 ps the other methods are beginning to show signs of convergence, however, we see that AIM is forming a second distribution and the width of the distribution of the other methods has not condensed or flattened along the horizontal axis, suggesting convergence has not yet occurred for any of the methods. Despite this, the average of the methods are in agreement within uncertainty (variance in the mean of the estimated free energy). At 1 ns per λ all methods are similarly converged, indicating the longer simulation times has reduced the variance from the mean.

GAG to GVG Mutation

For the GAG to GVG mutation we first tested a distribution of 41 λ values averaged over 8 trial simulations of 1 ps and 100 ps per λ; see Fig. 3. By reviewing the smoothness of the function we concluded that 41 λ values was sufficient and continued the simulations for 1 ns per λ. Fig. 4 shows distribution of the convergence over time for each method. We see that AIM has mostly converged at 100 ps per λ and all methods have similarly converged for 1 ns per λ.

DISCUSSION

In the limit of infinite sampling, all rigorous methods (i.e., statistical mechanics-based methods), performed properly with the same force-field and parameters, will yield the same result within uncertainty. For fixed

λ simulations the sampling time is typically the same for each λ state. Sampling time must be increased whenever convergence has not been achieved. However, if bias is introduced by using an insufficient number of λ values in regions of high curvature, increased sampling leads to radical convergence problems (Shyu and Ytreberg, 2009; Steinbrecher and Labahn, 2010). If the curvature of the underlying free energy slope valuess is large, averaging over a state space that is not dense enough to fully describe the state function propagates this bias requiring significantly increased sampling time to achieve convergence. For TI, the bias will persist even for infinite sampling. In addition, increasing sampling time may not be realistic when dealing with limited computational resources. Paliwal and Shirts (2011) make a detailed argument to why convergence may not be possible for all systems due to hard limitations in computational resources.

In particular, both TI and AIM are calculating the same slope averages and should agree very well for simple systems and reasonably long simulation times. However, due to the fact that AIM spends more time in some regions, we should not expect the approximation of AIM to exactly match TI with similar sampling time until the number of λ values has been sufficiently increased in high curvature regions. Once we have properly chosen the λ values then reasonably long simulations should lead to highly similar results between these two methods.

Since AIM is a Monte Carlo approach, the approximation of a given intermediate state is adaptively expanded whenever a better approximation of the state is needed. This allows AIM to more efficiently sample λ space compared to fixed λ simulations. AIM is able to smooth the underlying free energy function by spending more time at points with large curvature. This means that AIM requires fewer λ states to estimate the free energy of a system because the overall variance of the free energy estimate is minimized by adjusting the sampling for each λ state.

The observant reader may note that AIM violates detailed balance since the acceptance criterion contains the free energy estimates that are updated continuously. AIM does however obey detailed balance asymptotically. As simulation time increases, the average free energy differences between λ values reach an equilibrium and detailed balance is satisfied.

CONCLUSION

In this report we have implemented the adaptive integration method (AIM) for calculating free energy differences in GROMACS and applied it to two molecular systems. We have shown agreement within statistical uncertainty between AIM and a suite of standard fixed λ methods for methane solvation and an GAG to GVG mutation. We have also shown that AIM is more efficient than the other tested methods. That is, for a given amount of simulation time, AIM has a higher level of accuracy and precision compared to other methods.

Further, we found that running longer simulations with too few intermediate λ states generated results that were inconsistent between methods. The density and sampling convergence of the λ states directly influences the agreement between all the tested methods. Since some states will contribute disproportionately to the variance of the estimate, we found that generating short test simulations of different λ densities before attempting longer simulations is advisable.

ACKNOWLEDGMENTS

Support for this research was provided by the National Science Foundation (DEB 1521049 and OIA 1736253) and the Center for Modeling Complex Interactions sponsored by the National Institutes of Health (P20 GM104420). Computer resources were provided in part by the Institute for Bioinformatics and Evolutionary Studies Computational Resources Core sponsored by the National Institutes of Health (P30 GM103324).

REFERENCES

Abraham, M., van der Spoel, D., Lindahl, E., and Hess, B. (2016). *GROMACS User Manual version 5.1.4*.

