Bioinformatics, 31(9), 2015, 1428–1435
doi: 10.1093/bioinformatics/btv007
Advance Access Publication Date: 7 January 2015
Original Paper

OXFORD

Gene expression

Inferring single-cell gene expression mechanisms using stochastic simulation

Bernie J. Daigle, Jr.[1], Mohammad Soltani[2], Linda R. Petzold[3] and Abhyudai Singh[2,*]

[1]Institute for Collaborative Biotechnologies, University of California, Santa Barbara, CA 93106, [2]Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 and [3]Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

*To whom correspondence should be addressed.
Associate Editor: Alfonso Valencia

Received on June 11, 2014; revised on November 5, 2014; accepted on January 4, 2015

Abstract

Motivation: Stochastic promoter switching between transcriptionally active (ON) and inactive (OFF) states is a major source of noise in gene expression. It is often implicitly assumed that transitions between promoter states are memoryless, i.e. promoters spend an exponentially distributed time interval in each of the two states. However, increasing evidence suggests that promoter ON/OFF times can be non-exponential, hinting at more complex transcriptional regulatory architectures. Given the essential role of gene expression in all cellular functions, efficient computational techniques for characterizing promoter architectures are critically needed.

Results: We have developed a novel model reduction for promoters with arbitrary numbers of ON and OFF states, allowing us to approximate complex promoter switching behavior with Weibull-distributed ON/OFF times. Using this model reduction, we created bursty Monte Carlo expectation-maximization with modified cross-entropy method ('bursty MCEM^2'), an efficient parameter estimation and model selection technique for inferring the number and configuration of promoter states from single-cell gene expression data. Application of bursty MCEM^2 to data from the endogenous mouse glutaminase promoter reveals nearly deterministic promoter OFF times, consistent with a multi-step activation mechanism consisting of 10 or more inactive states. Our novel approach to modeling promoter fluctuations together with bursty MCEM^2 provides powerful tools for characterizing transcriptional bursting across genes under different environmental conditions.

Availability and implementation: R source code implementing bursty MCEM^2 is available upon request.

Contact: absingh@udel.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The process of gene expression—whereby the information contained in a DNA sequence is converted into RNA and proteins—plays an essential role in the execution of nearly all cellular functions. As a result, the misregulation of this process underlies a large number of human diseases including cancer, diabetes and neurological disorders (Lee and Young, 2013). Despite its importance, the mechanistic details of gene expression are still not well understood. In particular, we lack a comprehensive molecular-level explanation for expression 'bursts'—periods of intense RNA and protein production separated by periods of quiescence—observed in pro- and eukaryotes (Cai *et al.*, 2006; Raj *et al.*, 2006). Precise characterization of the mechanisms underlying expression bursts is profoundly important, as the properties of these bursts have been implicated in

disease-related processes such as bacterial phenotype switching (Choi *et al.*, 2008) and HIV activation (Singh *et al.*, 2010).

Recent advances in single-cell monitoring and single-molecule detection have made possible the experimental characterization of gene expression bursts (Dar *et al.*, 2012; Golding *et al.*, 2005; Singh, 2014; So *et al.*, 2011; Suter *et al.*, 2011; Yu *et al.*, 2006). Specifically, Suter *et al.* (2011) have quantified transcriptional bursts from 11 endogenous mouse promoters, demonstrating that each observed expression pattern can be approximated using a stochastic two-state model of promoter architecture. This commonly used 'random telegraph' model assumes that each promoter can exist in one of two states—'ON' or 'OFF'—with synthesis of RNA only possible in the ON state. Because of intrinsic noise exhibited by the small numbers of molecules involved in transcription (e.g. 1–2 copies of DNA, few available copies of transcriptional regulators) (Raser and O'Shea, 2005), the promoter produces expression bursts by switching randomly between the transcriptionally active (ON) and inactive (OFF) states according to kinetic parameters (rate constants) that can be estimated from single-cell time series data (Suter *et al.*, 2011).

