Simulated tempering distributed replica sampling: 
A practical guide to enhanced conformational sampling

Sarah Rauscher1,2 and Régis Pomès1,2

1 Molecular Structure and Function, Hospital for Sick Children, 555 University Avenue, Toronto, ON, Canada
2 Department of Biochemistry, 1 King’s College Circle, University of Toronto, Toronto, ON, Canada

Email: pomes@sickkids.ca

Abstract. Simulated tempering distributed replica sampling (STDR) is a generalized-ensemble method designed specifically for simulations of large molecular systems on shared and heterogeneous computing platforms [Rauscher, Neale and Pomès (2009) J. Chem. Theor. Comput. 5, 2640]. The STDR algorithm consists of an alternation of two steps: (1) a short molecular dynamics (MD) simulation; and (2) a stochastic temperature jump. Repeating these steps thousands of times results in a random walk in temperature, which allows the system to overcome energetic barriers, thereby enhancing conformational sampling. The aim of the present paper is to provide a practical guide to applying STDR to complex biomolecular systems. We discuss the details of our STDR implementation, which is a highly-parallel algorithm designed to maximize computational efficiency while simultaneously minimizing network communication and data storage requirements. Using a 35-residue disordered peptide in explicit water as a test system, we characterize the efficiency of the STDR algorithm with respect to both diffusion in temperature space and statistical convergence of structural properties. Importantly, we show that STDR provides a dramatic enhancement of conformational sampling compared to a canonical MD simulation.

1. Introduction
Biomolecular systems are challenging to study using molecular dynamics (MD) simulations because of their rugged energy landscapes [1]. The energetic barriers that separate conformational states are often larger than the available thermal energy, and therefore transitions between conformations are statistically rare events. As a consequence, achieving complete Boltzmann sampling of all important conformational states often requires prohibitively long simulation times. Without reaching statistical convergence, it is not possible to draw meaningful, quantitative conclusions regarding the relative populations of different conformational states [2].

Using state-of-the-art supercomputing, atomistic MD simulations with an explicit representation of solvent are currently limited to the nanosecond to microsecond timescale. Therefore, the timescales of many important biomolecular processes are still inaccessible to conventional MD. For example, β-hairpin folding occurs on the microsecond timescale [1], while the folding of two-state proteins occurs over a range of timescales spanning six orders of magnitude, from microseconds to seconds [3]. Sampling the conformational space of folded proteins is computationally demanding. Yet it is perhaps even more challenging to adequately sample the complex conformational landscape of disordered proteins, which lack a well-defined folded state. Molecular simulations of disordered proteins are
complicated by the fact that not one but possibly many thousands of states must be sampled in order to characterize a structurally heterogeneous ensemble of conformations [4].

It has recently become clear that the disordered states of proteins have significant biological relevance. While the majority of proteins are folded in their native state, a growing body of evidence from spectroscopy and bioinformatics indicates that a significant fraction of proteins are natively disordered [5]. Disordered proteins are abundant in all kingdoms of life: more than one-third of mammalian proteins [6] and more than 75% of cancer-associated proteins are predicted to contain disordered regions [5]. Despite their biological importance, disordered proteins are poorly understood: they are notoriously difficult to study using experimental approaches. Molecular simulations therefore have the potential to offer tremendous insight into the structural and physicochemical properties of disordered proteins [4]. However, conventional “brute force” MD simulations are insufficient for investigations of these proteins because they often remain trapped in a single conformational basin [7]. In order to provide meaningful insight into the structural properties of disordered proteins, molecular simulation approaches that enhance conformational sampling are required.

