On-the-fly learning and sampling of ligand binding by high-throughput molecular simulations

S. Doerr and G. De Fabritiis*

Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain

E-mail: gianni.defabritiis@upf.edu

Abstract

High-throughput molecular dynamics (MD) simulations are a computational method consisting of using multiple short trajectories, instead of few long ones, to cover slow biological timescales. Compared to long trajectories this method offers the possibility to start the simulations in successive batches, building a knowledgeable model of the available data to inform subsequent new simulations iteratively. Here, we demonstrate an automatic, iterative, on-the-fly method for learning and sampling molecular simulations in the context of ligand binding for the case of trypsin-benzamidine binding. The method uses Markov state models to learn a simplified model of the simulations and decide where best to sample from, achieving a converged binding affinity in approximately one microsecond, one order of magnitude faster than classical sampling. This method demonstrates for the first time the potential of adaptive sampling schemes in the case of ligand binding.

*To whom correspondence should be addressed

1 Introduction

Molecular simulations have recently been able to resolve biological processes in the timescale of microseconds to milliseconds. Some of the most important achievements came from the new capability to perform longer trajectories using specialized hardware,[1] e.g. in protein folding,[2] ion permeation,[3] receptor activation,[4] etc. Parallel to this, a different approach was developed which consisted of running a high number of shorter simulations, achieving similar throughput of special hardware at lower costs. There are examples of this approach from several years ago[5,6] but recently they became more interesting due to the fact that individual simulations are becoming longer and analysis tools based on Markov models have been shown to be capable of reconstructing a full model of the process from these trajectories.[7–10]

Markov state models provide a means to interpret and derive observables from large amounts of simulations. This is done by combining all simulations into a single kinetic model of the simulated system. From this model, long-time kinetic information, pathways, free energies and kinetic rates can be derived.[7,11,12] Applications have spanned from ligand binding,[13,14] to protein folding,[8,15,16] and more. However, the timescales of most biological processes can be orders of magnitude longer than the currently accessible timescales.

To reach these long timescales, in some of the previously mentioned cases, results were obtained by manually respawning simulations from interesting, poorly sampled, configurations.[15] A useful property of unbiased simulations is that the dynamics are not altered and thus the amount of simulation data can be increased at will by restarting from any previously seen configuration. These examples can be seen as a preliminary version of iterative sampling schemes where human supervision is still necessary to learn the model and decide from where to respawn. Ultimately, this approach is limited by the human intervention and allows for just a few resampling iterations. The aim of this paper is to show a completely automated protocol designed to do that but in a fast iterative manner. Similar approaches have been shown already and are usually called adaptive sampling methods.

Several strategies have been developed for adaptive sampling over a Markov state model rep-

resentation of the simulation. These include sampling from the states that contribute the most to the statistical error of the mean first passage times,[17] the estimation error of the eigenvalues and eigenvectors,[18] uniform sampling from all states, sampling from states with low populations[19] and adjacency based sampling.[20] In most cases, the adaptive sampling scheme was simulated a posteriori by using a pre-simulated classically sampled dataset. The a posteriori adaptive sampling was done using Markov chain Monte Carlo (MCMC) sampling of trajectories already existing in the dataset.[18,20,21] The test systems comprise a terminally blocked alanine peptide,[18] the Villin headpiece,[18–21] the Fs peptide,[20] the WW domain[20] and small RNA systems.[22,23] These adaptive tests, when compared to a single long simulation, report a speed up in compute-time by a factor of two. Another approach was proposed in ref.,[24] in which they combine rounds of sampling from the conformations kinetically furthest from the initial conformation, with rounds of refinement sampling based on a cut-based free energy profile. This method was applied on the alanine dipeptide and a three-stranded antiparallel beta-sheet peptide of 20 residues showing a speedup of one to two orders of magnitude in simulation time compared to classical parallel simulations.

In this work we apply a new adaptive sampling method on ligand binding processes in the case of trypsin-benzamidine. The scheme is tested on real compute infrastructures like a cloud-like interface using the molecular dynamics code ACEMD[25] and on the distributed community project GPUGRID.[26] In both cases, the learning and sampling is carried out unattended. To our knowledge, this is the first application of automatic adaptive sampling for ligand binding processes.

