Population Annealing for Molecular Dynamics Simulations of Biopolymers

Henrik Christiansen,*,^{†,¶} Martin Weigel,^{*,‡} and Wolfhard Janke^{*,†}

†Institut für Theoretische Physik, Universität Leipzig, Postfach 100 920, 04009 Leipzig, Germany

‡Applied Mathematics Research Centre, Coventry University, Coventry CV1 5FB, England ¶Corresponding author.

E-mail: henrik.christiansen@itp.uni-leipzig.de; martin.weigel@complexity-coventry.org; wolfhard.janke@itp.uni-leipzig.de

Abstract

We adapt population annealing, a simulation scheme originating from Monte Carlo simulations, to all-atom molecular dynamics sim-We demonstrate its excellent perulations. formance in computer simulations of biopolymers by investigating the folding of the pentapeptide met-enkephalin, a common test protein. The method is compared to the well established parallel tempering approach of accelerating simulations, and it is found to yield a similar broadening of the accessible configuration space compared to canonical simulations. In contrast to parallel tempering, however, population annealing scales to a nearly arbitrary number of parallel processors and it is thus able to tap into the massively parallel computing power available in petaflop and future exaflop machines.

Graphical TOC Entry



Keywords

Population Annealing, Molecular Dynamics, Protein Folding

Introduction

Simulating properties of biomolecules such as the folding of proteins is one of the computationally most challenging problems, largely as a consequence of the rugged free-energy landscape of such systems. A range of methods has been developed to overcome the problem of the peptide getting trapped in a local minimum.^{1–4} The most popular choice is parallel tempering 2,5,6 (also known as replica exchange) which has been shown to successfully sample a broad configuration space when applied to peptides.^{7,8} This method uses a small number of replicas which are, a priori, simulated independently at different temperatures. At regular intervals the replicas exchange configurations (or temperatures) with a probability adjusted to their relative Boltzmann weight. Although this approach is easily parallelized, the number of processors that can be reasonably used in parallel is limited by the increasing time it takes for a system to traverse the whole temperature range when the number of temperature points becomes too large.

Population annealing 9^{-14} is another generalizedensemble simulation scheme, which was originally introduced in the context of Monte Carlo simulations. While it was found to be of similar efficiency at dealing with complex free-energy landscapes as the parallel tempering method, it can easily make use of many thousands of processors including GPUs¹⁵ and scales extremely well. The method is based on the principles of sequential Monte Carlo. It consists of setting up an ensemble of R independent configurations at high temperatures where equilibration is straightforward. The population is then sequentially cooled down to lower temperatures in small steps. At each step, the population is resampled according to the Boltzmann weight of configurations at the lower temperature and then evolved for a number of simulation steps. This keeps the population in equilibrium and thus observables can be calculated as population averages at each of the temperatures. Parallelization is over independent members of the population, and this allows for the good scaling properties as population sizes R as large

as 10^6 and beyond are not uncommon.

Population annealing has been shown to perform well, e.g., in the simulation of spin glasses.^{10,12} Here we show how it can be adapted to molecular dynamics simulations and thereby unleash the power of massively parallel computations for applications in protein folding and simulations of other biomolecular systems.

Methods

The population annealing (PA) method proposed here is a straightforward extension of canonical (typically NVT) molecular dynamics (MD) simulations. It achieves substantially improved equilibration properties by following a population of systems that are independently evolved with the underlying MD algorithm while periodically replicating particularly well equilibrated copies (reminiscent of "go with the winners" strategies 16). The temperature annealing run starts with a population of R system replicas that are initially equilibrated at some high temperature T_0 where relaxation times are sufficiently small. In practice, it suffices for this purpose to run a single copy at a large T_0 and sample independent configurations out of the time series as the starting configurations for the R replicas, potentially followed by a number of MD steps for each replica to fully ensure statistical independence. The actual annealing process then proceeds by successively lowering the temperature from T_{i-1} to $T_i < T_{i-1}$, resampling the population with the relative Boltzmann weight of configurations at T_i , and simulating each new replica independently for θ MD steps.

This scheme for population annealing molecular dynamics (PAMD) simulations can be summarized as follows:

- 1. Set up an equilibrium ensemble of R independent copies of the system at some high temperature T_0 .
- 2. Resample the ensemble of systems to a temperature $T_i < T_{i-1}$ by replicating each copy a number of times proportional to the relative Boltzmann weight

 $au_j = e^{-(1/k_B T_i - 1/k_B T_{i-1})E_j}/Q$, where $Q = \sum_{j=1}^R e^{-(1/k_B T_i - 1/k_B T_{i-1})E_j}$ is a normalization factor and E_j is the potential energy of the *j*th replica.

- 3. Update each copy with θ simulation steps of the underlying MD algorithm.
- 4. Calculate observables \mathcal{O} at temperature T_i as population averages.
- 5. Goto step 2 until T_i reaches or falls below the target temperature T_N .

