Edinburgh Research Explorer

Parameter estimation for biochemical reaction networks using Wasserstein distances

Parameter estimation for biochemical reaction networks using Wasserstein distances

Kaan Öcal[1,2], Ramon Grima[2] and Guido Sanguinetti[1]

[1] School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, United Kingdom
[2] School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3JH, United Kingdom

E-mail: [1]kaan.ocal@ed.ac.uk, [2]ramon.grima@ed.ac.uk, [3]g.sanguinetti@ed.ac.uk

Abstract. We present a method for estimating parameters in stochastic models of biochemical reaction networks by fitting steady-state distributions using Wasserstein distances. We simulate a reaction network at different parameter settings and train a Gaussian process to learn the Wasserstein distance between observations and the simulator output for all parameters. We then use Bayesian optimization to find parameters minimizing this distance based on the trained Gaussian process. The effectiveness of our method is demonstrated on the three-stage model of gene expression and a genetic feedback loop for which moment-based methods are known to perform poorly. Our method is applicable to any simulator model of stochastic reaction networks, including Brownian Dynamics.

1. Introduction

Modern experimental methods such as flow cytometry and fluorescence in-situ hybridization (FISH) allow the measurement of cell-by-cell molecule numbers for RNA, proteins and other substances for large numbers of cells at a time, opening up new possibilities for the quantitative analysis of biological systems. Of particular interest is the study of biological reaction systems describing processes such as gene expression, cellular signalling and metabolism on a molecular level. It is well established that many of these processes are inherently stochastic [1–3] and that deterministic approaches can fail to capture properties essential for our understanding of these systems [4, 5]. Despite recent technological and conceptual advances, modelling and inference for stochastic models of reaction networks remains challenging due to additional complexities not present in the deterministic case. The Chemical Master Equation (CME) [6] in particular, while frequently used to model many types of reaction networks, is difficult to solve exactly, and parameter inference in practice often relies on a variety of approximation schemes whose accuracy and efficiency can vary widely and unpredictably depending on the context [7].

The diffusion approximation and the system size expansion [6] are two well-known approximations to the CME and frequently used for inference [8, 9]. However, in the presence of bimolecular reactions such as enzyme-substrate or protein-DNA interactions these approximations are unable to deal with low copy numbers frequently found in biological systems, rendering them unsuitable for inference in reaction systems involving species such as e.g. individual genes or mRNA, which is often present in copy numbers of less than 20 per cell [10].

Other methods for parameter inference rely on fitting moments of the particle number distributions returned by the CME to experimental data [11–14]. Moments can often be computed or approximated by solving a set of coupled equations, bypassing expensive simulations of the system in question. However, as pointed out in [13, 15, 16] these moment-based methods are not always suitable for inference. Computing the moments for reaction systems with bimolecular interactions usually necessitates the use of so-called moment closure approximations, validity conditions for which are not well-understood [17–19]. Given the wide variety of moment closure schemes it is not generally clear *a priori* which, if any, will prove suitable for a given reaction system, and the right method is usually chosen empirically based on its performance [20]. In addition, moment closure typically results in a set of coupled nonlinear equations which can have multiple different solutions, further complicating their use in parameter inference. We will provide an example of a genetic feedback loop based on [20] for which many commonly used moment approximations break down or provide inaccurate moment estimates.

In this paper we propose a method to estimate parameters for the Chemical Master Equation from population snapshot data by matching steady state distributions using Wasserstein distances [21], also known as Earth Mover's distances in the literature. Since Wasserstein distances can in general not be computed analytically we take sample-based estimates and emulate the complex dependency of the output on the parameters by using a Bayesian regression approach. We train a Gaussian process (GP) to learn the distances between the observed data and the steady-state distributions at different parameter settings, obtained using simulations, and apply Bayesian optimization (BO) to find the parameters minimizing this distance, sequentially selecting the next parameter settings to evaluate until the optimum is found. Our approach requires orders of magnitude fewer evaluations than grid searches and is suitable for any simulator-based model of reaction networks, including models such as Brownian Dynamics for which the previously mentioned inference methods are not available.

The idea of performing parameter inference based on considering full distributions has been previously explored in [15] and [16], where it is shown that estimating parameters by matching moments can result in reduced predictive power and inaccurate fits of the actual distributions. Both [15] and [16] perform parameter inference by maximizing the likelihood of the observed data; this requires computing the likelihood function which is commonly done by solving a finite-dimensional approximation of the Chemical Master Equation, the so-called Finite State Projection (FSP) [22]. The FSP involves solving a large system of coupled ordinary differential equations and scales poorly for larger reaction systems as the number of equations grows exponentially with the number of species considered. Our method in contrast does not require the likelihood function and instead relies on empirically approximating the steady state distribution using simulations, rendering it more scalable and more flexible than current likelihood-based approaches.

One drawback of the method we present is that is only provides point estimates for parameters and does not return a measure of confidence that would enable one to assess uncertainty. While Bayesian optimization is used to infer the location of the optimal parameters it does not yield a posterior distribution over parameters as in Bayesian inference. To the knowledge of the authors efficient Bayesian inference for the CME based on steady-state observation data remains an intractable problem in general.

2. Background

2.1. The Chemical Master Equation

In this section we briefly review biochemical reaction networks and the Chemical Master Equation, referring to [7] for a more comprehensive treatment. A reaction network consists of species S_j, $j = 1, \ldots, s$ and reactions R_i, $i = 1, \ldots, r$ of the form

$$a_{i,1}S_1 + \ldots + a_{i,s}S_s \longrightarrow b_{i,1}S_1 + \ldots + b_{i,s}S_s \tag{1}$$
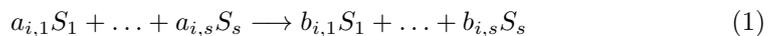
where $\vec{a}_i := (a_{i,1}, \ldots, a_{i,s})$ and $\vec{b}_i := (b_{i,1}, \ldots, b_{i,s})$ are vectors of nonnegative integers, the stoichiometric coefficients of the reaction. The Chemical Master Equation approach models the reaction network as a Markov chain whose states are given by tuples $\vec{n} := (n_1, \ldots, n_s) \in \mathbb{N}^s$ defining the number of particles of each species present at each time. The transitions of the Markov chain correspond to reactions, with the transition rate of reaction R_i determined by the state-dependent propensity function $\rho_i(\vec{n})$. The forward Kolmogorov equation for this Markov chain is called the Chemical Master Equation and reads:

$$\frac{\partial}{\partial t}P(\vec{n}, t) = \sum_{i=1}^{r}\left[\rho_i\left(\vec{n} - \vec{S}_i\right)P\left(\vec{n} - \vec{S}_i, t\right) - \rho_i(\vec{n})P(\vec{n}, t)\right] \tag{2}$$

Here $\vec{S}_i := \vec{b}_i - \vec{a}_i$ describes the net change in reactant numbers during reaction i.

The form of the transition functions $\rho_j(\vec{n})$ depends on the specific reaction, but the most commonly used transition functions are given by the mass-action law,

$$\rho_i(\vec{n}) := \lambda_i \binom{n_1}{a_{i,1}} \ldots \binom{n_s}{a_{i,s}} \tag{3}$$

for rate constants $\lambda_i > 0$. While our approach can handle general transition functions, in what follows we restrict ourselves to mass-action propensities of the form (3). With this setup the task of inferring parameters for the CME reduces to finding the appropriate rate constants λ_i.

We remark that the steady state distribution of a reaction system does not change if all transition rates are rescaled by a common factor $c > 0$. Thus by observing the steady state one can only identify the rate constants up to a common scaling factor, which can be fixed if any one reaction rate is known. It is frequently possible to measure the degradation rate of reaction species experimentally, which removes this ambiguity - such an approach to inference is taken for example in [23]. In the remainder of this paper we will always assume that one reaction rate is given and estimate the remaining rate constants.
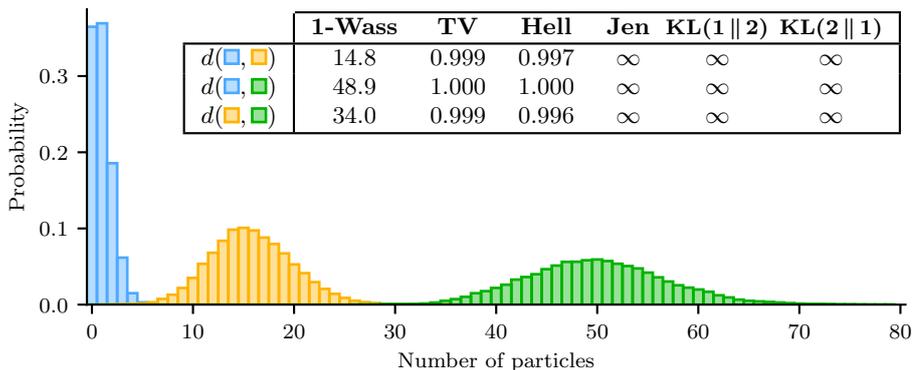
	1-Wass	TV	Hell	Jen	KL(1 ‖ 2)	KL(2 ‖ 1)
$d(\blacksquare, \blacksquare)$	14.8	0.999	0.997	∞	∞	∞
$d(\blacksquare, \blacksquare)$	48.9	1.000	1.000	∞	∞	∞
$d(\blacksquare, \blacksquare)$	34.0	0.999	0.996	∞	∞	∞

Figure 1: Common discrepancy measures for probability distributions do not provide usable distance metrics for simulated data. The histograms show empirical estimates of the steady-state distributions for a simple birth-death process with three different ratios of birth and death rates r ($r_\blacksquare = 1$, $r_\blacksquare = 16$, $r_\blacksquare = 50$). The table compares their 1-Wasserstein (1-Wass), total variation (TV) and Hellinger (Hell) distances, their Jensen divergences (Jen) and their Kullback-Leibler divergences (KL). Even though the two outer histograms are significantly further apart than the neighbouring pairs, the total variation and Hellinger distances in all cases differ by less than 1%, and the Kullback-Leibler and Jensen divergences between any two of these histograms are infinite. The 1-Wasserstein distance on the other hand captures an intuitive notion of distance between these histograms.