2 Methods

The adaptive method for MD simulations we have developed follows the following scheme. We perform multiple iterations (epochs) of relatively short parallel simulations. We first select a set of distinct starting conformations and send them to GPUGRID for simulation. Then we retrieve the simulations and analyze them using a Markov state model. From this analysis we extract a new set of starting conformations which are used for the next epoch of parallel simulations etc. The

method with which the conformations for each epoch are selected is the key to speeding up the sampling. The first epoch is naively initiated from a set of random conformations derived from short simulations. After the first epoch the next epochs use the Markov model to select the starting conformations based on criteria explained in the following sections.

2.1 Setup

We define an adaptive run A_i as an instance of our adaptive sampling method. Each such run consists of a number of epochs e and each epoch consists of N parallel simulations of t ns each. For an adaptive run A_i of trypsin-benzamidine, we run $e = 10$ epochs, each consisting of $N = 10$ parallel simulations of $t = 10ns$. Therefore an adaptive run A_i at epoch 10 consists of a total of 100 trajectories of 10 ns equaling 1 μs of simulated time. Only one adaptive run is needed in general, but here to verify the convergence of the results, we perform 10 independent adaptive runs called A_1-A_{10}. All 10 runs use for their first epoch the same 10 initial conformations obtained from 10 ns simulations but after that diverge as they are run independently. The setup and equilibration of the molecular system was the same as in ref.[13] .

The length of $t = 10ns$ for the individual trajectories was decided based on the fact that it is possible to build converged Markov models with a lag time of 5 ns for this system (see implied time scale plot for reference data in Fig. S1). In more complex systems this could be longer. A rule of thumb to use twice the expected lag time seems reasonable. As shown in,[7] the better the discretization of slow processes is, the shorter the required lag time can be and thus the shorter the simulations that can be used. For very poor discretizations, it might not make sense to run adaptive sampling schemes because the trajectories will have to be too long and we will thus lose in efficiency. Methods to improve discretization are available[27,28] however they were not required for this study.

2.2 Learning the model

In an adaptive run A_i, after each epoch of simulations, the resulting data of the N parallel simulations is post-processed to calculate a metric for each conformation. For trypsin-benzamidine the chosen metric is the contacts between atoms of benzamidine (Fig. S2) and all of the CA atoms of the protein using a threshold of 8 Å. Then, all conformations seen during the simulations of all past epochs are clustered using a k-center method into K clusters. For this specific system we used $K = 50$. Clusters containing fewer than 5 conformations are joined into the closest neighboring cluster.

Using this conformational clustering, a Markov state model is built.[7] The count and transition probability matrices are estimated using a lag time $t_{lag} = 100$ ps which is the simulation sampling rate. Note that at this lag time, the model is not Markovian, however this increases statistics and helps to produce a connected transition matrix between newly discovered states. This is important as when the clusters are disconnected we are unable to calculate any information about them. The rough Markov model used to decide from which part of the configuration space to respawn is not used to produce quantitative results, but just a ranking of relevant states to select from which ones to spawn.

2.3 Sampling method

Given the Markov model of the current available dynamics at epoch e, the starting conformations for the next epoch $e+1$ are selected proportionally to a given ranking. The rationale of the adaptive sampling scheme in Eq. 1 is that we would like to sample proportionally to the free energy of all states in order to select metastable states. We assume that the free energy surface is rough and thus expect to overcome barriers faster by starting from local minima along the pathway to the bound state. We define

$$p(m) \propto k_B T log(\frac{k_{on}^m}{k_{off}^m}), \tag{1}$$

where k_B is the Boltzmann constant, T the temperature and k_{off}^m, k_{on}^m the unbinding and binding rates in $A+B \rightleftharpoons AB$, where m designates all conformational clusters except the unbound state. The estimation of k_{on}^m proves to be really sensitive to the size of cluster m, especially in the case of data scarcity, i.e. a very poorly defined cluster has a higher size and probability to be visited even if there is no significant interaction there. This does not happen for $k_{off}^m = 1/\tau_{off}^m$, where τ_{off}^m is the mean first passage time from a state m to bulk. The proposed scheme is therefore the following. From the K available conformational clusters a new configuration is chosen proportionally to its mean residence time,

$$R(m) \propto log(\tau_{off}^m), \tag{2}$$

where m is the microstate and τ_{off}^m is computed in the current model at $t_{lag} = 100ps$. To calculate the τ_{off} of all clusters, the cluster with the least contacts is designated as the bulk cluster. Then by calculating the mean first passage times,[8] from the other $K-1$ clusters to the bulk cluster, we obtain the residence time for each cluster. By normalizing Eq. 2, we obtain a probability distribution $p(m)$. We then define a multinomial distribution on the random variables X_m, with associated probabilities $p(m)$ with N trials.