Although conceptually useful and amenable to analytical characterization, the random telegraph model is an oversimplification of the architecture of most promoters. Because of simultaneous regulation by multiple transcription factors as well as chromatin modifications, the effective number of states for most promoters is thought to be larger than two (Zhang *et al.*, 2012). A recent study of the human prolactin gene supports this assertion, where the distribution of time its promoter spent in an inactive state was inferred to be strongly non-exponential and thus indicative of multiple, sequential OFF states (Harper *et al.*, 2011). Classical examples of multi-state promoters include that of P_{RM} in phage lambda, whose complex mechanism of regulation gives rise to 128 regulatory states (Sanchez *et al.*, 2013) and the *Endo16* gene in sea urchin, whose cis-regulatory domain contains >30 binding sites for 15 different proteins that perform combinatorial regulation (Yuh *et al.*, 1998). In light of these observations and the increasing availability of single-cell expression data, computational methods are needed for characterizing complex promoter architectures and efficiently simulating their behavior.

An ideal candidate for such a method would (i) leverage the intrinsic noise of the system to better identify underlying mechanisms (Munsky *et al.*, 2009), (ii) simultaneously infer both the configuration of promoter states and the associated kinetic parameters needed for predictive simulation and (iii) provide computationally efficient performance for a wide range of architectures. Currently existing methods satisfy only a subset of these requirements. Suter *et al.* (2011) performed hidden Markov model parameter inference for two- and three-state promoter architectures, but their models assume constant (noise-free) promoter activity and RNA levels between discretely observed time points and they do not provide an efficient means to characterize architectures with larger numbers of states. We previously developed Monte Carlo expectation-maximization (MCEM) with modified cross-entropy method (MCEM^2), which uses statistically exact stochastic simulations to infer kinetic parameters from single-cell time series data (Daigle *et al.*, 2012). However, the original version of MCEM^2 does not enable characterization of promoter architecture. Toni *et al.* (2009) developed an approximate Bayesian computation-based method for inferring both parameters and model structure using stochastic simulations. Unfortunately, when using this method to discriminate between promoter models with increasing numbers of states, the addition of each state increases the number of unknown kinetic parameters (e.g. switching rates). In the presence of limited amounts of experimental data, this quickly renders more complex (and thus more realistic) models non-identifiable. We note that this drawback applies to any inference method that represents transitions between individual promoter states explicitly. Finally, stochastic simulation of multi-state promoter architectures suffers from a linear increase in computational cost with the addition of each promoter state, making the study of more complex models difficult.

Because of the limitations described above, our goal in this work was to develop a computationally efficient method for characterizing gene expression bursts by inferring the number and configuration of promoter states from single-cell time series data.

2 Results

Our results are structured as follows: we first devise a novel model reduction strategy that represents arbitrary numbers of promoter states by a single state accompanied by a time-dependent switching rate. As we demonstrate below, this strategy enables both efficient simulation and parameter inference. Next, we incorporate this model reduction into MCEM^2 and augment the method to perform model selection for configuration of promoter states. We demonstrate the resulting approach—'bursty MCEM^2'—by inferring the effective numbers of promoter states underlying simulated single-cell expression data. In addition, we use bursty MCEM^2 to characterize the architecture of the endogenous mouse glutaminase promoter given experimentally observed time series data (Suter *et al.*, 2011).

2.1 Model reduction for multi-state promoters

The random telegraph model of transcription can be represented by the following four chemical reactions:

$$\begin{aligned} \mathrm{DNA_{OFF}} &\underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} \mathrm{DNA_{ON}} \\ \mathrm{DNA_{ON}} &\xrightarrow{k_{\mathrm{m}}} \mathrm{DNA_{ON}} + \mathrm{mRNA} \\ \mathrm{mRNA} &\xrightarrow{\gamma_{\mathrm{m}}} \emptyset, \end{aligned} \tag{1}$$

where the promoter randomly switches between OFF and ON states according to rates k_{on} and k_{off}. RNA synthesis occurs at rate k_{m} from the ON state and expressed mRNAs live for an exponentially distributed time interval with mean lifetime $1/\gamma_{\mathrm{m}}$, where γ_{m} is the mRNA degradation rate.

