Supporting Information

RNA hairpin

1. Initial Configurations

We started our ST simulations from two different initial configurations as shown in Figure S1: a near-native state and a random coil. The near-native state of the UUUU tetraloop was created based on the NMR structure of the GCAA tetraloop (first structure of PDB code 1zih [1]) by replacing the loop nucletides from GGCA to UUUU using the Nucleic Acid Builder[2]. The random coil conformation was also created with the Nucleic Acid Builder [2].

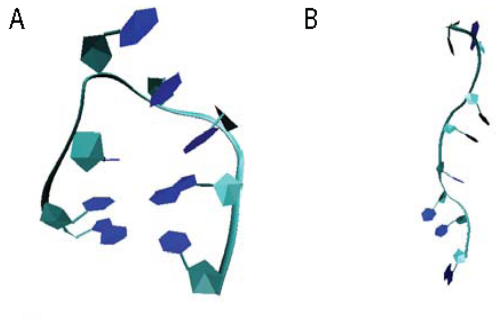


Figure S1. The two initial structures used in this study: A) A near-native conformation and B) a random coil conformation.

2. The Convergence of Weights in Simulated Tempering (ST)

2.1. Simulated Tempering

In Simulated Tempering (ST) [3, 4], configurations are sampled from a mixed canonical ensemble in which the canonical ensembles with different temperatures are weighted differently as defined by a generalized Hamiltonian:

$$\Xi_i(X,p) = \beta_i H(X,p) - g_i \qquad (1)$$

Where $\beta_i = 1/(k_B T_i)$, H(X, p) is the Hamiltonian for the canonical ensemble at temperature T_i. X denotes the conformation and p is the momentum. A priori determined constant g_i is the weight for the temperature T_i.

ST works as follows: a single simulation starts from a particular temperature (T_i) and an attempt is made periodically to change the configuration (X_n) to another temperature (T_j) according to a well defined transition probability by satisfying the detailed balance condition.

$$P_i(X_n, p_n)P(i \to j) = P_j(X_n, p_n')P(j \to i) \qquad (2)$$

The probability of configuration X_n at temperature T_i for the expanded canonical ensemble is,

$$P_i(X_n, p_n) = \frac{1}{Z}\exp(-\Xi_i(X_n, p_n)) = \frac{1}{Z}\exp(-\beta_i H(X_n, p_n) + g_i) \qquad (3)$$

where p_n is the momentum and Z is the partition function for the expanded canonical ensemble. $H(X_n, p_n)$ is the sum of kinetic energy (K) and potential energy (U), and $H(X_n, p_n) = K(p_n) + U(X_n)$

A re-scaling of the momentum ($p_n' = \sqrt{T_j / T_i}\, p_n$) following the exchange causes the kinetic energy to cancel out in the detailed balance equation, and the transition probability after applying the Metropolis criterion is shown below,

$$P_{i \to j} = \min\{1, e^{-(\beta_j - \beta_i)U(X_n) + (g_j - g_i)}\} \qquad (4)$$

where $U(X_n)$ is the potential energy for configuration X_n, which is sampled from the canonical ensemble at T_i. A set of weights need to be pre-determined to calculate these transition probabilities. Without proper weighting, ST simulations will be constrained to a subset of the temperature space and become inefficient. [4, 5] It was shown that weights

leading the system to perform a random walk in temperature space equal the unit-less free energies at different temperatures. [3, 4]

2.2. Simulated Tempering Equal Acceptance Ratio (STEAR) method

It is not an easy task to determine the free energy weights enabling system to perform a random walk in temperature space. The Simulated Tempering Equal Acceptance Ratio (STEAR) method for determining the free energy weights is adopted in this study.[5, 6] This method is based on the property that the free energy weights leading to uniform sampling must yield the same acceptance ratios for both forward and backward transitions from T_i to T_j as shown below.

$$< P_{i\to j}(g_j - g_i, U_i) >=< P_{j\to i}(g_i - g_j, U_j) > \tag{5}$$

where

$$\begin{aligned} < P_{i\to j} >&= \int_{-\infty}^{\infty} P_{i\to j}(g_j - g_i, U_i) P(U_i) dU_i \\ < P_{j\to i} >&= \int_{-\infty}^{\infty} P_{j\to i}(g_i - g_j, U_j) P(U_j) dU_j \end{aligned} \tag{6}$$

where U_i is the potential energy for a configuration sampled from the canonical ensemble at temperature T_i and $P(U_i)$ is the potential energy distribution function (PEDF) at T_i. PEDFs for each temperature are initially estimated from short trial MD simulations and then updated during an equilibration phase preceding the production phase, which uses a static set of weights. By solving Eq. 3, we can obtain a set of near free energy weights.

2.3. Detailed Procedure to update the Weights

The ST algorithm was implemented in version 3.1.4 of the GROMACS[7] molecular dynamics simulation package modified for the Folding@Home[8] infrastructure (http://folding.stanford.edu). In our ST simulations, the temperature list (T_1 … T_n)

containing 56 temperatures is roughly exponentially distributed between 270 and 592 K. The detailed procedure to determine the weights using STEAR is described as below

Obtaining the initial weights. For each of the two initial configurations (see Figure. S1), one 2 ns NVT simulation was carried out at each of 56 temperatures on a computer cluster. Potential energies collected every 0.1 ps from the last nanosecond of these simulations were used to get a rough approximation of the energy distribution at each temperature. The weight (g_i) that gives an equal acceptance ratio for transitions from T_i to T_{i+1} and vice versa is found using Newton's method (See Equation (5)) and g_1 is set to zero.