Aminpour, M., Montemagno, C., and Tuszynski, J. A. (2019). An overview of molecular modeling for drug discovery with specific illustrative examples of applications. *Molecules*, 24(9).

Bennett, C. H. (1976). Efficient Estimation of Free Energy differences from Monte Carlo Data. *Journal of Computational Physics*, 22:245–268.

Berendsen, H. J., van der Spoel, D., and van Drunen, R. (1995). GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1-3):43–56.

Bruckner, S. and Boresch, S. (2011). Efficiency of alchemical free energy simulations. I. A practical comparison of the exponential formula, thermodynamic integration, and Bennett's acceptance ratio method. *Journal of Computational Chemistry*, 32(7):1303–1319.

Chodera, J. D., Mobley, D. L., Shirts, M. R., Dixon, R. W., Branson, K., and Pande, V. S. (2011). Alchemical free energy methods for drug discovery: Progress and challenges. *Current Opinion in Structural Biology*, 21(2):150–160.

Chodera, J. D. and Shirts, M. R. (2011). Replica exchange and expanded ensemble simulations as Gibbs sampling: Simple improvements for enhanced mixing. *Journal of Chemical Physics*, 135(19):0–15.

Cournia, Z., Allen, B., and Sherman, W. (2017). Relative Binding Free Energy Calculations in Drug Discovery: Recent Advances and Practical Considerations. *Journal of Chemical Information and Modeling*, 57(12):2911–2937.

Fasnacht, M., Swendsen, R. H., and Rosenberg, J. M. (2004). Adaptive integration method for Monte Carlo simulations. *Physical Review E*, 69(5):56704.

Gapsys, V., Michielssens, S., Seeliger, D., and De Groot, B. L. (2015). pmx: Automated protein structure and topology generation for alchemical perturbations. *Journal of Computational Chemistry*, 36(5):348–354.

Geng, C., Xue, L. C., Roel-Touris, J., and Bonvin, A. M. (2019). Finding the ΔΔG spot: Are predictors of binding affinity changes upon mutations in protein–protein interactions ready for it? *Wiley Interdisciplinary Reviews: Computational Molecular Science*, (December 2018):1–14.

Gonçalves, P. F. B. and Stassen, H. (2004). Calculation of the free energy of solvation from molecular dynamics simulations. *Pure and Applied Chemistry*, 76(1):231–240.

Hossain, S., Kabedev, A., Parrow, A., Bergström, C. A., and Larsson, P. (2019). Molecular simulation as a computational pharmaceutics tool to predict drug solubility, solubilization processes and partitioning. *European Journal of Pharmaceutics and Biopharmaceutics*, 137(February):46–55.

Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236.

Kaus, J. W., Arrar, M., and McCammon, J. A. (2014). Accelerated adaptive integration method. *Journal of Physical Chemistry B*, 118(19):5109–5118.

Kaus, J. W. and Mccammon, J. A. (2015). Enhanced ligand sampling for relative protein-ligand binding free energy calculations. *Journal of Physical Chemistry B*, 119(20):6190–6197.

Klimovich, P. V., Shirts, M. R., and Mobley, D. L. (2015). Guidelines for the analysis of free energy calculations. *Journal of Computer-Aided Molecular Design*.

Kofke, D. A. (2005). Free energy methods in molecular simulation. *Fluid Phase Equilibria*, 228-229:41–48.

Lyubartsev, A. P., Førrisdahl, O. K., and Laaksonen, A. (1996). Calculations of solvation free energies by expanded ensemble method . *2nd International Conference on Natural Gas Hydrates. Toulouse (France), June 2-6, 1996*, (6):311–318.

Mackerell, A. D., Banavali, N., and Foloppe, N. (2001). Development and Current Status of the CHARMM Force Field for Nucleic Acids. *Biopolymers (Nucleic Acid Sciences*, 56:257–265.

Miller, C. R., Lee, K. H., Wichman, H. A., and Ytreberg, F. M. (2014). Changing folding and binding stability in a viral coat protein: A comparison between substitutions accessible through mutation and those fixed by natural selection. *PLoS ONE*, 9(11).

Mobley, D. L., Liu, S., Cerutti, D. S., Swope, W. C., and Rice, J. E. (2012). Alchemical prediction of hydration free energies for SAMPL. *Journal of Computer-Aided Molecular Design*, 26(5):551–562.

