JCTC Journal of Chemical Theory and Computation

Efficient Generalized Born Models for Monte Carlo Simulations

Julien Michel,[†] Richard D. Taylor,[‡] and Jonathan W. Essex*,[†]

School of Chemistry, University of Southampton, Highfield, Southampton SO17 1BJ, U.K., and Astex Therapeutics Ltd., 436 Cambridge Science Park, Cambridge CB4 0QA, U.K.

Received February 17, 2006

Abstract: The Generalized Born Surface Area theory (GBSA) has become a popular method to model the solvation of biomolecules. While efficient in the context of molecular dynamics simulations, GBSA calculations do not integrate well with Monte Carlo simulations because of the nonlocal nature of the Generalized Born energy. We present a method by which Monte Carlo Generalized Born simulations can be made seven to eight times faster on a protein−ligand binding free energy calculation with little or no loss of accuracy. The method can be employed in any type of Monte Carlo or Hybrid Monte Carlo-molecular dynamics simulation and should prove useful in numerous applications.

Introduction

A proper representation of water is essential for the study of numerous important biomolecular processes. The traditional microscopic approach explicitly models thousands of water molecules to solvate a protein and then applies periodic boundary conditions to the whole system. While rigorous, this approach entails a heavy computational cost. Continuum electrostatics provides an elegant way to model the influence of the solvent on the solute structure and dynamics. In this approach the solvent is considered as a high dielectric continuum that is interacting with partial charges that are embedded in a solute molecule of lower dielectric. The solute response to the reaction field of the solvent dielectric can then be modeled by applying laws of classical electrostatics. The dramatic reduction in the number of degrees of freedom of the simulated system increases the rate of convergence of properties that are averaged over many snapshots. Among the many flavors of continuum electrostatics theories that have been developed in the last two decades, the Generalized Born Surface Area (GBSA) method has become popular in biomolecular simulations.[1] This is in no doubt due to the efficiency of the method and its reasonable accuracy. Furthermore, under certain approximations, the GBSA theory can be expressed in a pairwise form that can be implemented relatively easily within traditional biomolecular simulation packages.[2−4]

In a GBSA simulation, the force field is augmented with an extra term, the solvation energy ΔG_{solv}, taken as the sum of a polarization energy ΔG_{pol} and a nonpolar energy term ΔG_{nonpol}.

$$\Delta G_{solv} = \Delta G_{pol} + \Delta G_{nonpol} \tag{1}$$

In eq 1 ΔG_{nonpol} is often taken as proportional to the solvent accessible surface area (SASA) of the system. Alternative more elaborate treatments have recently been proposed in the literature.[3] The term ΔG_{pol} is obtained from eq 2.

$$\Delta G_{pol} = -\frac{1}{2}\left(\frac{1}{\epsilon_{vac}} - \frac{1}{\epsilon_{solv}}\right)\sum_i\sum_j \frac{q_i q_j}{\sqrt{r^2_{ij} + B_i B_j e^{(-r^2_{ij}/4B_iB_j)}}} \tag{2}$$

ϵ_{vac} and ϵ_{solv} are the dielectric constants of the vacuum and the solvent respectively, q_i is the atomic partial charge of atom i, r_{ij} is the distance between a pair of atoms ij, and B_i is the effective Born radius of atom i.

The effective Born radius B_i is in essence the spherically averaged distance of the solute atom to the solvent. Modifications to eq 2 have been occasionally proposed,[5] but much of the improvements in the accuracy of the Generalized Born

* Corresponding author e-mail: J.W.Essex@soton.ac.uk.
[†] University of Southampton.
[‡] Astex Therapeutics Ltd.

10.1021/ct600069r CCC: $33.50
Published on Web 04/08/2006

models has come from a better calculation of the Born radii.[6−8] One method to compute this quantity is the Pairwise Descreening Approximation (PDA) of Hawkins et al.[2] described by eq 3. While the PDA is not the most accurate way to calculate a Born radius, it is one of the fastest and has proven popular.

$$\frac{1}{B_i} = \frac{1}{\alpha_i} - \frac{1}{2}\sum_{j\neq i}\frac{1}{L_{ij}} - \frac{1}{U_{ij}} + \frac{r_{ij}}{4}\left(\frac{1}{U_{ij}^2} - \frac{1}{L_{ij}^2}\right) + \frac{1}{2r_{ij}}\ln\frac{L_{ij}}{U_{ij}} + \frac{S_j^2\alpha_j^2}{4r_{ij}}\left(\frac{1}{L_{ij}^2} - \frac{1}{U_{ij}^2}\right) \quad (3)$$

$$L_{ij} = 1 \text{ if } r_{ij} + S_j\alpha_j \leq \alpha_i$$

$$L_{ij} = \alpha_i \text{ if } r_{ij} - S_j\alpha_j \leq \alpha_i < r_{ij} + S_j\alpha_j$$

$$L_{ij} = r_{ij} - S_j\alpha_j \text{ if } \alpha_i \leq r_{ij} - S_j\alpha_j$$

$$U_{ij} = 1 \text{ if } r_{ij} + S_j\alpha_j \leq \alpha_j$$

$$U_{ij} = r_{ij} + S_j\alpha_j \text{ if } \alpha i < r_{ij} + S_j\alpha_j$$

In eq 3 r_{ij} is the distance between a pair of atoms ij and α_i is the intrinsic Born radius of atom i, that is, the Born radius that atom i would adopt if it were completely isolated. Finally S_j is a scaling factor which compensates for systematic errors introduced by this approximate Born radii calculation.