A simplified version of (1) that is often used to model transcriptional bursting of mRNAs is:

$$\begin{aligned} \mathrm{DNA_{ON}} &\xrightarrow{c_1} \mathrm{DNA_{ON}} + \beta \times \mathrm{mRNA} \\ \mathrm{mRNA} &\xrightarrow{\gamma_{\mathrm{m}}} \emptyset \\ \beta &\sim \mathrm{Geometric}(c_3), \end{aligned} \tag{2}$$

where c_1 and β denote the burst frequency and size, respectively. In the stochastic formulation of (2) [see Gillespie (2007) for background information], mRNA bursts arrive at exponentially distributed time intervals with rate c_1. Each expression event generates a geometrically distributed number of transcripts β with mean value $(1-c_3)/c_3$ (Evans et al., 2000). Model (2) provides an increasingly accurate approximation of (1) as $k_{\mathrm{off}} \to \infty$, with $c_1 = k_{\mathrm{on}}$ and $c_3 = k_{\mathrm{off}}/(k_{\mathrm{off}} + k_{\mathrm{m}})$. A sample trajectory of mRNA

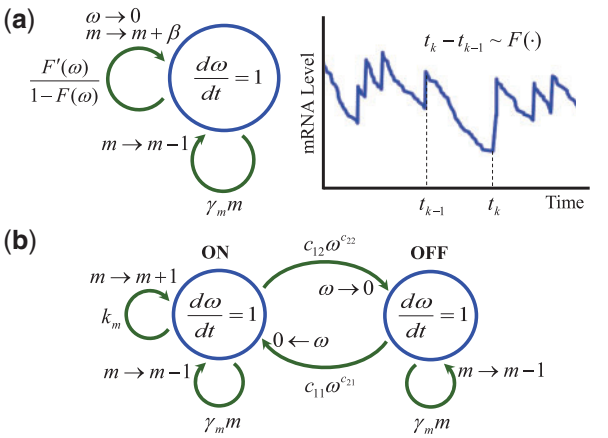


Fig. 1. Schematic of bursty gene expression models. (**a**) Transcriptional bursting model where each expression event generates a geometrically distributed burst of mRNAs. Each mRNA degrades at a constant rate γ_m. Bursting and decay reactions are represented by arrows with propensity functions directly adjacent (e.g. $\gamma_m m$). The state changes at the end of the arrows (e.g. $m \to m-1$) denote the change in $m(t)$ (mRNA level at time t) and/or $\omega(t)$ (timer state) when the reactions occur. The timer is set to zero when mRNA bursts are produced, and it increases as $\dot{\omega}(t)=1$ between bursts. An example of mRNA trajectory is displayed in the inset plot. The desired CDF $F(\cdot)$ of the inter-burst arrival time ($t_k - t_{k-1}$ in the plot) is obtained by setting the burst event propensity function to $F'(w)/(1-F(w))$. (**b**) Promoter-switching model where the promoter randomly toggles between ON and OFF states. Choosing the ON–OFF transition propensity functions as monomials in $\omega(t)$ results in Weibull-distributed ON and OFF times

copy numbers with transcriptional bursts followed by decay is illustrated in Figure 1.

Given the evidence discussed above that promoters exhibit multiple, sequential OFF states, a more realistic representation of the reversible reaction in (1) is the following:

$$\begin{aligned} &\mathrm{DNA}_{\mathrm{OFF}_1} \xrightarrow{k_1} \mathrm{DNA}_{\mathrm{OFF}_2} \\ &\quad\vdots \\ &\mathrm{DNA}_{\mathrm{OFF}_{N-1}} \xrightarrow{k_{N-1}} \mathrm{DNA}_{\mathrm{ON}} \\ &\mathrm{DNA}_{\mathrm{ON}} \xrightarrow{k_{\mathrm{off}}} \mathrm{DNA}_{\mathrm{OFF}_1}, \end{aligned} \tag{3}$$

where the promoter exhibits a multi-step OFF to ON transition with $N-1$ OFF states and a single ON state. In contrast to (1), the distribution of time spent in OFF states between transcription events is now non-exponential. Specifically, these time increments follow the hypoexponential distribution, which approaches an Erlang distribution (Evans *et al.*, 2000) when the switching rates $(k_1, \ldots, k_{N-1})$ are roughly equal.

