Statistica Sinica **34** (2024), 1263-1280
doi:https://doi.org/10.5705/ss.202022.0340

ESTIMATING BOLTZMANN AVERAGES FOR PROTEIN STRUCTURAL QUANTITIES USING SEQUENTIAL MONTE CARLO

Zhaoran Hou and Samuel W.K. Wong*

University of Waterloo

Abstract: Sequential Monte Carlo (SMC) methods are widely used to draw samples from intractable target distributions. Weight degeneracy can hinder the use of SMC when the target distribution is highly constrained. As a motivating application, we consider the problem of sampling protein structures from the Boltzmann distribution. This paper proposes a general SMC method that propagates multiple descendants for each particle, followed by resampling to maintain the desired number of particles. A simulation study demonstrates the efficacy of the method for tackling the protein sampling problem, compared to existing SMC methods. As a real data example, we estimate the number of atomic contacts for a key segment of the SARS-CoV-2 viral spike protein.

Key words and phrases: Monte Carlo methods, particle filter, protein structure analysis, SARS-CoV-2.

1. Introduction

Sequential Monte Carlo (SMC) methods, also known as particle filters, are simulation-based Monte Carlo algorithms for sampling from a target distribution. SMC originated from on-line inference problems in dynamic systems, where observations arrive sequentially and interest lies in the posterior distribution of hidden state variables (Liu and Chen (1998)). Subsequent developments include the Rao-Blackwellised particle filter and its extensions (Casella and Robert (1996); Chen and Liu (2000); Andrieu and Doucet (2002); Chen et al. (2010); Johansen, Whiteley and Doucet (2012)) and the class of particle Markov Chain Monte Carlo algorithms (Andrieu, Doucet and Holenstein (2010); Kantas et al. (2015); Chopin and Singh (2015)). These methods specialize in handling dynamic systems and their structure of hidden states. SMC has also been adapted as a useful approach for sampling from general high-dimensional probability distributions (Liu (2001); Del Moral, Doucet and Jasra (2006); Wang, Wang and Bouchard-Côté (2020)); this is the setting considered in this paper.

We begin with a review of the relevant SMC concepts for general sampling problems following Doucet, de Freitas and Gordon (2001). Assume we have a vector of random variables $(\mathbf{x}_0, \ldots, \mathbf{x}_T)$, denoted by $\mathbf{x}_{0:T}$, with continuous support

*Corresponding author.

$\mathcal{X}^{T+1}$, and we wish to draw samples from the target distribution $p(\mathbf{x}_{0:T})$. Let $f_T : \mathcal{X}^{T+1} \to \mathbb{R}^{n_{f_T}}$ denote a square integrable function of interest, then its expectation with respect to $p(\mathbf{x}_{0:T})$ is given by

$$E_p\left\{f_T\left(\mathbf{x}_{0:T}\right)\right\} = \int f_T\left(\mathbf{x}_{0:T}\right) p\left(\mathbf{x}_{0:T}\right) d\mathbf{x}_{0:T}. \tag{1.1}$$

Since this integration is usually analytically intractable, the goal of SMC is to produce a set of particles $\{(\mathbf{x}_{0:T}^{(n)}, w(\mathbf{x}_{0:T}^{(n)}))\}_{n=1}^{N}$ that is proper with respect to $p(\mathbf{x}_{0:T})$ (Liu and Chen (1998); Liu (2001); Liu, Chen and Logvinenko (2001)), i.e., $E\{f_T(\mathbf{x}_{0:T}^{(n)})w(\mathbf{x}_{0:T}^{(n)})\}$ and $E\{w(\mathbf{x}_{0:T}^{(n)})\}$ do not depend on n and satisfy

$$\frac{E\{f_T(\mathbf{x}_{0:T}^{(n)})w(\mathbf{x}_{0:T}^{(n)})\}}{E\{w(\mathbf{x}_{0:T}^{(n)})\}} = E_p\left\{f_T(\mathbf{x}_{0:T})\right\}, \tag{1.2}$$

then an estimate to (1.1) is given by $\widehat{E}_p\{f_T(\mathbf{x}_{0:T})\} = \{\sum_{n=1}^{N} f_T(\mathbf{x}_{0:T}^{(n)})w(\mathbf{x}_{0:T}^{(n)})\}/\sum_{n=1}^{N} w(\mathbf{x}_{0:T}^{(n)})$. Often, $p(\mathbf{x}_{0:T})$ does not adopt a form from which we can efficiently sample in practice (e.g., Jacquier, Polson and Rossi (2002); Carvalho et al. (2010)). In this case, a sequence of importance distributions $\eta(\mathbf{x}_0), \eta(\mathbf{x}_1 \mid \mathbf{x}_0), \ldots, \eta(\mathbf{x}_T \mid \mathbf{x}_{0:T-1})$ and a set of extended auxiliary distributions $\{p_t(\mathbf{x}_{0:t})\}_{t=0}^{T}$, with $p_t(\mathbf{x}_{0:t})$ defined on $\mathcal{X}^{t+1}$ and $p_T(\mathbf{x}_{0:T}) = p(\mathbf{x}_{0:T})$, can be introduced (Liu (2001)). To then construct $\{\mathbf{x}_{0:T}^{(n)}\}_{n=1}^{N}$, SMC generates particles according to the auxiliary distributions via a sequence of propagation and resampling steps, e.g., via the sequential importance sampling with resampling (SISR) algorithm (e.g., Liu and Chen (1995, 1998), see Section S1 in the Supplementary Material for a summary).

The motivating application of the SMC method proposed in this paper is for sampling 3-D structures of proteins. In this context, $\mathbf{x}_{0:T}$ is a vector of dihedral angles (with each angle defined on the space [-180°, 180°)) which corresponds to a *conformation*, i.e., an arrangement of the protein's atoms in 3-D space. The sampling target is the Boltzmann distribution (Boltzmann (1868); Landau and Lifshitz (2013)), defined by

$$p(\mathbf{x}_{0:T}) \propto \exp\left\{\frac{-H(\mathbf{x}_{0:T})}{\lambda}\right\} \tag{1.3}$$

where H is a given energy function, and λ is the effective temperature which can be taken to be 1 by appropriately scaling H. Based on a properly weighted Monte Carlo sample, we can estimate $E_p\{f(\mathbf{x}_{0:T})\}$ with respect to (1.3), where f is a quantity of interest (i.e., a given function evaluated on conformations $\mathbf{x}_{0:T}$); these are known as *Boltzmann averages* that represent the conformational equilibrium of the quantity f (Zhou and Berne (1997)). In practice, much of the space of (1.3) has density zero due to atomic and geometric constraints, so sampling from the Boltzmann distribution for protein structures is a challenging

task. While related to the famous *protein folding problem*, the latter focuses on finding the most likely conformation, i.e., argmax $p(\mathbf{x}_{0:T})$ (Anfinsen (1973); Onuchic, Luthey-Schulten and Wolynes (1997)), and is thus distinct from our goal. The problem of estimating Boltzmann averages is traditionally handled by using molecular dynamics (MD) or Markov chain Monte Carlo (MCMC) simulations (Adcock and McCammon (2006)). However, these methods tend to be time-consuming and suffer from being trapped in local modes; an effective SMC method to circumvent these difficulties has only been proposed for sampling simplified discrete representations of protein structures (Zhang et al. (2007)).