GBSA is widely used in the context of molecular dynamics simulations. For example, some interesting studies of GBSA molecular dynamics simulations of RNAs are discussed by Sorin et al.,[9,10] while Felts et al. used GBSA molecular dynamics to study the potential of mean force of small peptides.[11] To date, few Monte Carlo GBSA simulations have been reported in the literature. The flexible docking algorithm of Taylor et al. uses Monte Carlo moves and a GBSA model of water.[12] The Concerted Rotation with Angles (CRA) algorithm of Ulmschneider et al. uses novel Monte Carlo moves to fold peptides in a GBSA force field.[13,14] While the treatment of small peptides with a Monte Carlo GBSA method is still efficient compared to the explicit solvent alternative, or desirable in the case of Monte Carlo protein backbone moves, the method quickly loses its appeal as the system size increases. A Monte Carlo simulation of a biomolecule requires many more moves than molecular dynamics time steps because only portions of the system under study are updated at every move. Because most of the system does not change coordinates, and the force field terms are usually separable, it is generally sufficient to calculate only the change in energy of the part that has moved, which is very efficient. However, inspection of eq 3 shows that the Born radius of atom i depends on the position of every other atom j in the system. In turn, this means that the pairwise energies from eq 2 have the same dependency. As a result, the energy between atoms that did not move must be recomputed after every Monte Carlo move, and a full GB energy calculation must be performed after every Monte Carlo move. For even a mid-sized protein the computational cost can be very high. This is not a problem in a molecular dynamics simulation because the total energy of the system is calculated after every step in any case.

The aim of this article is to introduce methods that can overcome the limitations of a standard GBSA implementation within the framework of a Monte Carlo simulation. This work is motivated by the availability of powerful Monte Carlo methods such as concerted rotations[13] or configurational bias for sampling protein systems,[15] and the efficiency that can be attained by combining them with a GBSA model.

System Setup

In this work, the GBSA method was implemented in a modified version of the Monte Carlo package ProtoMS2.1.[16] Polarization energies were computed using eq 2, and the Born radii were calculated with eq 3. The Surface Area calculations were implemented using the method of Shrake and Rupley,[17] and a probe of 1.4 Å radius was used. The parameter set used for this GBSA model comes from a previous study ("PDAnum2" in ref 18).

To test the approximations introduced below we selected as a test case a set of protein−ligand relative binding free energy calculations which are shown in Figure 1. The perturbations are typical of the mutations performed in a protein ligand binding free energy calculation and cover apolar to apolar (**1to2**), polar to apolar (**1to3** and **4to5**), and polar to polar (**4to6**) perturbations. The two different proteins considered exhibit a very different binding site. Neuraminidase has a polar, solvent exposed, binding site, while cyclooxygenase-2 has a buried, fairly hydrophobic binding site. The test case should therefore represent a broad class of protein−ligand interactions that are studied by free energy perturbation methodologies. The binding mode of the inhibitors was inferred on the basis of a similar ligand complexed to a monomer of the N2 strain of influenza A (PDB code 1BJI)[19] or the PDB structure of murine cox2 complexed to SC-558 (PDB code 1CX2).[20] When necessary, hydrogens were added to the crystallographic structure using the program reduce.[21] Sugars, cofactors, crystallographic waters, and ions were removed. The protein was set up with the AMBER99 force field, inhibitors were set up with the GAFF force field, and the atomic partial charges were derived using the AM1/BCC method[22] as implemented in the package AMBER8.[23] The system was energy minimized using the Sander module of AMBER8 and a Generalized Born force field (the igb keyword was set to 1).[23] The backbone of the energy minimized protein was kept rigid for subsequent Monte Carlo simulations which were conducted with a modified version of the ProtoMS2.1 package.[16] To reduce the computational cost, only the protein residues that have one heavy atom within 15 Å of any heavy atom of the ligands were conserved. The bond angles and torsions of the protein side chains within 10 Å of any heavy atom of the ligand and all the bond angles and torsions of the ligand were sampled during the simulation, with the exception of rings. The bond lengths of the protein and ligand were kept rigid. The total charge of the system was brought to zero by neutralizing lysine residues lying in the outer (frozen) part of the scoop (residues number 511 and 532 for cox2, 432 and 273 for neuraminidase). The protonation state of the

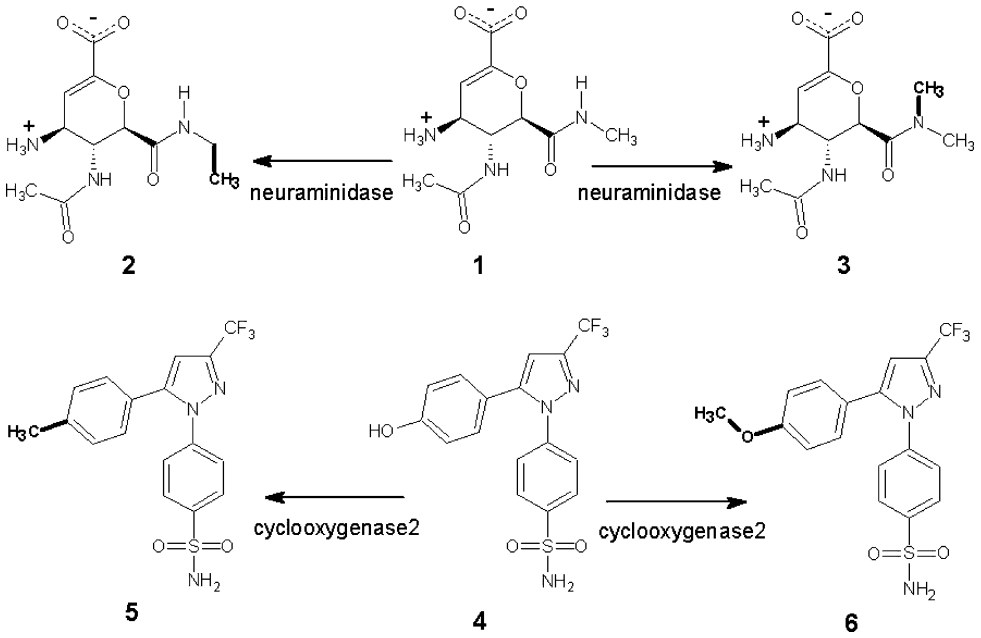


Figure 1. Representation of the ligands considered in this free energy study. For visual emphasis, the parts of the ligands being perturbed are highlighted by bold, straight lines.