Error due to insufficient sampling can be alleviated by utilizing generalized-ensemble algorithms. These methods use a generalized Hamiltonian for the purpose of achieving uniform sampling along a reaction coordinate of interest. Generalized-ensemble algorithms in which replicas of the system execute a random walk in temperature rely on the fact that the free energy surface becomes less rugged at high temperature, resulting in an increased frequency of interconversion between conformational states [8]. Two commonly-used generalized-ensemble algorithms that utilize a random walk in temperature are simulated tempering (ST) [9, 10] and replica exchange (RE) [11-14]. An RE simulation consists of identical copies of the system (replicas) sampling canonical ensembles at different temperatures [12-14]. Exchanges are performed between replicas at neighboring temperatures according to a Metropolis criterion. Each temperature exchange requires synchronization, which means that RE requires a large, dedicated, and homogeneous computing cluster to function efficiently when applied to complex systems. While ST has no such requirement for synchronization, it does require extensive initial simulations to accurately compute the Helmholtz free energy of the system at each temperature [8, 15]. We have shown previously that ST is the preferred generalized-ensemble algorithm for simple systems [7]. This is because ST exhibits the fastest temperature diffusion, which is correlated with faster convergence of structural properties. However, the computational cost of the initial simulations becomes prohibitive for all but the simplest biomolecular systems [7]. In this paper, we present our implementation of simulated tempering distributed replica sampling (STDR), which is a generalized-ensemble algorithm that combines ST with distributed replica sampling (DR) [7, 16]. STDR offers numerous practical advantages compared to both ST and RE that become important for simulations of complex biomolecular systems [7]. Unlike ST, STDR requires minimal initial simulation; unlike RE, STDR does not require synchronization of replicas or a fixed number of replicas. It accommodates fluctuations in CPU availability and requires minimal communication between replicas. STDR is therefore well-suited for shared computing platforms [7, 16].

This paper provides a guide to applying STDR to complex biomolecular systems and demonstrates the efficiency of the method using a challenging model system. We begin by briefly discussing the details of the algorithm, which has previously been derived [7, 16]. Next, we provide a description of our STDR implementation. We quantify the computational overhead for performing STDR compared to conventional MD simulations and we determine the amount of processor time required for each step of the algorithm. For a specific test case (a disordered peptide in explicit water), we evaluate the random walk in temperature space, the convergence of structural properties, and the enhancement of conformational sampling compared to a canonical MD simulation.

2. The STDR Algorithm
In contrast to RE, in which pairwise exchanges of replicas are attempted, STDR considers stochastic moves of individual replicas one at a time [7, 16]. The stochastic move of one replica is coupled to the
distribution of all other replicas through a pseudo-energy, called the distributed replica potential energy (DRPE) [16]. In STDR, replicas perform temperature jumps based on the current temperature location of all other replicas. The probability of accepting a move from a temperature $T_i$ to a neighbouring temperature, $T_j$ is:

$$p(T_i \rightarrow T_j) = \min \left\{ 1, e^{-(\beta_j - \beta_i)E + (a_j - a_i)(DRPE_j - DRPE_i)} \right\}$$  \hspace{1cm} (1)

where $E$ is the potential energy of the system at the end of the previous MD step, $\beta_i$ and $\beta_j$ are the inverse temperatures, and $a_i$ and $a_j$ are dimensionless Helmholtz free energies corresponding to temperatures $T_i$ and $T_j$, respectively. The first two terms in the exponent are identical to the exchange probability from ST, with the addition of the difference in DRPE between the states for which the replica is at temperature $T_i$ ($DRPE_i$) and temperature $T_j$ ($DRPE_j$) [7]. The DRPE depends upon the current temperatures of all replicas as follows [16, 17]:

$$DRPE = c_1 \sum_{m=1}^{M} \sum_{n=1}^{M} \left[ (\lambda_{m,\text{linear}} - \lambda_{n,\text{linear}}) - (m - n) \right] + c_2 \left[ \sum_{m=1}^{M} \lambda_{m,\text{linear}} - \sum_{m=1}^{M} m \right]^2$$  \hspace{1cm} (2)

Replicas are labeled by indices $m$ and $n$, where $M$ is the number of replicas. The values of $\lambda_{m,\text{linear}}$ refer to a linearly-spaced temperature coordinate. In this coordinate, the lowest temperature has $\lambda_{m,\text{linear}}=1$, and the highest temperature has $\lambda_{m,\text{linear}}$ equal to the number of temperatures. This procedure transforms exponentially-spaced temperatures into a uniformly-spaced coordinate. In STDR, the DRPE enforces approximately homogeneous sampling of a set of temperatures. The effect of the first term is to impose a pseudo-energetic penalty for two replicas sampling the same temperature simultaneously, while the second term imposes a penalty for a concerted drift of replicas towards high or low temperature. The two constants in the equation, $c_1$ and $c_2$, control the magnitude of the DRPE and the relative weight of the two terms [16, 17].