2.2. Wasserstein distances

We perform parameter estimation based on population snapshot data by minimizing the discrepancy between the observed distribution over particle numbers and the distributions returned by the simulator. In this section we motivate and describe our choice of discrepancy measure, namely Wasserstein distances and refer to [24] for a general discussion of parameter estimation in biology.

There are a variety of commonly used discrepancy measures in the literature that can be used to compare and match distributions directly. Information-theoretic measures like the Kullback-Leibler or Jensen divergences often become infinite if the compared distributions do not have identical support (which is rarely the case for the empirical distributions we consider), rendering them essentially useless for our purposes. Other metrics like the total variation and Hellinger distances do not provide a usable measure of distance for distributions without significant overlap, a common scenario where particle numbers can vary over orders of magnitude (see Fig. 1). In contrast, Wasserstein distances are generally well-defined and provide an interpretable distance metric between distributions. This is important for our global approach used for optimization described in Section 2.3.

The steady-state distribution for a reaction system can be represented as a normalized histogram over the state space $\mathbb{N}^s$, where s is the number of reactant species. Wasserstein distances measure the discrepancy between two such histograms by considering how much and how far probability mass has to be moved in order to reshape one histogram into the other; for this reason they are often called Earth

Mover's Distances (see Fig. 2). The concept of moving mass between histograms is formalized by transport plans introduced below.

Consider two histograms P, Q over $\mathbb{N}^s$; the value of the histogram P at $\vec{i} = (i_1, \ldots, i_s) \in \mathbb{N}^s$ is denoted $P_{\vec{i}}$. A transport plan T between P and Q is a histogram on $\mathbb{N}^s \times \mathbb{N}^s$ whose first and second marginals are P and Q, respectively,

$$\sum_{\vec{j}} T_{\vec{i},\vec{j}} = P_{\vec{i}} \qquad\qquad \sum_{\vec{i}} T_{\vec{i},\vec{j}} = Q_{\vec{j}} \qquad\qquad (4)$$

The value $T_{\vec{i},\vec{j}}$ can be viewed as the amount of probability mass that has to be moved from $\vec{i}$ to $\vec{j}$ in order to convert the histogram P into Q; Eq. (4) then represents the conservation of probability mass during this process. The simplest transport plan between P and Q is the independent coupling given by

$$(P \otimes Q)_{\vec{i},\vec{j}} = P_{\vec{i}} \cdot Q_{\vec{j}} \qquad\qquad (5)$$

which specifies that the probability mass in every bin of P is to be distributed evenly across Q. We denote the space of transport plans between P and Q by $U(P, Q)$.
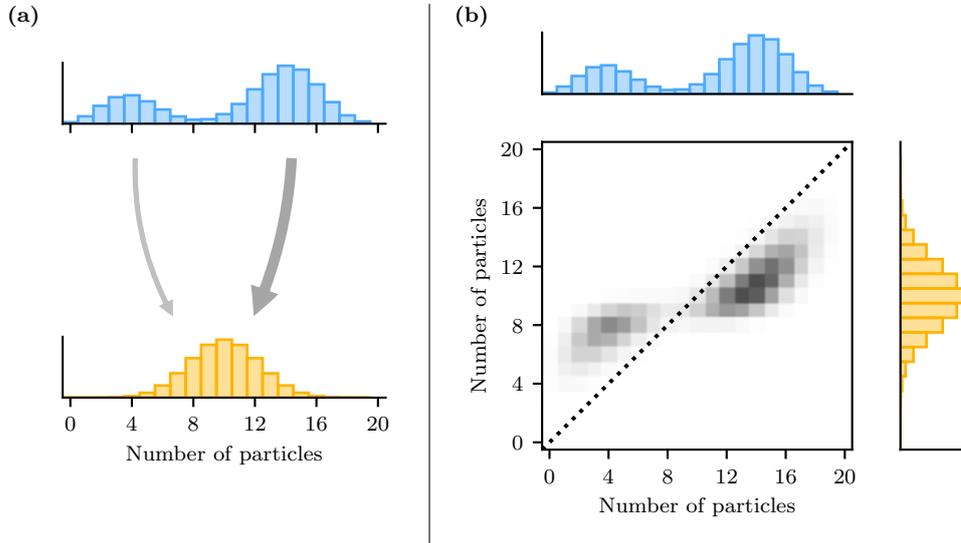
Figure 2: **(a)** Optimal transport distances between histograms measure how much mass has to be moved in order to convert one histogram into the other. **(b)** Illustration of a transport plan between the two histograms in (a). The joint histogram shows the amount of mass transported between different locations in the histograms. Mass on the diagonal (dotted line) is not moved during transport.

Optimal transport maps are defined by assigning to each move a certain cost. We define a cost function C to be a nonnegative function on $\mathbb{N}^s \times \mathbb{N}^s$, where $C_{\vec{i},\vec{j}}$ represents the cost involved in transporting a unit of probability mass from $\vec{i}$ to $\vec{j}$. While this is not necessary for the theory we will assume that the cost function is a distance metric on the ground space $\mathbb{N}^s$, ie. there is a metric d on $\mathbb{N}^s$ such that $C_{\vec{i},\vec{j}} = d(\vec{i}, \vec{j})$. The

optimal transport problem with cost function C reads

$$\mathcal{W}_C(P,Q) := \inf_{T \in U(P,Q)} \langle C, T \rangle = \inf \left\{ \sum_{\vec{i},\vec{j}} C_{\vec{i},\vec{j}} T_{\vec{i},\vec{j}} : T_{\vec{i},\vec{j}} \geq 0, \right.$$

$$\left. \sum_{\vec{j}} T_{\vec{i},\vec{j}} = P_{\vec{i}}, \sum_{\vec{i}} T_{\vec{i},\vec{j}} = Q_{\vec{j}} \right\} \qquad (6)$$

One can check that the Wasserstein distance $\mathcal{W}_C$ defines a metric on the space of probability distributions on $\mathbb{N}^s$. More generally one can verify that the p-Wasserstein distance

$$\mathcal{W}_C^{(p)}(P,Q) := \mathcal{W}_{C^p}(P,Q)^{1/p} \qquad (7)$$

defines a metric on the space of probability distributions on $\mathbb{N}^s$ for all $p \geq 1$. Here the cost is given by $C_{\vec{i},\vec{j}}^p = d(\vec{i},\vec{j})^p$.

A commonly used class of metrics on the space $\mathbb{N}^s$ is given by the weighted ℓ^q-metrics defined by

$$d_{\ell^q}^{(\mathbf{w})}(\vec{i},\vec{j}) = \left(\sum_{k=1}^{s} w_k^q |i_k - j_k|^q \right)^{\frac{1}{q}} \qquad (8)$$

for $q \geq 1$ and positive weights w_k. These metrics are all equivalent and yield equivalent classes of p-Wasserstein metrics for fixed p; the exact choice of q and the weights is therefore not important in most applications. We will use the notation $\mathcal{W}_p(P,Q)$ for the p-Wasserstein distance when the ground metric is understood.

2.3. Bayesian optimization

Since the dependence of the Wasserstein distance between observed data and simulator output on the parameters of the CME is not available in closed form and can only be evaluated by running simulations we are faced with the task of minimizing a function that is expensive to evaluate and about which no gradient information is available. In order to do this efficiently we rely on Bayesian optimization, a method for efficiently optimizing expensive black-box functions in low to moderate dimensions based on a Gaussian process surrogate of the function to be optimized. See [25] for a comprehensive reference on Gaussian processes and [26] for an overview of Bayesian optimization going beyond the description in this paper.

In order to apply Bayesian optimization to our problem we start by placing a Gaussian process prior on the loss function $L(\mathbf{x})$, which is defined on a space $\mathcal{X}$, that measures the discrepancy with the observed data:

$$\hat{L} \sim \mathrm{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \qquad (9)$$

with mean function $\mu(\mathbf{x})$ and covariance kernel $k(\mathbf{x}, \mathbf{x}')$. Thus $\hat{L}$ is a statistical model of the true function L. In our case L will be the Wasserstein loss measuring the Wasserstein distance between the steady-state distributions, $\mathcal{X}$ will be the chosen

space of parameters and the $\mathbf{x}$ will be individual parameter settings. We assume that we can use a simulator to compute noisy observations of $L(\mathbf{x}_i)$:

$$\tilde{L}(\mathbf{x}_i) = L(\mathbf{x}_i) + \epsilon_i \tag{10}$$

at any given point $\vec{\boldsymbol{x}}_i$, where the ϵ_i are observation noise. We assume that the ϵ_i are iid. normal random variables with mean zero; the standard deviation of the observation noise can be interpreted as a hyperparameter of the Gaussian process. With this setup our Gaussian process $\hat{L}$ can be updated by obtaining data points $\mathcal{D}_i = \{\mathbf{x}_i, \tilde{L}(\mathbf{x}_i)\}$ for different $\mathbf{x}_i$ and computing the posterior $\hat{L} \mid \mathcal{D}$.