At every epoch a certain number of parallel simulations N are restarted. As our number of samples N is very low, in this study 10, sampling from the multinomial distribution causes large fluctuations, producing instances where the most resident state is not actually sampled. Furthermore, at the beginning of the sampling the stationary distribution is only approximate and even states with high occupancy (bound) might be only marginally preferred compared to others where most of the mass of the distribution is. Instead of sampling directly from X_m we respawn new states m by the mean of the multinomial distribution $E(X_m) = p(m)N$,

$$S(m) = floor(Np(m)) \tag{3}$$

where $S(m)$ indicates how many states need to be respawned from state m and $floor$ truncates the value to the lower integer. The truncation is done to prevent respawning from the tail of

the distribution which corresponds to very transient states. After we have obtained the number of simulations to start from each state m, initial configurations are then selected randomly from inside each conformational cluster m. These configurations are then sent to GPUGRID for simulation and are the input of the next epoch of the adaptive run A_i.

2.4 Accurate model

After each adaptive run, we want to obtain a more accurate Markov state model which can provide more correct kinetics and free energy estimates. This can be done by using a higher lag time and varying the number of clusters. For an example of the effect of the lag time on the kinetics estimates see Figure S3. Therefore, to create an accurate model we used as lag time

$$t_{lag} = min(c, L/2) \tag{4}$$

where c is the maximum lag time before disconnecting the state involved in the slowest process and L is the trajectory length. This lag time gives us a model where the slowest process is still connected in the transition matrix with the lag time small enough to not reduce our statistics significantly. For the accurate models we also varied the number of clusters depending on the amount of aggregate simulation time. This means that for epoch 1 we used 50 clusters while for epoch 10 we used 150 with number of clusters of in-between epochs scaling linearly. This is better because a larger number of clusters can discretize the state space more finely, improving the model quality. However in the low epochs we are lacking data and thus a high number of clusters would significantly reduce statistics between states. Therefore the number of clusters has to scale by the aggregate simulation time.

Lastly, in the accurate model the standard binding free energy for the predicted bound state was calculated as in the supplementary of ref.[13] based on the equilibrium distribution using the equation

$$\Delta G^0 = \Delta G + \Delta G_V = -k_B T log(\frac{p_{eq}^{bound}}{p_{eq}^{bulk}} \frac{V_u}{V_0}) \tag{5}$$

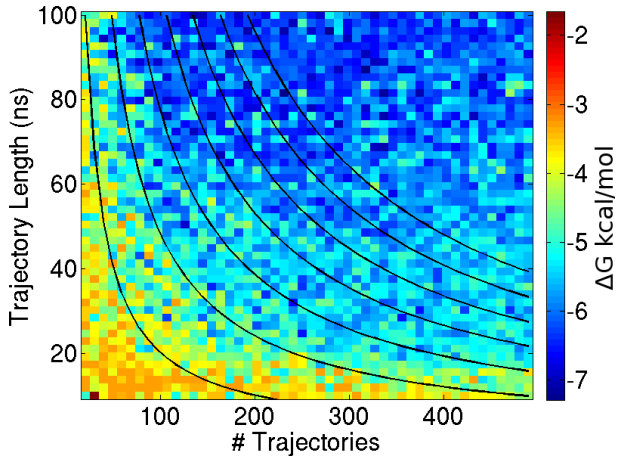

Figure 1: Bootstrapping calculations were done to demonstrate the effect of simulation length and amount of simulations on the estimate of the standard free energy of binding. The black curves are isocontours of equal computing cost corresponding to 2, 4.8, 7.7, 10.6, 13.4, 16.3, 19.2 μs from left to right. Free energy values are averaged over 4 different clusterings with a standard deviation of 0.84 kcal/mol.

where $\Delta G_V = -k_B T log(V_u/V_0)$ is the standard volume correction, p_{eq}^{bound} the equilibrium probability of the bound state, p_{eq}^{bulk} the equilibrium probability of the bulk, V_u is the unbound volume and $V_0 = 1661$ Å^3 is the standard volume at 1 molar. The unbound volume can be measured in terms of the number of water molecules (55.55 waters in V_0).