The construction of a simplified transcriptional bursting model of (3) that mirrors (2) requires the generation of inter-burst arrival times from an arbitrary distribution. This can be realized by making the burst frequency time dependent (Fig. 1a). Let $\omega(t)$ denote the state of a timer at time t measuring time elapsed since the last burst event: $\omega(t)$ is reset to zero whenever bursts occur and $\mathrm{d}\omega/\mathrm{d}t = 1$ in between burst events. The desired inter-burst arrival time cumulative distribution function (CDF) $F(\cdot)$ is obtained by setting the propensity function of the transcription reaction to $F'(\omega)/(1-F(\omega))$, a quantity also known as the hazard function or failure rate (Evans *et al.*, 2000). Thus, the probability that an mRNA burst event will occur in the next time interval $[t, t+\mathrm{d}t)$ is $F'(\omega)/(1-F(\omega))\mathrm{d}t$.

In this work, we consider transcription propensity functions of the form $c_1\omega(t)^{c_2}$, where $c_1 > 0$ and $c_2 \geq 0$. The corresponding simplified transcription reaction is represented as:

$$\mathrm{DNA}_{\mathrm{ON}} \xrightarrow{c_1\omega(t)^{c_2}} \mathrm{DNA}_{\mathrm{ON}} + \beta \times \mathrm{mRNA}. \tag{4}$$

For this class of propensity functions, the inter-burst arrival time distribution can be obtained as follows:

$$\frac{\mathrm{d}F(\omega)}{\mathrm{d}\omega}\frac{1}{1-F(\omega)} = c_1\omega^{c_2} \tag{5}$$

$$\Rightarrow F(\omega) = 1 - \exp\left(-\frac{c_1}{c_2+1}\omega^{c2+1}\right). \tag{6}$$

Equation (6) is the CDF of the Weibull distribution with shape parameter k_{W} and scale parameter λ_{W}, where $c_1 = k_{\mathrm{W}}/\lambda_{\mathrm{W}}^{k_{\mathrm{W}}}$ and $c_2 = k_{\mathrm{W}} - 1$ (Evans *et al.*, 2000). The mean (represented by $E(\cdot)$) and coefficient of variation squared ($\mathrm{CV}^2 \equiv \mathrm{variance/mean}^2$) of this distribution can be expressed as follows:

$$E(\omega(t_k)) = \left(\frac{c_2+1}{c_1}\right)^{\frac{1}{c_2+1}}\Gamma\left(\frac{c_2+2}{c_2+1}\right) \tag{7}$$

$$CV^2(\omega(t_k)) = \frac{\Gamma\left(\frac{c_2+3}{c_2+1}\right)}{\Gamma^2\left(\frac{c_2+2}{c_2+1}\right)} - 1, \tag{8}$$

respectively, where t_k denotes the kth mRNA burst time and $\Gamma(\cdot)$ is the gamma function. When $c_2 = 0$ (time-independent burst frequency), the Weibull distribution reduces to an exponential distribution and $\mathrm{CV}^2 = 1$. As c_2 increases, CV^2 monotonically decreases to zero. Thus, for large values of c_2, mRNA bursts arrive at deterministic time intervals. The Weibull distribution is known to provide an accurate approximation of the Erlang distribution (Malhotra and Reibman, 1993). This property allows us to mimic the behavior of an N-state promoter with roughly equal switching rates (3) using a simplified model of transcription with the propensity function $c_1\omega(t)^{c_2}$. Specifically, by increasing the value of c_2, we can simulate expression from promoters with larger numbers of states. The relationship between c_2 and N is given by the following expression, which is obtained by equating the CV^2 of the Weibull and Erlang distributions:

$$\frac{\Gamma\left(\frac{c_2+3}{c_2+1}\right)}{\Gamma^2\left(\frac{c_2+2}{c_2+1}\right)} = \frac{N}{N-1}, \tag{9}$$

where N is one greater than the Erlang shape parameter (k_{E} in the Erlang CV^2 formula: $1/k_{\mathrm{E}}$).