SMC is an intuitive strategy for sampling from (1.3) since proteins are by nature a sequence of amino acids (see Section 3 for scientific background). However, a simple SISR algorithm suffers from weight degeneracy, i.e., many of the intermediate particle weights decay to zero as a result of the highly constrained support of (1.3) (Wong, Liu and Kou (2018)). Some general techniques have been developed for handling weight degeneracy. Different resampling schemes have been proposed to improve the performance of SISR (Gordon, Salmond and Smith (1993); Kitagawa (1996); Liu and Chen (1998); Li et al. (2022)), but resampling alone cannot thoroughly solve weight degeneracy for all situations. The lookahead strategies considered in Lin, Chen and Liu (2013) incorporate information from future steps into current particles to help reduce weight degeneracy, but our experiments (Section 4) show only limited success for protein sampling. Annealed importance sampling can be adopted within an SMC algorithm to temper the auxiliary distributions (Neal (2001); Del Moral, Doucet and Jasra (2006); Dai et al. (2022)), but tempering is ineffective when many of the particle weights are exactly zero. Fearnhead and Clifford (2003) proposed to exhaustively explore the space by generating multiple descendants for each particle and then resampling; this strategy can help circumvent weight degeneracy but is only applicable to finite spaces.

In this paper, we adopt the idea of generating multiple descendants and resampling from Fearnhead and Clifford (2003), and extend it to continuous spaces as required for sampling protein conformations from (1.3). The proper weighting condition is maintained, so that we may obtain consistent estimates of Boltzmann averages. While motivated by the sampling problem in the protein context, the proposed SMC strategy is generally applicable for sampling from multivariate continuous distributions to compute Monte Carlo integrals. It may be especially effective when the support of the target distribution is highly constrained and weight degeneracy renders the SISR algorithm inapplicable.

The remainder of the paper is laid out as follows. In Section 2, we present the construction of our SMC method. In Section 3, we introduce the scientific background of proteins and the quantities of interest that may be computed from protein structures. In Section 4, we implement a simulation study that illustrates the advantages of the proposed method compared to existing ones for

sampling protein structures. In Section 5, we apply the proposed SMC method to estimate the number of atomic contacts for a key segment of the SARS-CoV-2 viral spike protein. In Section 6, we briefly summarize the paper and its contributions and discuss some potential future directions. Proofs are provided in the Supplementary Material.

2. Methodology

2.1. Review of Fearnhead and Clifford's SMC method

To introduce the key ideas, we review the relevant details of the SMC method proposed by Fearnhead and Clifford (2003), which was developed from the (partial) rejection control method (Liu, Chen and Wong (1998); Liu, Chen and Logvinenko (2001)). Consider a hidden Markov model setup with hidden process $\{\mathbf{x}_t\}_{t=0}^T$, each with finite state space $\mathcal{X}$ such that $|\mathcal{X}| = M < \infty$, observations $\{y_t\}_{t=1}^T$, distribution of the initial state $p(\mathbf{x}_0)$, transition probability $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and observation probability $l(y_t \mid \mathbf{x}_t)$ for $0 < t \leq T$. Their goal was to sample from the posterior

$$p(\mathbf{x}_{0:T} \mid y_{1:T}) \propto p(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) l(y_t \mid \mathbf{x}_t).$$

Given a set of weighted particles $\{\mathbf{x}_{0:t-1}^{(n)}\}_{n=1}^N$ up to index $t-1$, each $\mathbf{x}_{0:t-1}^{(n)}$ is used to produce M distinct descendants, denoted by $\mathbf{x}_t^{(n,m)}$, one for each value of $\mathcal{X}$. The weight of the propagated particle $\mathbf{x}_{0:t}^{(n,m)} = (\mathbf{x}_{0:t-1}^{(n)}, \mathbf{x}_t^{(n,m)})$ is given by

$$w(\mathbf{x}_{0:t}^{(n,m)}) = w(\mathbf{x}_{0:t-1}^{(n)}) q(\mathbf{x}_t^{(n,m)} \mid \mathbf{x}_{0:t-1}^{(n)}) l(y_t \mid \mathbf{x}_t^{(n,m)}). \tag{2.1}$$

To resample N particles from the NM candidates, the constant c_t in

$$\sum_{n=1}^{N} \sum_{m=1}^{M} \min\left\{c_t w(\mathbf{x}_{0:t}^{(n,m)}), 1\right\} = N \tag{2.2}$$

is calculated. Let L be the number of particles with weights greater than or equal to $1/c_t$; these L particles are all preserved, and stratified sampling (Carpenter, Clifford and Fearnhead (1999)) is used to resample another $N - L$ particles without replacement. Fearnhead and Clifford (2003) showed that this downsampling scheme minimizes

$$E\left[\sum_{n=1}^{N} \sum_{m=1}^{M} \left\{Q(\mathbf{x}_{0:t}^{(n,m)}) - \gamma_t^{(n,m)}\right\}^2\right]$$

where $\gamma_t^{(n,m)}$ denotes the weight with respect to p_t and $Q(\mathbf{x}_{0:t}^{(n,m)})$ is the stochastic weight of the sampled particle, and thus is optimal among downsampling schemes.

For a chosen particle size N, the computational complexity of the algorithm is $O(NM)$ since the evaluation of the incremental weights in (2.1) tends to be the most expensive part of the computation in practice; in contrast, the cost of SISR is $O(N)$ in this regard.

As a generalization of these key ideas, we shall, in the following context, define *upsampling* to be a propagation step that samples $M \geq 1$ descendants from each of the N existing particles, and *downsampling* to be the step of resampling N particles from the NM total descendants. An *upsampling-downsampling framework* shall refer to an SMC algorithm that combines these upsampling and downsampling features.

2.2. A general upsampling-downsampling SMC framework

Our goal is to sample from a general multivariate distribution $p(\mathbf{x}_{0:T})$ where $\mathbf{x}_t$ is a random vector with continuous support, with the help of an upsampling-downsampling SMC framework (or *UDSMC* for short). In this situation, it is impossible to explore every value of $\mathcal{X}$, so we use importance distributions $\eta(\mathbf{x}_0), \eta(\mathbf{x}_1 \mid \mathbf{x}_0), \ldots, \eta(\mathbf{x}_T \mid \mathbf{x}_{0:T-1})$ to facilitate sampling.