Updating the weights. Once an initial set of weights has been chosen, we start 1120 ST simulations from each initial configuration on the Folding@Home distributed computing environment. In these simulations, a temperature swap is attempted every 0.2 ps. At regular intervals (about every 300ns of simulation in total) all the new data is collected and only new data is used to refine the approximation of the energy distribution at each temperature. Newton's method is then used to update the weights to satisfy the equal acceptance ratio criterion given the new energy distributions as shown in Equation (4).

2.4. Convergence of the Weights

The weights obtained from two independent sets of ST simulations starting from different

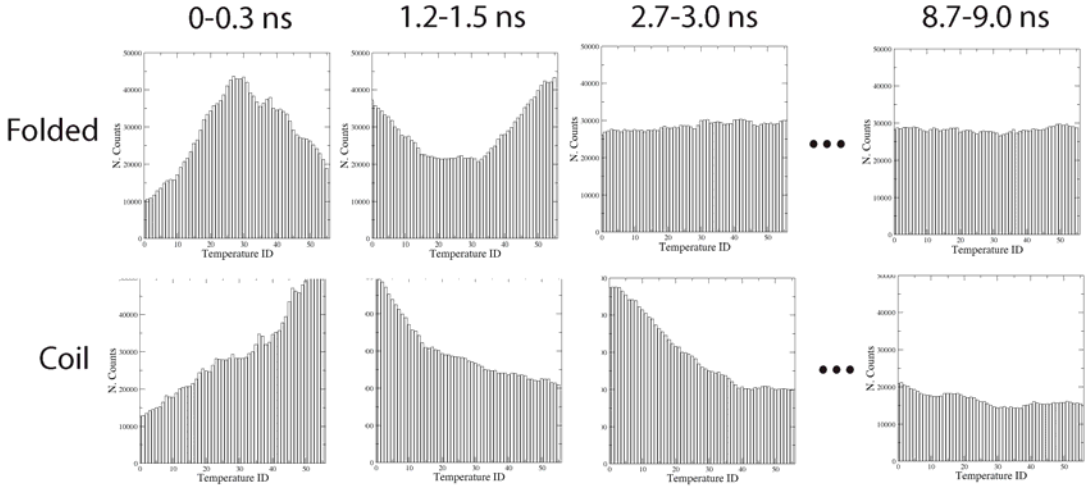


Figure S2. Amount of sampling at different temperatures for ST simulations started from the native (top row) and coil configurations (bottom row) computed from different segment of simulation time 0-0.3ns, 1.2-1.5 ns, 2.7-3.0 ns, and 8.7-9.0ns are displayed. Uniform sampling is reached for both sets of ST simulations indicating the weights are converged.

initial configurations are converged well as shown in Table S1. The weights converge at about 9 ns for each initial configuration. As described before, a set of converged weights, i.e. free energy weights should induce a uniform sampling of the temperature space. As shown in Figure S2, both sets of simulations achieve uniform sampling at about 9ns. Thus, after about 9 ns, the weights are held static and the simulations are continued in what is called the production phase.

3. Molecular Dynamics (MD) Simulation Details

Our MD simulations used the nucleic acid parameters from the AMBER99 force field.[9, 10] The RNA molecule was solvated in a water box with 2543 TIP3P[11] waters and 7 Na^+ ions. The simulation system was minimized using a steepest descent algorithm, followed by a 100ps MD simulation applying a position restraint potential to the RNA heavy atoms. All NVT simulations were coupled to a Nose-Hoover thermostat with a coupling constant of $0.02ps^{-1}$.[12] A cutoff of 10 Å was used for both vdW and short range electrostatic interactions. Long-range electrostatic interactions were treated with the Particle-Mesh Ewald (PME) method.[13] Nonbonded pair-lists were updated every

10 steps with an integration step size of 2 fs in all simulations. All bonds were constrained using the LINCS algorithm.[14]

4. Hierarchical K-medoids clustering algorithm

A hierarchical K-medoids clustering algorithm developed by Boxer *et .al.* [15] is used in this study. In K-medoids clustering one starts by choosing some number of random conformations to be generators. All remaining conformations are then assigned to the generator that they are most similar to, thus forming a state corresponding to each generator. Each generator is then updated by choosing a number of random conformations from its corresponding state and selecting the one that is closest to every other conformation in the state (i.e. the one that is closest to the center of the state) as the new generator. This updating procedure may be continued for some predetermined number of iterations or until the answer converges. The basic idea of hierarchical clustering is to perform K-medoids clustering on the entire dataset and then to recursively perform K-medoids clustering on each state until every state has fewer conformations than some threshold. This threshold is set as an input parameter for the K-medoids clustering algorithm.