Note that the replication in step 2 includes the case of making zero copies, corresponding to pruning the corresponding configuration from the population. While a number of different implementations for the resampling step are possible, 11,12,15 we here draw R samples from a multinomial distribution with probabilities τ_i , $j = 1, \ldots, R$. This ensures that the total population size is constant, thus simplifying parallel implementations on distributed machines. The choice of temperature steps T_i follows the same rules as for the parallel tempering (PT) method, i.e., sufficient overlap of the energy histograms is required. Temperature steps can also be chosen automatically based on an overlap condition. 15,17

The vast majority of the simulation time is spent in the MD steps advancing the systems. To get good performance there, we rely on the OpenMM package¹⁸ for this part of the simulation. OpenMM is also capable of making use of GPUs, such that the simulation can be easily run on clusters consisting of hundreds or thousands of GPUs. It is important to note that in PAMD simulations the thermostat has to be stochastic in nature: If more than one copy of a replica is created during resampling these are identical initially and the noise from the thermostat is required to make sure that they decorrelate over time.

The possibility of creating more than one copy implies that, after resampling, the population is no longer totally uncorrelated, and these correlations need to be taken into account for the analysis of statistical errors. This is achieved here via a binning analysis, details are described in Ref. 17. During the population annealing simulation the values of the normalizing constants Q are separately recorded. As is easily seen,^{11,15} these estimate the partition function ratios

$$\langle Q(T,T')\rangle = \frac{\mathcal{Z}(T)}{\mathcal{Z}(T')}$$
 (1)

or, equivalently, the free-energy differences $F(T')/k_BT' - F(T)/k_BT = \ln \mathcal{Z}(T) - \ln \mathcal{Z}(T')$. This allows us to not only determine free energies directly, but also to apply the weighted histogram analysis method (WHAM).^{19,20} In practice, we refine the results by a few iterations of the self-consistency equations of Refs. 19,20, but the above procedure already provides excellent starting values.

Results and Discussion

As a test system for demonstrating the efficiency of our method for relaxing the simulated system and broadening of the sampled configuration space, we consider the penta-peptide met-enkephalin in vacuo. For this peptide we cap the ends with acetyl respectively N-methyl Met-enkephalin has the protein segroups. quence Tyr-Gly-Gly-Phe-Met. Below we will focus on the quality of the relaxation of the dihedral angles in the inner amino acids GLY-2, GLY-3, and PHE-4. To model the interactions we employ the AMBER force-field ff94.²¹ The temperature set is adapted from a parallel tempering simulation of the same peptide⁸ and reads $T_i = 700, 585, 489, 409, 342, 286,$ 239, 200 K. We initialize the population of replicas from a canonical, equilibrium simulation at $T_0 = 700$ K. The distribution of dihedral angles Ψ and Φ of GLY-2 from this initial simulation is shown in the Ramachandran plot of Fig. 1. A full parallel tempering simulation performed with the above temperature set results in a virtually identical angle distribution, providing additional evidence for the fact that the simulation is equilibrated at T_0 .

For the population annealing simulation we take configurations from this initial run at T_0 to initialize $R = 10\,000$ replica, and additionally subject them to θ MD steps of the molecular



Figure 1: Ramachandran plot for the amino acid GLY-2 of met-enkephalin at $T_0 = 700$ K obtained from a canonical simulation. The Ramachandran plot obtained from parallel tempering at T_0 looks virtually identical.

dynamics each to ensure that they are statistically independent (the choice of θ is discussed below). We used the stochastic Langevin thermostat with a different pseudo random number generator seed for each of our replica. The friction coefficient of the Langevin thermostat is set to $\gamma = 1/ps$ and the system is advanced using the velocity-Verlet method with a time step of dt = 0.5 fs. In Fig. 2 we show the measured temperature vs. the given heat-bath temperature for two different numbers of MD steps, $\theta = 1000$ and $\theta = 4375$, between the resampling as well as for an annealing scheme with $\theta = 4375$, but without resampling. As is evident from this plot, $\theta = 1000$ MD steps are not sufficient for the equilibration of this system, but $\theta = 4375$ are, such that we use the latter parameter for all simulations discussed below. The plot also shows that omitting the resampling step of PAMD, corresponding to the pure annealing of the population of independent runs (denoted by "anneal" in Fig. 2), worsens the equilibration behavior.

