

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

[View the peer-reviewed version](https://peerj.com/articles/cs-264) (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 Mirabzadeh¹, 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.

1 Implementation of Adaptive Integration 2 Method for Free Energy Calculations in 3 Molecular Systems

4 CA Mirabzadeh¹ and FM Ytreberg²

5 ^{1,2}Department of Physics, University of Idaho, Moscow ID

6 Corresponding author:

7 FM Ytreberg²

8 Email address: ytreberg@uidaho.edu

9 ABSTRACT

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

20 INTRODUCTION

21 Measuring free energy differences using computer simulations can be computationally expensive, yet is
22 useful for many different applications (see e.g., Steinbrecher and Labahn (2010); Chodera et al. (2011);
23 Mobley et al. (2012); Zhan et al. (2013); Miller et al. (2014); Petukh et al. (2015); Zhan and Ytreberg
24 (2015); Wichman et al. (2016); Cournia et al. (2017); Hossain et al. (2019); Aminpour et al. (2019)).
25 Specific examples include determining protein conformational preferences, virtual screening for drug
26 design or drug discovery (Steinbrecher and Labahn, 2010; Chodera et al., 2011; Zhan and Ytreberg,
27 2015; Śledź and Caffisch, 2018; Aminpour et al., 2019; Zhang et al., 2019). Of specific relevance to the
28 current study is that free energy calculations allow prediction of how amino acid mutations may modify
29 protein-protein binding (Zhan et al., 2013; Miller et al., 2014; Petukh et al., 2015; Wichman et al., 2016;
30 Geng et al., 2019). We are particularly interested in developing and implementing efficient methods
31 for calculating free energy differences and using them to understand how amino acid mutations modify
32 protein-protein and protein-substrate interactions.

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

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

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

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

59 METHODS

60 For this study we performed alchemical free energy simulations where the system is changed from a
 61 reference state to an end state by constructing a reaction pathway that modifies, adds or removes atoms.
 62 Such alchemical simulations are non-physical, i.e., the simulation does not represent what could occur
 63 naturally. Since the free energy is a state variable, it is independent of the path taken, and we may provide
 64 any path we wish. To perform these simulations the reaction pathway is divided into many separate,
 65 non-physical, λ states between a reference state and an end state. The λ states represent the progress
 66 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,

$$F = U - TS, \quad (1)$$

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 ,

$$\Delta F \equiv \Delta F_{A \rightarrow B} = F_B - F_A = \Delta U - T \Delta S. \quad (2)$$

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,

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

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

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

67 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,

$$0 \leq \lambda \leq 1. \quad (5)$$

68 This pathway connects the two end states of the system. In the case of poor overlap, where the end states
 69 may be separated by a high energy barrier, $|U_B - U_A| \gg k_B T$, this pathway mitigates the otherwise very
 70 slow convergence of free energy estimates (Shirts et al., 2007). Care should be taken when choosing
 71 intermediate states such that there is adequate overlap in the conformation space between the end states
 72 (Shirts et al., 2007; Klimovich et al., 2015). For our simulations the number of λ values and time per λ
 73 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,

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

74 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
75 configuration for λ_i . Unlike some other methods, exponential averaging has an exact solution since it
76 is only used to evaluate the difference between two states. However, it is the least efficient method and
77 should not be used if difference in potential energies are much larger than $k_B T$ Shirts and Pande (2005).
78 In addition, exponential averaging can be noisy, biased and dependent on the tails of the distribution of λ
79 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 ,

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

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

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

80 where the $\langle \cdot \rangle_{\lambda}$ notation represents the ensemble average at a given intermediate state, λ . The free energy
81 is estimated by numerically integrating Eq. (8) after running equilibrium simulations at each intermediate
82 λ state. Since numerical integration is required, TI can be biased by the chosen method of integration.
83 Some of that bias can be removed by using cubic-spline interpolation or more complex integration
84 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 ,

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

C is a shift constant,

$$C = k_B T \ln \frac{n_j}{n_i}, \quad (10)$$

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

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

85 Equation (9) is the ratio of canonical averages of two different potentials U_i and U_j acting on the same
86 configuration space meaning it requires information from two neighboring states. However, this limitation
87 is not too much of a concern with a trivial coordinate transformation or when using dummy coordinates in
88 alchemical simulations. MBAR, an extension of BAR, differs in that it takes data from more than two
89 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

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

90 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
91 estimated free energy difference based on the current running averages of $\partial U / \partial \lambda$.