Our goal is to minimize $L(\mathbf{x})$ with as few evaluations of $\tilde{L}(\mathbf{x})$ as possible. Bayesian optimization consists of a procedure for sequentially choosing the points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ at which $\tilde{L}(\mathbf{x})$ is to be evaluated in order to decrease the uncertainty about the location of the optimum, based on the Gaussian process $\hat{L}$. This is done by considering a so-called acquisition function $\alpha(\mathbf{x}; L \mid \mathcal{D})$ depending on the collected observations $\mathcal{D}$ and choosing the next point to evaluate as

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathbf{x}; L \mid \mathcal{D}_{1:n}) \tag{11}$$

The acquisition function returns a point $\mathbf{x}_{n+1}$ such that computing $\tilde{L}(\mathbf{x}_{n+1})$ yields additional knowledge about the minimum of $L(\mathbf{x})$, e.g. by choosing a point which is likely to be near the true minimum. It should be simpler to evaluate and optimize than the target function so that one can use standard optimization methods for finding $\mathbf{x}_{n+1}$ with little overhead. After finding $\mathbf{x}_{n+1}$ and running the simulator to compute $\tilde{L}(\mathbf{x}_{n+1})$ one updates the Gaussian process $\hat{L}$ with the data $\mathcal{D}_{n+1} = \{\mathbf{x}_{n+1}, \tilde{L}(\mathbf{x}_{n+1})\}$ and repeats this procedure until the true optimum of $L(\mathbf{x})$ is found. An illustration of Bayesian optimization can be seen in Fig. 3.

A common choice for the acquisition function α is Expected Improvement, defined by the following formula:

$$\alpha_{\mathrm{EI}}(\mathbf{x}; L \mid \mathcal{D}) := \mathbb{E}_{L|\mathcal{D}} \left[\left(\min_{\mathbf{x}_i \in \mathcal{D}} \tilde{L}(\mathbf{x}_i) - \hat{L}(\mathbf{x}) - \beta \right)^+ \right] \tag{12}$$

Here $\beta \geq 0$ is a small "jitter" parameter used to reduce the time spent in local optima and increase exploration. With this acquisition function the predicted optimum of $L(\mathbf{x})$ is typically computed as:

$$\mathbf{x}^* := \arg\min_{\mathbf{x}_i \in \mathcal{D}} \tilde{L}(\mathbf{x}_i) \tag{13}$$

Since Eq. (12) can be computed in closed form the expected improvement at a point $\mathbf{x}$ can be evaluated quite cheaply, and gradients can be computed at little additional cost. It is known that Bayesian optimization using this acquisition function is guaranteed to find the optimum of the target function L under some mild assumptions on L and the Gaussian process prior [27]. This combined with its simplicity and empirical performance properties make Expected Improvement a popular choice of acquisition function in Bayesian optimization. Other common acquisition functions are Upper Confidence Bound, Probability of Improvement and Knowledge Gradient [26], which we shall not consider here.
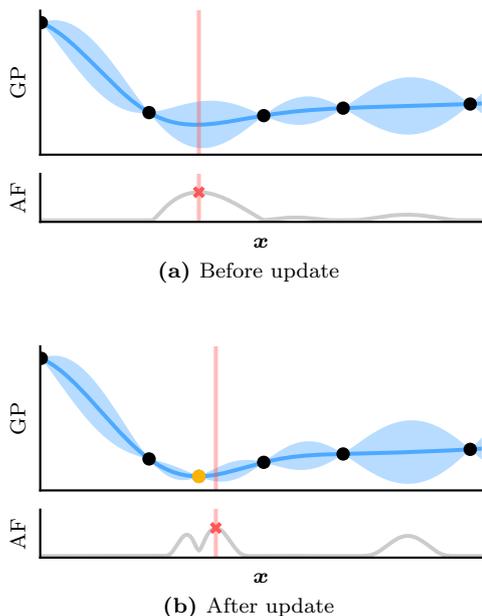
(a) Before update

(b) After update

Figure 3: Illustration of Bayesian optimization in one dimension. Plotted are a Gaussian process (GP) and its acquisition function (AF) before and after an update step. The shaded area represents two standard deviations around the mean. Each round consists of computing the loss function at the point maximizing the acquisition function (vertical line) and updating the GP with the computed value. After the update step the acquisition function changes to reflect the information gained in the process, and a new point is chosen for the next round.

3. Methods

3.1. Computing Wasserstein distances

Computing Wasserstein distances by finding an explicit optimal transport map in Eq. (6) requires solving a constrained linear optimization problem with a large number of variables and constraints. As this approach quickly gets impracticable for realistic histogram sizes we use an adaptation of the Sinkhorn algorithm presented in [28], which computes the optimum of a relaxed version of the transport problem:

$$\mathcal{W}_C^{(\epsilon)}(P,Q) = \inf_{T \in U(P,Q)} \langle C, T \rangle - \epsilon H(T) \tag{14}$$

where the regularizer $H(T)$ is defined as

$$H(T) = -\sum_{\vec{i},\vec{j}} T_{\vec{i},\vec{j}} \log T_{\vec{i},\vec{j}} \tag{15}$$

with the convention that $H(T) = -\infty$ if one of the entries of T is not positive. One can derive error bounds on the difference $\mathcal{W}_C^{(\epsilon)}(P,Q) - \mathcal{W}_C(P,Q)$ and show that the solutions to Eq. (14) converge to the solution of the unregularized problem as $\epsilon \to 0$. We can thus compute Wasserstein distances approximately by solving Eq. (14) for small enough ϵ.

    The Sinkhorn algorithm is an iterative solver for Eq. (14), a description of which can be found in [28]. One drawback of this algorithm is that the number of iterations required for convergence increases as $\epsilon \to 0$; in order to compute the solution to Eq. (14) for small ϵ we therefore use an annealing procedure starting with a large value of ϵ (typically $\epsilon = 10$) and multiplying it by an annealing factor $\delta < 1$ at each step. We run each step until the two margin constraints (4) are satisfied to a specified

tolerance ϵ' in the ℓ^1-norm. To improve convergence speed for small ϵ we use the overrelaxation method presented in [29].

Due to the large numerical range encountered in the Sinkhorn algorithm all computations are performed in log-space. In addition, when computing the p-Wasserstein distance with a weighted ℓ^q ground metric, for $p = q$ the cost matrix $C^p_{\vec{i},\vec{j}} = d^{(\mathbf{w})}_{\ell^q}(\vec{i},\vec{j})^p$ decomposes as a sum of terms, one for each histogram dimension,

$$C^p_{\vec{i},\vec{j}} = \sum_{k=1}^{s} w_k^p |i_s - j_s|^p \tag{16}$$

which permits vectorization of the relevant matrix-vector products in the Sinkhorn algorithm. For this reason and since the exact choice of q does not matter (see Section 2.2) we will generally set $p = q$ for efficiency.

We remark that (unweighted) Wasserstein distances in one dimension can be evaluated using a simpler and more straightforward algorithm: if F and G are the cumulative distribution functions of two probability distributions f and g on $\mathbb{R}$, respectively, then one can prove [21] that

$$\mathcal{W}_p(f,g) = \left(\int_0^1 \left| F^{-1}(x) - G^{-1}(x) \right|^p \, \mathrm{d}x \right)^{1/p} \tag{17}$$

For discrete histograms this integral can be computed exactly with little overhead, making Wasserstein distances in one dimension especially suited for computational purposes.

3.2. Bayesian optimization

At this stage we are given experimentally observed data P (a histogram over particle numbers) and a parametrized simulator model $Q(\mathbf{x})$ whose output is a histogram depending on the parameters $\mathbf{x} \in \mathcal{X}$. In our case the function $Q(\mathbf{x})$ is the steady state distribution of the system with parameters $\mathbf{x}$. Our goal is to find parameters $\mathbf{x}$ minimizing a chosen Wasserstein distance

$$L(\mathbf{x}) := \mathcal{W}_p(Q(\mathbf{x}), P) \tag{18}$$

We can approximate L for any set of parameters by simulating the Chemical Master Equation for sufficiently long times using the Stochastic Simulation Algorithm [30]. As described in Section 2.3 we optimize the loss using Bayesian optimization; in this section we describe the details of our setup.

Assuming our task is to infer d different parameters we start by choosing a (bounded) search space $\mathcal{X} \subseteq \mathbb{R}^d$. In practice one should choose a reasonably large region in which the true parameters are expected to be found; if this is not the case after optimization one can enlarge the search space and continue optimization until the optimum is found. We sample m points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ spread across the search space and evaluate $\tilde{L}(\mathbf{x}_1), \ldots, \tilde{L}(\mathbf{x}_m)$ in order to pre-train the GP. The choice of m usually depends on the dimension and the expected roughness of the loss landscape. We sample the points using Latin hypercubes in order to achieve uniform coverage of the parameter space.