For the actual benchmark, we compared the PAMD simulations to a sequence of canonical simulations at the chosen temperatures as well as to a PT simulation at the same temperature



Figure 2: Measured temperature $T_{\rm m}$ vs. the heat-bath temperature T for PAMD simulations. For $\theta = 4375$ the simulations are well in equilibrium, while this is not the case for $\theta = 1000$. Simulations without resampling ("anneal") are not as well equilibrated as PAMD runs with the same θ .

points, and finally to the variant of the PAMD simulations with the resampling step disabled. To ensure fair rules for the competition, all methods were assigned the same computational budget corresponding to a total of 200 ns of MD simulation — the computational overhead of the swaps in parallel tempering and the resampling steps in population annealing is negligible. For each simulation method, the budget is distributed differently. In the canonical simulations the protein was equilibrated for 12.5 ns for each of the 8 temperatures, followed by another 12.5 ns of simulation during which 10000 measurements were taken. For population annealing (and the annealing simulation without resampling) we used the configurations from this canonical simulation at T_0 as our R = 10000start configurations (see Fig. 1), and subjected them to an additional $\theta = 4375$ MD steps each to improve decorrelation. The actual simulations (including the initial 4375 MD steps) then ran for a total of 175 ns, corresponding to the above mentioned 4375 MD steps per replica and temperature. In the parallel tempering simulation a total of 50 ns were spent in the equilibration of the system. The remaining 150 ns were equally distributed on the 8 temperatures. Measurements were taken before each parallel



Figure 3: Energy histograms at the lowest temperature, T = 200 K, as obtained from the population annealing (PA), parallel tempering (PT), population annealing without resampling ("anneal"), and canonical ("canon.") simulations, respectively.

tempering exchange step. In total, this step was performed 10 000 times, amounting to 3750 MD steps between the exchanges.

In Fig. 3 we show the distribution of the potential energy of met-enkephalin at the lowest temperature T = 200 K from the canonical simulation, the population annealing with and without resampling as well as the parallel tempering simulation. The histograms obtained from the population annealing and parallel tempering are compatible with each other, while it is apparent that both the canonical simulation and the annealing simulation without resampling do not reproduce this distribution but are shifted towards higher energies. For the canonical simulation this indicates that the simulation was trapped in a local energy minimum from which it was unable to escape. The annealing simulation distribution indicates that without resampling, the $\theta = 4375$ MD steps at each temperature are not sufficient to keep the system in equilibrium at the given cooling rate, consistent with the observation in Fig. 2.

In Fig. 4 we present Ramachandran plots for the three significant amino acids, GLY-2, GLY-3, and PHE-4 at the lowest temperature T = 200 K from the population annealing as compared to the canonical simulation. As expected, the distributions of the dihedral angles Φ and Ψ for the two sets of simulations differ substantially. While the canonical simulation got trapped and thus only simulated a fraction of the available conformations, population annealing sampled a wide configuration space. With the same amount of computational resources thus a much better sampling of the configuration space is achieved.

While PAMD is hence clearly superior to a purely canonical simulation, it is important to check how it competes against parallel tempering as the *de facto* standard method for accelerated, generalized-ensemble simulations. A corresponding comparison can be found in Fig. 5, again showing Ramachandran plots of the three peptides GLY-2, GLY-3, and PHE-4 at the lowest temperature T = 200 K. It is evident that both methods sample a wide configuration space with a few areas having been discovered only by PT and others only by PA. Overall, the quality of sampling provided by the two techniques is comparable for this system.

The crucial advantage of PA lies in the fact that the sampling can be improved essentially arbitrarily by adding parallel resources used to increase the population size. This is illustrated in Fig. 6, where we compare the results for GLY-3 of a PAMD simulation with twice the number of replicas $R = 2 \times 10^4$, but otherwise using identical simulation parameters and the same initial canonical simulation. The results are compared to the PT simulation employing 200 ns of MD already shown in Fig. 5. Given sufficient parallel resources, the wall-clock time of this enlarged PAMD simulation is approximately the same as that for the previous PAMD run. As is clearly seen from Fig. 6, however, the sampling of configuration space is significantly improved, and the area of $\Phi \approx -90$ and $\Psi \approx 0$ that is difficult to observe is now sampled significantly better than with either the shorter PA or the PT runs.

Conclusion

We have shown that the combination of population annealing with molecular dynamics simulations is a very promising new tool for com-



Figure 4: Ramachandran plots for (a) GLY-2, (b) GLY-3, and (c) PHE-4 at T = 200 K obtained from the population annealing (PA) and canonical simulation. The total run time of both methods was set to 200ns. The configuration space sampled by the poulation annealing simulation is far superior to the space sampled by a canonical simulation.