Assume for a chosen particle size N, upsample size M, and $t > 0$, we have a set of generated particles $\{\mathbf{x}_{0:t-1}^{(n)}\}_{n=1}^{N}$ with weights $\{w(\mathbf{x}_{0:t-1}^{(n)})\}_{n=1}^{N}$. For each $\mathbf{x}_{0:t-1}^{(n)}$, we sample M descendants $\{\mathbf{x}_t^{(n,m)}\}_{m=1}^{M}$ from $\eta(\mathbf{x}_t \mid \mathbf{x}_{0:t-1}^{(n)})$ and define $\mathbf{x}_{0:t}^{(n,m)} = (\mathbf{x}_{0:t-1}^{(n)}, \mathbf{x}_t^{(n,m)})$ with

$$w(\mathbf{x}_{0:t}^{(n,m)}) = \frac{w(\mathbf{x}_{0:t-1}^{(n)}) p_t(\mathbf{x}_{0:t}^{(n,m)})}{p_{t-1}(\mathbf{x}_{0:t-1}^{(n)}) \eta(\mathbf{x}_t^{(n,m)} \mid \mathbf{x}_{0:t-1}^{(n)})}. \tag{2.3}$$

These NM propagated particles with the upsampled weights in (2.3) are then carried to the downsampling step to resample N particles.

The downsampling step resamples N particles from the set $\{\mathbf{x}_{0:t}^{(m,n)}, n = 1, \ldots, N \text{ and } m = 1, \ldots, M\}$; the resampled particles are then denoted by $\{\mathbf{x}_{0:t}^{(n)}\}_{n=1}^{N}$. There are two possible cases after each upsampling step: (i) at least N of $w(\mathbf{x}_{0:t}^{(n,m)})$'s are positive and (ii) less than N of $w(\mathbf{x}_{0:t}^{(n,m)})$'s are positive. (Note that Fearnhead and Clifford (2003) did not need to consider case (ii) as the Gaussian observation likelihood in their application always produces positive weights after upsampling.) Handling case (ii) ensures a valid algorithm when weight degeneracy is very severe.

For case (i), the downsampling step follows the method of Fearnhead and Clifford (2003). After the t-th upsampling step, the threshold c_t in the equation

$$\sum_{n=1}^{N} \sum_{m=1}^{M} \min\left\{c_t w(\mathbf{x}_{0:t}^{(n,m)}), 1\right\} = N$$

is calculated. Let L be the number of particles whose weights are greater

Algorithm 1: UDSMC

Require: particle size N, upsample size M;

Initial upsampling: Sample $\{\mathbf{x}_0^{(n,m)}, n = 1, \ldots, N \text{ and } m = 1, \ldots, M\}$ from $\eta(\mathbf{x}_0)$, each with the weight $w(\mathbf{x}_0^{(n,m)}) = p_0(\mathbf{x}_0^{(n,m)})/\eta(\mathbf{x}_0^{(n,m)})$;

Initial downsampling: Resample N particles, denoted by $\{\mathbf{x}_0^{(n)}\}_{n=1}^N$, from $\{\mathbf{x}_0^{(n,m)}, n = 1, \ldots, N \text{ and } m = 1, \ldots M\}$ with weights $\{w(\mathbf{x}_0^{(n)})\}_{n=1}^N$ using the proposed scheme;

for $t = 1, \ldots, T$ **do**

Upsampling: Sample $\{\mathbf{x}_t^{(n,m)}\}_{m=1}^M$ from $\eta(\mathbf{x}_t \mid \mathbf{x}_{0:t-1}^{(n)})$ and set $\mathbf{x}_{0:t}^{(n,m)} = (\mathbf{x}_{0:t-1}^{(n)}, \mathbf{x}_t^{(n,m)})$ for each n, each with a weight $w(\mathbf{x}_{0:t}^{(n,m)}) = w(\mathbf{x}_{0:t-1}^{(n)}) p_t(\mathbf{x}_{0:t}^{(n,m)})/\{p_{t-1}(\mathbf{x}_{0:t-1}^{(n)})\eta(\mathbf{x}_t^{(n,m)} \mid \mathbf{x}_{0:t-1}^{(n)})\}$;

Downsampling: Resample N particles, denoted by $\{\mathbf{x}_{0:t}^{(n)}\}_{n=1}^N$, from $\{\mathbf{x}_{0:t}^{(n,m)}, n = 1, \ldots, N \text{ and } m = 1, \ldots M\}$ with weights $\{w(\mathbf{x}_{0:t}^{(n)})\}_{n=1}^N$ using the proposed scheme;

end

than or equal to $1/c_t$; these are preserved together with the weights $w(\mathbf{x}_{0:t}^{(n,m)})$. $N - L$ particles are sampled without replacement from the remaining $NM - L$ particles proportional to their weights $w(\mathbf{x}_{0:t}^{(n,m)})$ and are assigned new weights $1/c_t$. Therefore, after the downsampling step, each $\mathbf{x}_{0:t}^{(n,m)}$ has a stochastic downsampled weight $Q(\mathbf{x}_{0:t}^{(n,m)})$ as

$$Q(\mathbf{x}_{0:t}^{(n,m)}) = \begin{cases} \dfrac{w(\mathbf{x}_{0:t}^{(n,m)})}{q(\mathbf{x}_{0:t}^{(n,m)})} & \text{with probability } q(\mathbf{x}_{0:t}^{(n,m)}) \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where $q(\mathbf{x}_{0:t}^{(n,m)}) = \min\{c_t w(\mathbf{x}_{0:t}^{(n,m)}), 1\}$. This resampling scheme ensures that the downsampled weights $Q(\mathbf{x}_{0:t}^{(n,m)})$ are proper with respect to p_t. This resampling scheme is optimal in terms of minimizing the expected squared error loss when resampling N distinct particles from the given collection of NM upsampled particles. These results are formalized in Propositions 1 and 2 below. The N downsampled particles are then the collection of those with realizations $Q(\mathbf{x}_{0:t}^{(n,m)}) > 0$, which we set to be $\{\mathbf{x}_{0:t}^{(n)}\}_{n=1}^N$ with corresponding weights $\{w(\mathbf{x}_{0:t}^{(n)})\}_{n=1}^N$.

Proposition 1. *For $t \in \{0, \ldots, T\}$ and the auxiliary distribution p_t, suppose there are at least N positive upsampled weights for all $s \in \{0, \ldots, t\}$. Let $\mathbf{x}_{0:t}$ be any particle from the space $\mathcal{X}^{t+1}$, then its downsampled particles $\{(\mathbf{x}_{0:t}^{(n,m)}, Q(\mathbf{x}_{0:t}^{(n,m)})); n = 1, \ldots, N, m = 1, \ldots, M\}$ given in Equation (2.4) are proper with respect to p_t.*

Proof. See Section S2 in the Supplementary Material.

