Simulation-based inference with approximately correct parameters via maximum entropy

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 025006 https://doi.org/10.1088/2632-2153/ac6286

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
22 October 2021

REVISED
18 March 2022

ACCEPTED FOR PUBLICATION
30 March 2022

PUBLISHED
27 April 2022

PAPER

Simulation-based inference with approximately correct parameters via maximum entropy

Rainier Barrett[1], **Mehrad Ansari**[1], **Gourab Ghoshal**[2,3] and **Andrew D White**[1,*]

[1] Department of Chemical Engineering, University of Rochester, Rochester, NY, 14627, United States of America
[2] Department of Physics and Astronomy, University of Rochester, Rochester, NY, 14627, United States of America
[3] Department of Computer Science, University of Rochester, Rochester, NY, 14627, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: andrew.white@rochester.edu

Keywords: simulation-based inference, maximum entropy, likelihood-free, derivative-free

Supplementary material for this article is available online

Abstract

Inferring the input parameters of simulators from observations is a crucial challenge with applications from epidemiology to molecular dynamics. Here we show a simple approach in the regime of sparse data and approximately correct models, which is common when trying to use an existing model to infer latent variables with observed data. This approach is based on the principle of maximum entropy (MaxEnt) and provably makes the smallest change in the latent joint distribution to fit new data. This method requires no likelihood or model derivatives and its fit is insensitive to prior strength, removing the need to balance observed data fit with prior belief. The method requires the ansatz that data is fit in expectation, which is true in some settings and may be reasonable in all settings with few data points. The method is based on sample reweighting, so its asymptotic run time is independent of prior distribution dimension. We demonstrate this MaxEnt approach and compare with other likelihood-free inference methods across three systems: a point particle moving in a gravitational field, a compartmental model of epidemic spread and molecular dynamics simulation of a protein.

1. Introduction

Simulation-based inference (SBI) is a class of methods that infer the input parameters and unobservable latent variables in a simulator from observational data. SBI is different from traditional statistical inference or machine learning because simulators are typically not differentiable and their likelihoods are intractable. There have been great strides in methods for SBI and a recent review may be found in [1]. Most SBI methods are concerned with finding a few simulator parameters from a rich set of observations [2–4]. Here, we consider updating a simulator with many trusted parameters to match a sparse set of observations. The ancestor for this line of research is in molecular dynamics simulations of proteins. These simulations require thousands of parameters and the observed data (macroscopic experimental values) is often on the order of 10 to 100 data points (e.g. Reißer *et al* [5]). An approach that has emerged in molecular dynamics simulations is maximum entropy (MaxEnt) biasing [6–9]. MaxEnt biasing minimially modifies the simulator to match observations. The premise of MaxEnt is that the original model is approximately correct and observations should be matched in expectation, which is not the usual approach in SBI. These two assumptions lead to a unique bias [10] to the simulator that is independent of the parameters and can be implemented as a simple reweighting procedure. The MaxEnt method's run-time scales only with sample number, rather than the number of model parameters which is atypical of most SBI methods because they require joint sampling.

Our MaxEnt method reweights a black-box simulator to agree with observed data in a provably minimal way. The reweighted simulator can then be used to infer either better input parameters or other simulation outputs. The two conditions are that (i) the simulator is accurate enough that the observed data could have been derived from an average of runs of the simulator; and (ii) predicted values for the observed data can be

computed from the outcome of the simulator. The MaxEnt method results in an ensemble of outcomes from the simulator whose means agree with data and provide a regressed agreement to observed data while being as close to the original simulator outcomes as possible. The method is efficient, provides uncertainty estimates, and can account for unknown systematic errors.

This paper focuses on a setting where distribution moments (e.g. population average) are the data for fitting a posterior. This is a common setting of MaxEnt and it has a number of advantageous properties. Finding the MaxEnt posterior is equivalent to maximizing the likelihood function under a distribution family (exponential in this work) [11]. The MaxEnt posterior is the closest to the prior distribution (under KL divergence) under the constraint of fitting the population averages [12]. The MaxEnt posterior exactly fits the distribution moments under mild assumptions [10][4]. Examples of MaxEnt in this setting can be seen in statistical mechanics as described above, biology [13], natural language processing [11], and ecology [14]. Any application of maximum likelihood on distribution moments can be recast as MaxEnt. Our contribution to this setting is to summarize the general theory and provide an efficient and simple implementation that is system independent.

The second setting of MaxEnt is to make an ansatz that an observation can be substituted as a distribution moment. For example, consider observing a particle trajectory and we would like to make inferences about where the particle will go next. Our observations are specific pairs of time and position that describe the trajectory. If we have a good model for how the particle behaves, MaxEnt will minimally change the model to agree with the particle trajectory *on average*. The MaxEnt posterior will agree in expectation exactly with the observed trajectory. Thus, from a practical point of view, MaxEnt provides an accurate description of the data and a probability distribution for the posterior. The inferred continuation of the trajectory will come from the expectation of that posterior. An alternative would be treating the observation as exact and regressing the prior model, which would not give a posterior but instead a mode (most likely trajectory). Yet this gives no uncertainties with the predictions and can lead far away from the prior model, leading to issues like overfitting and covariate shift. Bayesian inference could be used to fit the particle trajectory by supposing a measurement error distribution. Yet this creates an over-constrained problem where a weak error distribution reduces the agreement with the observed trajectory and a strong error distribution reduces the agreement of the prior model. In essence, by 'relaxing' the observation to be an average we enable agreement with the observation exactly, maximize the agreement with the prior, and do not require potentially ad-hoc construction of error distributions[5]. This can be justified through the principle of maximum parsimony: the MaxEnt formulation requires the fewest input parameters. If multiple observations are gathered, then the Bayesian inference setting is more appropriate because the distribution moment ansatz would exclude information about variance in multiple observations. Another potential application area could be in few-shot regression with Bayesian network models [15], where only a few examples are available in a new task. MaxEnt provides a way to fit a previously trained Bayesian network to those few examples, balancing agreement with them exactly and while minimizing the effect on the trained model.

The MaxEnt method presented here has a run time scaling that is independent of the number of model/prior parameters; it acts entirely on samples. This also means that intractable or infinite dimensional priors (such as sampling both models/priors) can be treated with MaxEnt. This can be a large advantage over other approximate inference methods like approximate Bayesian computing (ABC) and likelihood free inference.