For interpretation purposes, we lump the conformational clusters together based on kinetic similarity using the PCCA algorithm.[29] The PCCA algorithm uses the eigenvectors of the Markov state model to lump together clusters which are kinetically close, resulting in a set of so-called macro-states. In our case we arbitrarily chose to lump the clusters into 8 macro-states as they are enough to represent the binding process and give a clear image of the metastable states of benzamidine on trypsin.

3 Results

The simplest case of an iterative sampling scheme is one in which at each epoch the simulations are started from the same initial conditions used in the first epoch. This is equivalent of course to sam-

pling all epochs at once, corresponding to the current main mode of operation in high-throughput molecular dynamics simulations. Therefore, in order to have a reference dataset, we have generated 489 runs of 100 ns of the trypsin-benzamidine system starting from 10 initial configurations and we analyzed the convergence properties by varying the length of the runs and the number of trajectories used in the analysis. A similar calculation was previously performed on Villin folding,[21] using a relative entropy metric, showing that many shorter simulations can at least sample as well as fewer long ones. Instead of the relative entropy to a golden standard Markov model, we calculated for each trajectory length and trajectory count combination the standard free energy of the state with the highest equilibrium probability which should correspond to the bound pose. This is more intuitive to understand as we are interested in getting a correct estimate of the binding free energy. In this case for ligand binding the results are shown in Figure 1. For this analysis we used the same lag time equation used in the accurate adaptive models (see Eq. 4) and averaged the values over 4 different bootstrapping iterations.

From Figure 1 we can see that even though the hyperbolic curve of equal computing cost is roughly visible, it is skewed, thus giving better free energy estimates for longer simulations. This is indicative of the fact that the short simulations don't have time to sample the bound state enough to correctly estimate the free energy. Therefore, even for fast processes such as benzamidine binding to trypsin, it would require a very large amount of short simulations to sample the process well enough. However, in the iterative sampling scheme, it will be possible to obtain good estimates even with few, very short simulations and trajectories of 10 ns.

Given the reference dataset, we now look at a single adaptive instance in order to demonstrate how the adaptive sampling scheme progresses epoch after epoch. The ten starting conformations at epoch 1 are all shown superimposed in Figure 2 epoch 1a, corresponding to simply random configurations of the ligand from the bulk (water not shown for clarity). Note that each simulation consists of a single ligand and protein but ligands are superimposed for visualization. After running the 10 simulations for 10 ns, we can build the first Markov model. The resulting kinetic macrostates are shown in Figure 2 epoch 1b with all ligand conformations superimposed. Ligand poses with the

