

A simple method for automated equilibration detection in molecular simulations

John D. Chodera^{1,*}

¹Computational Biology Program, Sloan Kettering Institute,
Memorial Sloan Kettering Cancer Center, New York, NY 10065

(Dated: May 31, 2015)

Molecular simulations intended to compute equilibrium properties are often initiated from configurations that are highly atypical of equilibrium samples, a practice which can generate a distinct initial transient in mechanical observables computed over the timecourse of the simulation. Traditional practice in simulation data analysis recommends this initial portion be discarded to *equilibration*, but no simple, general, and automated procedure for this process exists. Here, we suggest a conceptually simple, automated procedure that does not make strict assumptions about the distribution of the observable of interest, in which the equilibration region is chosen to maximize the number of effectively uncorrelated samples in the production portion used to compute equilibrium averages. We present a simple reference implementation of the procedure in Python, and demonstrate its utility on both synthetic and real simulation data.

Keywords: molecular dynamics (MD); Metropolis-Hastings; Monte Carlo (MC); Markov chain Monte Carlo (MCMC); equilibration; timeseries analysis; statistical inefficiency; integrated autocorrelation time

INTRODUCTION

Molecular simulations use Markov chain Monte Carlo (MCMC) techniques [1] to sample configurations x from an equilibrium distribution $\pi(x)$, either exactly (using Monte Carlo methods such as Metropolis-Hastings) or approximately (using molecular dynamics integrators without Metropolization) [2].

Due to the sensitivity of the equilibrium distribution $\pi(x)$ to small perturbations in configuration x and the difficulty of producing sufficiently good guesses of typical equilibrium configurations, these molecular simulations are often started from highly atypical initial conditions. For example, simulations of biopolymers might be initiated from a fully extended conformation unrepresentative of behavior in solution, or a geometry derived from a fit to diffraction data collected from a cryocooled crystal; solvated systems may be prepared by periodically replicating a small solvent box equilibrated under different conditions, yielding atypical densities and solvent structure; liquid mixtures or lipid bilayers may be constructed by using methods that fulfill spatial constraints (e.g. PackMol [3]) but create locally atypical geometries, requiring long simulation times to relax to typical configurations.

As a result, traditional practice in molecular simulation has recommended some initial portion of the trajectory be discarded to *equilibration* (also called *burn-in*¹ in the MCMC literature [4]). While this practice is strictly unnecessary for the time-average of quantities of interest to eventually converge to the desired expectations [5], and statisticians often do not recommend this procedure in best practices [4], this habit may be a result of the relative simplicity of probabilistic densities that statisticians must sample from compared

to the highly complex densities faced in nearly all molecular simulations. In the simulation field, discarding initial samples to equilibration nevertheless often allows the practitioner to avoid what may be impractically long run times to eliminate the bias in computed properties in finite-length simulations induced by atypical initial starting conditions.

As an illustrative example, consider the computation of the average density of liquid argon under a given set of reduced temperature and pressure conditions Figure 1. To initiate the simulation, an initial dense liquid geometry at reduced density $\rho^* \equiv \rho\sigma^3 = 0.960$ was prepared and subjected to local energy minimization. Figure 1 (top) depicts the relaxation behavior of 100 simulations initiated from the same configuration with different random initial velocities and integrator random number seeds. The average (black line) and standard deviation (shaded grey) shows that all realizations of this simulation show a characteristic relaxation behavior away from the initial density toward a new equilibrium density. The expectation of the running average of the density over many realizations of this procedure (Figure 1, bottom) significantly deviates from the actual expectation, which would lead to biased estimates unless simulations were sufficiently long to eliminate this starting point dependent bias. Note that this significant bias is present because the *same* atypical starting condition is used for every realization of this simulation process.