Proposition 2. *For $t \in \{0, \ldots, T\}$ and upsampled particles $\{\mathbf{x}_{0:t}^{(n,m)}; n = 1, \ldots, N, m = 1, \ldots, M\}$, suppose there are at least N positive upsampled weights at step t. When each individual downsampled weight $Q(\mathbf{x}_{0:t}^{(n,m)})$ is assigned by Equation (2.4) with the constraint that no more than N elements of $\{Q(\mathbf{x}_{0:t}^{(n,m)}); n = 1, \ldots, N, m = 1, \ldots, M\}$ are positive, then the downsampled weights minimize the conditional expected squared error loss*

$$E\left[\sum_{n=1}^{N}\sum_{m=1}^{M}\left\{Q(\mathbf{x}_{0:t}^{(n,m)}) - \gamma_t^{(n,m)}\right\}^2 \,\middle|\, \{\mathbf{x}_{0:t}^{(n,m)}; n = 1, \ldots, N, m = 1, \ldots, M\}\right]$$

where $\gamma_t^{(n,m)} = \{p_t(\mathbf{x}_{0:t}^{(n,m)})/\eta(\mathbf{x}_{0:t}^{(n,m)})\}/\sum_{n=1}^{N}\sum_{m=1}^{M}\{p_t(\mathbf{x}_{0:t}^{(n,m)})/\eta(\mathbf{x}_{0:t}^{(n,m)})\}$.

Proof. See Section S3 in the Supplementary Material.

For case (ii), we have less than N particles with positive weights so cannot preserve N distinct particles via downsampling; instead, resampling N particles with replacement is needed, similar to the resampling step in SISR. The rationale for resampling in this case is to effectively duplicate particles with high weights, thereby retaining more potentially feasible particles as starting points for the following propagation step. We use multinomial resampling for simplicity, which mimics the resampling scheme of the bootstrap filter (Gordon, Salmond and Smith (1993)), but other schemes can also be used. The key difference between case (i) and (ii) is that there are duplicates of particles with equal weights after downsampling for case (ii).

Finally to initialize the algorithm, NM realizations of $\mathbf{x}_0$, i.e. $\{\mathbf{x}_0^{(n,m)}, n = 1, \ldots, N$ and $m = 1, \ldots, M\}$, are first sampled from $\eta(\mathbf{x}_0)$ (each with importance weight $p_0(\mathbf{x}_0)/\eta(\mathbf{x}_0)$), followed by a downsampling step to obtain N properly weighted particles representing $p_0(\mathbf{x}_0)$. Algorithm 1 summarizes UDSMC.

Analogously to that described in Section 2.1, UDSMC has computational complexity $O(NM)$ due to the evaluation of the incremental weights in (2.3). Thus, our method is best suited for situations where running SISR with a very large particle size (e.g., larger than NM in Algorithm 1) does not produce satisfactory results; the subsequent protein application is one such situation. Moreover, the performance of Algorithm 1 will be influenced by the choice of the upsample size M. For a finite space $\mathcal{X}$, a natural choice for M can be to take $M = |\mathcal{X}|$; however, there may not be an intuitive choice for M for a continuous space $\mathcal{X}$. Thus to choose a reasonable value of M in a given application, we may run preliminary experiments using different values of M (with a fixed computational budget, i.e., holding MN constant), as we subsequently demonstrate.

3. Estimating Protein Structural Quantities

Proteins have a crucial role in carrying out biological processes and their functions are dependent on their 3-D structures. A protein is composed of a sequence of amino acids, by which its 3-D structure is essentially determined (Anfinsen (1973)). However, it is also well-known that protein structures are not static; some dynamic movement is observed (Fraser et al. (2009, 2011); Bu and Callaway (2011)) and the relative probability of different conformations can be characterized by the Boltzmann distribution defined in (1.3), i.e., conformations with lower energy are more favorable. We focus on the 3-D structures of key protein segments due to their important biological functions, e.g., the highly dynamic region of the coronavirus spike protein that binds with human cells (Lan et al. (2020)). While the Protein Data Bank (PDB) (Berman et al. (2000)) is the source of known protein structures obtained from laboratory work, these should be considered as only static snapshots of a given protein. To study the dynamic movement of a protein and estimate Boltzmann averages, efficient computational approaches are needed for sampling conformations, which motivates our current work.

The amino acid indexed by t in the sequence is composed of four *backbone* atoms (the nitrogen atom denoted by N^t, the carbon atoms denoted by C_α^t and C^t, and the oxygen atom denoted by O^t) and a *side chain* denoted by R^t. Figure 1 illustrates how these backbone atoms and side chains are connected, with the solid lines indicating bonds. Successive amino acids are also connected, e.g., the bond between C^t and N^{t+1} connects the amino acids indexed by t and $t+1$ in Figure 1. The 3-D coordinates of the backbone atoms $a_t = (\mathrm{C}^t, \mathrm{O}^t, \mathrm{N}^{t+1}, \mathrm{C}_\alpha^{t+1})$ can be equivalently specified using a vector of dihedral angles $(\phi_t, \psi_t, \omega_t)$, where ϕ_t represents the dihedral angle of $\mathrm{C}^{t-1} - \mathrm{N}^t - \mathrm{C}_\alpha^t$ and $\mathrm{N}^t - \mathrm{C}_\alpha^t - \mathrm{C}^t$; ψ_t represents the dihedral angle of $\mathrm{N}^t - \mathrm{C}_\alpha^t - \mathrm{C}^t$ and $\mathrm{C}_\alpha^t - \mathrm{C}^t - \mathrm{N}^{t+1}$; ω_t represents the dihedral angle of $\mathrm{C}_\alpha^t - \mathrm{C}^t - \mathrm{N}^{t+1}$ and $\mathrm{C}^t - \mathrm{N}^{t+1} - \mathrm{C}_\alpha^{t+1}$. Figure 1 demonstrates an example of how ϕ_t is determined, and ψ_t is determined in a similar way by shifting the planes up by one atom. Since bond lengths and bond angles are essentially fixed, ϕ_t governs the distance between C^{t-1} and C^t; ψ_t governs the distance between N^t and N^{t+1} and also determines the coordinates of O^t. The bond connecting C^t and N^{t+1} is nearly non-rotatable, and thus the dihedral angle ω_t that governs the distance between C_α^t and C_α^{t+1} is usually close to 180° (Esposito et al. (2005)). In contrast, ϕ_t and ψ_t can take a wide range of values over the continuous interval [-180°, 180°).

Within a given protein, suppose that the segment of interest consists of the amino acids indexed from j to k, denoted by $(a_j, \ldots, a_k)$. The backbone structure of the segment can then be parameterized by $\mathbf{x}_{0:T} = (\mathbf{x}_0, \ldots, \mathbf{x}_T)$ with $T = k - j$, where $\mathbf{x}_t$, $t \in \{0, \ldots, T\}$, represents the three dihedral angles for the amino acid a_{j+t}, i.e., $\mathbf{x}_t = (\phi_{j+t}, \psi_{j+t}, \omega_{j+t})$. To focus on the properties of the segment, the