3. Implementation Details

3.1. Test System: A Disordered Elastin-like Peptide in Explicit Water

Because one of the main goals of this study is to develop and test a methodology that will provide structural insight into disordered states of proteins, we use a disordered peptide as our test system. Specifically, we use a 35-residue peptide with a sequence derived from elastin, (GVPGV)$_7$ [18]. Elastin is an intrinsically disordered protein that is responsible for the elasticity of the lungs, arteries, and skin. Its structural disorder is essential to its biological role: the increased entropy of the relaxed state relative to the stretched state makes elastin an entropic elastomer [19]. This elastin-like peptide has a complex conformational landscape, which makes it ideally suited to evaluate STDR’s ability to enhance conformational sampling. This paper does not focus on the detailed structural properties of this peptide, which will form the basis of future studies.

3.2. Computational Platform

The general purpose cluster (GPC) at the University of Toronto’s SciNet consortium was used to perform large-scale STDR simulations of the (GVPGV)$_7$ peptide in water. Each replica was run on an individual node of the GPC. Each Nehalam node has eight 2.5 GHz CPUs and 16 GB of memory. Because the STDR algorithm requires only minimal communication between nodes, the portion of the GPC connected with gigabit Ethernet was utilized. In total, 106 nodes were used for the simulations described here: 105 nodes were used to simulate individual replicas, with one additional node designated as a server. In fact, the number of nodes running concurrently occasionally is slightly less than 106 due to rare node failure and fluctuations in node availability (since the SciNet GPC is a shared computing resource). This is not problematic, however, since STDR is well-suited for such a
3.3. MD Simulation Details

The simulation system consists of the (GVPGV)$_7$ peptide capped with an acetyl group at the N-terminus and an NH$_2$ group at the C-terminus in a rhombic dodecahedral box with 9663 water molecules. Simulations were performed with an accurate leap-frog stochastic dynamics integrator [20] using GROMACS version 4.0.5 [21]. The OPLS-AA/L force field [22] and the TIP3P model [23] were used for the peptide and water, respectively. Periodic boundary conditions were applied. Calculation of electrostatic forces utilized the particle mesh Ewald summation method [24] with a Fourier spacing of 1.2 Å and a fourth-order interpolation. The real-space Coulombic cutoff was 11 Å. Lennard Jones interactions were cut off at 14 Å. The LINCS algorithm [25] was used to constrain covalent bonds involving hydrogen atoms, and the SETTLE algorithm [26] was used for water, allowing the use of a 2 fs integration time step. The solvated system was initially equilibrated in the isothermal-isobaric ensemble at 300 K and 1 atm for 0.5 ns using the Parrinello-Rahman barostat [27, 28]. All STDR simulations were performed in the canonical ensemble using stochastic dynamics as a thermostat. In addition, a long-time canonical MD simulation was performed for 700 ns, starting with a randomly-selected configuration from the STDR simulation at 300 K.

For the STDR simulation, 105 temperatures exponentially-spaced between 266 K and 749 K were used. At each of these temperatures, a preliminary canonical MD simulation was performed for 15 ns. The lists of potential energies obtained from these simulations were used to compute the dimensionless Helmholtz free energies in equation (1) using the method of Park and Pande [15]. A set of 105 random starting configurations (one for each replica) was obtained from ten 10 ns simulations at 1000 K. At the beginning of the STDR simulation, each replica was started at one of the 105 temperatures. The DRPE constants $c_1$ and $c_2$ were set to 0.004 and 0.002, respectively. Each replica performed $\sim 10^5$ 8 ps MD runs and attempted temperature jumps according to equations (1) and (2), which corresponds to a total of $\sim 800$ ns/replica. In total, this resulted in an accumulated simulation time of 84 $\mu$s for the entire STDR simulation. This is an order of magnitude more simulation than our previous study [7]. The implementation of STDR makes use of several GROMACS programs including g_energy, trjconv, trjcat, eneconv, and trjorder. These programs are used for analysis and data management, as described in detail in section 3.4.