Consider successively sampled configurations x_t from a molecular simulation, with $t = 1, \dots, T$. We presume we are interested in computing the expectation $\langle A \rangle \equiv \int dx A(x) \pi(x)$ of a mechanical property $A(x)$. For convenience, we will refer to the timeseries $a_t \equiv A(x_t)$, with $t = 0$. The estimator $\hat{A} \approx \langle A \rangle$ constructed from the entire dataset is given by

$$\hat{A}_{[1,T]} \equiv \frac{1}{T} \sum_{t=1}^T a_t. \quad (1)$$

While $\lim_{T \rightarrow \infty} \hat{A}_{[1,T]} = \langle A \rangle$ for an infinitely long simula-

* Corresponding author; john.chodera@choderalab.org

¹ The term *burn-in* comes from the field of electronics, in which a short “burn-in” period is used to ensure that a device is free of faulty components—which often fail quickly—and is operating normally [4].

tion², the bias in $\hat{A}_{[1,T]}$ may be significant in a simulation of finite length T .

By discarding samples $t < t_0$ to equilibration, we hope to eliminate the initial transient and provide a less biased estimate of $\langle A \rangle$,

$$\hat{A}_{[t_0,T]} \equiv \frac{1}{T - t_0 + 1} \sum_{t=t_0}^T a_t. \quad (2)$$

We can quantify the bias in an estimator \hat{A} by the expected error $\delta^2 \hat{A}$,

$$\delta^2 \hat{A} \equiv E_{x_0} \left[\left(\hat{A} - \langle A \rangle \right)^2 \right]. \quad (3)$$

where $E_{x_0}[\cdot]$ denotes the expectation over independent realizations of the simulation from the same initial configuration x_0 but different initial velocities and random number seeds.

Here, we concern ourselves with this question: Is there a simple approach to choosing an optimal equilibration time t_0 that provides an improved estimate $\hat{A}_{[t_0,T]}$, such that $\delta^2 \hat{A}_{[t_0,T]} < \delta^2 \hat{A}_{[1,T]}$? We note that, for cases in which the simulation is not long enough to reach equilibrium, no choice of t_0 will eliminate bias completely; the best we can hope for is to minimize this bias.

While several automated methods for selecting the equilibration time t_0 have been proposed, these approaches have shortcomings that have greatly limited their use. The reverse cumulative averaging method [6], for example, uses a statistical test for normality to determine the point before which the observable timeseries deviates from normality. While this concept may be reasonable for experimental data, where measurements often represent the sum of many random variables such that the central limit theorem's guarantee of asymptotic normality ensures the distribution of the observable will be approximately normal, there is no such guarantee that instantaneous measurements of a simulation property of interest will be normally distributed. In fact, many properties will be decidedly *non-normal*. For a biomolecule such as a protein, for example, the radius of gyration, end-to-end distance, and torsion angles sampled during a simulation will all be highly non-normal. Instead, we require a method that makes no assumptions about the nature of the distribution of the property under study.

AUTOCORRELATION ANALYSIS

The set of successively sampled configurations $\{x_t\}$ and their corresponding observables $\{a_t\}$ compose a correlated

timeseries of observations. To estimate the statistical error or uncertainty in a stationary timeseries free of bias, we must be able to quantify the *effective number of uncorrelated samples* present in the dataset. This is usually accomplished through computation of the *statistical inefficiency* g , which quantifies the number of correlated timeseries samples needed to produce a single effectively uncorrelated sample of the observable of interest. While this concept is well-established for the analysis of both Monte Carlo and molecular dynamics simulations [9–12], we review it here for the sake of clarity.

For a given equilibration time choice t_0 , the statistical uncertainty in our estimator $\hat{A}_{[t_0,T]}$ can be written as,

$$\begin{aligned} \delta^2 \hat{A}_{[t_0,T]} &\equiv E_{x_0} \left[\left(\hat{A}_{[t_0,T]} - \langle \hat{A} \rangle \right)^2 \right] \\ &= E_{x_0} \left[\hat{A}_{[t_0,T]}^2 \right] - E_{x_0} \left[\hat{A}_{[t_0,T]} \right]^2 \\ &= \frac{1}{T_{t_0}^2} \sum_{t,t'=t_0}^T [\langle a_t a_{t'} \rangle - \langle a_t \rangle \langle a_{t'} \rangle] \\ &= \frac{1}{T_{t_0}^2} \sum_{t=t_0}^T [\langle x_t^2 \rangle - \langle x_t \rangle^2] \\ &\quad + \frac{1}{T_{t_0}^2} \sum_{t \neq t'=t_0}^T [\langle a_t a_{t'} \rangle - \langle a_t \rangle \langle a_{t'} \rangle]. \end{aligned} \quad (4)$$

where $T_{t_0} \equiv T - t_0 + 1$, the number of correlated samples in the timeseries $\{a_t\}_{t_0}^T$. In the last step, we have split the double-sum into two separate sums—a term capturing the variance in the observations a_t , and a remaining term capturing the correlation between observations.