92 Implementation

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

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

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

- 109 1. Start the simulation from an equilibrated configuration at $\lambda=0$ and perform one molecular dynamics
110 step.
- 111 2. Randomly choose a trial move in λ space. For example, if our λ spacing is 0.05, a move from
112 $\lambda=0.35$ to 0.4 or 0.3 may be attempted but not to 0.45.
- 113 3. Calculate the difference in potential energy between the trial and current λ values.
- 114 4. Estimate the free energy difference between the trial and current λ values using the running averages
115 of $\partial U/\partial\lambda$ and the trapezoidal rule.
- 116 5. Accept λ trial with probability given in Eq. (12).
- 117 6. If the move is accepted then λ is updated to the trial value, otherwise the simulations stays at the
118 current λ .
- 119 7. The running average of $\partial U/\partial\lambda$ is updated.

120 Simulation Details

121 All simulations described in this paper were performed using the molecular dynamics package GROMACS
122 5.1.4. The simulations were carried out at 300 K and solvated in a dodecahedron box with TIP3P waters.
123 The molecule was parameterized using the OPLS (Optimized Potential for Liquid Simulations) force
124 field (Jorgensen et al., 1996). The OPLS force field was chosen for this study because it is known to
125 perform well on small molecules (Shirts et al., 2003). In future studies, we anticipate using AIM on
126 protein systems where other force fields are more appropriate such as AMBER (Salomon-Ferrer et al.,
127 2013) and CHARMM (Mackerell et al., 2001). Since all molecular dynamics force fields have similar
128 form and number of parameters, it is expected that the performance of AIM would not depend on the
129 force field chosen.

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

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

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

147 **Methane and GAG to GVG Solvation Free Energy**

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

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

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

161 **RESULTS**

162 **Methane**

163 After conducting short simulations, generating plots to determine locations of high curvature and in-
164 creasing λ density in those regions, we averaged eight trial simulations of 100 ps per λ for separate λ
165 distributions (see Fig. 1). We found, by progressively increasing the λ density between $\lambda = 0.5$ and $\lambda =$
166 1.0 , that a distribution of 31 λ values gave us a dense enough distribution to properly compare AIM to
167 fixed λ methods for the methane simulations.

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

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

183 **GAG to GVG Mutation**

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

189 **DISCUSSION**

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

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

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

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

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

218 CONCLUSION

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

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

230 ACKNOWLEDGMENTS

231 Support for this research was provided by the National Science Foundation (DEB 1521049 and OIA
232 1736253) and the Center for Modeling Complex Interactions sponsored by the National Institutes of
233 Health (P20 GM104420). Computer resources were provided in part by the Institute for Bioinformatics
234 and Evolutionary Studies Computational Resources Core sponsored by the National Institutes of Health
235 (P30 GM103324).

236 REFERENCES

- 237 Abraham, M., van der Spoel, D., Lindahl, E., and Hess, B. (2016). *GROMACS User Manual version*
238 *5.1.4*.
- 239 Aminpour, M., Montemagno, C., and Tuszynski, J. A. (2019). An overview of molecular modeling for
240 drug discovery with specific illustrative examples of applications. *Molecules*, 24(9).
- 241 Bennett, C. H. (1976). Efficient Estimation of Free Energy differences from Monte Carlo Data. *Journal*
242 *of Computational Physics*, 22:245–268.