3.4. Implementation of the STDR Algorithm

3.4.1 Parallelization

Our STDR implementation incorporates two levels of parallelization:

1. First, the STDR algorithm is inherently highly parallel: the system is simulated as a set of replicas, each occupying a single node. Each replica executes alternating MD simulations and stochastic temperature jumps. Minimal information sharing between replicas is required: only the current temperature location of all replicas is required to compute the DRPE in equation 2. In the present implementation, a server node collects the temperature information and trajectories of all replicas.

2. Parallelization also occurs at the level of the individual replica node. Each 8 ps MD simulation is run in parallel using all 8 cores of the node. The time required for 1 ns of simulation is 17.98 hours and 2.12 hours for one core and eight cores, respectively. Thus, the parallel efficiency is 106%. Superlinear scaling is achieved using GROMACS version 4.0, which has been highly optimized to scale well on parallel machines [21].
3.4.2 The replica nodes

The specific steps executed by a single STDR replica are illustrated in the flow chart in figure 1. The cycle of steps in a single temperature move begins with a short MD simulation (8 ps in length). The final potential energy from the last step of this MD simulation is obtained using the GROMACS program g_energy. This potential energy value, along with an up-to-date list of all replicas’ temperatures (received from the server node), is used to compute the probability of accepting a proposed temperature jump (equations 1 and 2). Information about the temperature jump is stored, including the potential energy, DRPE, probability, and temperature. In order to begin the next MD step, grompp generates a run input file. To minimize the amount of hard disk space required for storage, the water molecules in the MD trajectory are ordered by proximity to the protein using trjorder: only the nearest water molecules are saved in the final trajectory using trjconv. The replica’s trajectories are concatenated locally (on the node on which the replica is running). Trajectories corresponding to the same temperature must also be concatenated; this occurs on the server node. The trajectory corresponding to each STDR step must therefore be sent to the server node over the network. The replica periodically backs up the data stored in memory to disk once every 4 hours. This checkpointing limits the amount of data lost in case of a node failure to a maximum of a few hours, which corresponds to a few hundred STDR steps. This approach was designed to minimize disk I/O, minimize the amount of long-term data storage required for trajectories, and maximize the fraction of compute time spent on MD (see figure 2). Because GPFS, the high performance file system of the SciNet GPC, is not designed for accessing many small files, it is important that all trajectory files be concatenated while still in memory. Data locality is maximized by passing the smallest possible amount of data between the server node and the replica nodes. All of the processes on the nodes are

Figure 1. Implementation of the STDR Algorithm. The cycle of steps involved in STDR is illustrated as a flow chart. Each cycle consists of two parts: (1) a brief MD simulation (8 ps in length) and (2) an attempted temperature jump. Associated with this temperature jump are several essential data processing steps. During the cycle, all data is stored in shared memory and no disk access is required. The cycle is repeated thousands of times, with occasional interruptions to back up essential data to disk (every few hours) and concatenate trajectories (every two days).
controlled by bash scripts that call GROMACS programs, as well as an in-house program that computes the STDR temperature jump criterion (equations 1 and 2). In figure 2, we assess the efficiency of the STDR implementation for the (GVPGV)$_7$ test system by determining the amount of processor time required for each step of the algorithm. The majority of the STDR overhead results from managing the trajectory files (trjorder, trjconv, and concatenation). These steps are not part of the core STDR algorithm. However, the trajectories must be the minimum size possible and concatenated to facilitate structural analysis and long term data storage.

![Figure 2](image.png)

**Figure 2.** Node time required for each step in STDR. (A) The percent of replica node time spent on each step in the STDR algorithm is represented as a pie chart (colours correspond to the same steps illustrated in figure 1). (B) Overall, 96.14% of node time is spent performing MD, while 3.86% is spent on all tasks associated with STDR, including management of trajectory files. Note that the mdrun step in figure 1 is coloured blue to corresponding to the overhead for starting and stopping mdrun indicated in figure 2A.

### 3.4.3 The server node

The server node fulfills two key roles in our implementation of STDR:

1. It stores the current temperature of each replica. The server maintains a current list of the temperatures of all replicas. This temperature list is frequently communicated to each replica
node, ensuring that an up-to-date temperature list is used in the calculation of the DRPE (equation 2). After every temperature jump, the replica communicates its current temperature to the server.