The mean of the Gaussian process $\hat{L}$ is set to a constant equal to the mean of the $\tilde{L}(\mathbf{x}_i)$. For the covariance kernel we initially choose a squared exponential function of the form

$$k(\mathbf{x}, \mathbf{x}') = \sigma_y^2 \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Lambda (\mathbf{x} - \mathbf{x}')\right] \tag{19}$$

The hyperparameters for this setup are the marginal variance σ_y^2 and the precision matrix Λ, restricted to be diagonal for simplicity. We fit the kernel hyperparameters by maximizing the marginal likelihood of the data $\mathcal{D}_{1:m}$, a common procedure for determining hyperparameters for Gaussian process regression [25].

Bayesian optimization now consists of repeatedly optimizing the acquisition function, computing the loss function $L(\mathbf{x}^*)$ at the optimum $\mathbf{x}^*$ and updating the Gaussian process $\hat{L}$ with this information. In order to improve the fit of the Gaussian process we periodically refit the kernel after sampling enough new points. The resulting procedure is summarized in Algorithm 1.

Algorithm 1 Bayesian optimization-based parameter estimation

Input: P_{obs} - observed histogram
Options: $N > 0$ - number of rounds before refitting GP hyperparameters
$\qquad\qquad m > 0$ - number of pre-training samples
$\qquad\qquad \epsilon > 0$ - tolerance
Output: $\mathbf{x}^*$ - parameter estimate

---

sample $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$
for all $i = 1, \ldots, m$ **do**
$\qquad$ **compute** $\tilde{L}(\mathbf{x}_i)$ by running simulator
$\qquad$ $\mathcal{D}_i \leftarrow \{\mathbf{x}_i, \tilde{L}(\mathbf{x}_i)\}$
end for
fit mean and kernel hyperparameters of $\hat{L}$ **to** $\mathcal{D}_{1:m}$
$n \leftarrow m$
loop
$\qquad$ **maximize** $\alpha(\mathbf{x}; \hat{L} \mid \mathcal{D}_{1:n})$
$\qquad$ $\mathbf{x}_{n+1} \leftarrow \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \hat{L} \mid \mathcal{D}_{1:n})$
$\qquad$ **compute** $\tilde{L}(\mathbf{x}_{n+1})$ by running simulator
$\qquad$ **if** $\tilde{L}(\mathbf{x}_{n+1}) < \epsilon$ **then**
$\qquad\qquad$ $\mathbf{x}^* \leftarrow \mathbf{x}_{n+1}$
$\qquad\qquad$ **break**
$\qquad$ **update** $\hat{L}$ with $\mathcal{D}_{n+1} = \{\mathbf{x}_{n+1}, \tilde{L}(\mathbf{x}_{n+1})\}$
$\qquad$ $n \leftarrow n + 1$
$\qquad$ **if** $n - m = 0 \,(\mathrm{mod}\, N)$ **then**
$\qquad\qquad$ **refit** kernel hyperparameters of $\hat{L}$ **to** $\mathcal{D}_{1:n}$
end loop
return $\mathbf{x}^*$

We often found the value of the loss function $L(\mathbf{x})$ to range over orders of magnitude, making it difficult to fit a Gaussian process to $L(\mathbf{x})$ in a way that models the function accurately around the minimum. We therefore use a modified version of the loss function given by $L_{\log}(\mathbf{x}) := \log(1 + L(\mathbf{x}))$ which satisfies $L_{\log}(\mathbf{x}) \approx L(\mathbf{x})$

for small $L(\mathbf{x})$ and grows more slowly for large values of L. This significantly reduces the dynamic range of the function modelled by the Gaussian process, improving the efficiency of Bayesian optimization on these problems.

3.3. Non-stationary Bayesian optimization

One issue with the squared exponential kernel commonly used in Gaussian process regression is that it is stationary, that is, the covariance $k(\mathbf{x}, \mathbf{x}')$ only depends on the relative difference $\mathbf{x} - \mathbf{x}'$. This makes it unsuitable for modelling functions which have different levels of roughness in different parts of parameter space. The loss functions we encountered often displayed a minimum located in a narrow valley surrounded by a large plateau where the loss showed little variation. A Gaussian process with a stationary kernel would either choose very short length scales in order to fit the valley accurately, resulting in a lot of unnecessary uncertainty far away from the minimum and an inefficient optimization procedure due to overexploration, or it would pick large length scales to fit the plateau and treat the observations around the valley as statistical outliers, rendering the optimization routine unable to find the minimum.

Following [31] we thus consider a weighted superposition of two independent Gaussian processes, $f = w_g f_g + w_l f_l$ with

$$f_g \approx \mathrm{GP}(0, k_g(\mathbf{x}, \mathbf{x}')) \qquad\qquad f_l \approx \mathrm{GP}(0, k_l(\mathbf{x}, \mathbf{x}')) \qquad\qquad (20)$$

and weight functions $w_g(\mathbf{x})$, $w_l(\mathbf{x})$ to be determined later. This enables us to decompose the Gaussian process into a global component $w_g f_g$ modelling the smooth large-scale behaviour of the loss function and a local component $w_l f_l$ that can fit the function accurately at the minimum. We choose squared exponential kernels $k_g(\mathbf{x}, \mathbf{x}')$ and $k_l(\mathbf{x}, \mathbf{x}')$ for these two Gaussian process components. The weights are parametrized as

$$w_g(\mathbf{x}) = \sqrt{\frac{1}{1 + \nu(\mathbf{x})}} \qquad\qquad w_l(\mathbf{x}) = \sqrt{\frac{\nu(\mathbf{x})}{1 + \nu(\mathbf{x})}} \qquad\qquad (21)$$

for a nonnegative function $\nu(\mathbf{x})$. We set $\nu(\mathbf{x})$ to be a squared exponential basis function of the form

$$\nu(\mathbf{x}) = \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}_\nu)^T \Lambda_\nu (\mathbf{x} - \mathbf{x}_\nu)\right] \qquad\qquad (22)$$

for Λ_ν a symmetric positive-definite matrix, chosen to be diagonal in our case, and an anchor point $\mathbf{x}_\nu$.

The kernel of the total Gaussian process $f = w_g f_g + w_l f_l$ can be computed to be

$$k(\mathbf{x}, \mathbf{x}') = w_g(\mathbf{x}) w_g(\mathbf{x}') k_g(\mathbf{x}, \mathbf{x}') + w_l(\mathbf{x}) w_l(\mathbf{x}') k_l(\mathbf{x}, \mathbf{x}') \qquad\qquad (23)$$

As before we fit the hyperparameters by maximum likelihood estimation, constraining $\mathbf{x}_\nu$ to be the location of the current best observation each time the kernel is refit. This is consistent with our observation that the loss function typically exhibits the largest amount of variation around the minimum.

4. Results

4.1. General setup

In all our experiments we chose the 1-Wasserstein distance with a weighted ℓ^1 ground metric on $\mathbb{N}^s$. The weight for each species is chosen to be inversely proportional to the mean particle number in the reference distribution, $w_i \propto \mathbb{E}[n_i]^{-1}$; this avoids scenarios where mismatches in abundant species are responsible for the bulk of the Wasserstein loss, leading the optimizer to ignore low-copy number species as it tries to make the largest gains in the loss function.

Our focus on 1-Wasserstein distances is due to the fact that the p-Wasserstein distances are numerically more difficult to compute using the Sinkhorn algorithm for $p > 1$. If the reference distribution can be matched exactly given the right parameters, these parameters will be the global minimum of the Wasserstein loss for all $p \geq 1$, so the choice of p will not affect results. If there is a model mismatch and the reference distribution cannot be reproduced with any set of parameters our algorithm will return parameters minimizing the p-Wasserstein distance to that distribution; in this case the minimum will in general depend on the chosen distance. While we have not investigated this issue in detail, we believe that the effect of outliers will be more pronounced for larger p and that $p = 1$ is therefore the most stable choice for limited sample sizes.

The steady state distribution of a reaction network can be obtained in two ways: one can perform simultaneous measurements for a population of cells in the steady state, or one can measure the time average of a single instance of the system over long time scales by the ergodic theorem. Population-level measurements are suitable for experimental data and are commonly realized using e.g. flow cytometry, while in our simulations we prefer to run one instance of the system for long times and compute the time average. Depending on the level of parallelization desired it is possible to run multiple independent simulations for each set of parameters and combine the results, speeding up the inference procedure. All simulations were performed using the Stochastic Simulation Algorithm (SSA) [30].

We check convergence to the steady state distribution by computing Wasserstein distances between the time averages at time points nT, $n = 1, 2, \ldots$, where T is the chosen epoch length. The epoch length T is chosen heuristically such that simulating the system for a few epochs yielded accurate estimates of the steady state distribution for the ground truth system. Simulations are stopped when the distance at two consecutive time points becomes less than 0.02. In order to avoid wasting computation time for parameter settings yielding very bad fits to the observed data we also stop simulations if the distance at two consecutive time points becomes less than 2% of the approximate Wasserstein distance to the observed data. As computing joint Wasserstein distances in multiple dimensions can be expensive, in our simulations we computed the sum of the Wasserstein distances of the marginals when checking for convergence; we found that this did not seem to measurably affect results at the chosen tolerances.

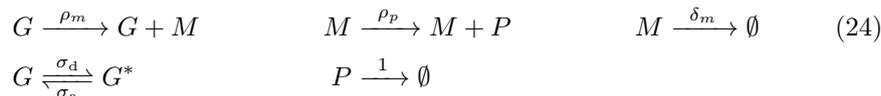The standard deviation of the observation noise in Eq. (10) for our Gaussian processes is set to 0.03, which is on the same order of magnitude as the typical measurement error due to finite simulation lengths. In general this hyperparameter can be fit together with the rest of the kernel parameters, but we found this value to work well across experiments. The jitter parameter β in Eq. (12) was set to 0.01, a value

commonly used in practice, being e.g. the default in the Python library `scikit-learn`.