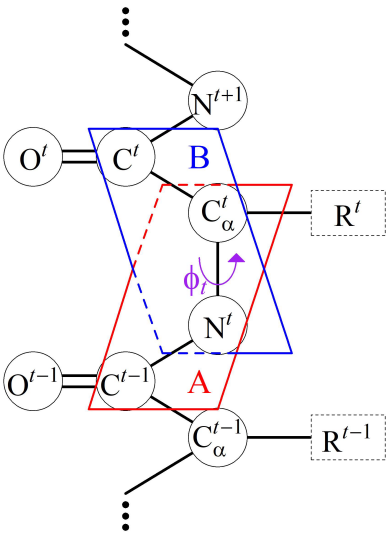

Figure 1. Illustration of connections and dihedral angles in a protein segment. Solid lines in the figure indicate bonds (connections). Within the amino acid indexed by t, N^t, C^t_α, C^t and O^t are successively connected, and the side chain R^t (that distinguishes different amino acid types) is connected to C^t_α; the double lines between C^t and O^t indicate a double-bond connection. Successive amino acids are connected via a C-N bond, e.g., amino acids indexed by t and $t+1$ are connected with the bond between C^t and N^{t+1}. To define the dihedral angle ϕ_t, consider the two planes marked in the figure: Plane A (red) consists of atoms C^{t-1}, N^t and C^t_α; Plane B (blue) consists of atoms N^t, C^t_α and C^t. Then ϕ_t is the angle between Plane A and Plane B when looking down the $\mathrm{N}-\mathrm{C}_\alpha$ bond.

rest of the protein is held fixed (e.g., at a static snapshot obtained from the PDB) while sampling conformations for the backbone of $(a_j, \ldots, a_k)$. For simplicity, the side chains R^t are not considered in this work.

The energy function H in (1.3) consists of two components: following Wong, Liu and Kou (2018), for a given backbone segment $\mathbf{x}_{0:T}$, the energy of atomic interactions $H_a(\mathbf{x}_{0:T})$ is based on pairwise atom distances (between atoms within the segment, and also atoms between the segment and the rest of the protein), and the energy of dihedral angles $H_\theta(\mathbf{x}_{0:T})$ is based on empirical distributions derived from historical protein data (Wong, Liu and Kou (2017)). The total energy of the segment is $H(\mathbf{x}_{0:T}) = H_a(\mathbf{x}_{0:T}) + H_\theta(\mathbf{x}_{0:T})$; detailed calculations for H_a and H_θ are presented in Section S4 of the Supplementary Material. It should be noted that compared to H_θ, H_a is expensive to compute since all pairwise distances between atoms need to be calculated; moreover, if two atoms violate geometric constraints (e.g., they are too close together in 3-D space), then the conformation will have $H_a = +\infty$ which implies a zero weight. This property of H_a leads to a highly constrained support for (1.3), which poses a challenge for sampling methods.

In protein structure analysis, summary statistics $f(\mathbf{x}_{0:T})$ can be computed; we call these *protein structural quantities*. The C_α backbone atoms are often treated as the representative of each amino acid for such computations (Simons et al. (1997); Di Lena, Nagata and Baldi (2012)). Summary statistics involving *atomic contacts*, i.e., the number of other atoms within a given radius of a selected atom, help determine whether the selected atom is close to the protein's surface (Karlin, Zhu and Baud (1999); Pintar, Carugo and Pongor (2002)). Within a given protein segment, C_α atoms with fewer atomic contacts tend to be on the surface of the protein and therefore more likely to interact with other molecules. To illustrate our methodology, we consider using the samples from (1.3) to estimate $E_p\{f(\mathbf{x}_{0:T})\}$ for the atomic contacts of C_α atoms within a radius of 7 Å.

4. Simulation Study

As a simulation study, we consider the length 8 segment from a_{284} to a_{291} of the protein with PDB ID *1ds1A*. The protein structural quantity of interest is the atomic contacts for each C_α in the segment, denoted by $n(C_\alpha^{285}), \ldots, n(C_\alpha^{292})$. This protein segment is of sufficient difficulty (the dimension of the Boltzmann distribution is 24 in this example) to showcase a comparison of different methods, while at the same time is simple enough such that the ground truth of the Boltzmann averages can be obtained via brute force.

To obtain the ground truth for this example, the brute force approach we used is as follows. We sequentially draw samples from the importance distributions $\eta(\mathbf{x}_0), \eta(\mathbf{x}_1 \mid \mathbf{x}_0), \ldots, \eta(\mathbf{x}_7 \mid \mathbf{x}_{0:6})$, defined by

$$\begin{aligned} \eta(\mathbf{x}_0) &\propto \exp\{-H_\theta(\mathbf{x}_0)\} \\ \eta(\mathbf{x}_t \mid \mathbf{x}_{0:t-1}) = \eta(\mathbf{x}_t) &\propto \exp\{-H_\theta(\mathbf{x}_t)\} \quad t = 1, \ldots 7, \end{aligned} \tag{4.1}$$

since H_θ is constructed to be independent for each amino acid position (see Section S4 of the Supplementary Material). Note that any importance distribution could be used here; H_θ is chosen for this purpose as it is easy to sample from the empirical distribution of dihedral angles and is more efficient than drawing $\mathbf{x}_t$ uniformly. Given that a partial sequence from $\mathbf{x}_0$ to $\mathbf{x}_s$ with $s \in \{0, \ldots, 6\}$ has been generated, we draw descendants from $\eta(\mathbf{x}_{s+1})$ until a sample with positive importance weight is obtained, i.e., $H(\mathbf{x}_{0:s+1}) < +\infty$; if it is not possible to obtain such a sample, we discard the partial sequence and start over. After successfully simulating a complete conformation, the total energy $H(\mathbf{x}_{0:7})$ is evaluated to obtain its importance weight.

Table 1 displays the ground truth for the Boltzmann averages of the eight quantities of interest, as approximated from a total of 10^8 samples (with positive importance weights). To verify that 10^8 samples provide a good proxy for the ground truth here, we randomly divided them into two subgroups of 5×10^7 samples, repeating 50 times to form a total of 100 groups of 5×10^7 samples. The

Table 1. Boltzmann averages of the eight quantities of interest. The 10^8 samples obtained using our brute force approach were used as a proxy for the ground truth.

$n(C_\alpha^{285})$	$n(C_\alpha^{286})$	$n(C_\alpha^{287})$	$n(C_\alpha^{288})$	$n(C_\alpha^{289})$	$n(C_\alpha^{290})$	$n(C_\alpha^{291})$	$n(C_\alpha^{292})$
35.135	43.042	45.313	35.236	38.300	55.303	53.273	65.142

same quantities were estimated from each of the 100 groups. For each quantity, the standard deviations of these estimates were less than 0.4% of the Boltzmann average computed from the full (10^8) sample, which indicates the stability desired. Overall, this computation of the ground truth was expensive: it required one week of compute time on a cluster of 300 Intel Xeon CPU cores.

Having established a proxy for the ground truth, we proceed to compare the performance of three different SMC methods in this section. For a fair comparison, the importance distributions for the propagation step are chosen to be the same for each SMC method, namely by adopting η as defined in (4.1). The first one is UDSMC. We implement UDSMC with an upsample size $M = 20$, based on the results of an experiment that assessed the Monte Carlo variance of different values of M with the overall computational budget fixed (see details in Section S5 of the Supplementary Material).