(2) The server node also collects the short MD trajectories of all replicas. Trajectories must be stored according to temperature to facilitate further structural analysis (described in detail in section 5 below). In order to concatenate trajectories by temperature, the trajectories of individual replicas must be collected in a central location (the shared memory of the server node). When the memory is almost full, the trajectories are concatenated by temperature and moved to disk. By performing all trajectory concatenation in memory on the nodes, disk I/O is kept to an absolute minimum.

4. Evaluating Temperature Diffusion

The quality of the random walk in temperature is assessed using several different metrics, which we have defined in our previous work [7]. These metrics are reported in Table 1. Importantly, the sampling of temperatures is very close to perfect homogeneity: the average deviation from homogeneity is only 1.3%. This indicates that the constants \( c_1 \) and \( c_2 \) in equation 2 have been appropriately selected to enforce homogeneous temperature sampling. Almost 30% of temperature jumps are accepted, which indicates good overlap of potential energy distributions of neighbouring temperatures, and therefore adequate temperature spacing. The majority of replicas have visited all temperatures at least once, and on average each replica has visited 94% of temperatures.

| Table 1. Evaluating the diffusion of replicas in temperature space. |
|---------------------------------------------------------------|
| Temperature Diffusion Properties                           |
| average deviation from homogeneity \( d = \frac{1}{M} \sum_{m=1}^{M} \left| \frac{N_m - \langle N_m \rangle}{\langle N_m \rangle} \right| \) | 0.013 |
| acceptance ratio = \frac{\text{number of accepted T jumps}}{\text{total number of attempted jumps}} | 0.292 |
| mean free path = \frac{\text{average distance travelled between rejected temperature jumps}}{\text{average distance travelled in 50 steps/50}} | 0.280 |
| replica speed = \frac{\text{average distance travelled in 50 steps}}{50} | 0.051 |
| diffusion coefficient = \frac{0.5 \times \text{slope in the linear region of the mean square displacement vs. steps plot}}{0.085} |
| average fraction of temperatures visited per replica | 0.94 |

\( N_m \) is the number of samples at temperature \( m \), and \( M \) is the number of temperatures.

In figure 3, the random walk in temperature is shown for one of the replicas. By varying the temperature, the STDR algorithm leads to exploration of conformational states with a wide range of radius of gyration, which is a measure of the spatial extension of the peptide. At low temperature, the peptide samples more compact conformations; increasing temperature allows the peptide to sample more expanded conformations. The random walk in temperature facilitates the peptide’s expansion and collapse, preventing it from becoming trapped in any specific conformational state.
Figure 3. A two-dimensional plot of the radius of gyration of the peptide vs. temperature. The complete STDR simulation of a single replica (750 ns in total) is shown in gray. Superimposed in black is the same trajectory including only every hundredth point with arrows for clarity.

5. Evaluating Convergence of the Structural Ensemble

5.1. Removal of unequilibrated structures
The STDR simulation of each replica began with a random configuration generated at high temperature. Because these configurations are not representative of the true conformational ensembles, an initial equilibration phase occurs during which these random configurations collapse to more realistic, compact structures. In order to delineate this equilibration phase of the STDR simulation, we look at the running average of radius of gyration, $R_g$, as a function of time (figure 4). Note that in this case, time refers to the accumulated simulation time at 300 K, and not to a continuous MD time trajectory. The average and standard deviation of $R_g$ from the second half of the STDR simulation are used for comparison. The point at which the $R_g$ falls within one standard deviation of the average (in this case, 64.9 ns) is taken to be the point at which the equilibration phase is complete. Configurations accumulated during the equilibration phase are discarded, and all other configurations are used for further structural analysis.
Figure 4. Running average of radius of gyration versus time. The radius of gyration is computed for the backbone Cα atoms. The red solid line indicates the average radius of gyration over the second half of the simulation, and the red dotted lines indicate one standard deviation above and below this average. The first point at which the average radius of gyration falls within this range is taken to be the end of the equilibration phase.