If t_0 is sufficiently large for the initial bias to be eliminated, the remaining timeseries $\{a_t\}_{t_0}^T$ will obey the properties of both *stationarity* and *time-reversibility*, allowing us to write,

$$\begin{aligned} \delta^2 \hat{A}_{[t_0,T]}^{\text{equil}} &= \frac{1}{T_{t_0}} [\langle a_t^2 \rangle - \langle a_t \rangle^2] \\ &\quad + \frac{2}{T_{t_0}} \sum_{n=1}^{T-t_0} \left(\frac{T_{t_0} - n}{T_{t_0}} \right) [\langle a_t a_{t+n} \rangle - \langle a_t \rangle \langle a_{t+n} \rangle] \\ &\equiv \frac{\sigma_{t_0}^2}{T_{t_0}} (1 + 2\tau_{t_0}) \\ &= \frac{\sigma_{t_0}^2}{T_{t_0}/g_{t_0}} \end{aligned} \quad (5)$$

where the variance σ^2 , statistical inefficiency g , and integrated autocorrelation time τ (in units of the sampling interval) are given by

$$\sigma^2 \equiv \langle a_t^2 \rangle - \langle a_t \rangle^2 \quad (6)$$

$$\tau \equiv \sum_{t=1}^{T-1} \left(1 - \frac{t}{T} \right) C_t \quad (7)$$

$$g \equiv 1 + 2\tau \quad (8)$$

² We note that this equality only holds for simulation schemes that sample from the true equilibrium distribution $\pi(x)$, such as Metropolis-Hastings Monte Carlo or Metropolized integration schemes. Molecular dynamics simulations utilizing finite timestep integration without Metropolization will produce averages that may deviate from the true expectation $\langle A \rangle$ [?].