Since reaction rates are positive and often range over orders of magnitude we use the log reaction rates for inference.

4.2. Three-stage gene expression model

Our first experiment consisted of identifying the parameters in the three-stage model of gene expression found in [32], described by the following reactions:
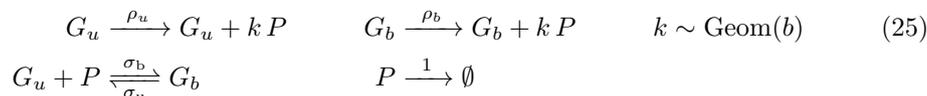
$$G \xrightarrow{\rho_m} G + M \qquad\qquad M \xrightarrow{\rho_p} M + P \qquad\qquad M \xrightarrow{\delta_m} \emptyset \qquad (24)$$

$$G \underset{\sigma_\mathrm{a}}{\overset{\sigma_\mathrm{d}}{\rightleftharpoons}} G^* \qquad\qquad P \xrightarrow{1} \emptyset$$

This model consists of four reactant species: a gene in an activated (G) and inactivated (G^*) form, mRNA (M) and protein (P). We fix the protein degradation rate to 1 and perform inference over the remaining five rate constants by observing joint distributions over mRNA and proteins.

We fix ground truth values for all parameters (taken from Fig. 3 in [32] with $\gamma = 1$) and use the SSA to obtain a reference steady state distribution. We then apply our method to recover the parameter values based on the observed distribution. The search range for the parameters is set to cover two orders of magnitude per dimension and includes the ground truth values; the results of this experiment can be seen in Fig. 4. We re-ran the same experiment comparing marginal mRNA and protein numbers (without using the joint distribution) and obtained very similar results, suggesting that it is not always necessary to measure joint distributions to perform parameter inference for the CME if the marginals are fit precisely.

A typical run for this experiment lasts about 9–10 hours (Intel Xeon CPU at 3.30 GHz, 32 GB RAM). In comparison, parameter inference for this system using moment equations takes a few minutes. Since the three-stage gene expression model is a linear reaction system it is possible to compute the steady state moments of protein and mRNA numbers exactly without having to resort to moment-closure approximations; parameters can then be estimated exactly using standard optimization methods. While moment-based inference provides a fast and accurate inference method for linear reaction systems, the three-stage gene expression model demonstrates that our method works for nontrivial examples. In the next example, however, we will encounter a nonlinear reaction system for which moment equations perform less well, requiring the use of alternative methods for parameter estimation.

4.3. Bursty feedback loop

In our second experiment we considered a bursty version of the genetic feedback loop described by the following list of reactions, taken from [33] and [20]:

$$G_u \xrightarrow{\rho_u} G_u + k\,P \qquad\qquad G_b \xrightarrow{\rho_b} G_b + k\,P \qquad\qquad k \sim \mathrm{Geom}(b) \qquad (25)$$

$$G_u + P \underset{\sigma_\mathrm{u}}{\overset{\sigma_\mathrm{b}}{\rightleftharpoons}} G_b \qquad\qquad P \xrightarrow{1} \emptyset$$

This system describes a protein which can bind to its gene and hence influence its own transcription rate. The number of proteins produced at each translation event follows
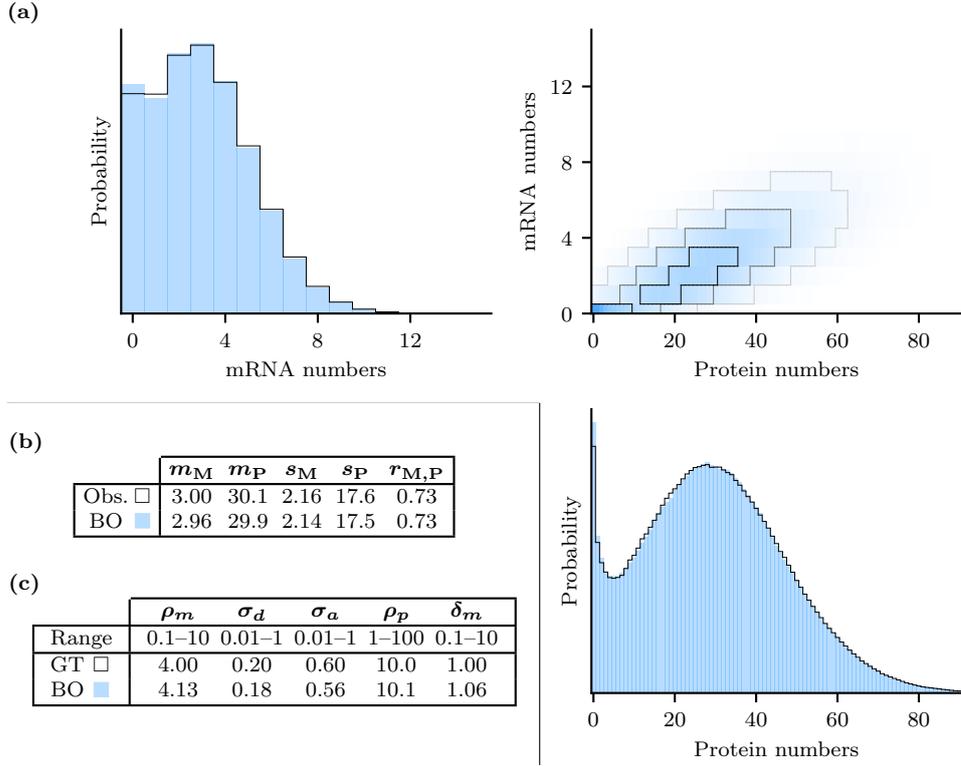
(a)

(b)

	m_{M}	m_{P}	s_{M}	s_{P}	$r_{\mathrm{M,P}}$
Obs. □	3.00	30.1	2.16	17.6	0.73
BO ▣	2.96	29.9	2.14	17.5	0.73

(c)

	ρ_m	σ_d	σ_a	ρ_p	δ_m
Range	0.1–10	0.01–1	0.01–1	1–100	0.1–10
GT □	4.00	0.20	0.60	10.0	1.00
BO ▣	4.13	0.18	0.56	10.1	1.06

Figure 4: Results for the three-stage gene expression model in Eq. (24). **(a)**: Steady steady state distribution over mRNA and protein numbers for the observed data (contours) and the parameters estimated using Bayesian optimization (shaded). **(b)**: Means ($\boldsymbol{m}$) and standard deviations ($\boldsymbol{s}$) of mRNA and protein numbers as well as their Pearson correlation coefficient ($\boldsymbol{r}$). Both the shape and the moments of the observed distribution are matched by our method. **(c)**: Ground truth and estimated parameters for the observed data and the chosen search ranges. The results were obtained after 362 rounds starting with 300 initial samples, where the GP kernel was refit every 75 rounds during optimization.

a geometric distribution with mean b,

$$p(k) = \frac{b^k}{(1+b)^{k+1}} \qquad (k \geq 0)$$

which is a common approximation of mRNA-mediated protein production when the lifetime of mRNA is very short compared to the mean protein lifetime [34].

The system (25) displays different types of behaviour depending on whether $\rho_u > \rho_b$ (negative feedback) or $\rho_u < \rho_b$ (positive feedback). In [20] the authors compare different moment closure approximations as well as the Linear Mapping Approximation [35] and show that for the negative feedback loop it is possible to efficiently obtain accurate parameter estimates using a suitable moment closure
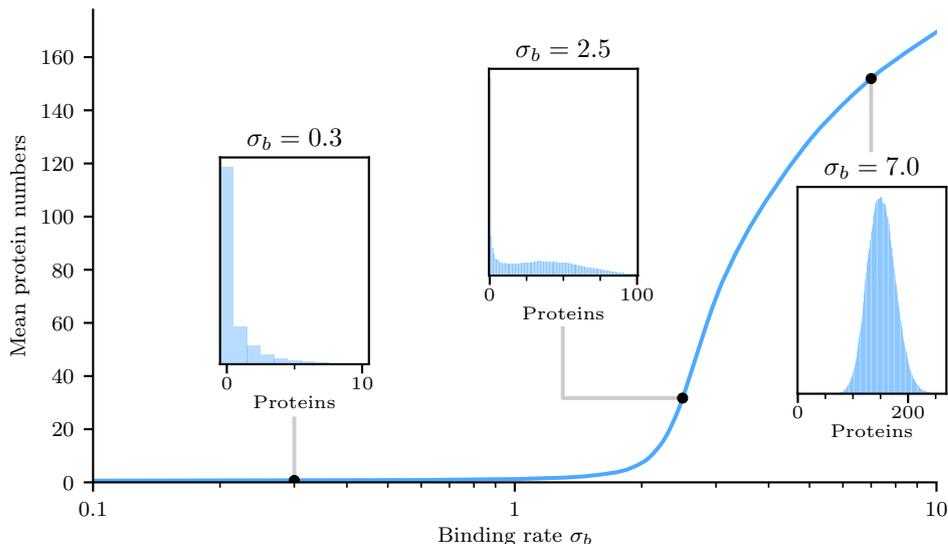
Figure 5: Mean protein numbers plotted against the binding rate σ_b for the bursty positive feedback loop. There is a sharp increase around $\sigma_b \approx 2$ as the system switches from being in the inactivated state most of the time to the activated state. The steady state distributions differ qualitatively depending on σ_b, with the intermediate region characterized by bimodal protein number distributions. The values of the remaining parameters are $\rho_u = 0.3$, $\sigma_u = 400$, $\rho_b = 105$, $b = 2$.

scheme. In this section we will focus on the positive feedback case, which we found to be more challenging for the approach presented in [20].