histidine residues was decided by visual inspection of the crystallographic structures. The resulting model of cox2 had 155 residues and neuraminidase 145 residues. A 10 Å switched residue based cutoff was employed in all simulations. In the Generalized Born simulations, a cutoff of 20.0 Å for the calculation of the Born radii was applied.

Replica exchange thermodynamic integration[24,25] (RETI) was applied to these systems, and the necessary ensemble of states were formed using Metropolis Monte Carlo sampling[26] at a temperature of 25 °C. In the RETI protocol, standard finite difference thermodynamic integration (FD-TI)[27] is performed at each value of the coupling parameter λ ($\Delta\lambda$=0.001). Occasionally, moves that exchange system coordinates between replica i at $\lambda = A$ of energy $E_A(i)$ and replica j at $\lambda = B$ of energy $E_B(j)$ are attempted, subject to the following acceptance test.

$$\exp[\beta([E_B(j) - E_B(i)] - [E_A(j) - E_A(i)])] \geq \text{rand}(0, 1) \quad (4)$$

The occasional exchange of coordinates between the different simulations enhances configurational sampling and hence convergence of the calculated properties, while the acceptance test ensures that each replica converges the simulation to the correct distribution of states.[24,25]

Solute moves were attempted 10% of the time, with the remainder being protein side chain moves. In the unbound state, 2000 (2K) moves of equilibration were performed before 200K moves of data collection. In the bound state, the system was preequilibrated at one value of λ for 600K moves. The resulting configuration was distributed over 12 values of the coupling parameter λ (0.00, 0.10, ..., 0.90, 0.95, 1.00), and further equilibration was performed for 100K moves. Data were collected over the remaining 900K moves. Replica exchange moves were attempted every 5K configurations.

The error on the free energy gradients was calculated by taking the standard error of batch averages (size 1K). The standard error of these averages was then integrated over the λ coordinate to yield the maximum error.

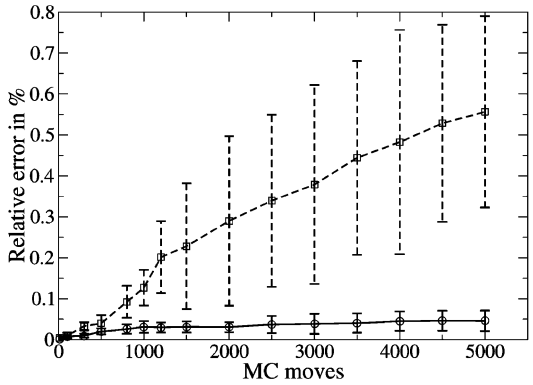
The speedup reported in the next sections are calculated as ratios of the time taken to complete 1000 MC moves on the test systems between two particular simulation protocols.

Approximated Generalized Born Potential

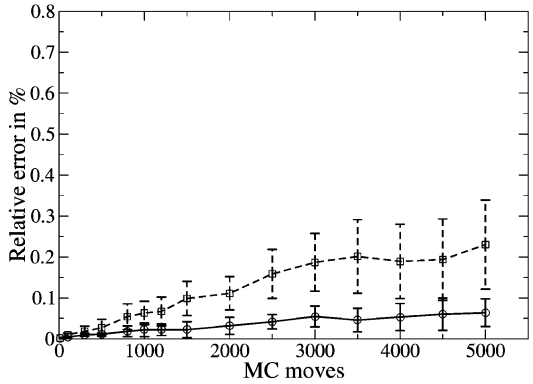
A rigorous GB calculation means that the GB energy between each pair of atoms must be recalculated after every move. However, the impact of a moving atom on the Born radius of a distant atom is small. We have therefore structured the implementation of the GB calculation such that the energy of a pair of atoms is recalculated only if the Born radius of either atom has changed by more than a specified threshold value after a MC move. A large number of pair interactions can be skipped in this fashion, resulting in a significant speedup. In this implementation, only the necessary old and new GB energy pair terms are recalculated to update the total GB energy. This keeps additional memory requirements low and makes the method easily applicable to larger system.

This approximation may have unwanted effects. For example, the total energy would not be completely conserved in a hypothetical Monte Carlo simulation in the NVE ensemble. However, the fact that useful results can be obtained from Molecular Dynamics simulations where fluctuations in the total energy are introduced because of errors in the integrator suggests that as long as the impact of the approximation is small, the resulting ensemble will closely mirror the correct one.