In model (2) and its modification (4), we assume that mRNA bursts are instantaneous. This assumption can be relaxed by generalizing (2) to:

$$\begin{aligned} &\mathrm{DNA}_{\mathrm{OFF}} \underset{c_{12}\omega(t)^{c_{22}}}{\overset{c_{11}\omega(t)^{c_{21}}}{\rightleftharpoons}} \mathrm{DNA}_{\mathrm{ON}} \\ &\mathrm{DNA}_{\mathrm{ON}} \xrightarrow{k_{\mathrm{m}}} \mathrm{DNA}_{\mathrm{ON}} + \mathrm{mRNA} \\ &\mathrm{mRNA} \xrightarrow{\gamma_{\mathrm{m}}} \emptyset \end{aligned} \tag{10}$$

where (c_{11}, c_{21}) and (c_{12}, c_{22}) contribute to the first and second reaction propensity, respectively, and the timer $\omega(t)$ resets to zero each time the promoter transitions between the OFF and ON states (Fig. 1b). By choosing the transition rates to be monomials in $\omega(t)$, the promoter resides in each state for a Weibull-distributed time interval. Setting $c_{21} = c_{22} = 0$ recovers the standard random telegraph model where promoter ON and OFF times are exponentially

distributed random variables. Note that when $c_{22} = 0$, in the limit of large c_{12} (i.e. promoter ON state is unstable), (10) reduces to the instantaneous mRNA burst model [modification (4) of (2)]

2.2 Bursty MCEM^2

Single-cell time course datasets are necessarily incomplete—at best, they provide the numbers of molecules for a subset of the system species at discrete time points. Because of the intractability of the incomplete data likelihood for stochastic biochemical systems (Wilkinson, 2006), the direct calculation of maximum likelihood parameter estimates (MLEs) is only possible given complete data. In light of this, we resort to an iterative, simulation-based approach for computing MLEs using the MCEM algorithm. Previously, we developed MCEM^2 (MCEM with modified cross-entropy method), a computationally efficient approach for estimating parameters of stochastic biochemical systems given incomplete data (Daigle *et al.*, 2012). We describe this approach in more detail in Supplementary Section S1, and below we present the modifications needed to accommodate the models of transcriptional bursting discussed in the previous section.

Given our time-dependent transcriptional model reduction, it is relatively straightforward to construct closed-form update formulas that enable MCEM^2 to infer maximum likelihood estimates for c_1 and the geometrically distributed burst size parameter c_3 ($\tilde{c}_1$ and $\tilde{c}_3$, respectively). Because of the occurrence of c_2 in an exponent, it is not possible to derive closed-form update formulas for this parameter, so we fix its value in the following derivation and select most probable values for c_2 separately via model selection. Specifically, we use MCEM^2 to identify which value of c_2 provides the lowest Akaike information criterion (AIC), a model selection score that quantifies both data likelihood and model complexity (see Supplementary Section S3 for details).

We begin by representing each of the system's M reaction propensity functions (where M is the total number of reactions in the system) as an explicitly state- and time-dependent function $a_j(t) = a_j(X(t), t)$ defined as follows:

$$a_j(X(t), t) \equiv c_{1j} h_j(X(t)) \times \omega_j(t)^{c_{2j}}, \tag{11}$$

where $X(t)$ represents the state of the system at time t, $h_j(X(t))$ is the system state-dependent portion of the propensity function and $\omega_j(t)$ is a reaction-specific timer that resets to zero each time the jth reaction fires. We note that c_{1j}, c_{2j} and $\omega_j(t)$ are designated with arbitrary subscripts only for the purposes of the below derivation; in practice, each promoter is associated with only one timer $\omega(t)$ and at most two values each of c_1 and c_2. For system reactions with mass action propensity functions (e.g. $a_j(t) = a_j(X(t)) = \theta_j h_j(X(t))$), the representation of Equation (11) is achieved by relabeling θ_j as c_{1j} and setting $c_{2j} = 0$. For the remaining system reactions that exhibit time-dependent propensities as outlined in the above model formulation, $h_j(X(t)) = 1$ and c_{2j} is nonzero.