The MaxEnt approach in simulation can be traced to Jayne's early work on deriving statistical physics from MaxEnt [16]. It was shown, for example, that the Boltzmann distribution could be derived by simply adding a restraint on average energy that must be satisfied in expectation, analogous to matching an observation. A similar method of incorporating observations in expectation returned 50 years later in determining how to match protein molecular dynamics simulations to observations [17]. This method was then recast as an approximation to MaxEnt [12]. Matching observations in molecular dynamics with MaxEnt was also shown in Pitera and Chodera [10]. This was followed by rapid progress to create practical methods for use in simulations [9, 18–20]. The MaxEnt method based on reweighting has been presented in the context of molecular dynamics simulations in many forms over the years [5, 21–29]. MaxEnt-based methods have a long history of use in the molecular dynamics community across various types of systems, and this approach is still widely used for biasing applications in modern molecular simulations, demonstrating ongoing interest and engagement in the community [30–32]. A review by Bonomi *et al* provides broader context for the use of MaxEnt and other similar methods in the molecular simulation community [7]. The review by Cesari, Reißer and Bussi [33] provides an overview of the mathematics of MaxEnt, its connections

[4] Bayesian inference for fitting distribution moments requires specifying an error distribution that requires additional system insight and its strength relative to the prior belief affects the agreement with the distribution moment data. See system 1.

[5] Our MaxEnt formulation does allow uncertainty on distribution moments if desired.

to Bayesian inference and maximum likelihood, and some discussion of the potential hurdles involved. Also, for a comparative study weighing the benefits of MaxEnt and restraint-based methods, see Rangan *et al* 2018 [34].

Our contribution here is deriving a general MaxEnt framework that is applicable to arbitrary simulators, demonstrating its application to areas outside of molecular dynamics, and showing one method of improving the support (sampling) of the posterior, which is important when the simulator is far from the observations. In the remainder of this work, we develop the theory, discuss sampling issues, and compare the MaxEnt method to ABC [3, 35–37], sequential neural likelihood (SNL) [38], and direct Bayesian inference when the likelihood is tractable. Additional background on these methods used for comparison can be found in the supporting information (available online at stacks.iop.org/MLST/3/025006/mmedia).

2. Theory

Given a simulator $f(\vec{\theta})$ with a set of parameters $\vec{\theta}$, we have a prior distribution of parameters $P(\vec{\theta})$. For example, the function $f(\vec{\theta})$ could be propagating a system of ODEs for some set number of timesteps or a molecular dynamics simulation with intrinsic noise.

Suppose we have some set of N observations, $\{\bar{g}\}_k$, $k \in [1, \ldots, N]$, which we would like to match with our model. Assume the measurement of each $\bar{g}_k$ has some uncertainty ε_k, where ε_k is a random variable distributed according to some prior distribution about uncertainty, $P_0(\epsilon_k)$. We would like to constrain our model such that

$$\int \mathrm{d}\vec{\theta}\mathrm{d}\vec{\epsilon} P'(\vec{\theta}) P_0(\epsilon_k) \left(g_k[f(\vec{\theta})] + \epsilon_k\right) = E[g_k + \epsilon_k] = \bar{g}_k \; \forall k. \tag{1}$$

This means that we want the average over the distribution of our updated models ($P'(\vec{\theta})$) to match the observations data, with an allowable average disagreement based on $\{\epsilon_k\}$. This is an unusual constraint and is weaker than most simulation inference methods. It reflects the strong belief in our prior model in this setting. Note that inclusion of the $P_0(\epsilon_k)$ and ε_k terms is optional: it is not necessary to allow disagreement on average with data, unlike in a Bayesian framework. This would be equivalent to setting the error distribution to a Dirac delta about 0: $P_0(\epsilon_k) = \delta(\epsilon_k = 0)$. Another difference is that this distribution of uncertainty is about bias. It accounts for systemic deviation in average agreement and does not describe the underlying variance of the observational data. This approach is analogous to Bayesian model averaging [39], in that it is an average over many model parameter settings, reweighted by the posterior likelihood.

The MaxEnt modification to the prior distribution $P(\vec{\theta})$ to satisfy the N constraints is given by [9, 10, 12, 40]:

$$P'(\vec{\theta}, \vec{\epsilon}) = \frac{1}{Z'} P(\vec{\theta}) \prod_k^N e^{-\lambda_k g_k[f(\vec{\theta})]} e^{-\lambda_k \epsilon_k} P_0(\epsilon_k), \tag{2}$$

$$Z' = \int \mathrm{d}\vec{\theta}\mathrm{d}\vec{\epsilon} P(\vec{\theta}) P_0(\epsilon) e^{-\sum_k \lambda_k (g[f(\vec{\theta})] + \epsilon_k)}, \tag{3}$$

where $Z^{'}$ is a normalization constant and λ_k are chosen such that $E[g_k + \epsilon_k] = \bar{g}_k$. The dependence on $\vec{\epsilon_k}$ can be removed by computing the marginal,

$$P'(\vec{\theta}) = \int \mathrm{d}\vec{\epsilon} P'(\vec{\theta}, \vec{\epsilon}) = \frac{1}{Z'} P(\vec{\theta}) \prod_k^N e^{-\lambda_k g_k[f(\vec{\theta})]} \int \mathrm{d}\epsilon_k e^{-\lambda_k \epsilon_k} P_0(\epsilon_k). \tag{4}$$

The problem is reduced to finding λ_k such that the constraint is satisfied. Again, we must remove $\vec{\epsilon_k}$ from $E[g_k] + E[\epsilon_k] = \bar{g}_k$, where $E[\epsilon_k]$ is:

$$E[\epsilon_k] = \frac{\int \mathrm{d}\epsilon_k e^{-\lambda_k \epsilon_k} P_0(\epsilon_k) \epsilon_k}{\int \mathrm{d}\epsilon_k e^{-\lambda_k \epsilon_k} P_0(\epsilon_k)}, \tag{5}$$

and is understood to still be a function of λ_k. If we define $\xi_k(\lambda_k) = E[\epsilon_k]$ the constraint equation can be rewritten as $E[g_k] + \xi_k(\lambda_k) = \bar{g}_k$. If the prior is an exponential family, the λ_ks will exist and be unique under some mild assumptions about support of the prior and covariance of observables (i.e. cannot have perfectly correlated observables with incompatible observations)[10, 12].