To assess the impact of this approximation, we have run a series of short free energy simulations with a GBSA force field at different values of the threshold parameter. The plots reported in Figure 2 are constructed by running a simulation for N steps with a specified value of the threshold parameter. The total energy of the last generated configuration is then recorded and compared to the value that is obtained by

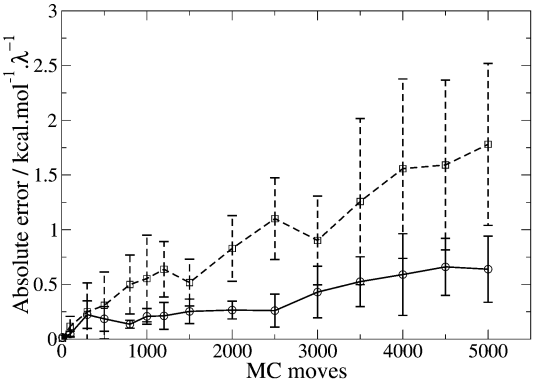


(a) Perturbation **1to3**, neuraminidase

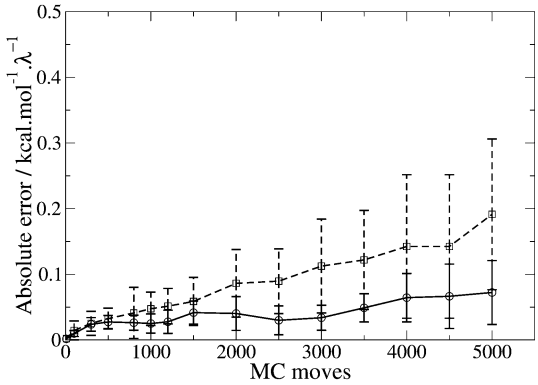


(b) Perturbation **4to5**, cox2

Figure 2. Relative error in percent of the total energy as a function of the number of Monte Carlo moves for a threshold of 0.005 Å (black line, circles) and 0.05 Å (dashed line, squares). Each point is the average of 10 different simulations, and the error bar represents the associated standard error. Similar plots for the perturbations **1to2** and **4to6** are observed (data not shown).



(a) Perturbation **1to3**, neuraminidase



(b) Perturbation **4to5**, cox2

Figure 3. Absolute error in the free energy gradients as a function of the number of Monte Carlo moves for a threshold of 0.001 Å (black line, circles) and 0.005 Å (dashed line, squares). Each point is the average of 10 different simulations, and the error bar represents the associated standard error. Similar plots for the perturbations **1to2** and **4to6** are observed (data not shown).

calculating the total energy with no approximation. This procedure is repeated 10 times for a number of values of *N*. An arguably acceptable error on the total energy would be about 0.1% since this is in the accepted range of MD integrator errors.[28] The systems run in neuraminidase are more sensitive to the approximation, and at a high threshold value, the deviations become quickly large. The systems run on cox2 appear much less sensitive. In both systems, at a threshold of 0.005 Å and up to 5000 MC moves the error is below 0.1%.

An added requirement for a free energy calculation is that the free energy gradients are not too sensitive to this approximation. In this application, we use a finite difference scheme, and the gradients are formed from the difference in total energy at a value of $\lambda - d\lambda$ and $\lambda + d\lambda$. In Figure 3 the protocol described previously is applied to report the free energy gradients accumulated at $\lambda = 0.50$. The gradients are formed from the difference of two large numbers and are therefore more sensitive to small errors in the total energies. The free energy gradients of the perturbations run on neuraminidase are seen to be much more sensitive to the threshold than those run on cox2. Because the binding site of neuraminidase is much more solvent exposed and comprises several polar amino acids, a rigorous treatment of the GB energy appears more important than for cox2, where the buried, hydrophobic binding site is less sensitive to solvent effects. After 1000 MC moves, at a threshold of 0.001, the average error on the free energy gradients is still 0.20 ± 0.07 kcal·mol^{-1}·λ^{-1} for **1to3**, while for **4to5** it is essentially negligible at high and low threshold values. From a computational perspective, the cost of a full GBSA calculation after 1000 approximate GBSA calculations is small. By updating completely the GB energy every 1K MC moves and with a suitably small threshold parameter, the errors on the total energy and the free energy gradients can be kept sufficiently low such that they have a small or negligible influence on the computed free energy.

To verify more rigorously the sensitivity of the systems to the threshold, a series of GBSA free energy simulations is run for each system with a varying threshold parameter. The impact of the threshold parameter on the calculated free energy is shown in Figure 4. For the perturbations in cox2, the calculated free energies are within the statistical error of the exact simulation over the range of thresholds studied. For the perturbations in neuraminidase, the free energy is more sensitive to the value of the threshold parameter, and a high value of the threshold yields results that deviate significantly from the rigorous calculations; this is more

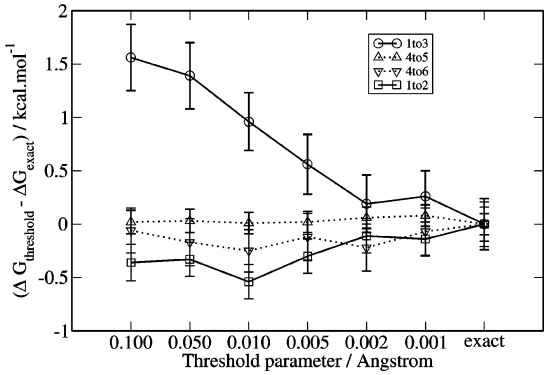


Figure 4. Influence of the threshold parameter on the calculated free energy for the selected perturbations in the bound state. The error bar represents the associated statistical error.