Paliwal, H. and Shirts, M. R. (2011). A benchmark test set for alchemical free energy transformations and its use to quantify error in common free energy methods. *Journal of Chemical Theory and Computation*, 7(12):4115–4134.

Petukh, M., Li, M., and Alexov, E. (2015). Predicting Binding Free Energy Change Caused by Point Mutations with Knowledge-Modified MM/PBSA Method. *PLoS Computational Biology*, 11(7):1–23.

Salomon-Ferrer, R., Case, D. A., and Walker, R. C. (2013). An overview of the Amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(2):198–210.

Shirts, M. R. and Chodera, J. D. (2008). Statistically optimal analysis of samples from multiple equilibrium

states. *Journal of Chemical Physics*, 129(12).

Shirts, M. R. and Mobley, D. L. (2013). An Introduction to Best Practices in Free Energy Calculations. In L., M. and E., S., editors, *Biomolecular Simulations. Methods in Molecular Biology (Methods and Protocols)*, volume 924, pages 271–311. Humana Press, Totowa, NJ.

Shirts, M. R., Mobley, D. L., and Chodera, J. D. (2007). Chapter 4 Alchemical Free Energy Calculations: Ready for Prime Time? *Annual Reports in Computational Chemistry*, 3(February 2016):41–59.

Shirts, M. R. and Pande, V. S. (2005). Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. *Journal of Chemical Physics*, 122(14):144107.

Shirts, M. R., Pitera, J. W., Swope, W. C., and Pande, V. S. (2003). Extremely precise free energy calculations of amino acid side chain analogs: Comparison of common molecular mechanics force fields for proteins. *Journal of Chemical Physics*, 119(11):5740–5761.

Shyu, C. and Ytreberg, F. M. (2009). Reducing the bias and uncertainty of free energy estimates by using regression to fit thermodynamic integration data. *Journal of Computational Chemistry*, 30(14):2297–2304.

Śledź, P. and Caflisch, A. (2018). Protein structure-based drug design: from docking to molecular dynamics. *Current Opinion in Structural Biology*, 48:93–102.

Steinbrecher, T. and Labahn, A. (2010). Towards accurate free energy calculations in ligand protein-binding studies. *Current medicinal chemistry*, 17(8):767–785.

Sun, Y., Spellmeyer, D., Pearlman, D. A., and Kollman, P. (1992). Simulation of the solvation free energies for methane, ethane, and propane and corresponding amino acid dipeptides: a critical test of the bond-PMF correction, a new set of hydrocarbon parameters, and the gas phase-water hydrophobicity scale. *Journal of the American Chemical Society*, 114(17):6798–6801.

Wichman, H. A., Miura, T. A., Ytreberg, F. M., Johnson, E. L., Burke, A. Z., Brown, C. J., Miller, C. R., and Martin, K. P. (2016). Initiating a watch list for Ebola virus antibody escape mutations. *PeerJ*, 4:e1674.

Ytreberg, F. M., Swendsen, R. H., and Zuckerman, D. M. (2006). Comparison of free energy methods for molecular systems. *Journal of Chemical Physics*, 125(18):1–11.

Zhan, Y. A., Wu, H., Powell, A. T., Daughdrill, G. W., and Ytreberg, F. M. (2013). Impact of the K24N mutation on the transactivation domain of p53 and its binding to murine double-minute clone 2. *Proteins: Structure, Function and Bioinformatics*, 81(10):1738–1747.

Zhan, Y. A. and Ytreberg, F. M. (2015). The cis conformation of proline leads to weaker binding of a p53 peptide to MDM2 compared to trans. *Archives of Biochemistry and Biophysics*, 575:22–29.

Zhang, H., Liao, L., Saravanan, K. M., Yin, P., and Wei, Y. (2019). DeepBindRG: a deep learning based method for estimating effective protein–ligand affinity. *PeerJ*, 7:e7362.

Zwanzig, R. W. (1954). High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases. *Journal of Chemical Physics*, 22(8):1420–1426.