Algorithm 1. MaxEnt weights with uncertain observations

Input $P(\vec{\theta})$, $f(\vec{\theta})$, M, N of $P_0(\epsilon_k)$, g_k, $\bar{g}_k$, η
Initialize $\lambda_k = 0 \forall k$
for $i \leftarrow 1$ **to** M **do**
  Sample $\vec{\theta}_i \sim P(\vec{\theta})$
  for $k \leftarrow 1$ **to** N **do**
    Evaluate $g_k[f(\vec{\theta}_i)]$
  end
end
While $\sum_i w_i g_k[f(\vec{\theta}_i)]/\sum_i w_i + \xi_k(\lambda_k) \neq \bar{g}_k$ for any k **do**
  for $i \leftarrow 1$ **to** M **do**
    for $k \leftarrow 1$ **to** N **do**
      $\lambda_k \leftarrow \lambda_k - \eta \frac{\partial}{\partial \lambda_k} \left(\sum_l \left(\bar{g}_l - \left[g[f(\vec{\theta}_i)] w_i / \sum_j w_j + \xi_l(\lambda_l) \right] \right)^2 \right)$
      where $w_i = \prod_k e^{-\lambda_k g_k[f(\vec{\theta}_i)]} \int \mathrm{d}\epsilon_k e^{-\lambda_k \epsilon_k} P_0(\epsilon_k)$
    end
  end
end
return $\vec{w}$

2.1. Computing weighted properties and sampling efficiency

In algorithm 1, we show the procedure for sampling from the MaxEnt distribution defined in equation (4) via importance sampling [41]. Here, $P(\vec{\theta})$ is the prior distribution over simulation parameters $\vec{\theta}$, f is the simulator, M is the number of samples from the prior to take (and hence the number of weights to be computed), N is the number of constraints, $P_0(\epsilon_k)$ is the error distribution, g_k is the kth observation, and $\bar{g}_k$ is the target observable value to which we would like to constrain our simulator. η is the learning rate. Note that the loop over M can be batched, as all samples of model parameters are independent. The output of this algorithm, $\{w_i\}$, are the weights of trajectories $\{f(\theta_i)\}$, and any desired property g can be computed as $\sum_i g[f(\vec{\theta}_i)] w_i / \sum_i w_i$.

The challenge of using MaxEnt is sampling from $P'(\vec{\theta})$. Our assumption thus far is that our prior $P(\vec{\theta})$ is approximately correct, so that samples from $P(\vec{\theta})$ should be similar to $P'(\vec{\theta})$. In this ideal case, the algorithm is simply a matter of reweighting. One samples $\vec{\theta}_i$, computes $f(\vec{\theta}_i)$, compute weights proportional to $w_i[P'] = \prod_k^N e^{-\lambda_k g_k[f(\vec{\theta})]}$ consistent with the experimental data (algorithm 1), and then any other property is reweighted with the same weights. In the non-ideal case (if for instance sampling is expensive, the space is high-dimensional, or the model is far from correct), there can be insufficient support to agree with the constraints. To treat insufficient support, we take a simple approach and use gradient descent to modify the sampling distribution parameters $\vec{\theta}^j$ to minimize the cross-entropy with $P'(\vec{\theta})$:

$$\vec{\theta}^{j+1} = \vec{\theta}^j - \eta \nabla_{\vec{\theta}^j} \sum_i w_i[P'] \ln P(\vec{\theta}_i), \tag{6}$$

where $w_i[P']$ depends on $\vec{\theta}^j$ via the expectation function. We remove the effect of the sampling distribution from the posterior via reweighting by $P(\vec{\theta})/P(\vec{\theta}^j)$ We refer to this approach as *variational*.

The good efficiency of MaxEnt is because samples from the prior and evaluation of the observations g_k can be done *once*, as can be seen in the separate loop at the beginning of algorithm 1. This reduces the evaluation of $g_k[f(\vec{\theta}_i)]$ to a table lookup. The asymptotic runtime complexity of the fitting loop of MaxEnt is thus $O(MNZ)$, where Z is the number of fitting steps required to reach convergence, which is unknown *a priori*. A maximum number of training iterations can be specified, and as noted previously, the M inner loops can be unrolled and performed concurrently, because samples from the prior are independent.

We will compare briefly with similar methods to give context, but recall that these methods have different assumptions/objectives and so it is not relevant to claim one is clearly more efficient than another. The SNL algorithm by Papamakarios *et al* [42] re-samples parameters for each iteration of training using a Markov process, running a new simulation for each sample, and trains a neural network to estimate the posterior using a dataset consisting of the cumulative sampled parameters and simulator results from each iteration. This precludes the ability to sample *a priori*, resulting in an asymptotic runtime complexity of $O(RN\log N) \times O(f) \times O(S)$, where R is the number of training rounds, N is the number of simulations per round, $O(f)$ is the simulator's runtime, and $O(S)$ is the runtime of the Markov chain Monte Carlo parameter sampling step.

The ABC algorithm also precludes *a priori* sampling. It employs several simulations per iteration, each with parameters drawn from some prior distribution, which are iteratively updated until they fall within some tolerance (ε) of the observation(s). This bias of the ABC estimate has been shown to be asymptotically proportional to $O(\epsilon^2)$ as ε decreases [43]. Thus, the runtime complexity becomes $O(RN) \times O(f) \times O(\epsilon^2)$, where again R is the number of iterations, $O(f)$ is the runtime complexity of the simulator f, and N is the number of simulations evaluated each round based on the number of samplings.

3. Methods

Here we present detailed descriptions of the methods used for each of the example systems described in the Results section.

3.1. Point particle gravitation simulation

For this simulation, the prior parameter distribution was taken as a multivariate normal distribution centered at $\{m_1 = 85, m_2 = 40, m_3 = 70, v_{0x} = 12, v_{0y} = -30\}$, with covariance matrix $\mathbf{I} \times 50$. This wide prior was chosen because MaxEnt needs parameter support that overlaps with the observations we would like to fit. Fitting was done using the SBI package for Python [44] with the SNL method, [38] and a custom implementation of MaxEnt reweighting using Keras [45, 46]. Both methods used 2048 prior samples for fitting. SNL used default parameters from the SBI package [44] and MaxEnt used the Adam optimizer [47] with a learning rate of 0.0001 with mean squared error for 30 000 epochs and batch size 2048.

3.2. Epidemiology modeling

Epidemic spreading in networks can be modeled as a reaction-diffusion process. The reaction corresponds to an infection caused by interactions of subjects within a fully-mixed region or patch of varying granularities (a meta-population), while diffusion corresponds to movement of people (of various infection states) between patches [48]. In this example, the meta-population system is comprised of three isolated local populations (patches) connected via flows corresponding to migrating individuals. The spreading process is represented through a temporally discretized ODE that includes the spatial distribution of the population as well as their mobility patterns [49].

In our simulation, the infection begins in patch 1, propagating to the other two patches according to a synthetic mobility matrix. This mobility matrix was randomly generated with dominating diagonal elements to satisfy the fully-mixed region assumption. Five uniformly random data points within the first half of the trajectory of the compartment **I** in patch 1 were considered as observations with a 5% random additive noise and Laplace prior of 0.01 (shown as restraints in figure 4(a). The true parameters for the reference epidemic trajectory are: $\{start_I = 0.02, start_A = 0.05, E_{period} = 7, A_{period} = 5, I_{period} = 14\}$. Parameters for this simulation were asymptomatic, infectious and exposed periods along with the fractional starting values for **I** and **A** compartments. The prior parameter distribution were taken as a truncated-normal distribution centered at $\{start_I = 0.001, start_A = 0.001, E_{period} = 2, A_{period} = 2, I_{period} = 10\}$, with variances of $\{0.8, 0.8, 1, 4, 5\}$, respectively. For the simulation, the pyABC [50] package was used with default parameters, and the same MaxEnt implementation was used with the Adam optimizer, a learning rate of 0.1, and loss of mean squared error for 1000 epochs with a batch size of 8192.