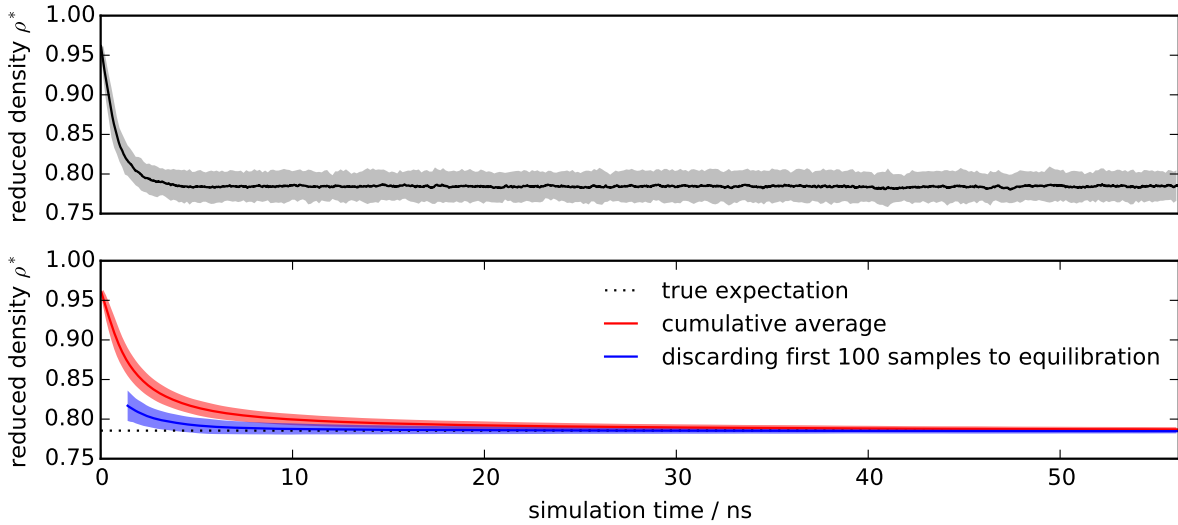


FIG. 1. Illustration of the motivation for discarding data to equilibration. To illustrate the bias in expectations induced by relaxation away from initial conditions, 100 replicates of a simulation of liquid argon were initiated from the same energy-minimized initial configuration constructed with initial reduced density $\rho^* \equiv \rho\sigma^3 = 0.960$ but different random number seeds for stochastic integration. **Top:** The average of the reduced density (black line) over the replicates relaxes to the region of typical equilibrium densities over the first few ns of simulation time. **Bottom:** If the average density is estimated by a cumulative average from the beginning of the simulation (red line), the estimate will be heavily biased by the atypical starting density even beyond 10 ns. Discarding even a small amount of initial data—in this case 100 initial samples (blue line)—results in a cumulative average estimate that converges to the true average (black dotted line) much more rapidly. Shaded regions denote 95% confidence intervals. Simulations were performed using a box of $N = 500$ argon atoms at reduced temperature $T^* \equiv k_B T / \epsilon = 0.850$ and reduced pressure $p^* \equiv p\sigma^3 / \epsilon = 1.266$ using a Langevin integrator [7] with timestep $\Delta t = 0.01\tau$, where characteristic oscillation timescale $\tau = \sqrt{mr_0^2/72\epsilon}$, with $r_0 = 2^{1/6}\sigma$ [8]. A Metropolis Monte Carlo barostat was used with box volume moves attempted every 25 timesteps. Densities were recorded every 25 timesteps.

with the discrete-time normalized fluctuation autocorrelation function C_t defined as

$$C_t \equiv \frac{\langle a_n a_{n+t} \rangle - \langle a_n \rangle^2}{\langle a_n^2 \rangle - \langle a_n \rangle^2}. \quad (9)$$

In practice, it is difficult to estimate C_t for $t \sim T$, due to growth in the statistical error, so common estimators of g make use of several additional properties of C_t to provide useful estimates [4].

The t_0 subscript for the variance σ^2 , the integrated autocorrelation time τ , and the statistical inefficiency t_0 mean that these quantities are only estimated over the production portion of the timeseries, $\{a_t\}_{t=t_0}^T$. Since we assumed that the bias was eliminated by judicious choice of the equilibration time t_0 , this estimate of the statistical error will be poor for choices of t_0 that are too small.

THE ESSENTIAL IDEA

Suppose we choose some arbitrary time t_0 and discard all samples $t \in [0, t_0)$ to equilibration, keeping $[t_0, T]$ as the dataset to analyze. How much data remains? We can determine this by computing the statistical inefficiency g_{t_0} for the interval $[t_0, T]$, and computing the effective number of uncorrelated samples $N_{\text{eff}}(t_0) \equiv (T - t_0 + 1)/g_{t_0}$. If we start at $t_0 \equiv T$ and move t_0 to earlier and earlier points in time,

we expect that the effective number of uncorrelated samples $N_{\text{eff}}(t_0)$ will continue to grow until we start to include the highly atypical initial data. At that point, the integrated autocorrelation time τ (and hence the statistical inefficiency g) will greatly increase, and the effective number of samples N_{eff} will start to plummet.

ILLUSTRATION

Molecular dynamics simulations were performed with OpenMM 6.2 [13] using the Python API. All scripts used to run simulations, analyze data, and generate plots—along with the simulation data itself—are available on GitHub at <http://github.com/choderalab/automatic-equilibration-detection>. The automated equilibration detection scheme is also available in the `timeseries` module of the `pymbar` package as `detectEquilibration()`:

```
from pymbar.timeseries import detectEquilibration
# determine equilibrated region
[t, g, Neff_max] = detectEquilibration(A_t)
# extract equilibrated region
A_t_eqilibrated = A_t[t:]
```