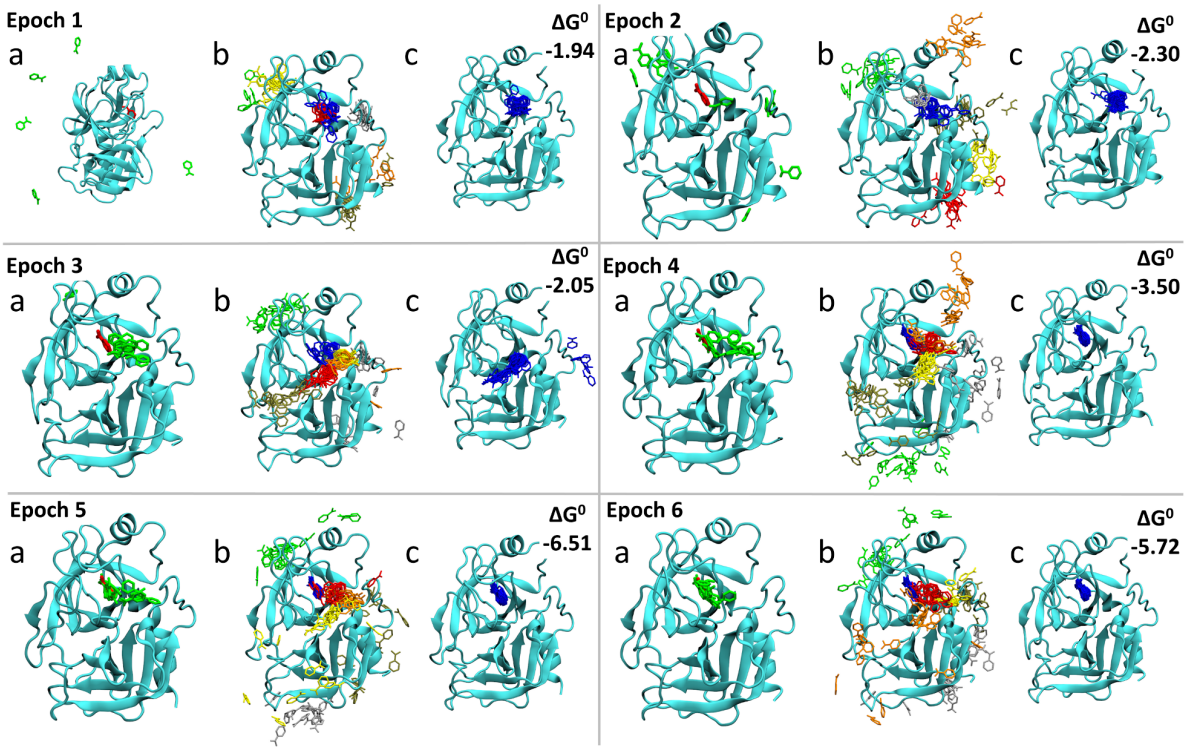

Figure 2: Six epochs of a single adaptive run are visualized. The starting conformations of each epoch are shown in subfigure a of each epoch in green; the bound pose is shown in red. The kinetic clusters produced by the Markov model are shown in subfigures b with all ligand conformations superimposed on one image and each cluster colored differently. In subfigures c, the cluster with the lowest free energy is shown together with the calculated value of the free energy of the cluster.

lowest ΔG^0 of the epoch 1 have been identified using the accurate Markov model (Figure 2 epoch 1c), but these poses are not in the binding pocket and accordingly their standard binding affinity is wrong. The most stable poses of epoch 1 are restarted using the adaptive scheme in Eq. 2 leading to the starting configurations of Figure 2 epoch 2a. None of the new configurations are starting from the bound pose as this pose was not identified yet in the Markov model, however one configuration is at the entrance of the pocket. Note that no information on the position of the binding pocket is given to the model a priori. The same procedure is followed for all following epochs. At epoch 3 and epoch 4 we can see that the starting configurations are all selected outside the binding pocket (Figure 2 epoch 3a,4a). Thus in epoch 4 the binding conformation is detected as can be seen in the kinetic macrostates (Figure 2 epoch 4b colored blue). This leads the accurate model in Figure

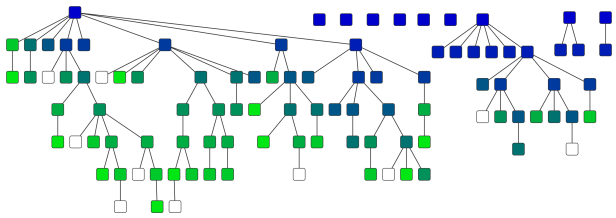

Figure 3: A genealogical tree of all simulations of a single adaptive run is shown. Starting from 10 initial trajectories (dark blue tree roots and single nodes), following epochs are colored on a gradient to green with the last epoch being white. Every simulation is connected to its parent simulation from which its starting configuration was chosen.

2 epoch 4c to detect correctly the configuration with the lowest free energy matching the crystal pose for the first time. However due to lack of sampling of the binding configuration and pathway, the free energy estimate of the binding pocket provided by the accurate model is still not very close to the experimental[30] (-3.5 vs experimental -6.2 kcal/mol). At epoch 5 and 6 with the additional sampling produced by spawning from within the bound state and the entry pathway (Figure 2 epoch 5,6a) we can see that the free energy estimate becomes more accurate. Next epochs are important to establish the stability of this metastable state and to better connect the transition matrix in order to build Markov models at longer lag times. After 10 epochs the lagtimes converge for most adaptive Markov models (i.e. see Figure S4). Having a connected Markov model at longer lag times allows us to compute more correct estimates from it.