3.3. MBP fragment molecular dynamics

Molecular dynamics was done with Gromacs 2020.03 [51–57] as driven by GromacsWrapper [58] with a timestep of 2 fs. Myelin basic protein (MBP) initial structure were generated with PeptideBuilder [59] and Packmol [60]. The CHARMM27 forcefield was used for [61, 62]. Canonical Sampling through Velocity Rescaling thermostat was used [63]. Long-range electrostatic forces were calculated with the particle mesh Ewald method [64]. Shifted Van der Waals and short-range electrostatics were used with a cutoff distance of 1 nm. Hydrogen containing covalent bonds were constrained using the LINear Constraint Solver algorithm [65]. MaxEnt implementation as described above was used with 500 epochs in Adam optimizer with learning rate of 0.1.

4. Results

4.1. Trivial simulation with gaussian noise

We first consider a toy simulator f that outputs a scalar r. We have a prior belief about the value of the constant as a normal distribution $\mathcal{N}(\hat{r}, \theta)$. This example serves to compare the MaxEnt approach with Bayesian inference. The observed data is a single point ($\bar{r}$) and we treat it as an average constraint in the MaxEnt. That is, we have a single observation and we constrain our simulator to on average match this

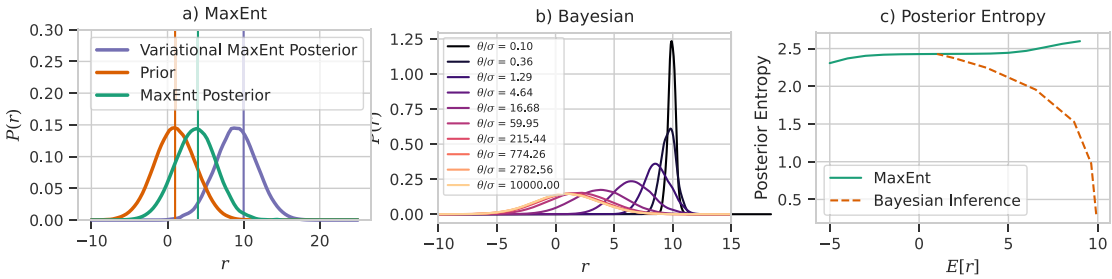

Figure 1. Comparison of Bayesian inference and MaxEnt reweighting. Panel (a) shows the simulator prior distribution in orange and the two versions of MaxEnt posterior with observations of 5 and 10. Panel (b) shows the interplay between strength of prior and assigned uncertainty to the observation at 10 for Bayesian inference. The value is arbitrary and chosen to illustrate how Bayesian inference strongly alters the shape of the posterior compared to the prior, whereas MaxEnt preserves the shape well. Note the scale change between panels (a) and (b). Panel (c) further illustrates this by comparing the posterior entropy of the two as a function of the observation location.

observation. Figure 1 panel (a) shows how the MaxEnt posterior changes with different observations ($\bar{r} = 5$ or $\bar{r} = 10$). The $r = 10$ observation requires the variational sampling (equation (6)) because the observed value is outside the sampled support of the prior. The expected value of $E[r]$ of the posterior always matches the observation and the moments of the posterior are identical to the prior, except the 1st moment (the mean). Although figure 1(a) is calculated with algorithm 1, the analytic equation for the posterior is simply $\mathcal{N}(\bar{r}, \theta)$ [10].

With Bayesian inference, we must assume some noise model of our simulator so that we can compute the probability of the single observation arising from the simulator, namely $P(\text{data}|\text{model})$ [66]. We take this to be $\mathcal{N}(\hat{r}, \sigma)$. The Bayesian posterior balances this evidence with the prior distribution:

$$P_B(r|\,\bar{r}) = \frac{1}{Z} e^{-\frac{(r-\bar{r})^2}{2\sigma}} e^{-\frac{(r-\hat{r})^2}{2\theta}}. \tag{7}$$

The expected value of $\hat{r}$ will not match the observed value except in the limit of σ/θ reaching 0. $E[\hat{r}]$ will be between the observed value and the prior belief expectation. Figure 1(b) compares the Bayesian inference and MaxEnt posteriors. Panel (a) shows how the MaxEnt method leaves the variance of $\hat{r}$ unchanged as we consider different observed values. Panel (b) shows the use of Bayesian inference to match the observation at $r = 10$. It requires extreme ratios between prior belief and experimental uncertainty to match the observation at 10. This is not necessarily a disadvantage, we simply are showing that observations are matched in expectation with MaxEnt and not with Bayesian inference. Panel (c) shows how the MaxEnt method keeps the posterior entropy maximized regardless of the observation value (x-axis), as expected for a MaxEnt method. Bayesian inference shows a more peaked distribution when the observed value is far away from the prior, giving less entropy. That is, to agree with the observation we must necessarily increase the strength of evidence, which peaks the posterior.

4.2. Example system 1: point particle gravitation simulation

For our first example system, figure 2 shows a comparison of SNL and MaxEnt reweighting on a unit mass particle in a gravitational field of three attractors. The simulator here is a point particle following Newtonian gravitational mechanics. The goal here is to modify the simulation trajectories to align with a small set of observations. An example task might be fitting the trajectory of a comet to a small number of observations separated by years.

The parameters for this simulation were m_1, m_2, m_3, v_{0x} and v_{0y}, the masses of the three attractors, and the initial velocity of the particle, respectively. The positions of the attractors and the initial position of the particle were all fixed. We treat these parameters as unknown, and the prior belief for them follows a normal distribution, shown in figure 2. Repeatedly sampling from this prior and running the simulator results in a distribution of trajectories, whose means are shown in figure 2(a). MaxEnt reweights this ensemble of trajectories to agree with five observed positions along the trajectory. (The mean path does not exactly pass through the observations because some zero-mean normally-distributed noise with standard deviation of 3 was added to each observation.) The average posterior trajectory indeed agrees with all observations. The prior and posterior for the parameters are shown in figure 2(b).