T_i	T_j	Δg_{ji}(Helical)	Δg_{ji}(Coil)	Δf_{ji}(Helical) -Δf_{ji}(coil)	kT_i
270	272	377.745	377.722	0.051	2.243
279	282	523.049	523.009	0.097	2.318
292	296	625.190	625.132	0.152	2.426
308	312	552.587	552.525	0.175	2.559
327	332	599.062	599.026	0.127	2.717
347	352	521.894	521.864	0.123	2.883
367	372	458.022	457.984	0.156	3.049
388	393	402.168	402.151	0.100	3.224
411	418	489.160	489.137	0.149	3.415
439	447	477.633	477.615	0.152	3.648
468	476	410.834	410.829	0.107	3.888
497	506	399.965	399.964	0.103	4.129
532	540	303.152	303.148	0.108	4.420
565	574	294.772	294.768	0.120	4.694
583	592	273.443	273.437	0.133	4.844

Table S1. Convergence of the weights is shown for representative temperatures $\Delta g = g_j - g_i$ obtained from distributed computing simulations starting from a helical structure (third column) and a coil structure (fourth column) at different temperature pairs. Differences between free energy differences $\Delta f_{ji} = g_j/\beta_j - g_i/\beta_i$ obtained from simulations starting from a helical structure and a coil structure are displayed in the 5th column. KT at temperature i is shown in the sixth column. Δf_{ji}(Helical)-Δf_{ji}(coil)(KJ/mol) is smaller than KT (KJ/mol) at all temperature pairs.

5. Markov State Models

A Markov model is basically a graph representing the structure and temporal connectivity of some dataset that consists of temporally ordered observations.[16, 17] In this case, each node corresponds to a set of kinetically similar conformations. These nodes are connected by directed edges with corresponding values equal to the probability of transitioning between them. For the model to be Markovian, the probability of transitioning to state j must depend solely on the previous state.

A Markov State Model (MSM) may also be represented by a transition probability matrix as (also see Equation 1 in the main text)

$$P(\Delta t) = T(\Delta t)P(0) \qquad (7)$$

where P(Δt) is a vector of state populations at time Δt, T is the column-stochastic transition probability matrix, and Δt is the lag time (or time step). Using this representation, the time evolution of a vector representing the population of each state may be calculated by repeatedly left-multiplying the column vector by the transition probability matrix. The model also has a corresponding *lag time*, which is effectively the time resolution of the model. Each step, or multiplication by the transition probability matrix, is equivalent to one lag time. For the model to be Markovian there must be a separation of timescales. That is, equilibration within states must occur on timescales faster than the lag time while transitions between states must occur on timescales longer than the lag time. The key is finding an appropriate balance between the number of states in the model and the lag time. A desirable Markov model has few enough states that it may be understood by a person and a lag time shorter than the timescale of the process of interest.

The eigenvalues (μ_k) of the transition matrix each imply a time scale (τ_k).

$$\tau_k = -\frac{\tau}{\ln \mu_k(\tau)} \qquad (8)$$

where μ_k is an eigenvalue of the transition matrix with the lag time τ.

The focus of the current study is thermodynamics instead of kinetics. The first left eigenvector of the transition matrix T_{ij} correspond to the equilibrium distribution[17].

5.1 Splitting into Microstates

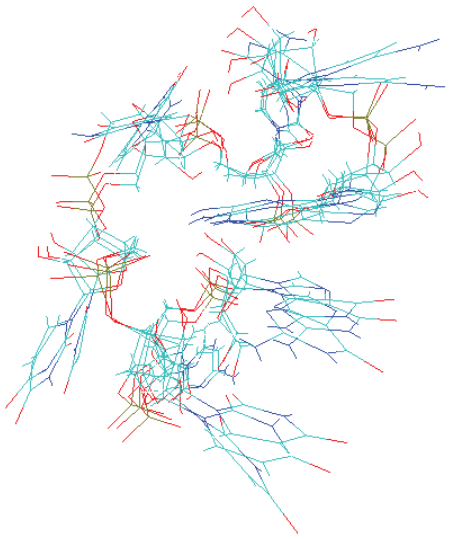


Figure S3. Three example structures from a single microstate.

The first step in our procedure to build an MSM is to divide all the conformations sampled into small sets of structurally similar configurations called microstates [16, 17]. This is accomplished using the hierarchical K-medoids clustering algorithm described in Section 3[15]. For example, by setting the threshold for the hierarchical K-medoids clustering to stop splitting a certain state as 2500 conformations, we divided 1.3 million conformations generated from long ASM seeding simulations into 1,597 microstates. Heavy atom RMSD is used as the distance metric, since it accounts for both local similarities between pairs of conformations as well as global ones,. This distance metric has also been shown to be able to distinguish between kinetically distinct conformations[17]. If the state population threshold is chosen to be small enough then the conformations in one microstate may be considered to be kinetically as well as structurally similar as it would require very few MD steps to get from one to another. As shown in Figure S3, overlaid structures from the same microstate have great structural similarity. Based on this assumption, one may build a microstate Markov model by using the original data to calculate the probability of transitioning between each pair of microstates (stored as a transition probability matrix). Because of the small size of each microstate, this Markov model will have too many states to provide any insight into the nature of the free energy landscape. To gain a clearer understanding of the free energy landscape one may lump

together kinetically similar microstates to form macrostates. These macrostates comprise a new MSM that hopefully has an appropriate separation of timescales.