It is helpful to visualize how the sampling scheme restarted simulations. In Figure 3, the complete respawn tree is shown from epoch 1 to epoch 10 for a single adaptive run. Of the ten initial simulations only two generate significant sampling. This is due to the fact that only few of the epoch 1 simulations found stable conformations and these conformations far outweigh the residence time of any other conformations on trypsin. Thus the next epochs sample heavily from those simulations and from the ones spawned from them. In the early epochs only the most resident clusters are selected for respawning, while later the states are better spread across the metastable states found. In general, it might be useful to proceed with a two phase exploration and refinement scheme as suggested in Ref.[24] meant to better connect the matrix. However this obliges to take a

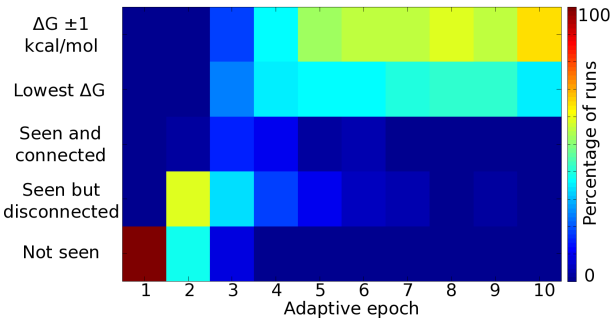

Figure 4: Progress of all adaptive runs through different stages of identification of the bound state, correct ranking and correct estimation of it's free energy over the epochs. Multiple MSMs of each adaptive run in each epoch were constructed to obtain more correct progress estimates.

decision when the exploration phase finishes and the refinement phase starts. The exploration phase is actually very fast as the mean first passage time of binding is of the order of half a microsecond, while a single epoch is already an aggregate hundred nanoseconds, therefore we keep the same scheme for simplicity and avoid as much as possible human intervention.

The previous analysis showed us a single case of adaptive sampling. Using GPUGRID we have replicated ten times the adaptive sampling scheme (adaptive runs A_1 to A_{10}) to obtain statistics of the reproducibility of the protocol. Each sampling run is performed independently. The progression towards an accurate model of the binding process is described in Figure 4 as the achievement of a series of steps: the bound state is seen but disconnected in the transition matrix, seen and connected, the state is also the most kinetically favorable and finally if the binding affinity is within ± 1 kcal/mol of the experimental value of -6.2 kcal/mol. In this analysis the accurate model was used, using as lag time equation 4 and varying the number of clusters depending on aggregate simulation. Values for each run and epoch were calculated for 20 clusterings to reduce sensitivity to the cluster definitions and values are reported as percentages.

At the first epoch, common to all adaptive runs, the bound pose has not been seen. In epoch 2 more than half of the A_i runs have seen the bound pose, but in most it is disconnected from the transition matrix due to lack of statistics and still none has reached close to the experimental free energy. At epoch 3, most runs still have the bound pose disconnected but more are connected and