- 243 Berendsen, H. J., van der Spoel, D., and van Drunen, R. (1995). GROMACS: A message-passing parallel
244 molecular dynamics implementation. *Computer Physics Communications*, 91(1-3):43–56.
- 245 Bruckner, S. and Boresch, S. (2011). Efficiency of alchemical free energy simulations. I. A practical
246 comparison of the exponential formula, thermodynamic integration, and Bennett's acceptance ratio
247 method. *Journal of Computational Chemistry*, 32(7):1303–1319.
- 248 Chodera, J. D., Mobley, D. L., Shirts, M. R., Dixon, R. W., Branson, K., and Pande, V. S. (2011).
249 Alchemical free energy methods for drug discovery: Progress and challenges. *Current Opinion in*
250 *Structural Biology*, 21(2):150–160.
- 251 Chodera, J. D. and Shirts, M. R. (2011). Replica exchange and expanded ensemble simulations as Gibbs
252 sampling: Simple improvements for enhanced mixing. *Journal of Chemical Physics*, 135(19):0–15.
- 253 Courmia, Z., Allen, B., and Sherman, W. (2017). Relative Binding Free Energy Calculations in Drug
254 Discovery: Recent Advances and Practical Considerations. *Journal of Chemical Information and*
255 *Modeling*, 57(12):2911–2937.
- 256 Fasnacht, M., Swendsen, R. H., and Rosenberg, J. M. (2004). Adaptive integration method for Monte
257 Carlo simulations. *Physical Review E*, 69(5):56704.
- 258 Gapsys, V., Michielssens, S., Seeliger, D., and De Groot, B. L. (2015). pmx: Automated protein
259 structure and topology generation for alchemical perturbations. *Journal of Computational Chemistry*,
260 36(5):348–354.
- 261 Geng, C., Xue, L. C., Roel-Touris, J., and Bonvin, A. M. (2019). Finding the $\Delta\Delta G$ spot: Are predic-
262 tors of binding affinity changes upon mutations in protein–protein interactions ready for it? *Wiley*
263 *Interdisciplinary Reviews: Computational Molecular Science*, (December 2018):1–14.
- 264 Gonçalves, P. F. B. and Stassen, H. (2004). Calculation of the free energy of solvation from molecular
265 dynamics simulations. *Pure and Applied Chemistry*, 76(1):231–240.
- 266 Hossain, S., Kabedev, A., Parrow, A., Bergström, C. A., and Larsson, P. (2019). Molecular simulation as
267 a computational pharmaceuticals tool to predict drug solubility, solubilization processes and partitioning.
268 *European Journal of Pharmaceutics and Biopharmaceutics*, 137(February):46–55.
- 269 Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and testing of the OPLS
270 all-atom force field on conformational energetics and properties of organic liquids. *Journal of the*
271 *American Chemical Society*, 118(45):11225–11236.
- 272 Kaus, J. W., Arrar, M., and McCammon, J. A. (2014). Accelerated adaptive integration method. *Journal*
273 *of Physical Chemistry B*, 118(19):5109–5118.
- 274 Kaus, J. W. and McCammon, J. A. (2015). Enhanced ligand sampling for relative protein-ligand binding
275 free energy calculations. *Journal of Physical Chemistry B*, 119(20):6190–6197.
- 276 Klimovich, P. V., Shirts, M. R., and Mobley, D. L. (2015). Guidelines for the analysis of free energy
277 calculations. *Journal of Computer-Aided Molecular Design*.
- 278 Kofke, D. A. (2005). Free energy methods in molecular simulation. *Fluid Phase Equilibria*, 228-229:41–
279 48.
- 280 Lyubartsev, A. P., Førrisdahl, O. K., and Laaksonen, A. (1996). Calculations of solvation free energies
281 by expanded ensemble method. *2nd International Conference on Natural Gas Hydrates. Toulouse*
282 *(France), June 2-6, 1996*, (6):311–318.
- 283 Mackerell, A. D., Banavali, N., and Foloppe, N. (2001). Development and Current Status of the CHARMM
284 Force Field for Nucleic Acids. *Biopolymers (Nucleic Acid Sciences)*, 56:257–265.
- 285 Miller, C. R., Lee, K. H., Wichman, H. A., and Ytreberg, F. M. (2014). Changing folding and binding
286 stability in a viral coat protein: A comparison between substitutions accessible through mutation and
287 those fixed by natural selection. *PLoS ONE*, 9(11).
- 288 Mobley, D. L., Liu, S., Cerutti, D. S., Swope, W. C., and Rice, J. E. (2012). Alchemical prediction of
289 hydration free energies for SAMPL. *Journal of Computer-Aided Molecular Design*, 26(5):551–562.
- 290 Paliwal, H. and Shirts, M. R. (2011). A benchmark test set for alchemical free energy transformations and
291 its use to quantify error in common free energy methods. *Journal of Chemical Theory and Computation*,
292 7(12):4115–4134.
- 293 Petukh, M., Li, M., and Alexov, E. (2015). Predicting Binding Free Energy Change Caused by Point
294 Mutations with Knowledge-Modified MM/PBSA Method. *PLoS Computational Biology*, 11(7):1–23.
- 295 Salomon-Ferrer, R., Case, D. A., and Walker, R. C. (2013). An overview of the Amber biomolecular
296 simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(2):198–210.
- 297 Shirts, M. R. and Chodera, J. D. (2008). Statistically optimal analysis of samples from multiple equilibrium