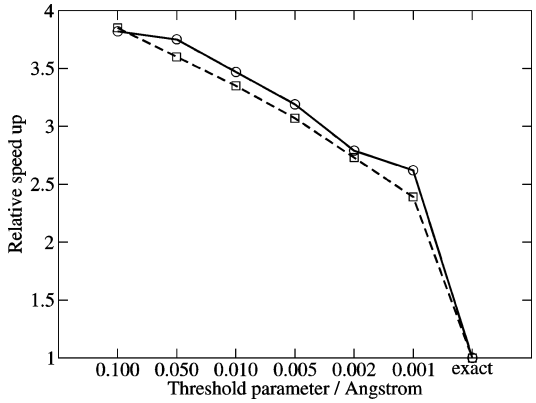


Figure 5. Relative speedup that can be achieved in the simulation of the perturbations **1to3** (circles, black line) and **4to6** (squares, dashed line) in the bound state with a varying threshold parameter. The speedups are based on the time taken to perform 1K Monte Carlo moves.

marked for **1to3** than **1to2**, with results agreeing to within a statistical sampling error from a threshold value of 0.002 Å or less. These results are consistent with the increased sensitivity of the free energy gradients to the threshold for the neuraminidase systems observed in Figure 3. Figure 5 shows the speedup relative to a full GBSA calculation. Because the computational expense is similar for the systems run on the same protein, speedups are shown for **1to3** and **4to5** only. Even with a threshold as low as 0.001 Å, a considerable speedup is achieved because the Born radii of several protein atoms are insensitive to the displacement of a distant residue. On these systems and over the range of thresholds studied, the simulations run 2.4−3.8 times faster.

On the range of systems studies here, the influence of the threshold parameter on the calculated binding free energies has been shown to be negligible (cox2) or minor (neuraminidase) and can be minimized by reducing sufficiently the threshold parameter, at the cost of additional simulation time. That a balance can be struck between speedup and accuracy can prove useful. In applications where accuracy is important, almost rigorous calculations can be made with a sufficiently low threshold. On the other hand, less accurate calculations that could be useful in the context of fast free energy calculations could be run with a higher threshold. The optimum value of the threshold varies according to the system but can be estimated rapidly by plotting the drifts in the free energy gradients for a series of short simulations. For the systems studied here, it appears that a good compromise would be achieved with a threshold of 0.005 Å.

Simplified Sampling Potential

Theory. A novel methodology to perform Monte Carlo simulations has recently been proposed by Gelb.[29] He shows that it is possible to perform a Monte Carlo simulation in which the potential energy is evaluated using an approximate, less expensive potential E_ζ than a more realistic potential E_π and still form an ensemble of states that are distributed according to the rigorous potential. The method is very powerful as in principle any kind of simplified potential/expensive potential combination can be devised.

The method can be briefly summarized as follows:

1. Start a simulation in state *i*.

2. Performs *N* steps of standard Metropolis sampling with a simple potential E_ζ of limiting distribution ζ until a state *j* is reached.

3. Set state i = state j with probability $\chi = (\pi_j\zeta_i)/(\pi_i\zeta_j)$. In this equation π_i and ζ_i are the probability of state *i* in the two different distributions π and ζ.

4. Accumulate any property of interest that is a function of the coordinates of state *i*.

5. Return to 1 or terminate after a number of iterations

In essence, a standard Monte Carlo simulation is conducted for *N* steps with a potential chosen for its convenience (usually computational efficiency). However, because the probabilities of state *i* and *j* in the two distributions π and ζ generally differ, it is necessary in step three to correct for any bias introduced by the potential E_ζ. This acceptance test makes sure that the ensemble formed during the simulation converges toward the distribution π instead of ζ. In the NVT ensemble, step three amounts to accepting state *j* according to

$$\exp[\beta([E_\pi(j) - E_\pi(i)] - [E_\zeta(j) - E_\zeta(i)])] \geq \text{rand}(0, 1) \quad (5)$$

The acceptance test is therefore based on the difference of differences of energies of state *i* and *j* between the two potentials E_π and E_ζ. With this method, no statistics for the target ensemble π can be collected during step two, and the number of data points accumulated is reduced compared to a traditional Monte Carlo simulation. This does not necessarily affect convergence because subsequent configurations in the Markov chain are typically highly correlated and do not contribute new information to the running average. That is to say, it is equally good to sample the distribution of interest less often if the samples are less correlated.

To date, applications of this methodology have been reported by Hetenyi et al.[30,31] (who seem to have developed a similar method independently). Hetenyi reported a 3.0−4.7 speedup in the simulation of a Lennard-Jones fluid using for E_ζ a potential similar to E_π but with a shorter cutoff. Gelb reported similar results on a similar system.[29] In a second application by Hetenyi, a MC Ewald sum simulation

of water was running 4.5−7.5 times faster using this methodology. This method has been employed by Iftimie et al. to perform ab initio simulations using a classical potential.[32]

If the change of energy in going from state *i* to *j* is similar with the two Hamiltonians, the probability of accepting the configuration *j* will be close to unity. On the other hand, if the two potentials differ too much, then the acceptance rate will drop, and the method will lose efficiency since all the steps performed with E_ξ have been wasted. Therefore a good approximate potential E_ξ must be faster than, and yet very similar to, E_π. This is of course difficult to achieve.

Application to a GBSA Model. The complete application of the GBSA theory requires the calculation of a Surface Area (SA) dependent term to yield a solvation free energy. The inclusion of this term can be expensive and becomes significant once the GB calculations have been accelerated with the use of a threshold. For example, the simulation of **4to5** with a GB threshold of 0.005 is about 1.8 times slower once the SA calculations are enabled.

The fluctuations in the SA term are known to be small compared to the other energy components of the force field. This observation has led other workers to devise schemes where the SA term is only periodically updated.[33,12,14] While reasonable, this approximation is not completely rigorous. Other workers have developed faster, approximate SASA calculations schemes, but these algorithms do not calculate reliably the small changes in SASA associated with the small conformational changes observed between MC moves.[34]

However, the effect of the SA term can be rigorously included in the GBSA simulation by adopting a particular simplified sampling potential methodology. The simple potential E_ξ corresponds to a GB simulation run without a SA term, while the correct potential E_π includes SA calculations.