The second one is the SISR algorithm, as summarized in Section S1 of the Supplementary Material. SISR is commonly used and easy to implement, but it can encounter problems when weight degeneracy is severe and can fail to complete (i.e., all particles have zero weights before any complete conformations are simulated). Here, if SISR fails to complete, we simply restart the algorithm from the beginning.

The third one is SISR with the lookahead strategy, or simply *lookahead SISR*, as detailed in Section S6 of the Supplementary Material. The lookahead strategy of Lin, Chen and Liu (2013) utilizes "future" information for inference on the "current" states, and this idea was also adapted by Wong, Liu and Kou (2018) for exploring protein structures in a finite space; here, we may apply this strategy in the context of SISR to alleviate the problem of severe weight degeneracy. After the t-th propagation step (i.e., producing $\mathbf{x}_{0:t}$), lookahead SISR implements an extra one-step exploration that evaluates multiple (L) "future" descendants (i.e., by generating $\{(\mathbf{x}_{0:t}, \mathbf{x}_{t+1}^l)\}_{l=1}^L$) and then implements a resampling step using the marginalized importance weights (i.e., $\sum_{l=1}^L w(\mathbf{x}_{0:t}, \mathbf{x}_{t+1}^l)$). In this way, the dead-end particles, i.e., where all L "future" descendants have zero weights, are discarded before resampling. We set $L = 20$ for lookahead SISR to mimic the upsample size of UDSMC in this simulation.

We run 100 repetitions of each method in the following and report the RMSEs of the estimates in Table 2. For UDSMC, we chose a range of N values from 5,000 to 20,000; the computation speed is fast, e.g., $N = 10000$ and $M = 20$ had a time cost of about 200 seconds per repetition on a single core of a Xeon Gold 6244 3.6

Table 2. RMSEs of the eight quantities, based on 100 repetitions of UDSMC ($M = 20$), SISR and lookahead SISR ($L = 20$) for the protein segment $\mathbf{x}_{284:291}$. The corresponding rows of each method have a similar computational cost: e.g., UDSMC with $N = 5000$ and $M = 20$, SISR with $N = 100000$, and lookahead SISR with $N = 5000$ and $L = 20$.

Method	N	$n(C_\alpha^{285})$	$n(C_\alpha^{286})$	$n(C_\alpha^{287})$	$n(C_\alpha^{288})$	$n(C_\alpha^{289})$	$n(C_\alpha^{290})$	$n(C_\alpha^{291})$	$n(C_\alpha^{292})$
UDSMC ($M = 20$)	5,000	0.612	1.576	1.556	1.372	2.794	2.701	1.539	0.576
	10,000	0.310	1.291	1.264	1.089	2.133	2.480	1.171	0.416
	20,000	0.283	0.759	0.773	0.749	1.414	1.626	0.855	0.263
SISR	100,000	4.825	8.382	9.638	10.058	15.342	15.236	10.019	3.347
	200,000	2.988	8.020	7.995	9.115	13.843	14.489	9.546	3.248
	400,000	1.528	7.773	7.213	8.701	11.944	13.372	9.019	2.404
Lookahead SISR ($L = 20$)	5,000	3.943	5.842	6.790	5.520	8.864	9.262	7.283	6.727
	10,000	1.190	2.879	3.858	3.375	5.482	6.070	4.408	1.088
	20,000	0.576	1.971	3.176	2.633	5.302	5.386	3.889	0.788

GHz processor. To fairly compare the three methods, we assess their performance given a similar computational budget: given the N particles obtained from a resampling step, all three methods sample the same total number of descendants for the subsequent propagation step (i.e., by setting NM for UDSMC, N for SISR, and NL for lookahead SISR to be equal), since the evaluation of the energy function (specifically, H_a) is the computational bottleneck. For example, UDSMC with $N = 5000$ and $M = 20$ has similar computational cost as SISR with $N = 100000$ and lookahead SISR with $N = 5000$ and $L = 20$.

In Table 2, we observe that RMSEs decrease as the particle size increases for each SMC method; however, this pattern is not uniform across the amino acids in the segment. The RMSEs tend to be higher for the amino acids in the middle of the segment; this is sensible because amino acids further from the anchors tend to be less geometrically constrained by the rest of the protein. The RMSEs for amino acids in the middle of the segment also tend to decrease more slowly as particle size increases. Comparing the rows that have similar computational cost, we observe that SISR yields higher RMSEs compared to UDSMC and lookahead SISR (e.g., the RMSEs for SISR with $N = 200000$ are substantively higher compared to UDSMC with $N = 10000$ and lookahead SISR with $N = 10000$), while UDSMC also outperforms lookahead SISR (with $> 50\%$ reduction in RMSE for every amino acid, compared to lookahead SISR for the same value of N). These simulation results indicate that UDSMC provides a more efficient approach to sampling conformations from the Boltzmann distribution compared to the two existing strategies. The relative inefficiency of SISR (even with a large N) makes sense: since when only a small fraction of the propagated particles have positive weights, there will be many duplicates among the N particles after a resampling step. This problem is exacerbated by successive iterations of propagation and resampling. For further discussion and a graphical

example of how particle diversity decays, see Section S7 of the Supplementary Material. Consequently, SISR can give poor estimates or even become entirely stuck in dead ends. Applying the lookahead strategy partially alleviates this problem, but is not nearly as effective as UDSMC for a given computational budget.

The protein segment used in this simulation study was short enough such that the ground truth could be obtained via brute force; in practice, target protein segments often have a much longer length (e.g., 12–20 amino acids long) so that the higher dimension of $\mathbf{x}$ poses a substantively more difficult sampling problem. The brute force approach used to facilitate benchmarking for this simulation study would no longer be applicable due to a prohibitive computational cost; e.g., it takes more than one hour to generate 10 valid samples for a typical length 18 segment (i.e., the length of our subsequent real data example) on a single core of a Xeon Gold 6244 3.6 GHz processor, and likely more than 10^8 samples would be needed to obtain stable estimates for the longer segment. Further, SISR will encounter such severe weight degeneracy that it cannot be run successfully; this point is also illustrated by the experiment in Section S5 in the Supplementary Material, which considers a length 10 segment. The results show that the SISR algorithm often fails to complete even with $N = 1000000$ (i.e., all particles end up with weight zero), so simply increasing N in SISR does not address the weight degeneracy problem for longer segments. In contrast, UDSMC provides stable estimates with a much smaller computational budget: using $N = 5000$ and $M = 20$ works well and costs 10 times less than SISR with $N = 1000000$.

5. Example: Estimating the Atomic Contacts in the SARS-CoV-2 Spike Protein

The COVID-19 pandemic was caused by the novel coronavirus SARS-CoV-2, with the first identified outbreak in Wuhan, China, in 2019 (Chen, Liu and Guo (2020)). This virus binds with a human host cell via an interaction between its spike protein's receptor-binding domain (RBD) and the host cell's angiotensin-converting enzyme 2 (ACE2) receptor (Lan et al. (2020)). The protein segments of the RBD involved in binding have thus been of scientific interest towards the development of therapeutics. Four key segments of amino acids have been previously identified as being involved in RBD–ACE2 binding; of these, several research groups have suggested that the segment 472–490 (known as *Loop 3*) exhibits the greatest dynamic movement (Ali and Vijayan (2020); Nguyen et al. (2020); Dehury et al. (2021); Williams et al. (2022)). The presence of such dynamic movement has implications for drug design; e.g., a drug could target a specific conformation of the segment to prevent the virus from being able to bind.