5.2 Lumping into Metastable States

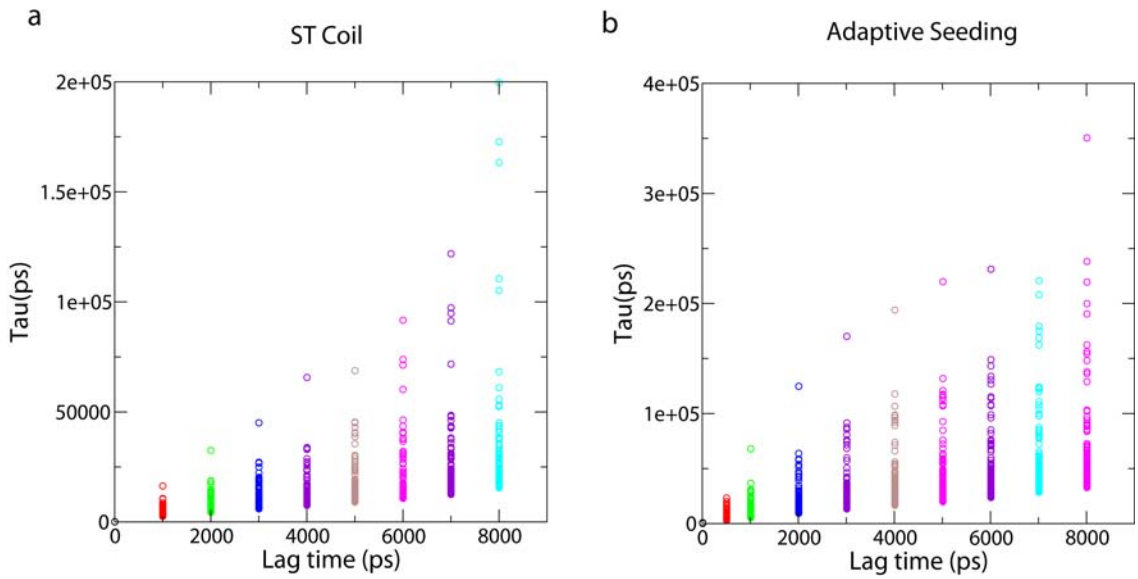


Figure S4. The largest one hundred implied timescales as a function of the lag time for (a) ST simulations starting from the coil initial configuration. (b) The long adaptive seeding microstate MSM.

Lumping is done by first calculating the eigenvalues and eigenvectors of the microstate transition probability matrix[18]. The eigenvalues are related to the timescale for interconverting between two sets of microstates while the corresponding eigenvectors indicate which microstates constitute these two sets if the model is Markovian at this timescale. We estimate the number of macrostates based on the gap in the implied timescales (see Equation (8)) of the microstate transition probability matrix as a function of the lag time. As shown in Figure S4, there are six macro states for the seeding simulations.

Sets of kinetically related microstates are grouped together into macrostates using a spectral clustering algorithm: Perron Cluster Cluster Analysis (PCCA)[19]. While generating the transition count matrix, all the recorded transitions are independent (i.e. transitions from time t to $2t$, $2t$ to $3t$, etc). The initial lumping calculated from this data is refined by using a Simulated Annealing (SA) scheme to maximize the metastability (Q)

of the model[17]. Twenty SA runs of 20,000 steps each are used. In each simulated annealing step, a microstate is randomly reassigned to a new macrostate and the move is accepted using the Metropolis criterion. The metastability is defined as the sum of the self-transition probabilities of each macrostate ($Q = \sum_{i=1}^{N} T_{ii}$). Maximizing the metastability is assumed to be a good way for maximizing the separation of timescales necessary for a valid MSM. The metastability is shown in table S2.

Table S2. Metastability (Q) and average self-transition probability <$P_{i\,i}$> between metastable states for the MSMs built from ST simulations and seeding simulations.

	N. Metastable States	Q	<P_{ii}>
ST (Native)	6	5.09	0.848
ST(Coil)	6	5.01	0.835
Seeding	6	5.61	0.935

5.3 Determining State Populations and Uncertainties

Simulation trajectories are used to estimate transitions between different metastable states in order to build a MSM. Such estimation induces uncertainties in any property computed from the model including the metastable state equilibrium population we pursued in this study. Therefore, obtaining the uncertainties is important to test the reliability of our results. In order to estimate these uncertainties, we employ a Bayesian method introduced by Noe[20]. Assuming that the system is Markovian at the given lag time, the method defines the following stochastic model for its parameters. The likelihood of any trajectory is simply the product of independent transition probabilities, as a consequence of the Markov property, and the transition probability matrix T is assigned an independent, symmetric Dirichlet prior in each row. This is the conjugate prior for the Markov likelihood, which means that the posterior distribution of T after observing a number of transitions has the same functional form as the prior. This method makes the further assumption that the system obeys detailed balance, so the distributions

of T are restrained to the space of reversible stochastic matrices. This distribution is difficult to normalize analytically, but it may be sampled using a Markov Chain Monte Carlo (MCMC) algorithm. It was shown[20] that the restriction to reversible matrices greatly reduces the uncertainty of many thermodynamic properties, which is why it was deemed necessary in our study. Using this method, we were able to sample from the posterior distribution of T, given our simulation data, to obtain stable Monte Carlo estimates of the deviations of equilibrium populations.

A simple model of non-Arrhenius, metastable dynamics

1. Simple potential

GE algorithms attempt to overcome the sampling problem by inducing a random walk in temperature space, where high temperatures help systems cross energetic barriers. However, it has been shown that GE simulations will provide little improvement when the folding kinetics are non-Arrhenius, and the dominant barriers are entropic at high temperatures. In order to demonstrate the efficiency of the ASM in comparison with the GE algorithms, we introduce a model 2D potential to fully contrast the convergence of equilibrium statistics from the different algorithms. The model is based on a discrete-state system introduced by Zwanzig[21] as a simple model for protein folding, which is similar in sprit to continuous-space models used to study anti-Arrhenius dynamics by the Levy group[22]. These models define an energy surface reminiscent of a golf-course, which is almost everywhere flat with some bias toward the folded state and has a sharp decline near the folded state. On the other hand, the degeneracy of the microstates increases sharply as we move away from the folded conformation, providing an entropic advantage that stabilizes the unfolded macrostate at higher temperatures.