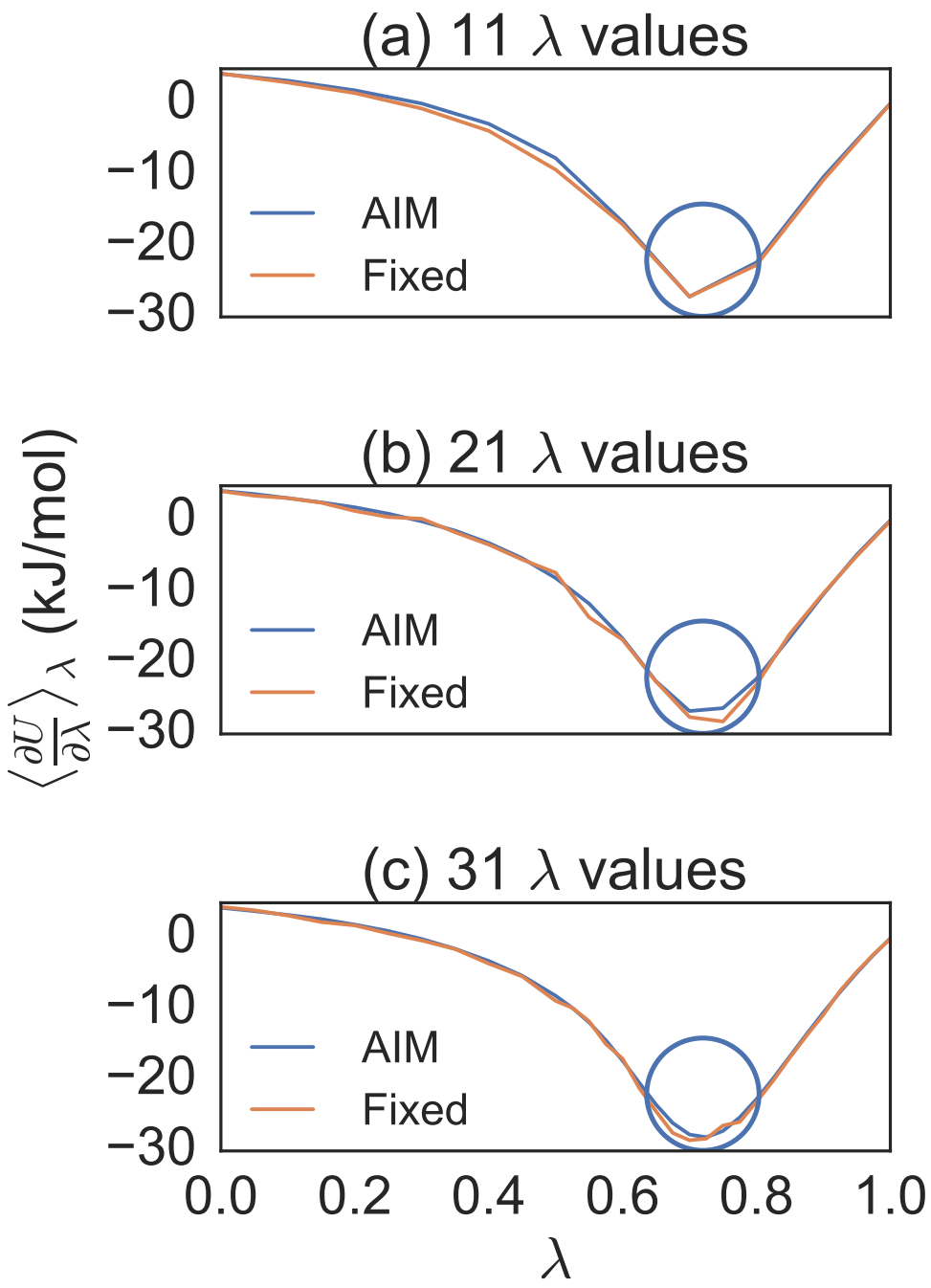

Figure 1. Different λ densities for methane solvation free energy calculations. Eight trial simulations of 100 ps per λ for 11, 21 and 31 λ values. This shows how the number of λ values were chosen to effectively compare AIM to fixed λ simulations. The circles indicate the region where the λ density needed to be increased.