- 298 states. *Journal of Chemical Physics*, 129(12).
- 299 Shirts, M. R. and Mobley, D. L. (2013). An Introduction to Best Practices in Free Energy Calculations.
300 In L., M. and E., S., editors, *Biomolecular Simulations. Methods in Molecular Biology (Methods and*
301 *Protocols)*, volume 924, pages 271–311. Humana Press, Totowa, NJ.
- 302 Shirts, M. R., Mobley, D. L., and Chodera, J. D. (2007). Chapter 4 Alchemical Free Energy Calculations:
303 Ready for Prime Time? *Annual Reports in Computational Chemistry*, 3(February 2016):41–59.
- 304 Shirts, M. R. and Pande, V. S. (2005). Comparison of efficiency and bias of free energies computed
305 by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. *Journal of*
306 *Chemical Physics*, 122(14):144107.
- 307 Shirts, M. R., Pitner, J. W., Swope, W. C., and Pande, V. S. (2003). Extremely precise free energy
308 calculations of amino acid side chain analogs: Comparison of common molecular mechanics force
309 fields for proteins. *Journal of Chemical Physics*, 119(11):5740–5761.
- 310 Shyu, C. and Ytreberg, F. M. (2009). Reducing the bias and uncertainty of free energy estimates by using
311 regression to fit thermodynamic integration data. *Journal of Computational Chemistry*, 30(14):2297–
312 2304.
- 313 Śledź, P. and Caflisch, A. (2018). Protein structure-based drug design: from docking to molecular
314 dynamics. *Current Opinion in Structural Biology*, 48:93–102.
- 315 Steinbrecher, T. and Labahn, A. (2010). Towards accurate free energy calculations in ligand protein-
316 binding studies. *Current medicinal chemistry*, 17(8):767–785.
- 317 Sun, Y., Spellmeyer, D., Pearlman, D. A., and Kollman, P. (1992). Simulation of the solvation free
318 energies for methane, ethane, and propane and corresponding amino acid dipeptides: a critical test of
319 the bond-PMF correction, a new set of hydrocarbon parameters, and the gas phase-water hydrophobicity
320 scale. *Journal of the American Chemical Society*, 114(17):6798–6801.
- 321 Wichman, H. A., Miura, T. A., Ytreberg, F. M., Johnson, E. L., Burke, A. Z., Brown, C. J., Miller, C. R.,
322 and Martin, K. P. (2016). Initiating a watch list for Ebola virus antibody escape mutations. *PeerJ*,
323 4:e1674.
- 324 Ytreberg, F. M., Swendsen, R. H., and Zuckerman, D. M. (2006). Comparison of free energy methods for
325 molecular systems. *Journal of Chemical Physics*, 125(18):1–11.
- 326 Zhan, Y. A., Wu, H., Powell, A. T., Daughdrill, G. W., and Ytreberg, F. M. (2013). Impact of the
327 K24N mutation on the transactivation domain of p53 and its binding to murine double-minute clone 2.
328 *Proteins: Structure, Function and Bioinformatics*, 81(10):1738–1747.
- 329 Zhan, Y. A. and Ytreberg, F. M. (2015). The cis conformation of proline leads to weaker binding of a p53
330 peptide to MDM2 compared to trans. *Archives of Biochemistry and Biophysics*, 575:22–29.
- 331 Zhang, H., Liao, L., Saravanan, K. M., Yin, P., and Wei, Y. (2019). DeepBindRG: a deep learning based
332 method for estimating effective protein–ligand affinity. *PeerJ*, 7:e7362.
- 333 Zwanzig, R. W. (1954). High-Temperature Equation of State by a Perturbation Method. I. Nonpolar
334 Gases. *Journal of Chemical Physics*, 22(8):1420–1426.