With all reaction propensities in the form of Equation (11), we can use the results of Anderson (2007) to express the distribution function of the time interval τ until the next reaction firing, given the system at the current time t, as:

$$F(\tau, t) = 1 - \exp\left(-\sum_{j=1}^{M} \int_t^{t+\tau} c_{1j} h_j(X(t)) \times \omega_j(s)^{c_{2j}} \mathrm{d}s\right), \tag{12}$$

where $X(t)$ is constant in the integral because no reactions take place within the time interval $[t, t+\tau)$. After differentiating Equation (12) with respect to τ, integrating and simplifying (given $\omega_j(t+\tau) = \omega_j(t) + \tau$), we obtain the corresponding density function:

$$\begin{aligned} f(\tau, t) = &\left(\sum_{j=1}^{M} c_{1j} h_j(X(t)) \times (\omega_j(t) + \tau)^{c_{2j}}\right) \\ &\times \exp\left(-\sum_{j=1}^{M} c_{1j} h_j(X(t)) \times \frac{(\omega_j(t) + \tau)^{\bar{c}_{2j}} - \omega_j(t)^{\bar{c}_{2j}}}{\bar{c}_{2j}}\right), \end{aligned} \tag{13}$$

where $\bar{c}_{2j} = c_{2j} + 1$. Given the time of the next reaction firing $t + \tau$, we can also express the probability mass function of the index of the next fired reaction j' as a simple categorical probability:

$$p(j', \tau, t) = \frac{c_{1j'} h_{j'}(X(t)) \times (\omega_{j'}(t) + \tau)^{c_{2j'}}}{\sum_{j=1}^{M} c_{1j} h_j(X(t)) \times (\omega_j(t) + \tau)^{c_{2j}}}. \tag{14}$$

Taken together, Equations (12–14) allow us to represent the likelihood of a fully observed stochastic trajectory as the following product:

$$\prod_{i=0}^{r_k - 1} \left(f(\tau_i, t_i) \times p(j'_{i+1}, \tau_i, t_i)\right) \times (1 - F(\tau_{r_k}, t_{r_k})), \tag{15}$$

where i indexes the events including the start of each simulation ($i = 0$), the total number of overall reaction firings r_k and arrival at the final time ($i = r_k + 1$), τ_i is the time interval between the ith and $(i+1)$th events, t_i represents the time immediately after the ith event and j'_i is the index of the ith reaction to fire. By taking the logarithm of Equation (15), differentiating with respect to c_{1j}, solving for the unique root and averaging across K' simulated trajectories, we obtain a closed-form Monte Carlo update formula:

$$\hat{c}_{1j}^{(1)} = \hat{c}_{1j}^{(0)} \times \frac{\sum_{k=1}^{K'} r_{jk}}{\sum_{k=1}^{K'} \left(\sum_{i=0}^{r_k} \hat{c}_{1j}^{(0)} h_j(X_k(t_{ik})) \times \frac{(\omega_{jk}(t_{ik}) + \tau_{ik})^{\bar{c}_{2j}} - \omega_{jk}(t_{ik})^{\bar{c}_{2j}}}{\bar{c}_{2j}}\right)}, \tag{16}$$

where $\hat{c}_{1j}^{(0)}$ and $\hat{c}_{1j}^{(1)}$ represent the initial guess and first update, respectively, for parameter c_{1j}, k indexes the K' simulated trajectories and r_{jk} is the number of times the jth reaction fires. The equivalence of Equation (16) and the update formula for parameters from mass action reactions [Equation (1) in the Supplementary Information] can be seen by setting $c_{2j} = 0$ and relabeling $\hat{c}_{1j}^{(0)}, \hat{c}_{1j}^{(1)}$ as $\hat{\theta}_j^{(0)}, \hat{\theta}_j^{(1)}$. For those reactions with time-dependent propensities, Equation (16) provides a closed-form expression for inferring the maximum likelihood estimate of c_{1j} using MCEM^2.

For time-dependent models that incorporate a geometrically distributed transcriptional burst reaction, inference of parameter c_{3j} proceeds by first modifying Equation (15) as follows:

$$\begin{aligned} &\prod_{i=0}^{r_k - 1} \left(f(\tau_i, t_i) \times p(j'_{i+1}, \tau_i, t_i) \times g(\beta_{i+1}, j'_{i+1})\right) \\ &\quad \times (1 - F(\tau_{r_k}, t_{r_k})), \end{aligned} \tag{17}$$

where $g(\beta, j')$ represents the geometric probability mass function evaluated at a burst size of β ($(1 - c_{3j'})^{\beta} c_{3j'}$) if j' is a transcriptional burst reaction and 1 otherwise. Following the same procedure as for c_{1j}, we obtain a closed-form update formula for c_{3j}:

$$\hat{c}_{3j}^{(1)} = \hat{c}_{3j}^{(0)} \times \frac{\sum_{k=1}^{K'} r_{jk}}{\sum_{k=1}^{K'} \left(r_{jk} + \sum_{i=1}^{r_k} \mathbb{1}_j(j'_{ik}) \times \beta_{ik}\right)}, \tag{18}$$

where $\mathbb{1}_j(j')$ is an indicator function that takes a value of 1 if $j' = j$ (0 otherwise).