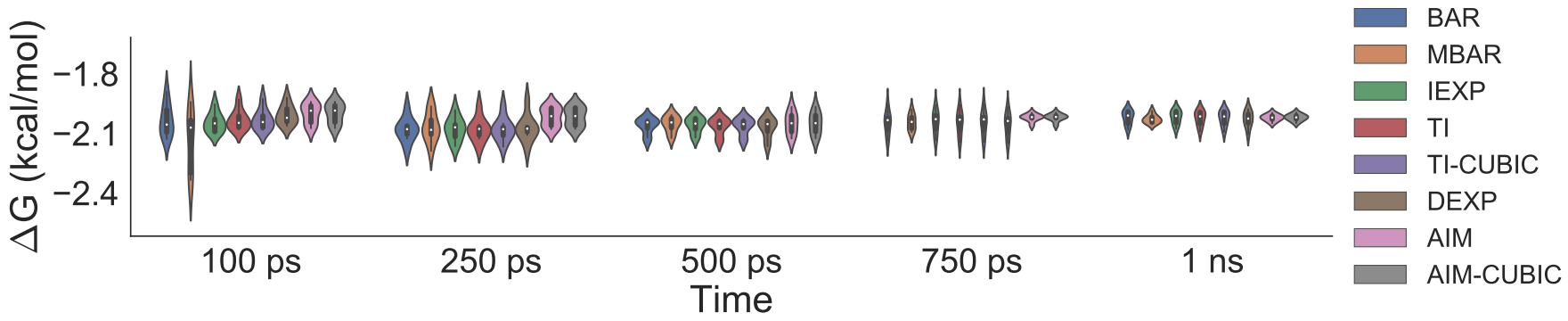

Figure 2. Violin plot showing methane solvation results for 31 λ values averaged over eight trials. A violin plot combines a box plot and a density plot to visualize the distribution and probability density. The graphic shows all methods have similarly converged at 1 ns per λ. AIM and AIM-CUBIC converge earlier than other methods at 750 ps per λ.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27935v1 | CC BY 4.0 Open Access | rec: 4 Sep 2019, publ: 4 Sep 2019