Figure 5: Ramachandran plots for (a) GLY-2, (b) GLY-3, and (c) PHE-4 at T = 200 K obtained from the population annealing (PA) and parallel tempering (PT) simulations. The total amount of parallel work allowed for each method was the equivalent of 200 ns of MD steps. The fraction of configuration space sampled by the two methods is approximately equivalent.

puter simulations of biomolecules. As demonstrated for the folding of a common test protein, the penta-peptide met-enkephalin, it yields a broadening of the accessible configuration space and faster relaxation on par with the well established parallel tempering heuristic, but with the potential of employing essentially arbitrarily large parallel resources that is lacking in parallel tempering. Population annealing combined with Monte Carlo simulations was already shown to scale to millions of threads for spin systems on GPU clusters.¹⁵ A wide range of additional improvements to this scheme come to mind, including more advanced and potentially self-adaptive schedules of simulation temperatures, sweep protocols and population sizes,^{15,17} and adaptations for other ensembles such as NVE and combinations with umbrella sampling. Crucially, however, even the extremely simple extension of the standard molecular dynamics simulation technique as presented and discussed here provides outstanding performance given sufficiently powerful parallel resources. For numerical studies of biopolymers and related macromolecular systems this opens the door to the world of highly efficient com-



Figure 6: Ramachandran plot for GLY-3 at T = 200 K obtained from a population annealing simulation with $R = 2 \times 10^4$ replicas as compared to the parallel tempering simulation of total run time 200 ns already shown in Fig. 5.

puter simulations on petaflop supercomputers of the present and the exaflop machines of the future.

Acknowledgement This project was funded by the Deutsche Forschungsgemeinschaft (DFG) under Grant No. SFB/TRR 102 (project B04), and further supported by the Leipzig Graduate School of Natural Sciences "Build-MoNa", the Deutsch-Französische Hochschule (DFH-UFA) through the Doctoral College "L⁴" under Grant No. CDFA-02-07, and the EU Marie Curie IRSES network DIONICOS under Grant No. PIRSES-GA-2013-612707.

References

- Berg, B. A.; Neuhaus, T. Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Phys. Rev. Lett.* **1992**, *68*, 9–12.
- (2) Hukushima, K.; Nemoto, K. Exchange Monte Carlo method and application to spin glass simulations. J. Phys. Soc. Jpn. 1996, 65, 1604–1608.
- (3) Wang, F.; Landau, D. P. Efficient,

multiple-range random walk algorithm to calculate the density of states. *Phys. Rev. Lett.* **2001**, *86*, 2050–2053.

- (4) Laio, A.; Parrinello, M. Escaping freeenergy minima. *Proc. Natl. Acad. Sci.* U.S.A. **2002**, 99, 12562–12566.
- (5) Swendsen, R. H.; Wang, J. S. Replica Monte Carlo simulation of spin glasses. *Phys. Rev. Lett.* **1986**, *57*, 2607–2609.
- (6) Geyer, C. J. Markov chain Monte Carlo maximum likelihood. In *Computing Sci*ence and Statistics: Proceedings of the 23rd Symposium on the Interface, edited by Keramidas, E. M. American Statistical Association, New York, 1991; pp. 156–163.
- (7) Hansmann, U. H. E. Parallel tempering algorithm for conformational studies of biological molecules. *Chem. Phys. Lett.* 1997, 281, 140–150.
- (8) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141–151.
- (9) Iba, Y. Population Monte Carlo algorithms. Trans. Jpn. Soc. Artif. Intell. 2001, 16, 279–286.
- (10) Hukushima, K.; Iba, Y. Population annealing and its application to a spin glass. *AIP Conf. Proc.* **2003**, 690, 200–206.
- (11) Machta, J. Population annealing with weighted averages: A Monte Carlo method for rough free-energy landscapes. *Phys. Rev. E* **2010**, *82*, 026704.
- (12) Wang, W.; Machta, J.; Katzgraber, H. G. Population annealing: Theory and application in spin glasses. *Phys. Rev. E* 2015, *92*, 063307.
- (13) Barash, L. Y.; Weigel, M.; Shchur, L. N.; Janke, W. Exploring first-order phase transitions with population annealing *Eur. Phys. J. Spec. Top.* **2017**, *226*, 595– 604.

- (14) Callaham, J.; Machta, J. Population annealing simulations of a binary hard sphere mixture. *Phys. Rev. E* 2018, 95, 063315.
- (15) Barash, L. Y.; Weigel, M.; Borovský, M.; Janke, W.; Shchur, L. N. GPU accelerated population annealing algorithm. *Comput. Phys. Commun.* **2017**, *220*, 341–350.
- (16) Grassberger, P. Go with the winners: A general Monte Carlo strategy. Comput. Phys. Commun. 2002, 147, 64–70.
- (17) Weigel, M.; Barash, L. Y.; Shchur, L. N.; Janke, W. In preparation.
- (18) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D. et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.* 2017, 13, 1–17.
- (19) Ferrenberg, A. M.; Swendsen, R. H. Optimized Monte Carlo data analysis. *Phys. Rev. Lett.* **1989**, *63*, 1195–1198.
- (20) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. J. Comput. Chem. 1992, 13, 1011–1021.
- (21) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. J. Am. Chem. Soc. 1995, 117, 5179–5197.