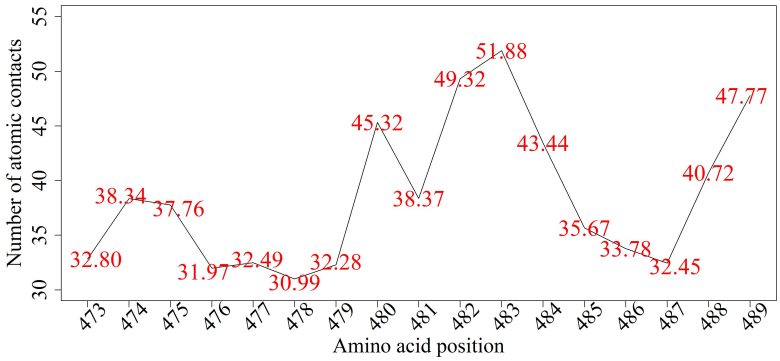

Figure 2. Estimated Boltzmann averages for the atomic contacts of C_α atoms at amino acid positions 473 to 489 of the SARS-CoV-2 spike protein, using the samples obtained from running UDSMC with $N = 500000$ and $M = 20$.

Since the initial COVID-19 outbreak, laboratory work (e.g., Wrapp et al. (2020)) has provided static snapshots for the overall 3-D structure of the SARS-CoV-2 spike protein, which are publicly available in the PDB. These static snapshots have been useful as a starting point for studying the dynamic movement of key segments; e.g., Williams et al. (2022) used MD simulations to explore the possible conformations of Loop 3. The importance of Loop 3 was further reinforced by the emergence of the Delta and Omicron variants of SARS-CoV-2; both of these variants had mutations (i.e., changes to the amino acid sequence) in Loop 3 that could induce changes to its conformation.

In this example, we took the same starting 3-D structure of the spike protein as Williams et al. (2022), and applied UDSMC to sample backbone conformations of Loop 3. We then estimated the Boltzmann averages for the atomic contacts for each C_α in the segment. Amino acid positions that have fewer contacts on average tend to be closer to the protein surface, and hence more likely to be directly involved in binding. We ran UDSMC with $M = 20$ and $N = 500000$, and the estimated Boltzmann averages for $n(C_\alpha^{473}), \ldots, n(C_\alpha^{489})$ are plotted in Figure 2. These results suggest that on average, position 478 is the most likely to be on the protein's surface (among several consecutive positions with a low number of contacts, namely 476–479), while position 483 is the least likely. Interestingly, Williams et al. (2022) also found that 476, 477, and 478, were among the top four amino acid positions where mutations occurred; from a biological point of view, this suggests that these surface amino acids are quite resilient to mutation.

6. Conclusion and Discussion

In this paper, we proposed an SMC method that features an upsampling-downsampling framework (UDSMC), with an emphasis on sampling backbone conformations of protein segments from the Boltzmann distribution. Existing

SMC methods are not effective for this problem: SISR is hindered by severe weight degeneracy while incorporating a lookahead strategy only partially alleviates the difficulty of sampling conformations. UDSMC preserves the collection of distinct particles needed for sampling from a highly constrained distribution and thereby provides an efficient approach for estimating Boltzmann averages. We show that UDSMC generates a properly weighted sample and illustrate the performance of the method in a simulation study. As a real data example, we used UDSMC to estimate the atomic contacts for a key segment of the SARS-CoV-2 spike protein. Our work is also distinct from methods designed for protein folding; the latter usually search for conformations with the lowest energy, without regard for properly weighted samples.

UDSMC provides an approach to estimate Boltzmann averages but comes with a price: its computational complexity is $O(NM)$. Thus, choosing a value of M involves a tradeoff between computing speed and exploration effort of high-density regions. It is sensible to fix the computational budget NM before selecting M for a given application. While there is no universally optimal choice of M for all SMC applications, we can provide some intuition. When the target distribution is more uniform, a larger N is preferred; in the extreme case where the target distribution is uniform over the support, the optimal choice is clearly $M = 1$ to maximize the number of samples N for a fixed NM. In contrast, when the target distribution has many sharp local modes or is highly constrained, a larger M is preferred to ensure that UDSMC sufficiently explores the regions with positive density during propagation steps, and a specific value of M could be chosen according to Monte Carlo variance. Therefore, while UDSMC is generally applicable, its efficacy will depend on the target distribution of interest. When sampling protein segments, geometric constraints lead to a rough energy surface with many local modes and large regions of zero density. This situation is well-suited for UDSMC, whereas other SMC methods could fail.

Potential future applied work could involve protein analyses with different structural quantities or energy functions. It is straightforward to adapt UDSMC for this purpose. The sampling framework could also be extended to consider both the backbone and the side chains. One approach would be to sample the side chains after sampling backbone conformations, effectively applying SMC twice and accumulating the energy contributions.

Supplementary Material

The Supplementary Material contains algorithmic descriptions for SISR in Section 1 and lookahead SISR in Section 4, the proofs of Propositions 1 and 2 in Section 2.2, the details of the protein energy function in Section 3, the experiment that assesses the Monte Carlo variance of different values of M with overall computational budget fixed in Section 4, and a graphical example of decay

in the diversity of the paths of the particles in Section 4.

Acknowledgments

We thank Martin Lysy and Glen McGee for constructive comments on the manuscript. This work was partially supported by Discovery Grant RGPIN-2019-04771 from the Natural Sciences and Engineering Research Council of Canada.

References

Adcock, S. A. and McCammon, J. A. (2006). Molecular dynamics: Survey of methods for simulating the activity of proteins. *Chemical Reviews* **106**, 1589–1615.

Ali, A. and Vijayan, R. (2020). Dynamics of the ACE2–SARS-CoV-2/SARS-CoV spike protein interface reveal unique mechanisms. *Scientific Reports* **10**, 14214.

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **64**, 827–836.

Andrieu, C., Doucet, A. and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**, 269–342.

Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science* **181**, 223–230.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H. et al. (2000). The protein data bank. *Nucleic Acids Research* **28**, 235–242.

Boltzmann, L. (1868). Studien uber das Gleichgewicht der lebenden Kraft. *Wissenschafiliche Abhandlungen* **1**, 49–96.

Bu, Z. and Callaway, D. J. (2011). Proteins move! Protein dynamics and long-range allostery in cell signaling. *Advances in Protein Chemistry and Structural Biology* **83**, 163–221.

Carpenter, J., Clifford, P. and Fearnhead, P. (1999). Improved particle filter for non-linear problems. *IEE Proceedings - Radar, Sonar and Navigation* **146**, 2–7.