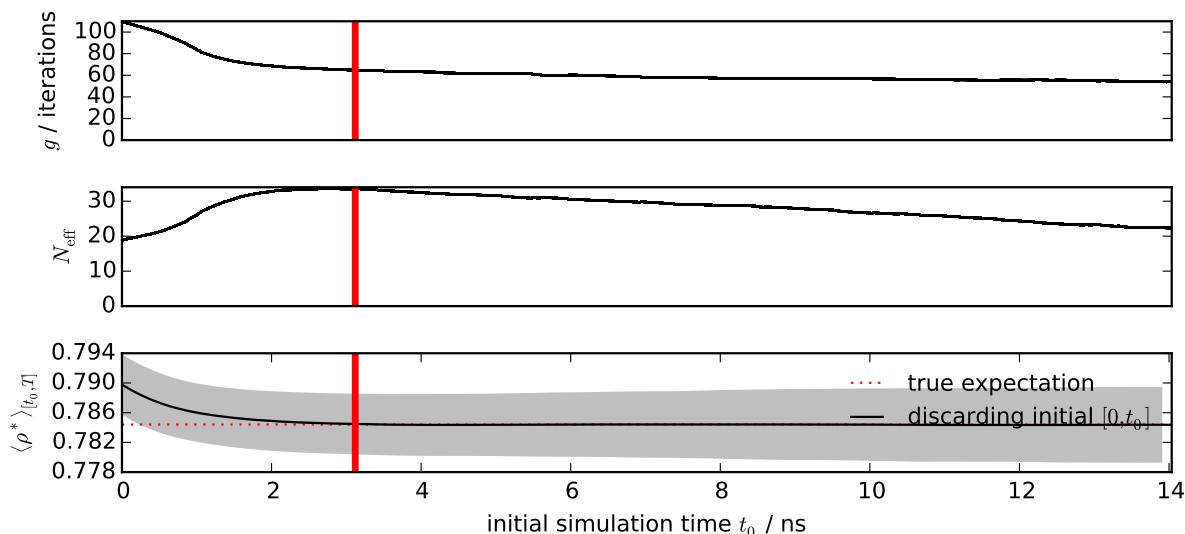


FIG. 2. Statistical inefficiency, number of uncorrelated samples, and bias for different burn-in times. Trajectories of length $T = 2\,000$ iterations (~ 14 ns) were analyzed as a function of equilibration time choice t_0 . Averages over all 100 replicate simulations (all starting from the same initial conditions) are shown as dark lines, with shaded lines showing the standard deviation among replicates. **Top:** The statistical inefficiency g as a function of equilibration time choice t_0 is initially very large, but diminishes rapidly after the system has relaxed to equilibrium. **Middle:** The number of effectively uncorrelated samples $N_{\text{eff}} = (T - t_0 + 1)/g$ shows a maximum at $t_0 = 222$ iterations, suggesting the system has equilibrated by this time. The red vertical line in all plots marks this choice of $t_0 = 222$. **Bottom:** The cumulative density average $\langle \rho^* \rangle$ computed over the span $[t_0, T]$ shows that the bias (deviation from the true estimate, shown as red dashed lines) is minimized for choices of $t_0 \geq 222$ iterations. The standard deviation among replicates (shaded region) grows with t_0 because fewer data are included in the estimate. The choice of optimal t_0 that maximizes N_{eff} (red vertical line) strikes a good balance between bias and variance. The true estimate (red dashed lines) is computed from averaging over the range $[5\,000, 10\,000]$ iterations over all 100 replicates.

ACKNOWLEDGMENTS

We are grateful to William C. Swope (IBM Almaden Research Center), Michael R. Shirts (University of Virginia),

David L. Mobley (University of California, Irvine), Kyle A. Beauchamp (MSKCC), and Robert C. McGibbon (Stanford University) for valuable discussions on this topic, and Joshua L. Adelman (University of Pittsburgh) for helpful feedback and encouragement.