In addition, we consider other means to further speed up the calculations by adopting a less rigorous solvation model for the simplified potential. Here two different simplified solvation models are investigated: a distance dependent dielectric (DDD) force field and a simplified GB force field (fastGB). In the DDD force field the GB equations are replaced by a $\epsilon(r) = 4r$ distance dependent dielectric. In the fastGB force field smaller cutoffs are applied: a residue based cutoff, Born radii cutoff, and thresholds of 6.0, 12, and 0.05 Å, respectively. Furthermore, since no statistics are collected with fast GB, it is not necessary to compute free energy gradients, which avoids the expensive GB energy calculations for the perturbed states. The rigorous potential is taken as a GB simulation with a threshold of 0.005 Å and a rigorous SA calculation.

Table 1 lists the average acceptance rate of the correction step for the two different potentials as a function of the number of moves performed. The speedup compared to the rigorous GBSA simulation is also reported. The parameter *M* is the number of moves performed with the quick potential before attempting to add the generated configuration to the ensemble. As this quantity increases, the acceptance rate diminishes. As has been pointed out, a tradeoff must be made between computational efficiency and sampling efficiency.[29]

Table 1. Acceptance Rate at the Correction Step and Relative Speedup for Different Combinations of Potentials and Number of Moves *M* with the Approximate Potential[a]

M	DDD rate[b] (%)	DDD speedup[c]	fastGB rate[b] (%)	fastGB speedup[c]
		1to3		
5	56.7	2.2	90.2	2.1
10	36.7	2.4	83.4	2.3
20	18.2	2.9	76.3	2.7
25	13.6	3.1	71.5	2.8
		4to5		
5	65.4	2.2	91.2	2.0
10	47.7	2.5	87.1	2.3
20	27.9	2.9	80.8	2.5
25	22.5	3.1	78.2	2.7

[a] The results for **1to2** and **4to6** are similar to **1to3** and **4to5**. [b] Average across all values of λ. [c] Relative to a GBSA simulation with a threshold of 0.005 Å.

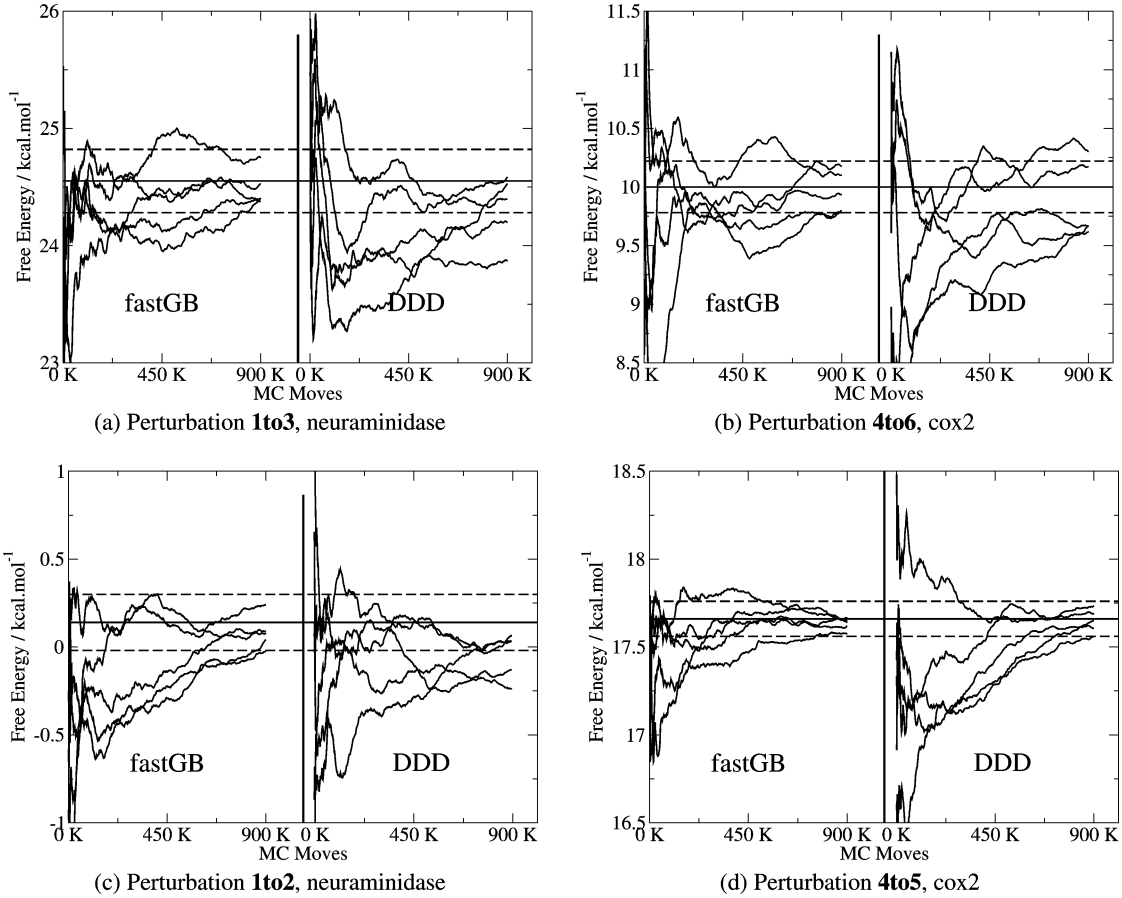
With the DDD model, the acceptance rate decreases faster than the speedup increases, and a short value of *M* is favored. Even after only 5 steps, the acceptance rate is only 55−65%. For the systems in neuraminidase, the acceptance rate of the correction step is actually similar to using vacuum conditions (data not shown). This illustrates that the configurations favored by a GBSA force field are rather different from those preferred by a DDD force field.

With the fastGB model the decreases in the acceptance rate are more or less counterbalanced by the increase in speedup, and no value of *M* is clearly favored. In addition, the acceptance rates are much higher and around 90% for *M* equal to 5.