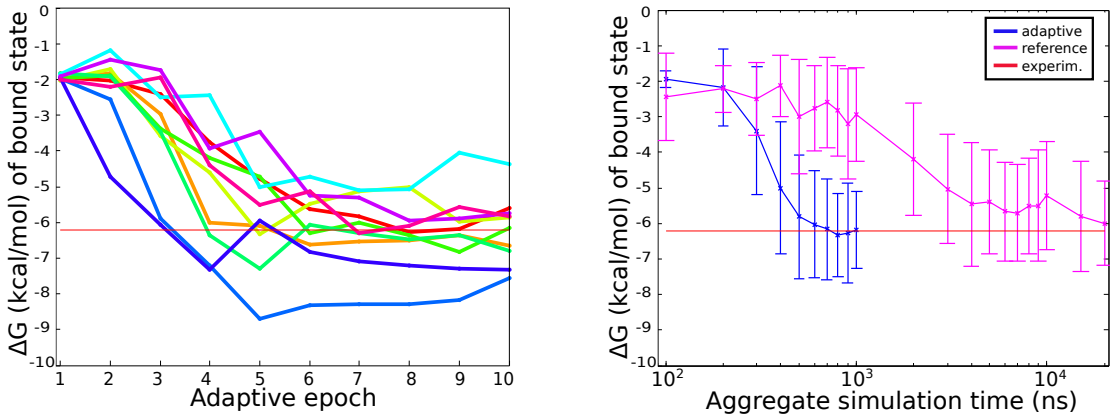

Figure 5: (a) For each adaptive run $A_1 - A_{10}$ we calculated the average standard free energy of the cluster closest to the bound pose at each epoch. To calculate the average, 20 clusterings were performed for each run on each epoch with a standard deviation of 0.71 kcal/mol. Progressively the standard free energies are reaching the experimental value. (b) Averaged adaptive free energies over the 10 runs are compared to random full-length non-adaptive simulations taken from the reference dataset with aggregate simulation times given in the x axis. The adaptive simulations converge close to the experimental free energy an order of magnitude faster than the non-adaptive.

a small percentage even reach the experimental free energy within 1 kcal/mol. Numbers are nicely progressing until epoch 10 where there are around 65% runs within 1 kcal/mol of the experimental value and around 35% runs determine that it is the most stable pose.

Across the epochs, the predicted binding affinity gets progressively more accurate. Figure 5a shows the estimated binding affinity for the best pose, for each adaptive run and for each epoch averaged over 20 clusterings. As before, we used the accurate Markov state models. The estimations quickly decrease towards the experimental value of -6.2 kcal/mol in just a few epochs. However, afterwards the exact convergence to this value seems to slow down. Nevertheless at epoch 10 only 1 microsecond of sampling has been generated. Compared to classical high-throughput sampling, this corresponds close to the bottom left corner of Figure 1 (100 trajectories of 10 ns). On the other hand the average free energy value from the 10 runs even at epoch 6 (just 600 ns data), gives the same accuracy as classically sampled Markov models of total 7.7 μs computation time using longer trajectories. This is a greater efficiency of roughly one order of magnitude and is corroborated by

Figure 5b where the average free energy of the ten adaptive runs is compared to non-adaptive simulations from the reference dataset of equal aggregate simulation time. Additionally in Figure S5 we show the convergence of the equilibrium probability of the bound state.

4 Conclusion

In this paper we demonstrated an automatic, iterative scheme for rapidly learning simplified (Markov) models of molecular simulations of a ligand binding process and using these models to direct the sampling of successive iterations.

The computational process of binding of benzamidine to trypsin is governed by the kinetic rates, $k_{on} = 4.4 * 10^8 M^{-1} s^{-1}$ and $k_{off} = 2.8 * 10^4 s^{-1}$ (see Figure S3). These rates allow us to determine the characteristic time scales in the simulation assuming a simple model of complex formation at the computational experimental concentration of $C_{comp} = 0.0037M$, $\tau_{on} = 618$ ns, $\tau_{off} = 28$ μs. In a classical high-throughput molecular dynamics setting, in Figure 1 we show that it requires at least 7.7 μs of sampling to have an accurate estimate of the binding affinity. This can be obtained by fewer long simulations or many short ones. The learning and sampling method shows that for the case of trypsin-benzamidine it is possible to choose an iterative sampling scheme which provides speed-ups of around one order of magnitude over trivial high-throughput molecular dynamics.

Ideally in an iterative sampling scheme, one would hope that the simulation time required is given by the highest barrier in the simplified Markov model, which might or might not correspond to the full binding event depending on the fine graining of the simplified model itself. On the other hand, if the simplified model decomposes this process in two slow modes, then the sampling time should be of the order of the slowest of these two. In some cases, especially the most complex and slow biological systems, this could lead to several orders of magnitude speed-up.

Acknowledgement

We finally thank all the volunteers of GPUGRID who donate GPU computing time to the project. GDF acknowledges support by the Spanish Ministry of Science and Innovation(Ref. BIO2011-27450). We thank Acellera Ltd for funding.

Supporting Information Available

Details on the construction of the accurate models of the adaptive runs. Five figures. S1 top timescales of naively sampled dataset. S2 Benzamidine atoms used for the contact metrics. S3 convergence of kinetic rates for naively sampled dataset based on lagtimes. S4 timescales of a single adaptive run after 10 epochs. S5 convergence of equilibrium probability of bound state by aggregate simulation time. This information is available free of charge via the Internet at http://pubs.acs.org This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Shaw, D. E. et al. *Commun. ACM* **2008**, *51*, 91–97.