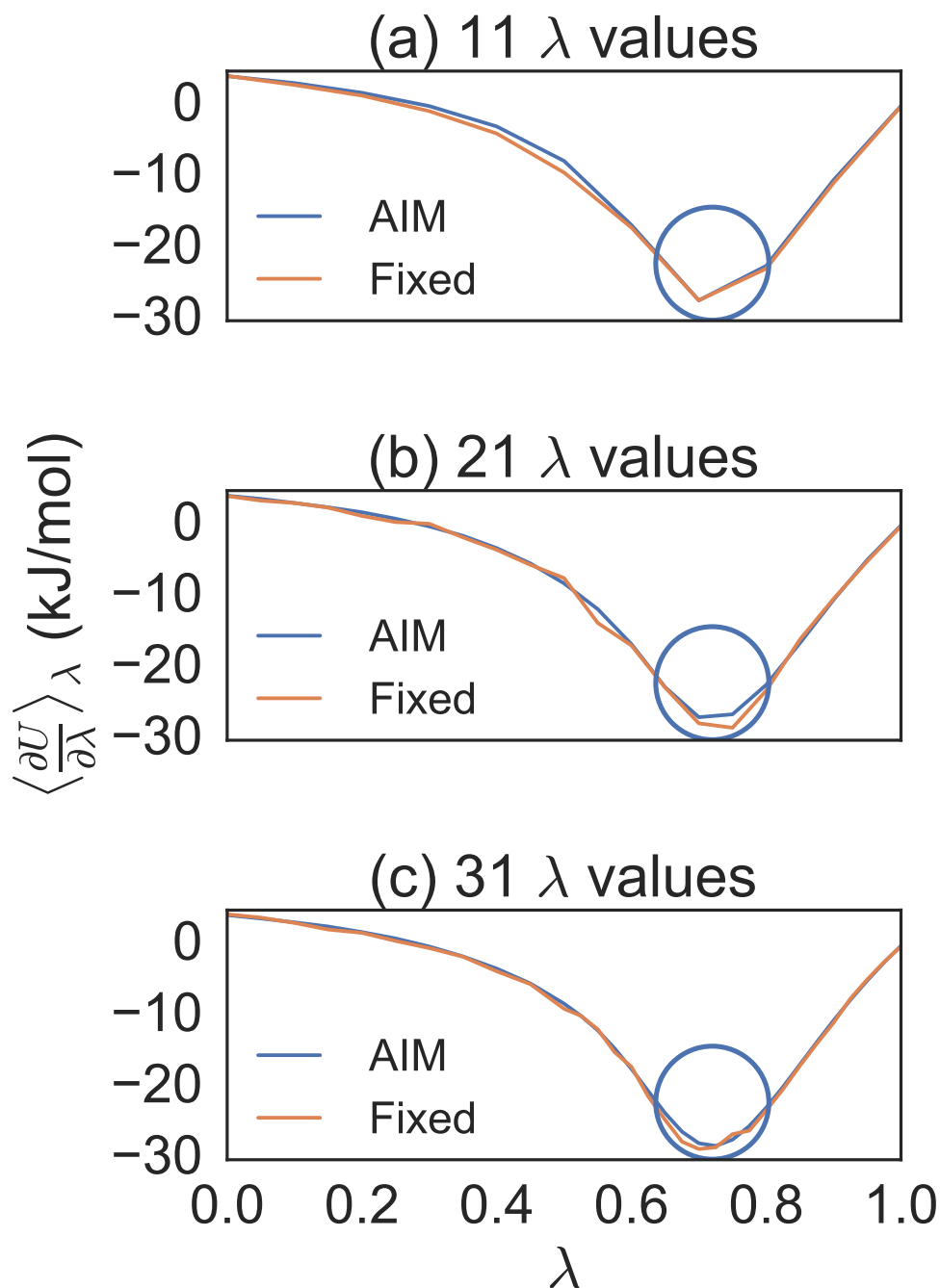


Figure 1. Different λ densities for methane solvation free energy calculations. Eight trial simulations of 100 ps per λ for 11, 21 and 31 λ values. This shows how the number of λ values were chosen to effectively compare AIM to fixed λ simulations. The circles indicate the region where the λ density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