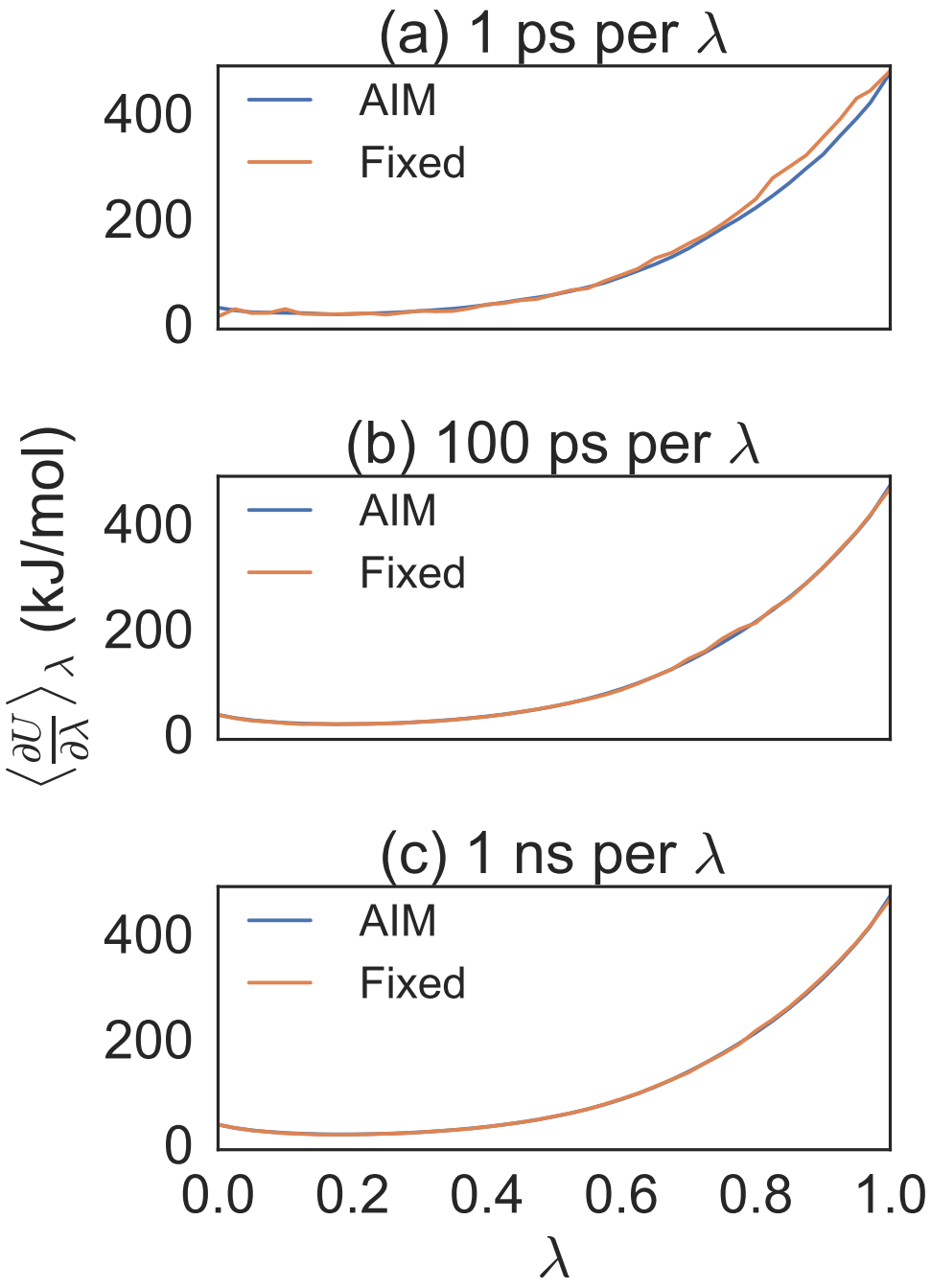

Figure 3. Different simulation times for alanine to valine mutation free energy calculations. Eight trial simulations of 41 λ values at 1 ps, 100 ps and 1 ns per λ. Note the smoothness of AIM versus fixed λ simulations. AIM requires less samples than fixed λ simulations to smooth the free energy function.

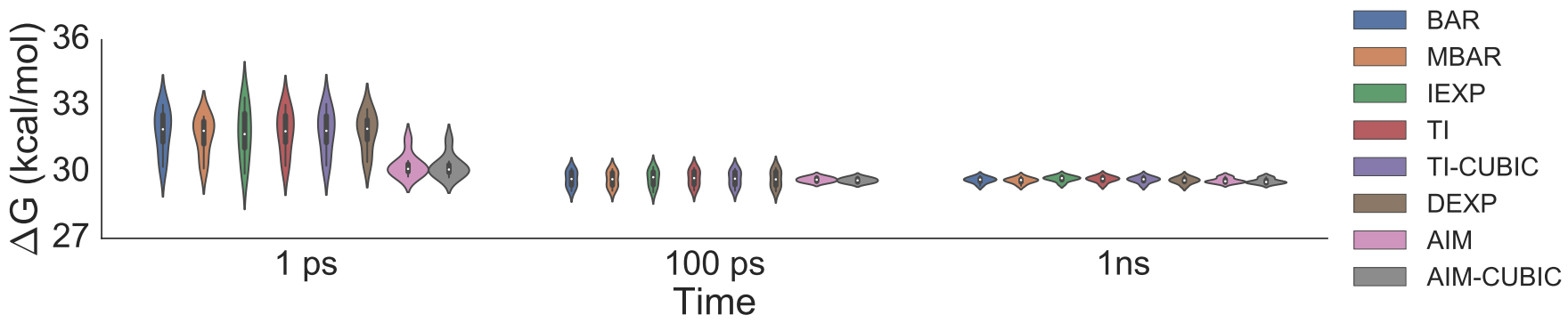

Figure 4. Violin plot showing alanine to valine mutation results for 41 λ values averaged over eight trials. The graphic shows all methods have similarly converged at 1 ns per λ. AIM and AIM-CUBIC converge more rapidly than other methods and are mostly converged at 100 ps per λ

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27935v1 | CC BY 4.0 Open Access | rec: 4 Sep 2019, publ: 4 Sep 2019