If we make the assumption that simulations run with the DDD and fastGB force field explore the configurational space at the same rate, then these results suggest that the combination of the two potentials fastGB/GBSA yields more efficient sampling than the DDD/GBSA combination.

To demonstrate this more decisively, in Figure 6 we investigate the convergence of the calculated free energies in the bound state for 5 independent simulations performed with the different protocols and a value of *M* set to 10. After 900K moves of data collection, almost all the fastGB simulations have converged to within the error bounds of the results obtained with GBSA 0.005 Å. With the DDD protocol, the results are more spread out, and several simulations are outside the error bounds. It is apparent that the fastGB protocol converges better the free energies than the DDD protocol for the same number of iterations.

Taken together, the results in Table 1 and Figure 6 suggest that a low acceptance rate for the correction step hinders convergence. The DDD simulations are slightly faster than the fastGB simulations. However, since the simulation results are much better converged with the fastGB protocol, it should be preferred over a DDD model. By combining the fastGB potential with the value of *M* set to 10 and with a GBSA 0.005 Å potential described in the previous section, an approximately 2.3-fold speedup over a standard MC simulation run with GBSA 0.005 Å can be achieved. The present results demonstrate that the simplified sampling potential methodology, applied here to increase the efficiency of



(a) Perturbation **1to3**, neuraminidase (b) Perturbation **4to6**, cox2

(c) Perturbation **1to2**, neuraminidase (d) Perturbation **4to5**, cox2

Figure 6. The convergence of the calculated free energies in the bound state using different sampling potentials. The estimated free energy is plotted as a function of the number of Monte Carlo moves performed. In each figure, on the left-hand side, 5 independent simulations run with fastGB are shown. On the right-hand side, 5 independent simulations run with DDD are shown. The horizontal line is the estimate of the free energy obtained after 900K moves with the potential GBSA 0.005 Å. The dashed lines represents the statistical error associated with this number.

Generalized Born calculations for the first time, allows significant computational savings without additional approximations.

Conclusion

A novel methodology by which free energy calculations in a Generalized Born framework can be made more efficient within Monte Carlo simulations has been proposed. It can be summarized as follows:

1. An approximate Generalized Born potential in which the energy of the system is only partially updated after a MC move—the impact of this approximation on the calculated free energies can be made arbitrarily small at the expense of computational time by adjusting a single parameter.

2. Sampling driven by an inexpensive potential with a special Monte Carlo acceptance test that removes any bias in the distribution introduced by the cheap potential—this allows in addition the rigorous incorporation of surface area calculations at a minimum computational cost.

In Table 2 timings for various combinations of these approximations on two of the four systems are reported. Protocol 4, which combines the two approximations is seven to eight times faster compared to protocol 1 which would correspond to a standard implementation of a GBSA force field. Protocol 4 is also only about 4.1−4.3 times slower than a simulation run in a vacuum (protocol 5) which compares favorably with the typical efficiency of molecular dynamics GBSA simulations.[23] While the increased efficiency has been demonstrated on a free energy calculation, improvements in standard MC simulations could be sought with the same method.

Table 2. Time Required To Complete a Block of 1K Moves for Selected Approximations[a]

protocol	solvation model	simplified potential	time **1to3** (s)	time **4to5** (s)
1	GBSA exact	no	633.4	746.3
2	GBSA threshold 0.001 Å	no	241.8	312.0
3	GBSA threshold 0.005 Å	no	198.6	243.0
4	GBSA threshold 0.005 Å	fastGB $M = 10$	79.8	104.2
5	vacuum	no	19.5	24.1

[a] ProtoMS2.1 on a Pentium IV 2 GHz compiled with g77.

We may ask whether the partial rigidity of the system and the simplified treatment of solvation affects the accuracy of the calculated binding free energy. The simulation results can be compared with experimental figures by constructing a thermodynamic cycle, which requires the ligand perturbations in the unbound state to be performed. We stress that free energy calculations in the unbound state are extremely rapid (on the order of a few minutes), and there is no need to introduce the methods developed to speed up the simula-

Table 3. Calculated and Experimental Binding Free Energies of the Tested Systems with the ApproxGB+SA Protocol[a]

perturbation	$\Delta\Delta G_{exp}$	$\Delta\Delta G_{bind}$	ΔG_{prot}	ΔG_{wat}
1to2	−1.6	-0.3 ± 0.3	0.1 ± 0.2	0.4 ± 0.1
1to3	−2.7	-2.7 ± 0.3	24.5 ± 0.3	27.2 ± 0.2
4to5	<−4.6	-2.4 ± 0.1	17.7 ± 0.1	20.1 ± 0.1
4to6	<−5.6	-3.1 ± 0.2	10.0 ± 0.2	13.1 ± 0.1

[a] The threshold was set to 0.005 Å. The figures are in kcal·mol^{-1}. The experimental figures were taken from ref 35 for the neuraminidase inhibitors and from ref 36 for the cox2 inhibitors.

tions in the bound state. The calculated binding free energies are listed in Table 3. The implicit solvent protocol reproduces well the relative binding free energy of **1to3** but underestimates somewhat the binding free energy of the three other systems, although the trends are respected. The main emphasis of this work was to introduce a novel methodology to perform Generalized Born Monte Carlo free energy calculations efficiently. A thorough investigation of the influence of the solvation model on the relative binding free energies will require the comparison of explicit and implicit solvent simulations on a larger set of systems. Such studies will be made computationally tractable by the present technique.

Acknowledgment. We thank Astex Therapeutics and the University of Southampton for funding this work and EPSRC (GR/R06137/01) for providing computational resources. J. Michel is grateful to Dr. C. Woods for helpful discussions.