The system of Zwanzig[21] was modified by introducing an additional, uncoupled degree of freedom, which has the effect of creating intermediate states between the folded and unfolded states. The energy as a function of the two independent parameters S and R is

$$E_{S,R} = SU + RU - \varepsilon\delta_{S0} - \varepsilon\delta_{R0} + (2-\rho)\varepsilon\delta_{R0}\delta_{S0} \qquad (9)$$

where $S \in \{0,....,N_s\}$ and $R \in \{0,....,N_R\}$. The constant U determines the slope of the energy function as we move away from the folded state along each coordinate; ε represents the drop in energy when one of the coordinates becomes 0, while $\rho\varepsilon$ is the depth of the energy well of the completely folded state, where both S and R equal 0. The degeneracy of each microstate is given by:

$$g_{S,R} = \nu^S \binom{N_S}{S} \nu^R \binom{N_R}{S} \qquad (10)$$

With all this information, it is straightforward to analytically derive the partition function

$$Q = \sum_{S=0}^{N_S}\sum_{R=0}^{N_R} e^{-\beta E(S,R)} g_{S,R} \\ = (e^{\beta\varepsilon} + (1+e^{-\beta U})^{N_S} - 1)(e^{\beta\varepsilon} + (1+e^{-\beta U})^{N_R} - 1) - e^{\beta 2\varepsilon} + e^{\beta\rho\varepsilon} \tag{11}$$

The equilibrium probability of each of the $(N_R+1)(N_S+1)$microstates is now easy to compute by

$$P(S,R) = \frac{e^{-\beta E(S,R)}}{Q} \tag{12}$$

In the current study, we select parameters ν=4, ε=100, ρ=1.5, U=1, and $N_R = N_S = 7$ for our purpose of mimicking the non-Arrhenius folding kinetics. The Potential of Mean Force (PMF) ($G = -\ln P(S,R)$) at a range of temperatures are displayed in Fig. S5. PMF plots suggest 4 metastable macrostates, shown in Fig. S5 as separated by black dashed lines (the state decomposition will be discussed in the next paragraph). The folded state where $S = R = 0$ *(state 1)*, the unfolded state where $S>0$ and $R>0$ *(state 4)*, and two

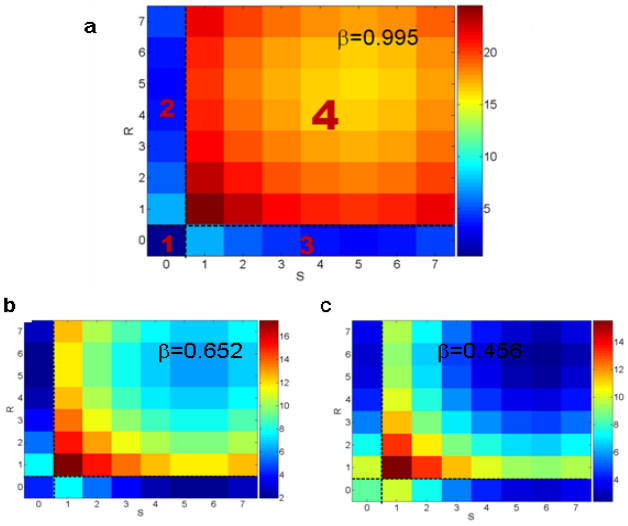


Figure S5. Potential of Mean Force (PMF) for the simple potential at β (1/KT) a. 0.995, b. 0.652, and c. 0.456. In part a, four metastable macrostates are separated by the dashed black lines and labeled.

intermediate states where either $S = 0$ *(State 2)* or $R = 0$ *(State 3)*.

As expected, the free energy of the folded state decreases as we increase the temperature, while the opposite is true of the unfolded state. This is also shown in Fig. S6 where the equilibrium populations of four macrostates are plotted as a function of $\beta=1/kT$. The populations of intermediate states 2 and 3 have low populations at both low and high temperatures, but reach the maximum values at medium temperatures with $\beta \approx 0.65$.

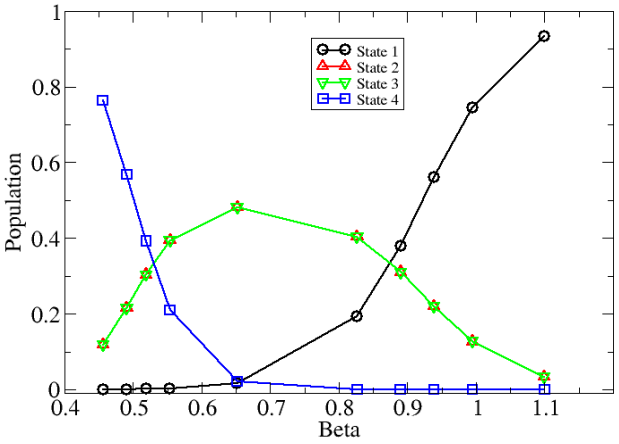


Figure S6. Populations of four macrostates as function of $\beta=1/kT$.