(2) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. *Science* **2011**, *334*, 517–520.

(3) Jensen, M.; Jogini, V.; Borhani, D. W.; Leffler, A. E.; Dror, R. O.; Shaw, D. E. *Science* **2012**, *336*, 229–233, PMID: 22499946.

(4) Arkhipov, A.; Shan, Y.; Das, R.; Endres, N. F.; Eastwood, M. P.; Wemmer, D. E.; Kuriyan, J.; Shaw, D. E. *Cell* **2013**, *152*, 557–569.

(5) Shirts, M.; Pande, V. S. *Science* **2000**, *290*, 1903–1904, PMID: 17742054.

(6) Hummer, G.; Kevrekidis, I. G. *J. Chem. Phys.* **2003**, *118*, 10762–10773.

(7) Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. *J. Chem. Phys.* **2011**, *134*, 174105–174105–23.

(8) Singhal, N.; Snow, C. D.; Pande, V. S. *J. Chem. Phys.* **2004**, *121*, 415–425, PMID: 15260562.

(9) Sriraman, S.; Kevrekidis, I. G.; Hummer, G. *J. Phys. Chem. B* **2005**, *109*, 6479–6484.

(10) Pan, A. C.; Roux, B. *J. Chem. Phys.* **2008**, *129*, 064107, PMID: 18715051 PMCID: PMC2674374.

(11) Schütte, C.; Fischer, A.; Huisinga, W.; Deuflhard, P. *J. Comput. Phys.* **1999**, *151*, 146–168.

(12) Noé, F.; Fischer, S. *Curr. Opin. Struct. Biol.* **2008**, *18*, 154–162.

(13) Buch, I.; Giorgino, T.; De Fabritiis, G. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 10184–10189.

(14) Silva, D.-A.; Bowman, G. R.; Sosa-Peinado, A.; Huang, X. *PLoS Comput. Biol.* **2011**, *7*, e1002054.

(15) Sadiq, S. K.; Noé, F.; De Fabritiis, G. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 20449–20454.

(16) Muff, S.; Caflisch, A. *J. Chem. Phys.* **2009**, *130*, 125104, PMID: 19334897.

(17) Singhal, N.; Pande, V. S. *J. Chem. Phys.* **2005**, *123*, 204909, PMID: 16351319.

(18) Hinrichs, N. S.; Pande, V. S. *J. Chem. Phys.* **2007**, *126*, 244101.

(19) Pronk, S.; Larsson, P.; Pouya, I.; Bowman, G. R.; Haque, I. S.; Beauchamp, K.; Hess, B.; Pande, V. S.; Kasson, P. M.; Lindahl, E. Copernicus: A New Paradigm for Parallel Adaptive Molecular Dynamics. 2011.

(20) Weber, J. K.; Pande, V. S. *J. Chem. Theory Comput.* **2011**, *7*, 3405–3411.

(21) Bowman, G. R.; Ensign, D. L.; Pande, V. S. *J. Chem. Theory Comput.* **2010**, *6*, 787–794.

(22) Bowman, G. R.; Huang, X.; Pande, V. S. *Biophys. J.* **2009**, *96*, 575a.

(23) Huang, X.; Bowman, G. R.; Bacallado, S.; Pande, V. S. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 19765–19769, PMID: 19805023.

(24) Zhou, T.; Caflisch, A. *J. Chem. Theory Comput.* **2012**, *8*, 2134–2140.

(25) Harvey, M. J.; Giupponi, G.; Fabritiis, G. D. *J. Chem. Theory Comput.* **2009**, *5*, 1632–1639.

(26) Buch, I.; Harvey, M. J.; Giorgino, T.; Anderson, D. P.; De Fabritiis, G. *J. Chem. Inf. Model.* **2010**, *50*, 397–403.

(27) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. *J. Chem. Phys.* **2013**, *7*, 07B604.

(28) Schwantes, C. R.; Pande, V. S. *J. Chem. Theory Comput.* **2013**, *9*, 2000–2009.

(29) Deuflhard, P.; Weber, M. *Linear Algebra Appl.* **2005**, *398*, 161–184.

(30) Mares-Guia, M.; Shaw, E. *J. Biol. Chem.* **1965**, *240*, 1579–1585, PMID: 14285494.