Carvalho, C. M., Johannes, M. S., Lopes, H. F. and Polson, N. G. (2010). Particle learning and smoothing. *Statistical Science* **25**, 88–106.

Casella, G. and Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika* **83**, 81–94.

Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**, 493–508.

Chen, T., Schon, T. B., Ohlsson, H. and Ljung, L. (2010). Decentralized particle filter with arbitrary state decomposition. *IEEE Transactions on Signal Processing* **59**, 465–478.

Chen, Y., Liu, Q. and Guo, D. (2020). Emerging coronaviruses: Genome structure, replication, and pathogenesis. *Journal of Medical Virology* **92**, 418–423.

Chopin, N. and Singh, S. (2015). On particle Gibbs sampling. *Bernoulli* **21**, 1855–1883.

Dai, C., Heng, J., Jacob, P. E. and Whiteley, N. (2022). An invitation to sequential Monte Carlo samplers. *Journal of the American Statistical Association* **117**, 1587–1600.

Dehury, B., Raina, V., Misra, N. and Suar, M. (2021). Effect of mutation on structure, function and dynamics of receptor binding domain of human SARS-CoV-2 with host cell receptor ACE2: A molecular dynamics simulations study. *Journal of Biomolecular Structure and Dynamics* **39**, 7231–7245.

Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of*

the Royal Statistical Society: Series B (Statistical Methodology) **68**, 411–436.

Di Lena, P., Nagata, K. and Baldi, P. (2012). Deep architectures for protein contact map prediction. *Bioinformatics* **28**, 2449–2457.

Doucet, A., de Freitas, N. and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice* (Edited by A. Doucet, N. de Freitas and N. Gordon), 3–14. Springer.

Esposito, L., De Simone, A., Zagari, A. and Vitagliano, L. (2005). Correlation between ω and ψ dihedral angles in protein structures. *Journal of Molecular Biology* **347**, 483–487.

Fearnhead, P. and Clifford, P. (2003). On-line inference for hidden Markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65**, 887–899.

Fraser, J. S., Clarkson, M. W., Degnan, S. C., Erion, R., Kern, D. and Alber, T. (2009). Hidden alternative structures of proline isomerase essential for catalysis. *Nature* **462**, 669–673.

Fraser, J. S., van den Bedem, H., Samelson, A. J., Lang, P. T., Holton, J. M., Echols, N. et al. (2011). Accessing protein conformational ensembles using room-temperature X-ray crystallography. *Proceedings of the National Academy of Sciences* **108**, 16247–16252.

Gordon, N. J., Salmond, D. J. and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE proceedings. Part F. Radar and Signal Processing* **140**, 107–113.

Jacquier, E., Polson, N. G. and Rossi, P. E. (2002). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics* **20**, 69–87.

Johansen, A. M., Whiteley, N. and Doucet, A. (2012). Exact approximation of Rao-Blackwellised particle filters. *IFAC Proceedings Volumes* **45**, 488–493.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J. and Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statistical Science* **30**, 328–351.

Karlin, S., Zhu, Z.-Y. and Baud, F. (1999). Atom density in protein structures. *Proceedings of the National Academy of Sciences* **96**, 12500–12505.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* **5**, 1–25.

Lan, J., Ge, J., Yu, J., Shan, S., Zhou, H., Fan, S. et al. (2020). Structure of the SARS-CoV-2 spike receptor-binding domain bound to the ACE2 receptor. *Nature* **581**, 215–220.

Landau, L. D. and Lifshitz, E. M. (2013). *Statistical Physics*. Elsevier.

Li, Y., Wang, W., Deng, K. and Liu, J. S. (2022). Stratification and optimal resampling for sequential Monte Carlo. *Biometrika* **109**, 181–194.

Lin, M., Chen, R. and Liu, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science* **28**, 69–94.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association* **90**, 567–576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* **93**, 1032–1044.

Liu, J. S., Chen, R. and Logvinenko, T. (2001). A theoretical framework for sequential importance sampling with resampling. In *Sequential Monte Carlo Methods in Practice* (Edited by A. Doucet, N. de Freitas and N. Gordon), 225–246. Springer.

Liu, J. S., Chen, R. and Wong, W. H. (1998). Rejection control and sequential importance sampling. *Journal of the American Statistical Association* **93**, 1022–1031.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing* **11**, 125–139.

Nguyen, H. L., Lan, P. D., Thai, N. Q., Nissley, D. A., O'Brien, E. P. and Li, M. S. (2020). Does SARS-CoV-2 bind to human ACE2 more strongly than does SARS-CoV? *The Journal of Physical Chemistry B* **124**, 7336–7347.

Onuchic, J. N., Luthey-Schulten, Z. and Wolynes, P. G. (1997). Theory of protein folding: The energy landscape perspective. *Annual Review of Physical Chemistry* **48**, 545–600.

Pintar, A., Carugo, O. and Pongor, S. (2002). CX, an algorithm that identifies protruding atoms in proteins. *Bioinformatics* **18**, 980–984.

Simons, K. T., Kooperberg, C., Huang, E. and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of Molecular Biology* **268**, 209–225.

Wang, L., Wang, S. and Bouchard-Côté, A. (2020). An annealed sequential Monte Carlo method for Bayesian phylogenetics. *Systematic Biology* **69**, 155–183.

Williams, J. K., Wang, B., Sam, A., Hoop, C. L., Case, D. A. and Baum, J. (2022). Molecular dynamics analysis of a flexible loop at the binding interface of the SARS-CoV-2 spike protein receptor-binding domain. *Proteins: Structure, Function, and Bioinformatics* **90**, 1044–1053.

Wong, S. W., Liu, J. S. and Kou, S. (2017). Fast de novo discovery of low-energy protein loop conformations. *Proteins: Structure, Function, and Bioinformatics* **85**, 1402–1412.

Wong, S. W., Liu, J. S. and Kou, S. (2018). Exploring the conformational space for protein folding with sequential Monte Carlo. *Annals of Applied Statistics* **12**, 1628–1654.

Wrapp, D., Wang, N., Corbett, K. S., Goldsmith, J. A., Hsieh, C.-L., Abiona, O. et al. (2020). Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science* **367**, 1260–1263.

Zhang, J., Lin, M., Chen, R., Liang, J. and Liu, J. S. (2007). Monte Carlo sampling of near-native structures of proteins with applications. *Proteins: Structure, Function, and Bioinformatics* **66**, 61–68.

Zhou, R. and Berne, B. (1997). Smart walking: A new method for Boltzmann sampling of protein conformations. *The Journal of Chemical Physics* **107**, 9185–9196.

Zhaoran Hou

Department of Statistics and Actuarial Science, University of Waterloo, Ontario N2L 3G1, Canada.

E-mail: zhaoran.hou@uwaterloo.ca

Samuel W.K. Wong

Department of Statistics and Actuarial Science, University of Waterloo, Ontario N2L 3G1, Canada.

E-mail: samuel.wong@uwaterloo.ca

(Received October 2022; accepted October 2023)