The observed points were synthesized by choosing a set of true parameter values and imposing zero-mean normally-distributed noise with standard deviation 3 on every 20th timestep on the 100-step simulation. Thus, one way to evaluate the MaxEnt performance is to see if the posterior means are close to

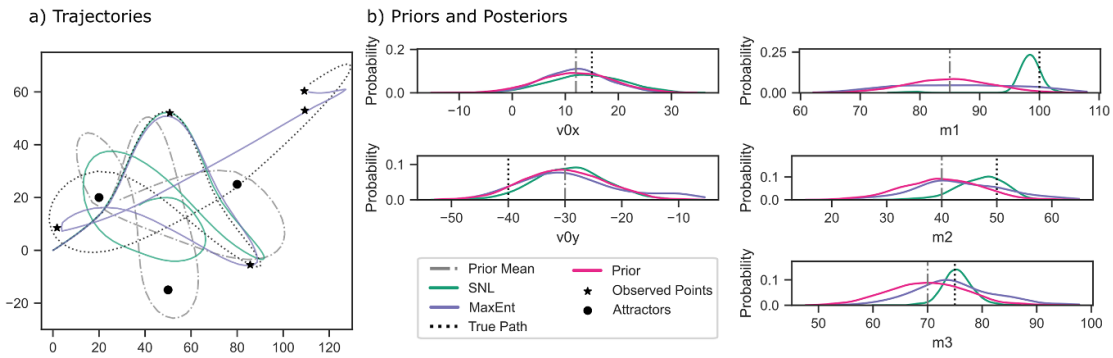

Figure 2. Comparison of SNL and MaxEnt methods on a gravitational field simulation of a particle moving through a fixed field with three attractors. All units are SI values (m, m s^{-1}, kg). (a) weighted mean paths generated by SBI with SNL (green) and MaxEnt (purple), alongside the path generated by the mean of the prior distribution (dash-dotted grey), and the true path used to generate observations (dashed black). Target points appear as black stars, and the attractors are black circles. (b) Kernel density estimate of the posterior distribution of parameters after fitting, alongside their respective priors.

these true values. We can see that the MaxEnt posteriors are closer to these values than the prior, but still largely in agreement with the prior. It fits the observations while staying as close to the prior as possible, because that maximizes entropy. In contrast, SNL results in a much narrower posterior around the true values, while diverging from the prior, because that maximizes likelihood.

We computed the cross-entropy of the prior and posterior produced by MaxEnt and SNL. These values were 5.09 for SBI with SNL, and 3.43 for MaxEnt reweighting. This demonstrates how MaxEnt minimally alters the prior distribution while still matching observations in expectation—the average path followed by the MaxEnt particle matches all target points, while matching the posterior to the prior's shape more closely than SNL.

This example illustrates two key points. Firstly, it shows that MaxEnt is robust to chaotic systems, as it is still able to match observations on average, with minimal change to the prior. However, it is also an example of when another SBI method may be preferable, depending on the goal. The goal of MaxEnt is, by construction, to alter the prior as little as possible while agreeing with observations on average. In cases where the true underlying parameters governing a model are in a low-density region of the prior, the posterior resultant from MaxEnt will therefore assign relatively low probability to these parameter values as well. Thus, in the sense of estimating likelihood when the prior is not close to the true values, other methods like SNL can be preferable to MaxEnt. We can see that while SNL makes a better estimate of the true parameters used to generate those observations, it does not reproduce a path that aligns with the observations. This presents a choice to the simulator. If the goal is to alter a model to agree with observations, MaxEnt is preferred, especially if the prior is strongly trusted. If the goal is accurate likelihood estimation, methods like SNL are preferred, especially if the prior is not strongly trusted.

4.3. Example system 2: epidemiological modeling

In our third example, we apply our framework to modeling the spread of a pathogen in vulnerable populations. We consider an SEAIR compartmental model of epidemic spread (figure 3) on metapopulations connected via a spatial network of patches. Each patch corresponds to a location such as a zipcode in a city, or a county, and connections between patches correspond to mobility flows of residents encoded in a $M \times M$ mobility matrix for M patches, where M_{ij} is the number of people moving from patch i to patch j in one time increment. Contacts within patches occur in a fully-mixed mean field manner where individuals can be in any one of five states of infection: Susceptible (**S**), Exposed (**E**), Asymptomatic (**A**), Infected (I), and Resolved (**R**). The choice for this particular combination of compartments was inspired by its relevance in modeling the evolution of the current SARS-CoV-2 pandemic [67, 68]. Each individual patch is represented with fractions of **S, E, A, I, R**, rather than the count of individuals within each compartment.

We first create a 'reference' trajectory that represents the true disease model. From this reference trajectory, we extract observations which are used as the input to the MaxEnt methods, by extracting values at specific timepoints in the reference trajectory. A challenge in modeling the spread of epidemics is associated with reporting of the empirical number of confirmed cases (compartment **I**), which is typically very noisy [69]. To simulate this uncertainty, we add random additive noise to the observations from the reference trajectory (see Methods for details). This reference trajectory is represented as dashed lines in figure 4(a). We choose 5 uniformly random data points within the first half of the trajectory of the compartment **I** in patch 1 as observations (represented as black dots). The performance of the model is evaluated by comparing the

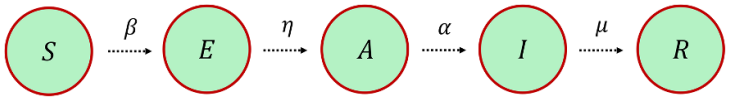

Figure 3. SEAIR model. Populations in each patch can be in any one of Susceptible (**S**), Exposed (**E**), Asymptomatic (**A**), Infected (**I**) and Resolved (**R**). Susceptible individuals can get exposed to the disease by having contacts with the asymptomatic or the infected at infectivity rate β. Once exposed, they become asymptomatic and infected at rates η and α. The infected finally recovers or dies at rate μ and becomes resolved.