Positive feedback in this system can result in strong sensitivity to parameter values (see Fig. 5). We chose parameters that resulted in the gene spending non-negligible amounts of time in both the bound and the unbound state. This regime (which we shall call the intermediate regime) is characterized by a bimodal steady state distribution over protein numbers which changes rapidly with σ_b and ρ_b.

In order to test how well moment closure methods can approximate the positive feedback loop we applied six different moment closure schemes from [20]: conditional derivative matching and conditional Gaussian [36], both conditioned on either the bound or the unbound states of the gene (denoted CDM1 and CDM2, resp. CG1 and CG2), as well as unconditional Gaussian (Gauss) and derivative matching (DM) [37]. We also considered the Linear Mapping Approximation (LMA) [35] as it yields a set of moment equations that can be solved directly, similar to classical moment closure schemes. In addition we tested the conditional negative binomial approximation, again conditioned on both the bound and the unbound state of the gene, which we respectively denote CNB1 and CNB2.

We found that none of the nine methods tested were able to accurately predict mean and standard deviation of the protein number distribution for our system (Fig. 6). While many were able to model the system outside the intermediate regime, the presence of large fluctuations in that regime significantly decreased the accuracy of the methods. The closed moment equations are nonlinear and usually admit multiple
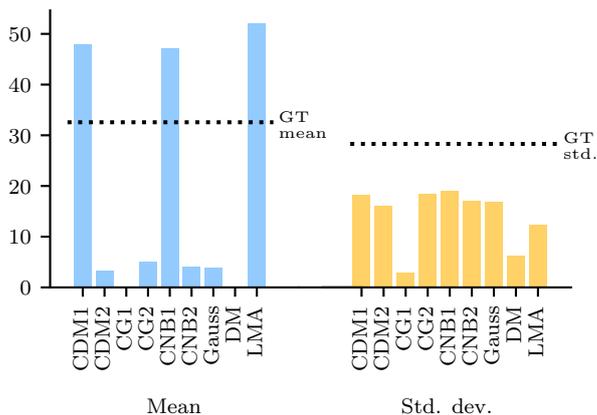
Figure 6: Eight different moment closure schemes and the LMA applied to the positive feedback loop, using ground truth the parameters given in Fig. 7. CG1 and DM failed to yield a solution predicting a positive mean. Even the best approximation (CNB1) is more than 30% off in its estimate of both the mean and the standard deviation of protein numbers.

solutions, yielding complex, negative or outlandishly high predictions for moments. This is similar to the scenario tested in [17] where the considered moment closure schemes failed to yield unique solutions in general. We found that with the exception of the LMA all tested moment closure methods broke down in different parts of the intermediate regime, and while the LMA itself always yields an interpretable solution it returns inaccurate predictions of moments for our system (cf. Fig. 6).

Given that none of the moment closure methods used for the negative feedback case capture the intermediate regime in the positive feedback case one has to rely on alternative inference methods for our problem. The complicated dependence of the steady state on the parameters makes this task challenging in general. In the intermediate regime very small changes in σ_b or ρ_b will typically lead to large changes in the steady state distribution, while in the regime where the gene is mostly unbound the system will virtually be independent of σ_b (cf. Fig. 5). Hence the loss landscape looks very different at different points in parameter space, causing problems for both global and local optimization approaches. A grid search for σ_b for example would need to sample values at very short intervals in order to find the correct value in the intermediate regime, while a local optimization routine would likely get stuck if initialized in a region where the loss function is flat.

We tested our method on this problem by performing joint inference over σ_b, ρ_u and σ_u in the intermediate regime, based on observing the (marginal) protein number distribution. The results are shown in Fig. 7. We recovered a set of parameters that yield a steady state protein distribution closely matching the input data and which themselves are in broad agreement with the ground truth. We suspect that the observed discrepancy of about $10-15\%$ is due to local non-identifiability of the parameters, consistent with previous observations for the negative feedback loop [20]. We remark that changing only σ_b from its ground truth value does not yield a similar steady state distribution as can be seen in Fig. 5, suggesting that the non-identifiability involves trade-offs between different parameters.

The time required for a typical run of our algorithm with this reaction system was 3.5 hours, which is noticeably faster than the three-stage gene expression model due to the reduced number of simulations needed for convergence.
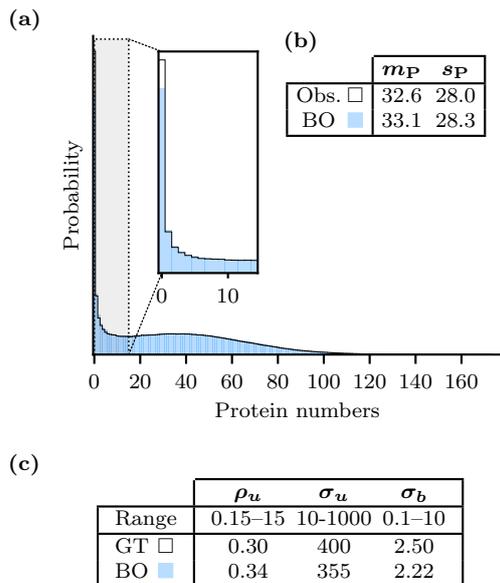
(a)

(b)

	$m_{\mathbf{P}}$	$s_{\mathbf{P}}$
Obs. □	32.6	28.0
BO ▪	33.1	28.3

(c)

	ρ_u	σ_u	σ_b
Range	0.15–15	10-1000	0.1–10
GT □	0.30	400	2.50
BO ▪	0.34	355	2.22

Figure 7: Results for the bursty positive feedback loop. **(a)**: Steady state distribution over protein numbers for the observed data (contours) and the estimated parameters (shaded). **(b)**: Means and standard deviations of the two distributions. **(c)**: Ground truth and estimated parameters for the observed data. The results were obtained after 130 rounds starting with 75 initial samples, where the kernel was refit every 25 rounds during optimization. The remaining parameters are $\rho_b = 105$, $b = 2$ (cf. Fig. 5).

4.4. Further experiments

In order to gain a better understanding of the relationship between parameter estimation using Wasserstein distances and existing methods we performed additional tests, investigating the accuracy of our method for limited sample sizes and for variations of the bursty feedback loop from Section 4.3.

In practical applications one does not usually have direct access to the steady-state distribution of a reaction system due to the finite number of cells that can be measured at once. The reference distribution will therefore always be an empirical measure of the ground truth, where sample sizes of at least 100 are common for population snapshot data. It is therefore important to understand how well our method performs when the input distribution is such an empirical distribution.

We tested our method on the three-stage gene expression model, the positive feedback loop and negative feedback loop considered in [20], taking as inputs empirical estimates of the ground truth distribution for different sample sizes. Results are displayed in Tables 1 and 2. In all cases the estimated parameters yield steady-state distributions that are similar to both the empirical (input) distribution and the original ground truth data, where the quality of the fit improves with the sample size. We found that larger sample sizes seem to be necessary where multiple species are observed simultaneously. The estimated steady-state distributions try to match the means and standard variations of the empirical data, which is not surprising as convergence in any of the Wasserstein distance generally implies convergence of the moments [21]. This suggests that Wasserstein-based inference behaves similarly to moment-based inference in the context of limited sample sizes.

$N = 500$									
Obs.					BO				
m_M	m_P	s_M	s_P	$r_{M,P}$	m_M	m_P	s_M	s_P	$r_{M,P}$
2.89	29.9	2.05	17.2	0.69	2.92	30.3	2.07	16.2	0.67

$N = 1000$									
Obs.					BO				
m_M	m_P	s_M	s_P	$r_{M,P}$	m_M	m_P	s_M	s_P	$r_{M,P}$
2.98	29.7	2.16	17.4	0.71	2.99	29.7	2.03	17.2	0.76

Table 1: Marginal means and standard deviations over mRNA and protein counts as well as their correlation coefficients for the three-stage gene expression network given empirical input distributions with sample sizes $N = 500$ and 1000. Our method was run for 325 rounds starting with 300 initial samples, with the remaining setup as in Section 4.2.

System	$N = 100$				$N = 500$				$N = 1000$			
	Obs.		BO		Obs.		BO		Obs.		BO	
	m_P	s_P	m_P	s_P	m_P	s_P	m_P	s_P	m_P	s_P	m_P	s_P
PFL	30.1	28.6	30.6	28.2	30.0	27.5	29.1	26.7	33.2	27.7	32.4	28.4
NFL	25.2	22.0	25.0	21.0	28.6	19.1	28.5	19.1	28.0	19.7	28.1	19.2

Table 2: Mean and standard deviation over protein counts for the three-stage gene expression network given empirical input distributions with sample sizes $N = 100$, 500 and 1000. Our method was run for 150 rounds starting with 75 initial samples, with the remaining setup as in Section 4.3.