Taken together, the above derivations combined with MCEM^2 enable the construction of 'bursty MCEM^2', a novel parameter estimation and model selection framework for inferring the number and configuration of promoter states from single-cell expression data. Supplementary Section S2 provides additional details of the method.

2.3 Simulation study

To illustrate the accuracy of our model reduction along with bursty MCEM^2's ability to select a plausible multi-state transcriptional model given observed data, we first performed a simulation study using three models with different numbers of promoter states. Model (A) represents a single-state promoter, which exhibits constant (non-bursty) production of mRNA:

$$\begin{aligned} \mathrm{DNA_{ON}} &\xrightarrow{5.0} \mathrm{DNA_{ON}} + \mathrm{mRNA} \\ \mathrm{mRNA} &\xrightarrow{.0001} \emptyset. \end{aligned} \tag{A}$$

In the above system, the transcription reaction produces an average of five mRNA molecules per unit of time, whereas mRNAs are degraded at a much lower rate. Model (B) contains a promoter with three states—two OFF states and a very short-lived ON state—which exhibits bursty mRNA production. It modifies (A) by replacing the first reaction with the following:

$$\begin{aligned} \mathrm{DNA_{OFF_1}} &\xrightarrow{2.0202} \mathrm{DNA_{OFF_2}} \xrightarrow{2.0202} \mathrm{DNA_{ON}} \\ \mathrm{DNA_{ON}} &\xrightarrow{100.0} \mathrm{DNA_{OFF_1}} \\ \mathrm{DNA_{ON}} &\xrightarrow{500.0} \mathrm{DNA_{ON}} + \mathrm{mRNA}. \end{aligned} \tag{B}$$

As before, the rates of (B) were chosen so that the promoter would produce five mRNA molecules per time unit on average. However, rather than exhibiting constant production, the promoter switches from OFF to ON once per time unit on average and quickly outputs a burst of mRNA molecules with mean burst size $500.0/100.0 = 5$ before returning to OFF. Finally, model (C) contains a promoter with six states that also exhibits bursty mRNA production. It modifies (B) by replacing the first two reactions with the following:

$$\begin{aligned} \mathrm{DNA_{OFF_1}} &\xrightarrow{5.0505} \mathrm{DNA_{OFF_2}} \xrightarrow{5.0505} \mathrm{DNA_{OFF_3}} \\ \mathrm{DNA_{OFF_3}} &\xrightarrow{5.0505} \mathrm{DNA_{OFF_4}} \xrightarrow{5.0505} \mathrm{DNA_{OFF_5}} \\ \mathrm{DNA_{OFF_5}} &\xrightarrow{5.0505} \mathrm{DNA_{ON}}. \end{aligned} \tag{C}$$

As in model (B), this promoter switches from OFF to ON once per time unit on average, and it outputs bursts of mRNA with mean size 5 before returning to OFF. The mRNA degradation rates of all three models are identical.

For each model, we first simulated a single trajectory over 100 time units and recorded the number of mRNA molecules at 400 equally spaced intervals. The starting conditions for each simulation were 0 mRNA molecules and 1 promoter in states $\mathrm{DNA_{ON}}, \mathrm{DNA_{OFF_1}}$ and $\mathrm{DNA_{OFF_1}}$ for models (A), (B) and (C), respectively. Supplementary Figure S1 (Supplementary Section S5) displays the three simulated trajectories.