The potential was equipped with a discrete-time, Metropolis Hastings Monte Carlo dynamics, where the proposal probabilities are proportional to the state degeneracy for states where at least one of S and R change by 1, and zero for all others. A Markovian transition probability matrix T_β was computed at each temperatures, from which we obtained evidence for non-Arrhenius behavior and metastability. The non-Arrhenius behavior can be seen in Fig. S7 where we plot the folding and unfolding rates at a function of temperature, computed as the inverse of the mean first passage times between the folded and unfolded states. The mean first passage times are computed using the method described by Singhal *et.al.*[23]. The unfolding rate increases with temperature. However, the folding rate decreases with temperature due to the high entropic barriers for refolding at high temperatures. Metastability for this system is confirmed by the large gap between the third and fourth timescales implied by T_β as shown in Fig. S8. At all temperatures, the third largest timescale is at least a factor of 5

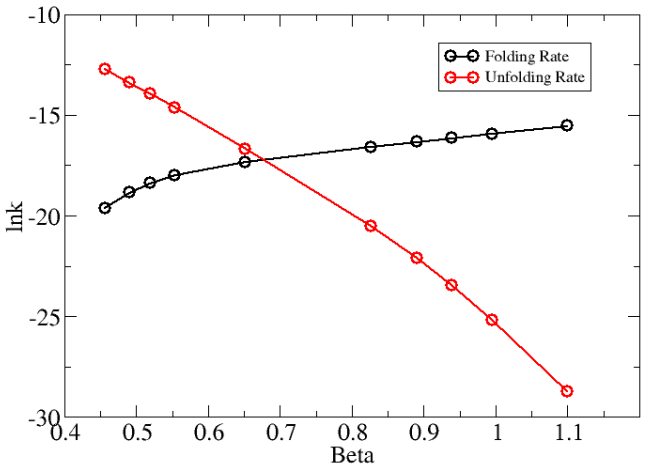


Figure S7. Folding (black) and unfolding (red) rates are plotted as a function of β=1/kT.

greater than the fourth implied timescale. Therefore, we confirm that there is a separation of timescales for this system, and it has four metastable macrostates. The first 3 implied timescales correspond to the transitions between macrostates, while other shorter implied timescales correspond to transitions within macrostates. State decomposition can be obtained by spectra clustering algorithm Perron Cluster Cluster Analysis (PCCA)[19]. and the resulting definition of the four metastable states is shown in Fig. S5 (a).

2. Comparing efficiency of ASM and GE using the simple potential.

To test our hypothesis that GE algorithms, in particular Simulated Tempering (ST), would exhibit a slower rate of convergence for equilibrium statistics than ASM, we simulated 1000 trajectories of 6×10^6 steps using each method. An optimal list of 10 temperatures with β = 1.1, 0.995, 0.939, 0.89, 0.827, 0.652, 0.554, 0.519, 0.491, and 0.456 are selected for ST to obtain acceptance ratios bigger than 40% between all neighboring temperatures. The weights (g_i) are chosen analytically from the partition function[5] to enable the system to uniformly sample every temperature.

$$g_i = -\ln Q(\beta) \tag{13}$$

An equal number of trajectories were started from each temperature, with temperature change proposals done every 10 steps of simulation. Two independent sets of ST

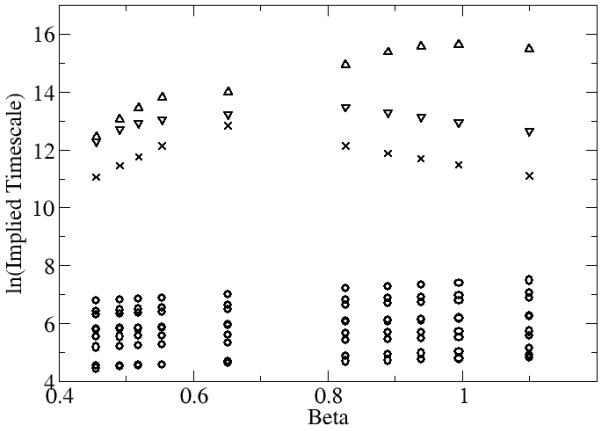


Fig. S8. Logarithms of the implied timescales as function of β for the 2D potential are displayed. The three slowest timescales are plotted using up triangle, down triangle, and cross points respectively.

simulations are performed with initial state 0 and 4 respectively.

For ASM, we simulated 250 trajectories from each of the 4 macrostates at a constant temperature of $\beta = 0.995$, at which the folded state is the dominant state in order to mimic the situation at physiological temperatures.

The convergence of the equilibrium populations from ST was analyzed in the following way. For a set of trajectories, we take a window of 50,000 steps, and compute the fraction of the configurations at a certain metastable state and temperature $\beta = 0.995$ within this window. By bootstrapping this estimator 100 times, we can determine distribution of the state populations as a function of simulation time (See Fig. S9). Populations obtained from the two independent sets of ST simulations are converged between 2.5×10^5 and 3×10^5 steps.

Similarly for ASM, we obtain a distribution for the equilibrium populations with different trajectory length for a certain number of trajectories, which is computed by a Bayesian method.[20] As shown in Fig. S10, it only takes about 4×10^4 steps for ASM to converge to the correct populations, which is much more efficient than ST. The populations in Fig. S10 are computed using a lag time of 1/3 of the trajectory length.