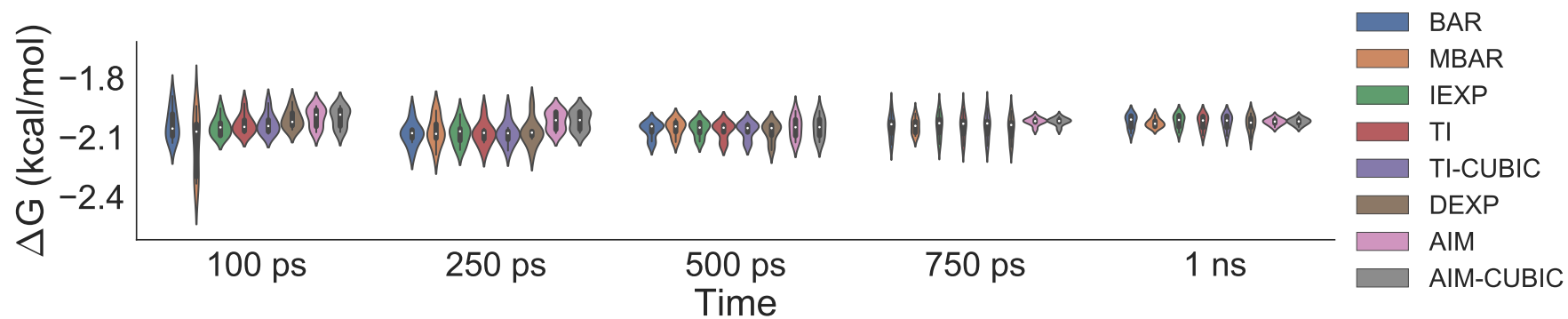


Figure 2. Violin plot showing methane solvation results for 31 λ values averaged over eight trials. A violin plot combines a box plot and a density plot to visualize the distribution and probability density. The graphic shows all methods have similarly converged at 1 ns per λ . AIM and AIM-CUBIC converge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