Figure 5. Probability distribution of the radius of gyration, $R_g$. Shown here is the $R_g$ distribution computed using incrementally increasing amounts of data from the STDR trajectory at 300 K from 64.9 ns to 763.8 ns (the data from the equilibration phase is not included). Error bars for the $R_g$ distribution computed using the complete trajectory (in red) are the standard error based on block averaging.
5.2 Convergence of the conformational ensemble

The probability distribution of the radius of gyration (figure 5) appears to be statistically converged, since it does not change with additional simulation time in the second half of the trajectory at 300 K (after 414.4 ns). It is always possible that longer simulations will reveal longer timescale phenomenon. Thus, it is not possible to be certain in declaring that a simulation or a structural property is converged [2]. Similarly, we note that it is important to investigate the convergence of several structural properties, since they may reach statistical convergence with differing amounts of conformational sampling [7].

6. Comparison ofCanonical MD and STDR

In order to quantify the sampling enhancement provided by STDR, we compare the conformational ensemble at 300 K with that of an MD simulation in the canonical ensemble at 300 K (figures 6 and 7). First, we monitored the number of intramolecular peptide hydrogen bonds observed with increasing simulation time, shown in figure 6 (a hydrogen bond was counted as observed if it was formed in at least one of the configurations in the trajectory). With the complete STDR trajectory at 300 K, more than 92% of all possible hydrogen bonds were observed in at least one configuration. In contrast, the canonical MD simulation at the same temperature only sampled 11.9% of all possible peptide-peptide hydrogen bonds. We also show the number of hydrogen bonds observed for the replica trajectory depicted in figure 3. This replica has sampled 95% of all possible hydrogen bonds. The STDR simulation has not yet sampled all possible hydrogen bonds, which indicates that unexplored regions of conformational space remain to be sampled. It is also possible that some of the intramolecular hydrogen bonds that have not yet been sampled cannot be made due to structural restraints. Nevertheless, the vast majority of intramolecular peptide hydrogen bonds have been sampled at least once.

![Figure 6](image.png)

**Figure 6.** Number of intramolecular peptide hydrogen bonds (HBs) observed with increasing simulation time for STDR (green and purple) and canonical MD (red). In purple, the number of HBs observed for the replica trajectory corresponding to figure 3 is shown. The number of possible HBs is computed based on the number of possible contacts in the HB contact plot, figure 7A and 7B (35 x 35 = 1225). From this value, we subtract the diagonal and the two off diagonals (35+34+34); these HBs cannot be formed due to conformational restrictions. Contacts that would involve the N-H of proline residues are also subtracted. Thus, there are 898 possible HBs.
Figure 7. Configurations of the disordered (GVPGV)$_7$ peptide from canonical MD and STDR simulations at 300 K. Intramolecular hydrogen-bonding contact maps for the ensembles from canonical MD and STDR are shown in (A) and (B), respectively. The probability of a particular hydrogen bond between backbone C=O and N-H groups is indicated by the colour of the corresponding square. Note that the contact maps have different colour scales; some of the contacts in the canonical MD simulation are highly populated, whereas the STDR ensemble has no contacts populated in more than 12% of structures. In (C) and (D), we show 100 randomly selected configurations from each ensemble and the average root mean-square deviation (RMSD) of backbone atoms compared to the starting configuration. Configurations are shown with a cartoon representation of the backbone using VMD [29] (glycine in orange, valine in purple, and proline in yellow).
The canonical MD simulation was 700 ns in length, which is the same amount of accumulated simulation time at 300 K after removing the initial unequilibrated configurations (refer to section 4). Representative sets of configurations from both the MD simulation and the STDR simulation are shown in figure 7, along with hydrogen-bonding contact maps for these ensembles. The canonical MD simulation exhibits quasi-nonergodicity [30]: even though the system is ergodic, it appears non-ergodic because it samples essentially a single conformational basin throughout the entire simulation (over hundreds of nanoseconds). The configurations from the canonical MD simulation do not differ significantly from each other based on RMSD. In contrast, the configurations from STDR represent a small subset of the thousands of configurations generated by the simulations, and illustrate the heterogeneous set of structures that this peptide can adopt. By comparing the hydrogen-bonding contact maps obtained using canonical MD and STDR (figure 7C and 7D), it is evident that canonical MD severely underestimates the structural heterogeneity of this disordered peptide, while STDR provides a dramatic enhancement in conformational sampling.