- [1] J. S. Liu, *Monte Carlo strategies in scientific computing*, 2nd ed. (Springer-Verlag, New York, 2002).
- [2] D. Sivak, J. Chodera, and G. Crooks, *Physical Review X* **3**, 011007 (2013), bibtex: Sivak:2013:Phys.Rev.X.
- [3] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, *J. Chem. Theor. Comput.* **30**, 2157 (2009).
- [4] S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, in *Handbook of Markov chain Monte Carlo*, Chapman & Hall/CRC *Handbooks of Modern Statistical Methods* (CRC Press, ADDRESS, 2011), Chap. Introduction to Markov chain Monte Carlo.
- [5] C. Geyer, Burn-in is unnecessary., <http://users.stat.umn.edu/~geyer/mcmc/burn.html>.
- [6] W. Yang, R. Bittetti-Putzer, and M. Karplus, *J. Chem. Phys.* **120**, 2618 (2004).
- [7] D. A. Sivak, J. D. Chodera, and G. E. Crooks, *J. Phys. Chem. B* **118**, 6466 (2014).
- [8] B. Veytsman and M. Kotelyanskii, Lennard-Jones potential revisited., <http://borisv.lk.net/matsc597c-1997/simulations/Lecture5/node3.html>.
- [9] H. Müller-Krumbhaar and K. Binder, *J. Stat. Phys.* **8**, 1 (1973).
- [10] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.* **76**, 637 (1982).
- [11] W. Janke, in *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms*, edited by J. Grotendorst, D. Marx, and A. Murmatsu (John von Neumann Institute for Computing, ADDRESS, 2002), Vol. 10, pp. 423–445.
- [12] J. D. Chodera, W. C. Swope, J. W. Pitera, C. Seok, and K. A. Dill, *J. Chem. Theor. Comput.* **3**, 26 (2007).
- [13] P. Eastman, M. Friedrichs, J. D. Chodera, R. Radmer, C. Bruns, J. Ku, K. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stitch, and C. Klein, *J. Chem. Theor. Comput.* **9**, 461 (2012).

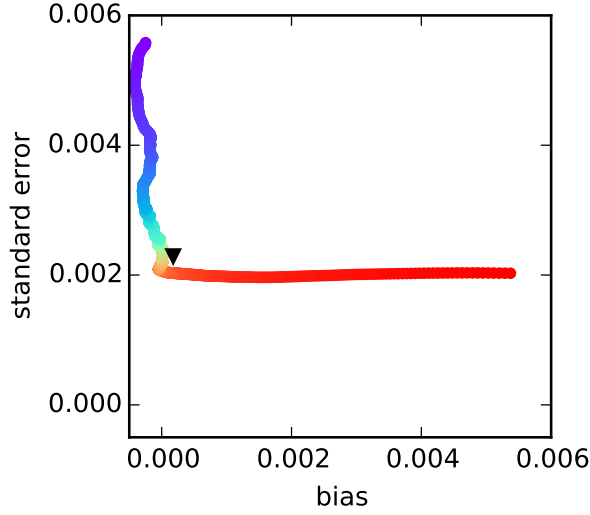


FIG. 3. Bias-variance tradeoff for fixed equilibration time versus automatic equilibration time selection. Trajectories of length $T = 2\,000$ iterations (~ 14 ns) were analyzed as a function of equilibration time choice t_0 . Using 100 replicate simulations, the average bias (average deviation from true expectation) and standard deviation (random variation from replicate to replicate) were computed as a function of a prespecified fixed equilibration time t_0 , with colors running from $t_0 = 0$ (red) to $t_0 = 1\,800$ iterations (blue). As is readily discerned, the bias for small t_0 is initially large, but minimized for larger t_0 . By contrast, the standard error (a measure of variance, estimated here by standard deviation among replicates) grows as t_0 grows above a certain critical time (here, ~ 222 iterations). If the t_0 that maximizes N_{eff} is instead chosen *individually* for each trajectory based on that trajectory's estimated statistical inefficiency $g_{[t_0, T]}$, the resulting bias-variance tradeoff (black triangle) does an excellent job minimizing bias and variance simultaneously, comparable to what is possible for a choice of equilibration time t_0 based on knowledge of the true bias and variance among many replicate estimates.