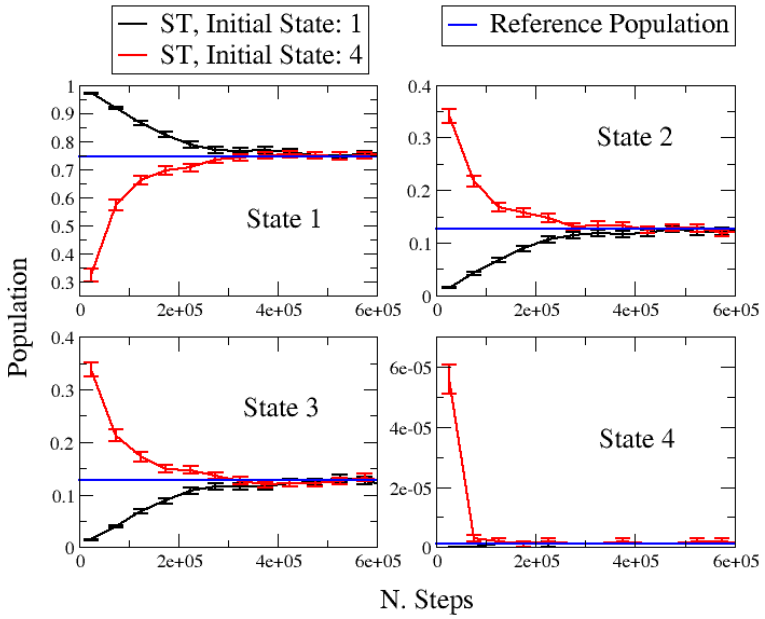


Figure S9. Populations computed from Simulated Tempering (ST) simulations for four metastable states of are plotted as a function of length of the simulation. The reference populations are shown in the solid lines and 1000 trajectories are used for this calculation. The error bars are the standard derivation obtained from bootstrapping 100 times with replacement.

However, we show in Fig. S11 that the populations are almost invariant to the lag time if it is longer than about 1/8 of the trajectory. We note that one has to choose a proper lag time in order to get good estimate of the populations. A good lag time has to be small enough so that there are enough transition counts, but not too small to have many correlated transition counts. In our RNA hairpin example, we use a small lag time but only a few transition counts are taken from each trajectory to make sure we only consider independent transition events. In that case, we can still estimate thermodynamic properties accurately even though the model is not Markovian under the lag time used.

To compare the efficiency of ASM and ST as a function of length and number of trajectories, we define a criterion for the convergence as following: the probability that the estimated populations for all states are within 5% of the actual equilibrium populations is bigger than 80%. The population distributions are computed the same way as in Fig S9 for ST and in Fig. S10 for ASM. As shown in Fig. S12, ASM is much more efficient than ST, and can reach the convergence using 4-7 times shorter simulations than ST. In addition, the efficiency of ST will not increase with the number of trajectories after 200, while the efficiency of ASM keeps increasing with number of

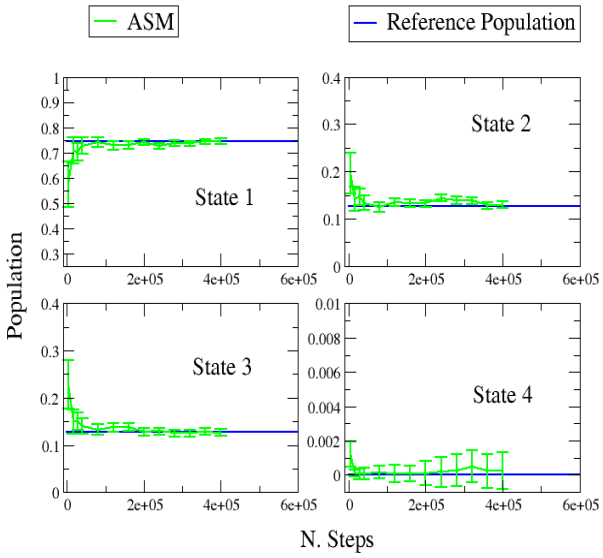


Figure S10. Populations computed from Adaptive Seeding Method (ASM) for four metastable states of are plotted as a function of length of the simulation. The reference populations are shown in the solid lines and 1000 trajectories are used for this calculation. The lag time is selected as 1/3 of the length of the simulation. The error bars are standard derivation obtained from a Bayesian method (See section 2.5.3 for details)(cite).

trajectories up to 600. We think ideally the length of the seeding simulations should lie in the major gap of the implied timescales, such that they are longer than the slowest intra-macrostate equilibration time to minimize the model error due to non-Markovian effects. In the current system, the minimum length of the simulations ($\sim 5\times10^4$) is indeed between 3rd (1.61×10^3) and 4th (9.58×10^4) slowest implied timescales. There is evidence from the RNA hairpin example and previous work on a water dewetting

transition in a carbon nanotube[24] that these requirements for the lag time may be relaxed for real systems, where the separation of timescales is less evident than in the model system studied here. Additionally, the number of seeding simulations has to be big enough to reduce the statistical error to a satisfactory level.

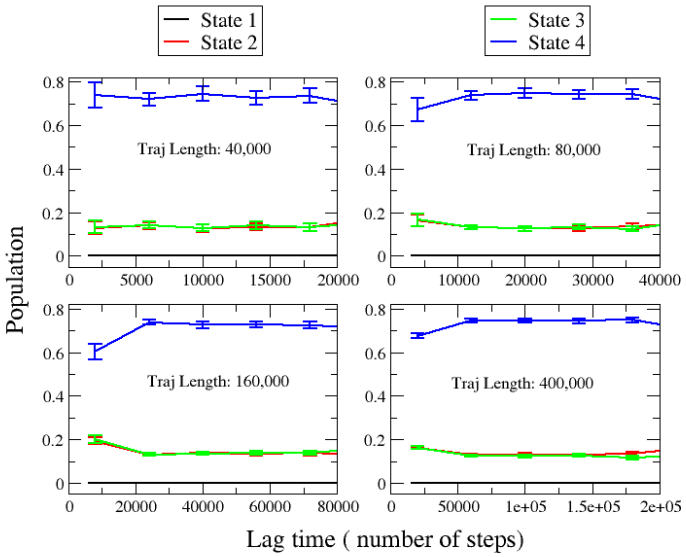


Figure S11. Populations computed from ASM simulations for four metastable states as a function of lag time.