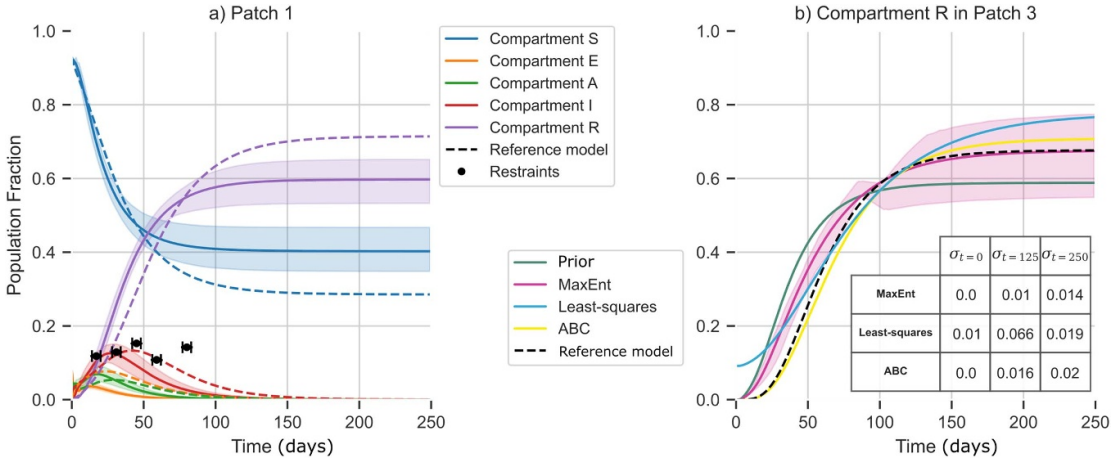

Figure 4. MaxEnt reweighting of disease trajectory in a meta-population SEAIR model. (a) Prior-generated trajectory in one of the spatial patches for compartments S (blue), E (orange), A (green), I (red) and R (purple) are shown with solid lines and the reference trajectory is in dashed lines. The colored area represents the one-third higher and lower quantiles than average. Observations shown as restraints (black circles) are selected randomly from compartment I with 5% additive noise and Laplace prior of 0.01. (b) Comparing the performance of MaxEnt (pink), Least-squares (blue), ABC (yellow) in fitting to reference model (black dashed line) in patch 3, based on observations in patch 1. Table inset shows standard deviations from five-fold cross validation of the observations at three different times. Shading on MaxEnt is from $\pm 67\%$ weighted quantiles.

predicted trajectory and the reference in a different patch (3). In figure 4(b) we compare the performance of MaxEnt, a least-squares fit, and ABC in fitting the prior to the observations. Compared to MaxEnt, the result from the least squares method was a poor fit with high variance, as it over-fits to observation noise. This was shown by doing a five-fold leave-one-out cross-validation of the observations and evaluating the standard deviation at times $t = 0, 125$ and 250 for each method (inset in figure 4(b)). Out of all methods evaluated, ABC had the least variance, but was computationally more expensive to run, whereas MaxEnt can include more model parameters without additional computational cost. Variational MaxEnt was also implemented to reweight the disease trajectory (See details in supporting information figure S2).

4.4. Example system 3: MBP fragment molecular dynamics

Finally, we consider an application from biophysical modeling of the MBP epitope fragment. MBP is a common autoimmune target for the disease multiple scelrosis [70]. Spyranti *et al* [71] characterized the specific region of MBP (83–99) that is the binding epitope for T-cell receptor recognition with solution nuclear magnetic resonance (NMR). NMR provides per-atom chemical shifts, which are population averages of a measurement of an atoms' local environment [72]. However, we must infer a specific structure to understand the molecular biology of MBP. In this example we use molecular dynamics as a prior model, the chemical shifts as MaxEnt restraints, and compute a posterior of protein configurations. MaxEnt analysis has been applied frequently already in molecular dynamics, although not this exact approach with uncertainty [34].

Our prior model is an empirical distribution consisting of MBP fragment atomic positions as sampled from molecular dynamics. The specific fragment sequence was ENPVVHFFKNIVTPRTP and the molecular dynamics was initialized from an extended conformation. Simulations was performed in NVT ensemble in Gromacs 2020 [51–57] with CHARMM27 force field at a density of 25 mg ml^{-1} [61, 62]. The simulation duration was 1.3 μs with frames saved for this analysis every 500 ps. To compute the chemical shift, g_k, we use a graph neural network that can compute chemical shift from atomic positions [73]. We only biased backbone HN atoms, due to their higher accuracy [73]. The first 6 HN atoms were biased (NPVVHF),

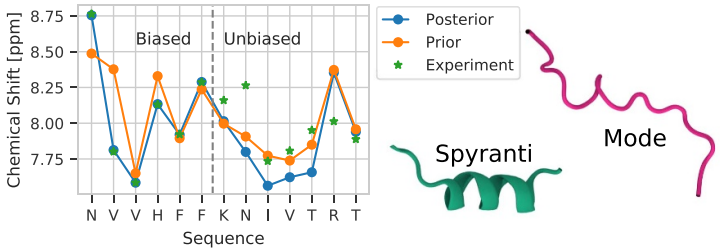

Figure 5. MaxEnt reweighting of protein molecular dynamics simulation. Molecular dynamics simulation of MBP epitope fragment was used to generate prior. Prior was reweighted with MaxEnt to have agreeing chemical shifts at indicated sequence positions. Chemical shifts are HN, as predicted by graph neural network [73]. Experiments from Spyranti *et al* [71]. Mode is most weighted structure after MaxEnt. Spyranti is from NMR structure refinement.

excluding the N-terminus. The remaining HN atom chemical shifts were unbiased (KNIVTRT). $P_0(\epsilon)$ was chosen to be a Laplace distribution with scale parameter 0.05—allowing a small amount of systematic disagreement.

Figure 5 shows the MaxEnt posterior average chemical shifts. As expected, exact agreement is found for the chemical shifts for which observations are provided. The posterior for which there are no observations follow the the prior closely (as expected in MaxEnt), although they move in the wrong direction at some positions. The protein structures are shown to the right of the plot. 'Spyranti' is a representative structure from the deposited structure constructed by Spyranti *et al* [71]. The posterior mode from MaxEnt is shown to the right. It has a helix, though not the α-helix as shown in Spyranti. An advantage of this MaxEnt approach to analyzing NMR data is that there are 6500 structures in the posterior, whereas the traditional approach of NMR structure refinement results in 5–20 structures. This large distribution can then be used for other tasks with better calibration, such as finding drugs to target the protein structure, predicting protein-protein interfaces, and assessing structural properties.

5. Discussion

We have presented MaxEnt reweighting as an inference method for altering an approximately-correct simulator to agree with observations. This method can be used on arbitrary simulators with arbitrary numbers of parameters, requiring only sufficient sampling of the prior distribution. The simulator need not have derivatives or tractable likelihoods. We demonstrated this by comparing with other SBI methods using three different simulators in different example contexts. The framework is particularly effective and robust when data is scarce or expensive (epidemic spreading being an archetypal example). MaxEnt provably changes the prior minimally to fit observations. While the method was initially developed for and particularly well-suited for molecular dynamics simulations—where experimental observations are much more costly and few in number compared to simulation—as demonstrated here, its applicability has potential for use in any setting of stochastic modeling where the derivative of the simulator's output with respect to latent variables is unavailable or intractable.

The approach to sampling described here is an implementation of variational inference to sample from the posterior. One could instead use Monte Carlo sampling. This would have the advantage of not requiring prior distribution derivatives, but since the derivatives here are closed-form it is computationally convenient to use importance sampling. MaxEnt's advantages over other widely used SBI methods, such as SNL, are that it is simple to implement, requires no hyperparameter choices like a neural network design, and can fit observations in expectation.

Data availability statement

Code for this work is available at https://github.com/ur-whitelab/maxent. The SEAIR model implementation used in this work is publicly available as a python package at https://github.com/ur-whitelab/py0.

Acknowledgment

Funding for this research was provided by National Science Foundation under Grant No. 2029095.