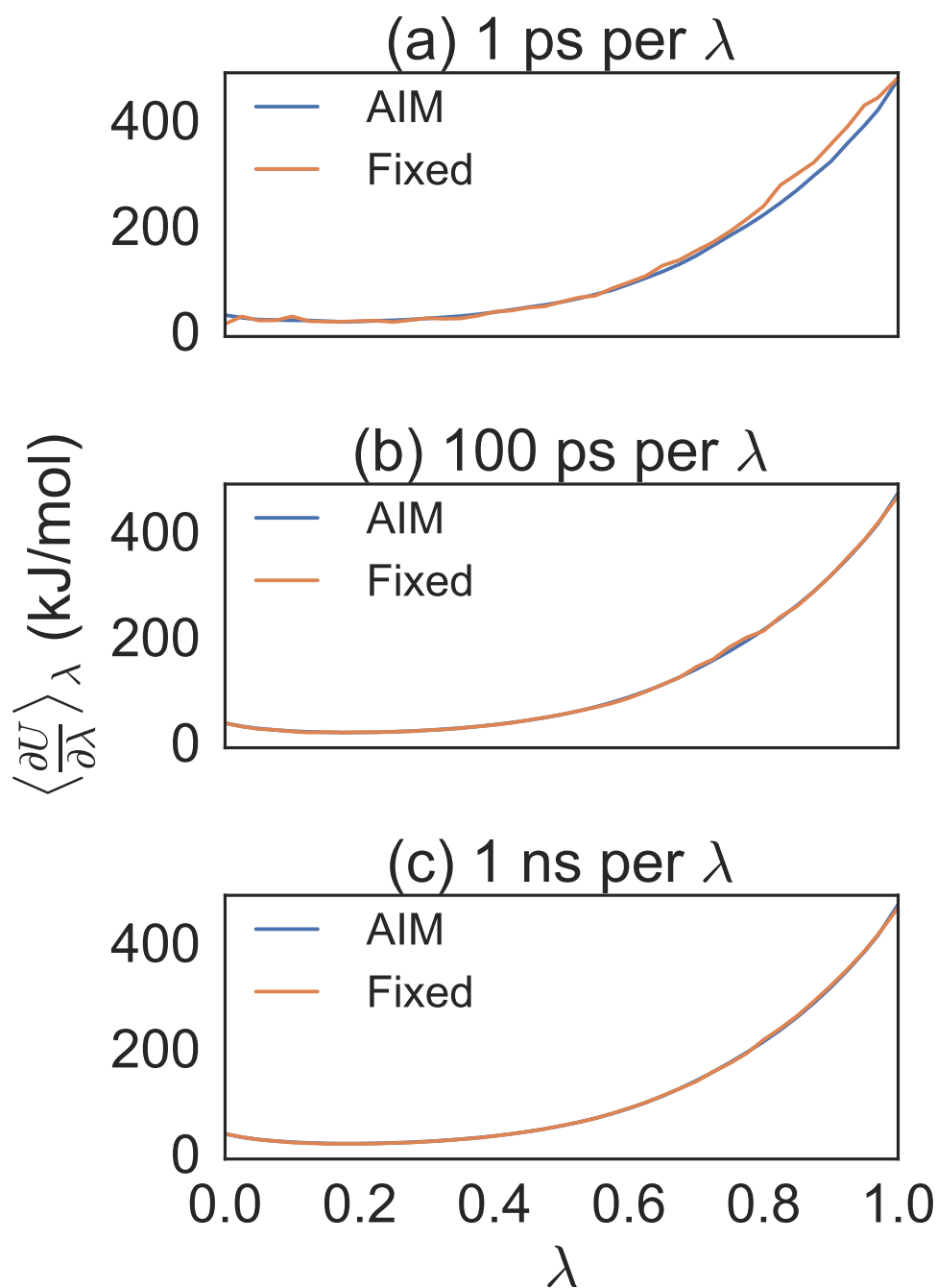


Figure 3. Different simulation times for alanine to valine mutation free energy calculations. Eight trial simulations of 41 λ values at 1 ps, 100 ps and 1 ns per λ . Note the smoothness of AIM versus fixed λ simulations. AIM requires less samples than fixed λ simulations to smooth the free energy function.

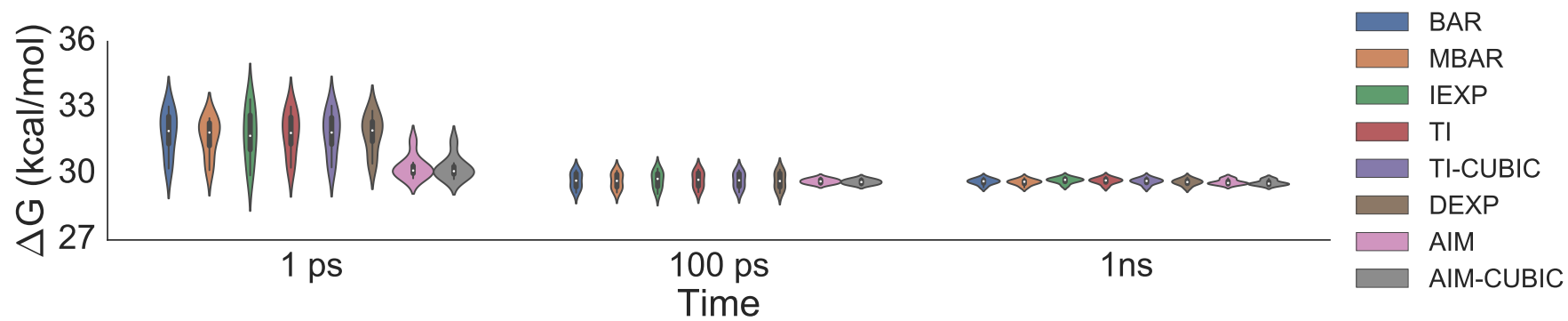


Figure 4. Violin plot showing alanine to valine mutation results for 41 λ values averaged over eight trials. The graphic shows all methods have similarly converged at 1 ns per λ . AIM and AIM-CUBIC converge more rapidly than other methods and are mostly converged at 100 ps per λ