7. Conclusions
We have presented our implementation of STDR on a large-scale computing cluster, demonstrating the computational efficiency of the approach with a detailed account of the overhead of each step in the algorithm. An atomic-level description of the disordered conformational ensemble of an elastin-like peptide was obtained. Our results push the boundaries of modern MD simulations with nearly 0.1 ms of sampling. By comparing the conformational ensembles obtained using STDR and canonical MD, the sampling enhancement provided by the random walk in temperature is clearly demonstrated. The detailed structural properties of the ensemble of conformations for this peptide will be the subject of future publications. Significantly, this study establishes a methodological framework for future applications of STDR to other disordered states of proteins.

Acknowledgements
We are grateful to Dr. Tom Rodinger, Chris Neale, and Grace Li for discussions on distributed replica sampling, and to Chris Madill for administration of our in-house data storage servers. We also thank the SciNet support team for helpful advice in tailoring our implementation of STDR to the SciNet GPC architecture. Computations were performed on the GPC supercomputer at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto. The Canadian Institutes of Health Research (Grant No. MOP84496) is gratefully acknowledged for support. S.R. is funded by a Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council (NSERC) and the Research Training Center at the Hospital for Sick Children. R.P. is a CRCP chairholder.

References
[1] S. Gnanakaran et al., Curr. Opin. Struct. Biol. 13, 168 (2003).
[2] E. Lyman, and D. M. Zuckerman, Biophys. J. 91, 164 (2006).
[3] B. Gillespie, and K. W. Plaxco, Annu. Rev. Biochem. 73, 837 (2004).
[4] S. Rauscher, and R. Pomès, Biochemistry and Cell Biology 88, 269 (2010).
[5] A. K. Dunker et al., BMC Genomics 9, 25 (2008).
[6] J. J. Ward et al., J. Mol. Biol. 337, 635 (2004).
[7] S. Rauscher, C. Neale, and R. Pomès, J. Chem. Theory Comput. 5, 2640 (2009).
[8] X. Huang, G. R. Bowman, and V. S. Pande, J. Chem. Phys. 128, 205106 (2008).
[9] E. Marinari, and G. Parisi, Europhys. Lett. 19, 451 (1992).
[10] A. P. Lyubartsev et al., J. Chem. Phys. 96, 1776 (1992).
[11] A. Mitsutake, Y. Sugita, and Y. Okamoto, Biopolymers 60, 96 (2001).
[12] Y. Sugita, and Y. Okamoto, Chem. Phys. Lett. 314, 141 (1999).
[13] M. C. Tesi et al., J. Stat. Phys. 82, 155 (1996).
[14] A. M. Ferrenberg, and R. H. Swendsen, Phys. Rev. Lett. 61, 2635 (1988).
[15] S. Park, and V. S. Pande, Phys. Rev. E 76, 5 (2007).
[16] T. Rodinger, P. L. Howell, and R. Pomès, J. Chem. Theory Comput. 2, 725 (2006).
[17] T. Rodinger, P. L. Howell, and R. Pomès, J. Chem. Phys. 129, 155102 (2008).
[18] S. Rauscher et al., Structure 14, 1667 (2006).
[19] L. D. Muiznieks, A. S. Weiss, and F. W. Keeley, Biochem. Cell Biol. 88, 239 (2010).
[20] W. F. van Gunsteren, and H. J. C. Berendsen, Molecular Simulation 1, 173 (1988).
[21] B. Hess et al., J. Chem. Theory Comput. 4, 435 (2008).
[22] G. A. Kaminski et al., J. Phys. Chem. B 105, 6474 (2001).
[23] W. L. Jorgensen et al., J. Chem. Phys. 79, 926 (1983).
[24] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. 98, 10089 (1993).
[25] B. Hess et al., J. Comput. Chem. 18, 1463 (1997).
[26] S. Miyamoto, and P. A. Kollman, J. Comput. Chem. 13, 952 (1992).
[27] M. Parrinello, and A. Rahman, J. Appl. Phys. 52, 7182 (1981).
[28] S. Nose, and M. L. Klein, Mol. Phys. 50, 1055 (1983).
[29] W. Humphrey, A. Dalke, and K. Schulten, J. Mol. Graph. 14, 33 (1996).
[30] C. Chipot, and A. Pohorille, Free Energy Calculations: Theory and Applications in Chemistry and Biology (Springer, Berlin, 2007).