ORCID iDs

Rainier Barrett https://orcid.org/0000-0002-5728-9074
Mehrad Ansari https://orcid.org/0000-0001-5696-9193
Gourab Ghoshal https://orcid.org/0000-0001-7593-5626
Andrew D White https://orcid.org/0000-0002-6647-3965

References

[1] Cranmer K, Brehmer J and Louppe G 2020 The frontier of simulation-based inference *Proc. Natl Acad. Sci.* **117** 30055–62
[2] Rubin D B 1984 Bayesianly justifiable and relevant frequency calculations for the applied statistician *Ann. Stat.* 12 1151–72
[3] Beaumont M A, Zhang W and Balding D J 2002 Approximate Bayesian computation in population genetics *Genetics* **162** 2025–35
[4] Diggle P J and Gratton R J 1984 Monte Carlo methods of inference for implicit statistical models *J. R. Stat. Soc.* B **46** 193–212
[5] Reißer S, Zucchelli S, Gustincich S and Bussi G 2020 Conformational ensembles of an RNA hairpin using molecular dynamics and sparse NMR data *Nucleic Acids Res.* **48** 1164–74
[6] Sormanni P *et al* 2017 Simultaneous quantification of protein order and disorder *Nat. Chem. Biol.* **13** 339–42
[7] Bonomi M, Heller G T, Camilloni C and Vendruscolo M 2017 Principles of protein structural ensemble determination *Curr. Opin. Struct. Biol.* **42** 106–16
[8] Olsson S, Frellsen J, Boomsma W, Mardia K V, Hamelryck T and Fernandez-Fuentes N 2013 Inference of structure ensembles of flexible biomolecules from sparse, averaged data *PLoS One* **8** e79439
[9] Amirkulova D B and White A D 2019 Recent advances in maximum entropy biasing techniques for molecular dynamics *Mol. Simul.* **45** 1285–94
[10] Pitera J W and Chodera J D 2012 On the use of experimental observations to bias simulated ensembles *J. Chem. Theory Comput.* **8** 3445–51
[11] Berger A, Della Pietra S A and Della Pietra V J 1996 A maximum entropy approach to natural language processing *Comput. Linguist.* **22** 39–71
[12] Roux Bît and Weare J 2013 On the statistical equivalence of restrained-ensemble simulations with the maximum entropy method *J. Chem. Phys.* **138** 02B616
[13] De Martino A and De Martino D M 2018 An introduction to the maximum entropy approach and its application to inference problems in biology *Heliyon* **4** e00596
[14] Banavar J R, Maritan A and Volkov I 2010 Applications of the principle of maximum entropy: from physics to ecology *J. Phys.: Condens. Matter.* **22** 063101
[15] Wilson A G and Izmailov P 2020 Bayesian deep learning and a probabilistic perspective of generalization (arXiv:2002.08791)
[16] Jaynes E T 1957 Information theory and statistical mechanics *Phys. Rev.* **106** 620
[17] Islam S M, Stein R A, Mchaourab H S and Roux Bît 2013 Structural refinement from restrained-ensemble simulations based on epr/deer data: application to t4 lysozyme *J. Phys. Chem.* B **117** 4740–54
[18] Cavalli A, Camilloni C and Vendruscolo M 2013 Molecular dynamics simulations with replica-averaged structural restraints generate structural ensembles according to the maximum entropy principle *J. Chem. Phys.* **138** 03B603
[19] Boomsma W, Ferkinghoff-Borg J, Lindorff-Larsen K and Levitt M 2014 Combining experiments and simulations using the maximum entropy principle *PLoS Comput. Biol.* **10** e1003406
[20] White A D and Voth G A 2014 Efficient and minimal method to bias molecular simulations with experimental data *J. Chem. Theory Comput.* **10** 3023–30
[21] Beauchamp K A, Pande V S and Das R 2014 Bayesian energy landscape tilting: towards concordant models of molecular ensembles *Biophys. J.* **106** 1381–90
[22] Różycki B, Kim Y C and Hummer G 2011 SAXS ensemble refinement of Escrt-III Chmp3 conformational transitions *Structure* **19** 109–16
[23] Leung H T A, Bignucolo O, Aregger R, Dames S A, Mazur A, Berneche S and Grzesiek S 2016 A rigorous and efficient method to reweight very large conformational ensembles using average experimental data and to determine their relative information content *J. Chem. Theory Comput.* **12** 383–94
[24] Choy W-Y and Forman-Kay J D 2001 Calculation of ensembles of structures representing the unfolded state of an SH3 domain *J. Mol. Biol.* **308** 1011–32
[25] Bernadó P, Mylonas E, Petoukhov M V, Blackledge M and Svergun D I 2007 Structural characterization of flexible proteins using small-angle x-ray scattering *J. Am. Chem. Soc.* **129** 5656–64
[26] Berlin K, Castaneda C A, Schneidman-Duhovny D, Sali A, Nava-Tudela A and Fushman D 2013 Recovering a representative conformational ensemble from underdetermined macromolecular structural data *J. Am. Chem. Soc.* **135** 16595–609
[27] Bertini I, Giachetti A, Luchinat C, Parigi G, Petoukhov M V, Pierattelli R, Ravera E and Svergun D I 2010 Conformational space of flexible biological macromolecules from average data *J. Am. Chem. Soc.* **132** 13553–8
[28] Pelikan M, Hura G L and Hammel M 2009 Structure and flexibility within proteins as identified through small angle x-ray scattering *Gen. Physiol. Biophys.* **28** 174–189
[29] Shaw D E *et al* 2010 Atomic-level characterization of the structural dynamics of proteins *Science* **330** 341–6
[30] Bottaro S, Bengtsen T and Lindorff-Larsen K 2020 Integrating molecular simulation and experimental data: a Bayesian/maximum entropy reweighting approach *Method. Mol. Biol.* **2112** 219–40
[31] Bradshaw R T, Marinelli F, Faraldo-Gómez Je D and Forrest L R 2020 Interpretation of HDX data by maximum-entropy reweighting of simulated structural ensembles *Biophys. J.* **118** 1649–64
[32] Lou H and Cukier R I 2018 Reweighting ensemble probabilities with experimental histogram data constraints using a maximum entropy principle *J. Chem. Phys.* **149** 234106
[33] Cesari A, Reißer S and Bussi G 2018 Using the maximum entropy principle to combine simulations and solution experiments *Computation* **6** 15
[34] Rangan R, Massimiliano B, Heller G T, Cesari A, Bussi G and Vendruscolo M 2018 Determination of structural ensembles of proteins: restraining vs reweighting *J. Chem. Theory Comput.*
[35] Blum M G B and Tran C 2010 HIV with contact tracing: a case study in approximate Bayesian computation *Biostatistics* **11** 644–60