Existing inference methods for the CME for steady-state data fall broadly into the category of moment-based inference methods and direct, likelihood-based methods. The direct approach in [15, 16] differs from our proposal in that it uses the Finite State Projection algorithm (FSP) to compute the steady-state distribution. The computational time required by the FSP depends on the number of states considered, which in turn is determined by the desired accuracy and typically increases exponentially in the number of species. As the time complexity of the FSP is cubic in the number of states it does not scale as well as the simulation-based approach for nontrivial reaction systems and we therefore concentrated on moment-based inference in the following comparison.

To investigate the accuracy of moment-based inference for a nonlinear reaction network we evaluated the performance of both approaches on four variations of the feedback loop considered in Section 4.3 with different parameters. The first is the negative feedback loop (NFL) considered in [20], and the remainder was obtained by randomly drawing parameters for the entire reaction system and taking the first three parameter settings that yielded nontrivial steady-state distributions ($m_P > 5$), which we call R1, R2 and R3. The chosen parameters and search ranges are given in Table 3. For moment-based inference we tested all of the 9 moment-closure schemes listed in Section 4.3. For our method the setup is the same as in Section 4.3, each run consisting of 150 optimization rounds starting with 75 initial samples.

We found moment-based inference to work well for the negative feedback loop, with all but two of the listed moment-closure schemes achieving accuracy comparable to our results (see Appendix Appendix A). In some of these cases the mean and

variance of the steady-state distribution were matched accurately, but the shape of the distributions differed perceptibly, which suggests that means and variances are not always enough to uniquely characterise steady-state distributions (see Fig. 8). Results for the remaining three networks were mixed: while some moment-closure schemes worked well for either R2 or R3, many introduced a perceptible bias into the results, and for R1 the resulting standard deviation was consistently too low. We report the results of our experiments in Tables 4 and 5, choosing the most accurate moment-closure scheme for each system as a representative of moment-based inference. Results for the remaining moment-closure schemes are given in Appendix Appendix A.
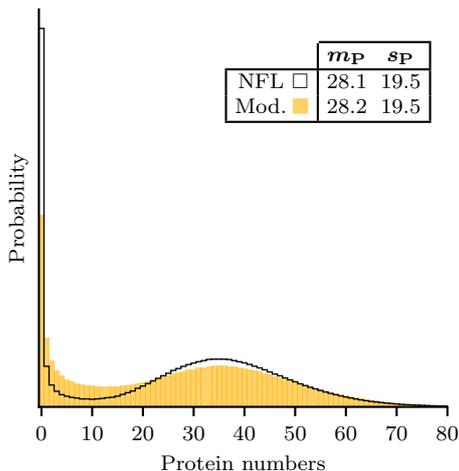

	$m_{\mathbf{P}}$	$s_{\mathbf{P}}$
NFL □	28.1	19.5
Mod. ■	28.2	19.5

Figure 8: Moments do not always distinguish steady-state distributions in practice. Shown are steady-state distributions for the original negative feedback loop (contours) and for a modification (shaded). While the two distributions are perceptibly different their means and standard deviations differ by less than 1%. The modified parameters are $\rho_u = 14.5$, $\sigma_u = 0.33$, $\sigma_b = 0.0048$.

Parameter values returned by either method frequently differed from their ground truth values even when the steady-state distributions matched up. This suggests that parameter identifiability is a common problem for this class of reaction networks (cf. [20]).

We point out that the accuracy of our method matches that of moment-based inference in all cases despite the fact that our approach is model-agnostic and can be applied to any reaction system without modifications, whereas all but two of the moment-closure schemes we used are tailored to the genetic feedback loop. The wealth of moment-closure schemes available for this reaction system is due to its structure as a genetic switch and conditional moment-closure schemes or the LMA [35] do not in general have equivalents for other types of reaction networks.

Runtimes for our method were on the order of one hour, except for the negative feedback loop (2.5 hours) and the positive feedback loop (3.5 hours). Moment-based inference was very fast, usually taking less than a minute per (convergent) run. We found that some moment-closure methods ran into numerical issues during optimization for some of the systems considered, consistent with the issues we reported for the positive feedback loop in Section 4.3 and leading to drastically increased runtimes.

System	ρ_u	Range	σ_u	Range	σ_b	Range	ρ_b	b
NFL	13	1–100	0.1	0.01–1	0.001	10^{-4}–0.01	0	3
R1	1.3	1–100	42	1–100	1.5	0.1–10	63	1.5
R2	9.3	0.1–10	1	0.1–10	0.7	0.1–10	0.4	8.3
R3	1	0.1–10	355	10–1000	7.9	0.1–10	20.1	2.4

Table 3: Ground truth parameter values and search ranges for the additional reaction networks tested in this section.

System	GT		BO		MBI		
	m_P	s_P	m_P	s_P	m_P	s_P	Method
PFL	32.6	28.0	33.1	28.3	31.9	21.4	CG1
NFL	27.9	19.5	27.9	19.5	28.5	19.3	CG2
R1	66.4	16.1	66.4	15.8	66.5	13.5	CG2
R2	11.4	10.1	11.3	9.9	11.6	10.1	CG2
R3	9.9	9.3	9.7	9.2	10.7	9.0	CG1

Table 4: Mean and standard deviation over protein counts for the ground truth (GT), our method (BO) and moment-based inference (MBI), as well as the corresponding moment-closure scheme in the last column.

System	GT			BO			MBI		
	ρ_u	σ_u	σ_b	ρ_u	σ_u	σ_b	ρ_u	σ_u	σ_b
PFL	0.3	400	2.5	0.34	355	2.22	1.41	1000.0	5.45
NFL	13	0.1	0.001	12.7	0.055	0.0006	12.9	0.071	0.0007
R1	1.3	42	1.5	22.2	15.1	0.28	40.2	50.2	0.17
R2	9.3	1	0.7	1.54	6.43	0.10	1.49	10.0	0.10
R3	1	355	7.9	1.14	88.2	1.87	1.36	326.4	6.67

Table 5: Parameters estimated by our method and using moment-based inference (see Table 4).

5. Conclusion

We presented a general-purpose method for parameter estimation for simulator-based stochastic reaction networks. We computed Wasserstein distances between steady state distributions to quantify the discrepancy between observed data and simulator output for different parameters and constructed a probabilistic model of the Wasserstein distance at unexplored parameter settings by training a Gaussian process with these data. Bayesian optimization was applied in order to (i) iteratively choose parameter settings that are likely to be close to the optimum, (ii) evaluate the system at the chosen parameters and (iii) update the model until the results are consistent with observations. We applied our method to two inverse problems for the Chemical Master Equation: a five-dimensional problem based on the classical three-stage gene expression model, and a three-dimensional problem based on a genetic feedback loop, recovering data consistent with observations in both cases. We performed further experiments that demonstrated that our method compares favourably with moment-based inference in terms of accuracy.

Our method only relies on having access to a simulator and does not require the computation of likelihoods. It is thus especially suited to models such as

Brownian Dynamics which can be sampled from but which are in general not tractable analytically. Given the fact that many simulator models are expensive to evaluate, Bayesian optimization provides an effective method to perform inference with a limited number of simulations. To our knowledge there is no previous literature on efficient parameter estimation for Brownian Dynamics or related models, and we hope that the approach presented will provide a first stepping stone in this direction.

Bayesian optimization has previously been used for likelihood-free inference e.g. in [38]. One potential limitation of global optimization approaches like Bayesian optimization is that they are often difficult to apply to high-dimensional problems. The number of evaluations needed until convergence usually scales with the dimension of the parameter space, reducing their usability for problems with many parameters. The effectiveness of Bayesian optimization in particular depends strongly on the ability to model complex high-dimensional functions using a Gaussian process given a limited amount of evaluations, a nontrivial task. We are positive that continuing research on non-stationary Bayesian optimization methods [31, 39] will enable us to deal with these problems more effectively in the future.

In this paper we used Wasserstein distances to match distributions, one problem with these being that the computations can become expensive for large histogram dimensions. There are multiple possible remedies for this. The Sinkhorn algorithm consists mostly of repeated matrix-vector operations and can be easily implemented on GPUs, potentially resulting in significant speed gains compared to using a CPU. For systems with large particle numbers one can coarse-grain the histogram by binning particle numbers and approximating the Wasserstein distances using the coarsened histograms. Finally one can compute Wasserstein distances between lower-dimensional marginals separately and minimize the sum of the distances. While this method may potentially lose information about correlations between different species we have observed it to yield near identical results for the three-stage gene expression model. Since it is difficult to experimentally measure joint distributions of several distinct species, fitting marginalized distributions is a reasonable approach in these situations and we found it to perform well in practice.

Data availability

The code for this project is available at `https://github.com/kaandocal/wasserstein_inference`.

Acknowledgments

References

[1] M. B. Elowitz, "Stochastic gene expression in a single cell," *Science*, **297**(5584): 1183–1186, 2002.

[2] P. J. Choi, L. Cai, K. Frieda, and X. S. Xie, "A stochastic single-molecule event triggers phenotype switching of a bacterial cell," *Science*, **322**(5900): 442–446, 2008.

[3]   D. J. Kiviet, P. Nghe, N. Walker, *et al.*, "Stochasticity of metabolism and growth at the single-cell level," *Nature*, **514**(7522): 376–379, 2014.

[4]   H. H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends in Genetics*, **15**(2): 65–69, 1999.

[5]   R. Ramaswamy, N. González-Segredo, I. F. Sbalzarini, and R. Grima, "Discreteness-induced concentration inversion in mesoscopic chemical systems," *Nat. Commun.* **3**: 779, 2012.