References

(1) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *J. Am. Chem. Soc.* **1990**, *112*, 6127−6129.

(2) Hawkins, C. J.; Cramer, D. *Chem. Phys. Lett.* **1995**, *246*, 122−129.

(3) Gallicchio, E.; Levy, R. M. *J. Comput. Chem.* **2004**, *25*, 479−499.

(4) Qiu, D.; Shenkin, P. S.; Hollinger, F. P.; Still, W. C. *J. Phys. Chem. A* **1997**, *101*, 3005−3014.

(5) Jayaram, B.; Liu, Y.; Beveridge, D. L. *J. Chem. Phys*. **1998**, *109*, 1465−1471.

(6) Ghosh, A.; Rapp, C. S.; Friesner, R. A. *J. Phys. Chem. B* **1998**, *102*, 10983−10990.

(7) Onufriev, A.; Case, D. A.; Bashford, D. *J. Comput. Chem.* **2002**, *23*, 1297−1304.

(8) Feig, M.; Onufriev, A.; Lee, M. S.; Im, W.; Case, D. A.; Brooks, L. C. *J. Comput. Chem.* **2004**, *25*, 265−284.

(9) Sorin, E. J.; Engelhardt, M. A.; Herschlag, D.; Pande, V. S. *J. Mol. Biol.* **2002**, *317*, 493−506.

(10) Sorin, E. J.; Rhee, Y. M.; Nakatani, B. J.; Pande, V. S. *Biophys. J.* **2003**, *85*, 790−803.

(11) Felts, A. K.; Harano, Y.; Gallicchio, E.; Levy, R. M. *Proteins: Struct., Funct., Bioinformatics* **2004**, *56*, 310−321.

(12) Taylor, R. D.; Jewsbury, P. J.; Essex, J. W. *J. Comput. Chem.* **2003**, *24*, 1637−1656.

(13) Ulmschneider, J. P.; Jorgensen, W. L. *J. Chem. Phys.* **2003**, *118*, 4261−4271.

(14) Ulmschneider, J. P.; Jorgensen, W. L. *J. Am. Chem. Soc.* **2004**, *126*, 1849−1857.

(15) Siepmann, J. I.; Frenkel, D. *Mol. Phys.* **1992**, *75*, 59−70.

(16) Woods, C. J.; Michel, J. ProtoMS2.1; in house Monte Carlo Code, 2005.

(17) Shrake, A.; Rupley, J. A. *J. Mol. Biol.* **1973**, *79*, 351−371.

(18) Michel, J.; Taylor, R. D.; Essex, J. W. *J. Comput. Chem.* **2004**, *25*, 1760−1770.

(19) Taylor, N. R.; Cleasby, A.; Singh, O.; Skarzynski, T.; Wonacott, A. J.; Smith, P. W.; Sollis, S. L.; Howes, P. D.; Cherry, P. C.; Bethell, R.; Colman, P.; Varghese, J. *J. Med. Chem.* **1998**, *41*, 798−807.

(20) Kurumbail, R. G.; Stevens, A. M.; Gierse, J. K.; Mcdonald, J. J.; Stegeman, R. A.; Pak, J. Y.; Gildehaus, D.; Miyashiro, J. M.; Penning, T. D.; Seibert, K.; Isakson, P. C.; Stallings, W. C. *Nature* **1996**, *384*, 644−648.

(21) Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. *J. Mol. Biol.* **1999**, *285*, 1735−1747.

(22) Jakalian, A.; Jack, D. B.; Bayly, C. I. *J. Comput. Chem.* **2002**, *23*, 1623−1641.

(23) Case, D. A.; Darden, T. A.; Cheatham, T. E., III; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Merz, K. M.; Wang, B.; Pearlman, D. A.; Crowley, M.; Brozell, S.; Tsui, V.; Gohlke, H.; Mongan, J.; Hornak, V.; Cui, G.; Beroza, P.; Schafmeister, C.; Caldwell, J. W.; Ross, W. S.; Kollman, P. A. AMBER 8; University of California, San Francisco, 2004.

(24) Woods, C. J.; King, M. A.; Essex, J. W. *J. Phys. Chem. B* **2003**, *107*, 13711−13718.

(25) Woods, C. J.; King, M. A.; Essex, J. W. *J. Phys. Chem. B* **2003**, *107*, 13703−13710.

(26) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087−1092.

(27) Leach, A. R. *Molecular Modelling, Principles and Applications;* Longman: Harlow, U.K., 1996.

(28) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids;* Oxford University Press: 1989.

(29) Gelb, L. D. *J. Chem. Phys.* **2003**, *118*, 7747−7750.

(30) Hetenyi, B.; Bernacki, K.; Berne, B. J. *J. Chem. Phys.* **2002**, *117*, 8203−8207.

(31) Bernacki, K.; Hetenyi, B.; Berne, B. J. *J. Chem. Phys.* **2004**, *121*, 44−50.

(32) Iftimie, R.; Salahub, D.; Wei, D. Q.; Schofield, J. *J. Chem. Phys.* **2000**, *113*, 4852−4862.

(33) Zhu, J. A.; Shi, Y. Y.; Liu, H. Y. *J. Phys. Chem. B* **2002**, *106*, 4844−4853.

(34) Weiser, J.; Shenkin, P. S.; Still, W. C. *J. Comput. Chem.* **1999**, *20*, 217−230.

(35) Wall, I. D.; Leach, A. R.; Salt, D. W.; Ford, M. G.; Essex, J. W. *J. Med. Chem.* **1999**, *42*, 5142−5152.

(36) Price, M. L. P.; Jorgensen, W. L. *J. Am. Chem. Soc.* **2000**, *122*, 9455−9466.

CT600069R