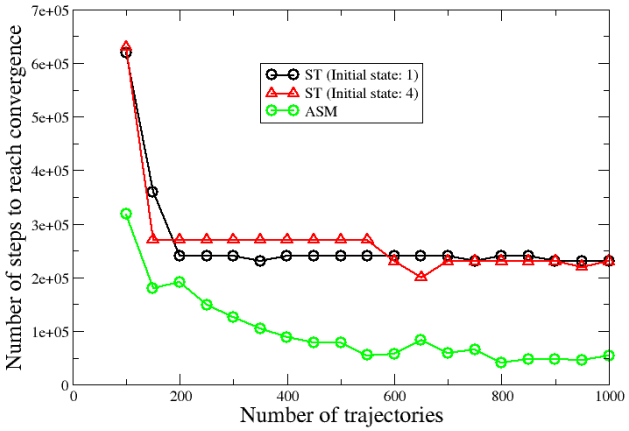


Figure S12. Number of steps taken to reach the convergence as a function of number of trajectories.

References

1. Jucker, F.M., et al., *A network of heterogeneous hydrogen bonds in GNRA tetraloops.* J Mol Biol, 1996. **264**(5): p. 968-80.
2. Macke, T. and D.A. Case, *Modeling unusual nucleic acid structures*. Molecular Modeling of Nucleic Acids, ed. J. N.B. Leontes and J. SantaLucia. 1998, Washington, DC: American Chemical Society.
3. Lyubartsev, A.P., et al., *New approach to Monte Carlo calculation of the free energy: Method of expanded ensembles.* The Journal of Chemical Physics, 1992. **96**(3): p. 1776-1783.
4. Marinari, E. and G. Parisi, *Simulated Tempering: a New Monte Carlo Scheme.* Europhysics Letters, 1992. **19**: p. 451-458.
5. Huang, X., G.R. Bowman, and V.S. Pande, *Convergence of folding free energy landscapes via application of enhanced sampling methods in a distributed computing environment.* J. Chem. Phys, 2008. **128**(20): p. 205106.
6. Bowman, G.R. and V.S. Pande, *Simulated tempering yields insight into the low-resolution Rosetta scoring functions.* Proteins, 2009. **74**(3): p. 777-88.
7. Lindahl, E., B. Hess, and D. van der Spoel., *GROMACS 3.0: a package for molecular simulation and trajectory analysis.* J. Mol.Modeling., 2001. **7**: p. 306-317.
8. Shirts, M. and V.S. Pande, *COMPUTING: Screen Savers of the World Unite!* Science, 2000. **290**(5498): p. 1903-1904.
9. DUAN, Y., et al., *A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations.* J. Comp. Chem. , 2003. **24**: p. 1999-2012.
10. Junmei Wang, P.C.P.A.K., *How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules?* 2000. p. 1049-1074.
11. Jorgensen, W.L.C., J.; Madura, J. D.; Impey, R. W.; Klein, M. L., J. Chem. Phys, 1983. **79**(926-935).
12. Hoover, W., Phys. Rev. A, 1985. **31**: p. 1695-1697.
13. Darden, T., D. York, and L. Pedersen., *A smooth particle mesh Ewald potential.* J. Chem. Phys., 1995. **103**: p. 3014-3021.
14. Hess, B., H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije., *LINCS: a linear constraint solver for molecular simulations.* J. Comput. Chem., 1997. **18**: p. 1463-1472.
15. Boxer, G., et al., *Hieratical K-medoids clustering algorithm.* Technique Report, 2009. **In prepration**.
16. Noe, F. and S. Fischer, *Transition networks for modeling the kinetics of conformational change in macromolecules.* Current Opinion in Structural Biology, 2008. **18**(2): p. 154-162.
17. Chodera, J.D., et al., *Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics.* J Chem Phys, 2007. **126**(15): p. 155101.

18. Deuflhard, P. and M. Weber, *Robust Perron cluster analysis in conformation dynamics.* Linear Algebra and Its Applications, 2005. **398**: p. 161-184.
19. Deuflhard, P., et al., *Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains.* Lin. Alg. Appl., 2000. **315**: p. 39–59.
20. Noe, F., *Probability distributions of molecular observables computed from Markov models.* J Chem Phys, 2008. **128**(24): p. 244103.
21. Zwanzig, R., *Simple model of protein folding kinetics.* Proc Natl Acad Sci U S A, 1995. **92**(21): p. 9801-4.
22. Zheng, W., et al., *Simple continuous and discrete models for simulating replica exchange simulations of protein folding.* J Phys Chem B, 2008. **112**(19): p. 6083-93.
23. Singhal, N., C.D. Snow, and V.S. Pande, *Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin.* J Chem Phys, 2004. **121**(1): p. 415-25.
24. Sriraman, S., I.G. Kevrekidis, and G. Hummer, *Coarse master equation from Bayesian analysis of replica molecular dynamics simulations.* J Phys Chem B, 2005. **109**(14): p. 6479-84.