[6]   N. van Kampen, *Stochastic Processes in Physics and Chemistry*, 3rd ed. Elsevier, 2007.

[7]   D. Schnoerr, G. Sanguinetti, and R. Grima, "Approximation and inference methods for stochastic biochemical kinetics - a tutorial review," *J. Phys. A*, **50**(9): 093001, 2017.

[8]   A. Golightly and D. J. Wilkinson, "Bayesian sequential inference for stochastic kinetic biochemical network models," *J. Comput. Bio.* **13**(3): 838–851, 2006.

[9]   ——, "Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo," *Interface Focus*, **1**(6): 807–820, 2011.

[10]   S. Marguerat, A. Schmidt, S. Codlin, *et al.*, "Quantitative analysis of fission yeast transcriptomes and proteomes in proliferating and quiescent cells," *Cell*, **151**(3): 671–683, 2012.

[11]   F. Fröhlich, P. Thomas, A. Kazeroonian, *et al.*, "Inference for stochastic chemical kinetics using moment equations and System Size Expansion," *PLOS Comput. Bio.* **12**(7): e1005030, 2016.

[12]   C. Zechner, J. Ruess, P. Krenn, *et al.*, "Moment-based inference predicts bimodality in transient gene expression," *PNAS*, **109**(21): 8340–8345, 2012.

[13]   J. Ruess and J. Lygeros, "Moment-based methods for parameter inference and experiment design for stochastic biochemical reaction networks," *ACM Trans. Model. Comput. Simul.* **25**(2): 8:1–8:25, 2015.

[14]   E. Cinquemani, "Identifiability and reconstruction of biochemical reaction networks from population snapshot data," *Processes*, **6**(9): 136, 2018.

[15]   G. Neuert, B. Munsky, R. Z. Tan, *et al.*, "Systematic identification of signal-activated stochastic gene regulation," *Science*, **339**(6119): 584–587, 2013.

[16]   B. Munsky, G. Li, Z. R. Fox, D. P. Shepherd, and G. Neuert, "Distribution shapes govern the discovery of predictive models for gene regulation," *PNAS*, **115**(29): 7533–7538, 2018.

[17]   D. Schnoerr, G. Sanguinetti, and R. Grima, "Comparison of different moment-closure approximations for stochastic chemical kinetics," *J. Chem. Phys.* **143**(18): 185101, 2015.

[18]   ——, "Validity conditions for moment closure approximations in stochastic chemical kinetics," *J. Chem. Phys.* **141**(8): 084103, 2014.

[19]   C. Schilling, S. Bogomolov, T. A. Henzinger, A. Podelski, and J. Ruess, "Adaptive moment closure for parameter inference of biochemical reaction networks," *Biosystems*, Selected papers from the Computational Methods in Systems Biology 2015 conference, **149**: 15–25, 2016.

[20]   Z. Cao and R. Grima, "Accuracy of parameter estimation for auto-regulatory transcriptional feedback loops from noisy data," *J. R. Soc. Interface*, **16**(153): 20180967, 2019.

[21]   C. Villani, *Optimal Transport: Old and New*, ser. Grundlehren der mathematischen Wissenschaften. Berlin Heidelberg: Springer, 2009.

[22]   B. Munsky and M. Khammash, "The Finite State Projection algorithm for the solution of the Chemical Master Equation," *J. Chem. Phys.* **124**(4): 044104, 2006.

[23] B. Schwanhäusser, D. Busse, N. Li, *et al.*, "Global quantification of mammalian gene expression control," *Nature*, **473**: 337–342, 2011.

[24] M. Ashyraliyev, Y. Fomekong-Nanfack, J. A. Kaandorp, and J. G. Blom, "Systems biology: Parameter estimation for biochemical models," *The FEBS Journal*, **276**(4): 886–902, 2009.

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, ser. Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2006.

[26] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, **104**(1): 148–175, 2016.

[27] E. Vazquez and J. Bect, "Convergence properties of the expected improvement algorithm with fixed mean and covariance functions," *J. Stat. Plan. Inference*, **140**(11): 3088–3095, 2010.

[28] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in Neural Information Processing Systems*, **26**: 2292–2300, 2013.

[29] A. Thibault, L. Chizat, C. Dossal, and N. Papadakis, "Overrelaxed Sinkhorn-Knopp algorithm for regularized optimal transport :" 2017. arXiv: `1711.01851`.

[30] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *J. Comput. Phys.* **22**(4): 403–434, 1976.

[31] R. Martinez-Cantin, "Local nonstationarity for efficient Bayesian optimization :" 2015. arXiv: `1506.02080`.

[32] V. Shahrezaei and P. S. Swain, "Analytical distributions for stochastic gene expression," *PNAS*, **105**(45): 17256–17261, 2008.

[33] R. Grima, D. R. Schmidt, and T. J. Newman, "Steady-state fluctuations of a genetic feedback loop: An exact solution," *J. Chem. Phys.* **137**(3): 035104, 2012.

[34] N. Friedman, L. Cai, and X. S. Xie, "Linking stochastic dynamics to population distribution: An analytical framework of gene expression," *Phys. Rev. Lett.* **97**(16): 168302, 2006.

[35] Z. Cao and R. Grima, "Linear mapping approximation of gene regulatory networks with stochastic dynamics," *Nat. Commun.* **9**(1): 3305, 2018.

[36] M. Soltani, C. A. Vargas-Garcia, and A. Singh, "Conditional moment closure schemes for studying stochastic dynamics of genetic circuits," *IEEE Trans. Biomed. Circuits Syst.* **9**(4): 518–526, 2015.

[37] A. Singh and J. P. Hespanha, "A derivative matching approach to moment closure for the stochastic logistic model," *Bull. Math. Biol.* **69**(6): 1909–1925, 2007.

[38] M. U. Gutmann and J. Corander, "Bayesian optimization for likelihood-free inference of simulator-based statistical models," *J. Mach. Learn. Res.* **17**(125): 1–47, 2016.

[39] J. Snoek, K. Swersky, R. Zemel, and R. Adams, "Input warping for Bayesian optimization of non-stationary functions," *Proceedings of the International Conference on Machine Learning*, **31**: 1674–1682, 2014.

Appendix A. Results for moment-based inference

Method	PFL			NFL			R1			R2			R3		
	ρ_u	σ_u	σ_b	ρ_u	σ_u	σ_b	ρ_u	σ_u	σ_b	ρ_u	σ_u	σ_b	ρ_u	σ_u	σ_b
GT	0.3	400	2.5	13	0.1	0.001	1.3	42	1.5	9.3	1	0.7	1	355	7.9
CDM1	0.15	1000.0	5.82	12.6	0.025	0.0002	43.2	100.0	0.10	3.26	1.62	0.26	0.10	356.5	10.0
CDM2	0.97	74.7	5.13	2.14	0.031	0.0076	1.20	54.1	0.71	1.65	0.19	0.11	3.73	12.1	2.22
CG1	1.41	1000.0	5.45	12.6	0.012	0.0001	43.2	100.0	0.10	3.32	0.66	0.11	1.36	326.3	6.67
CG2	7.78	619.6	2.07	12.9	0.070	0.0007	40.2	50.2	0.17	1.49	10.0	0.10	2.46	192.7	2.38
CNB1	0.15	1000.0	5.86	13.9	0.24	0.0032	43.2	100.0	0.10	4.21	1.25	0.30	0.10	341.3	10.0
CNB2	0.42	39.1	0.98	86.2	0.051	0.0027	4.38	31.3	7.91	3.45	1.27	7.77	0.62	552.5	0.30
Gauss	-	-	-	12.6	0.017	0.0002	43.2	100.0	0.10	-	-	-	-	-	-
DM	0.15	1000.0	5.90	14.3	0.29	0.0042	43.2	100.0	0.10	3.74	1.23	0.25	0.10	336.1	10.0
LMA	15.0	1000.0	0.79	13.7	0.068	0.0008	43.2	100	0.10	6.75	0.75	0.19	4.53	1000.0	0.10

Table A1: Ground truth and inferred parameters for the five reaction networks considered in Section 4.4 using the tested moment-closure schemes. Gaussian moment closure failed to converge in most cases.

Method	PFL		NFL		R1		R2		R3	
	m_P	s_P	m_P	s_P	m_P	s_P	m_P	s_P	m_P	s_P
GT	32.6	28.0	27.9	19.5	66.4	16.1	11.4	10.1	9.9	9.3
CDM1	15.8	22.8	28.4	19.2	66.5	13.1	11.8	9.8	3.5	7.6
CDM2	195	25.2	2.57	4.39	16.6	15.1	5.3	7.1	43.6	12.9
CG1	31.9	21.4	28.7	19.1	66.6	13.1	11.1	10.4	10.7	9.0
CG2	36.6	15.2	28.5	19.3	66.5	13.5	11.6	10.1	10.7	7.8
CNB1	13.2	20.7	27.8	19.6	66.7	13.3	11.8	10.2	3.7	7.9
CNB2	169	26	25.8	61.9	90.5	15.5	4.8	6.2	1.5	2.3
Gauss	-	-	29.4	19.0	66.4	13.1	-	-	-	-
DM	16.8	23.1	27.9	19.7	66.7	13.1	11.8	10.0	4.2	8.5
LMA	34.8	11	27.8	21.0	67.1	13.1	13.7	12.2	11.0	6.2

Table A2: Mean and standard deviation over protein counts for the ground truth parameters and those inferred using moment-based inference.