[36] Toni T, Welch D, Strelkowa N, Ipsen A and Stumpf M P H 2009 Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems *J. R. Soc. Interface* **6** 187–202
[37] Kypraios T, Neal P and Prangle D 2017 A tutorial introduction to Bayesian inference for stochastic epidemic models using approximate Bayesian computation *Math. Biosci.* **287** 42–53
[38] Papamakarios G, Sterratt D C, and Murray I 2019 Sequential neural likelihood: fast likelihood-free inference with autoregressive flows (arXiv:1805.07226)
[39] Gordon A and Izmailov W P 2020 Bayesian deep learning and a probabilistic perspective of generalization (arXiv:2002.08791v3)
[40] Cesari A, Gil-Ley A and Bussi G 2016 Combining simulations and solution experiments as a paradigm for rna force field refinement *J. Chem. Theory Comput.* **12** 6192–200
[41] Tokdar S T and Kass R E 2010 Importance sampling: a review *Wiley Interdiscip. Rev.-Comput. Stat.* **2** 54–60
[42] Papamakarios G, Sterratt D C and Murray I 2018 Sequential neural likelihood: fast likelihood-free inference with autoregressive flows *AISTATS 2019-22nd Int. Conf. on Artificial Intelligence and Statistics*
[43] Barber S, Voss J and Webster M 2015 The rate of convergence for approximate Bayesian computation *Electron. J. Stat.* **9** 80–105
[44] Tejero-Cantero A, Boelts J, Deistler M, Lueckmann J-M, Durkan C, Gonçalves P J, Greenberg D S and Macke J H 2020 SBI: a toolkit for simulation-based inference *J. Open Source Softw.* **5** 2505
[45] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems Software available from tensorflow.org
[46] keras Fçois C 2015 (available at: https://github.com/fchollet/keras)
[47] Kingma D P and Jimmy B 2014 Adam: a method for stochastic optimization *CoRR* abs/1412.6980
[48] Gómez-Gardenes Jus, Soriano-Panos D and Arenas A 2018 Critical regimes driven by recurrent mobility patterns of reaction–diffusion processes in networks *Nat. Phys.* **14** 391–5
[49] Arenas A, Cota W, Gomez-Gardenes J, Sergio Gómez, Granell C, Matamalas J T, Soriano-Panos D and Steinegger B 2020 A mathematical model for the spatiotemporal epidemic spreading of covid19 *MedRxiv*
[50] Klinger E, Rickert D and Hasenauer J 2018 pyABC: distributed, likelihood-free inference *Bioinformatics* **34** 3591–3
[51] Abraham M J, Murtola T, Schulz R, Páll S, Smith J C, Hess B and Lindahl E 2015 Gromacs: high performance molecular simulations through multi-level parallelism from laptops to supercomputers *SoftwareX* **1** 19–25
[52] Lindahl E, Hess B and Spoel D van der 2001 Gromacs 3.0: a package for molecular simulation and trajectory analysis *J. Mol. Model.* **7** 306–17
[53] Páll S, Abraham M J, Kutzner C, Hess B and Lindahl E 2015 Tackling exascale software challenges in molecular dynamics simulations with gromacs pp 3–27 (arXiv:1506.00716)
[54] Berendsen H J C, Spoel D van der and Drunen R van 1995 GROMACS: a message-passing parallel molecular dynamics implementation *Comput. Phys. Commun.* **91** 43–56
[55] David V D S, Lindahl E, Hess B, Groenhof G, Mark A E and Berendsen H J C 2005 GROMACS: fast, flexible and free *J. Comput. Chem.* **26** 1701–18
[56] Pronk S *et al* 2013 GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit *Bioinformatics* **29** 845–54
[57] Lindahl A and van der Spoel H 2020 Gromacs 2020.3 source code
[58] Beckstein O *et al* 2022 GromacsWrapper (available at: https://github.com/Becksteinlab/GromacsWrapper) (https://doi.org/10.5281/zenodo.17901)
[59] Tien M Z, Sydykova D K, Meyer A G and Wilke C O 2013 Peptidebuilder: a simple python library to generate model peptides *Wilke*
[60] Martínez L, Andrade R, Birgin E G and Martínez J M 2009 PACKMOL: a package for building initial configurations for molecular dynamics simulations *J. Comput. Chem.* **30** 2157–64
[61] MacKerell A D J, Bashford D, Bellott M L D R, Dunbrack R L J, Evanseck J D, Field M J, Fischer S, Gao J, Guo H, Ha S 1998 All-atom empirical potential for molecular modeling and dynamics studies of proteins *J. Phys. Chem.* B **102** 3586–3616
[62] Mackerell A D J, Feig M and Brooks C L I I I 2004 Extending the treatment of backbone energetics in protein force fields: limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations *J. Comput. Chem.* **25** 1400–15
[63] Bussi G, Donadio D and Parrinello M 2007 Canonical sampling through velocity rescaling *J. Chem. Phys.* **126** 014101
[64] Essmann U, Perera L, Berkowitz M L, Darden T, Lee H and Pedersen L G 1995 A smooth particle mesh Ewald method *J. Chem. Phys.* **103** 8577–93
[65] Hess B, Bekker H, Berendsen H J C and Fraaije J G E M 1997 LINCS: a linear constraint solver for molecular simulations *J. Comput. Chem.* **18** 1463–72
[66] Hummer G and Köfinger Jurgen 2015 Bayesian ensemble refinement by replica simulations and reweighting *J. Chem. Phys.* **143** 12B634_1
[67] Zhou P, Yang X-L, Wang X-G, Ben H, Zhang L, Zhang W, Hao-Rui S, Zhu Y, Bei Li, Huang C-L *et al* 2020 A pneumonia outbreak associated with a new coronavirus of probable bat origin *Nature* **579** 270–3
[68] Wu F *et al* 2020 A new coronavirus associated with human respiratory disease in China *Nature* **579** 265–9
[69] Lipsitch M, Swerdlow D L and Finelli L 2020 Defining the epidemiology of Covid-19—studies needed *New Engl. J. Med.* **382** 1194–6
[70] Bielekova B *et al* 2000 Encephalitogenic potential of the myelin basic protein peptide (amino acids 83–99) in multiple sclerosis: results of a phase ii clinical trial with an altered peptide ligand *Nat. Med.* **6** 1167–75
[71] Spyranti Z, Tselios T, Deraos G, Matsoukas J and Spyroulias G A 2010 NMR structural elucidation of myelin basic protein epitope 83–99 implicated in multiple sclerosis *Amino Acids* **38** 929–36
[72] Cavanagh J, Fairbrother W J, Palmer A G I I I and Skelton N J 1995 *Protein NMR Spectroscopy: Principles and Practice* (Amsterdam: Elsevier)
[73] Yang Z, Chakraborty M and White A D 2021 Predicting chemical shifts with graph neural networks *Chem. Sci.* **12** 10802–9