Given each synthetic data trajectory, we used our time-dependent transcriptional model reduction with bursty MCEM^2 to infer the unknown parameters (c_1, c_2, c_3) from models (D) and (E):

$$\begin{aligned} \mathrm{DNA_{ON}} &\xrightarrow{c_1} \mathrm{DNA_{ON}} + \mathrm{mRNA} \\ \mathrm{mRNA} &\xrightarrow{.0001} \emptyset \end{aligned} \tag{D}$$

$$\begin{aligned} \mathrm{DNA_{ON}} &\xrightarrow{c_1\omega(t)^{c_2}} \mathrm{DNA_{ON}} + \beta \times \mathrm{mRNA} \\ \mathrm{mRNA} &\xrightarrow{.0001} \emptyset \\ \beta &\sim \mathrm{Geometric}(c_3), \end{aligned} \tag{E}$$

where the mRNA degradation rate is given. The general version of model (E) is shown in Figure 1a. We used bursty MCEM^2 to perform parameter inference and model selection on each data trajectory using model (D) and three versions of model (E) with different values of c_2: (i) $c_2 = 0$, (ii) $c_2 = 0.4$ and (iii) $c_2 = 1.4$. These four model parameterizations correspond to promoters with one, two, three and six states, respectively. We set the initial guesses for parameters c_1 and c_3 to 1 and 0.5, respectively. Since the initial value of the reaction clock $\omega(0)$ was unobserved, we set it to 0 for all bursty MCEM^2 model simulations. Table 1 lists the resulting MLEs $(\tilde{c}_1, \tilde{c}_3)$, log likelihoods, AIC values and relative likelihoods for each data-model combination.

As this table shows, bursty MCEM^2 selected the model with the correct number of states as most probable for each synthetic data trajectory. For data from the one- (A) and six-state (C) models, competing models did not achieve a relative likelihood within an order of magnitude of a 0.368 cutoff (relative likelihood difference of 2 AIC units from best fitting model); for data from the three-state model (B), the two-state version of model (E) provided a second-best fit that was still substantially worse (likelihood = 0.052) than the cutoff. When inferring parameters of the one-state model (D) using data from model (A), $\tilde{c}_1 = 4.81$ is close to the true value of 5 and $\tilde{c}_1$'s 68% confidence interval (4.59–5.03) includes the true value. Similarly, the inferred mean burst sizes for the most probable models fitting data from models (B) and (C) (5.37 and 4.20, respectively) are close to the true value of 5 and the 68% and 95% confidence intervals, respectively, of these two estimates ([4.73–6.09] and [3.28–5.32]) both include the true value. Finally, we note that results for model (D) were unavailable when fitting data from models (B) and (C), as the lack of transcriptional bursts in a one-state model precluded the simulation of trajectories matching data with bursts.

2.4 Glutaminase data inference

Next, we applied bursty MCEM^2 to actual time-lapse microscopy data from a reporter gene driven by a mammalian promoter. Suter *et al.* (2011) measured gene expression at eight endogenous mouse promoters upstream of luciferase reporter genes. Using these data, the authors estimated the values of transcriptional parameters by modeling each promoter as occupying two or three states. Our goal was to use our model reduction with bursty MCEM^2 to infer whether one of these promoters likely occupies more than three states. To this end, we extracted a single trajectory of luminescence data collected once every 5 min from the glutaminase promoter (Fig. 1C in Suter *et al.*, 2011) and performed data smoothing and calibration to convert light intensity values to numbers of proteins (see Supplementary Section S4 for details). Figure 2 displays the glutaminase trajectory before and after preprocessing.

We then used bursty MCEM^2 to infer the unknown parameters $(c_1, c_2, c_3, k_{\mathrm{on}}, k_{\mathrm{off}}, k_{\mathrm{m}})$ from models (F), (G), (H) and (I) given the data from Figure 2b:

$$\begin{aligned} \mathrm{DNA_{ON}} &\xrightarrow{c_1} \mathrm{DNA_{ON}} + \mathrm{mRNA} \\ \mathrm{mRNA} &\xrightarrow{.924} \emptyset \\ \mathrm{mRNA} &\xrightarrow{12.6} \mathrm{mRNA} + \mathrm{Protein} \\ \mathrm{Protein} &\xrightarrow{1.98} \emptyset \end{aligned} \tag{F}$$

Table 1. Bursty MCEM^2 parameter inference and model selection results for simulation study

No. states data	No. states model	c_2	$\tilde{c}_1$	$\frac{1-e_3}{e_3}$	Log likelihood	AIC	Relative likelihood
1 (A)	**1 (D)**	**NA**	**4.81**	**NA**	**−566.94**	**1135.87**	**1**
	2 (E)	0	41.43	0.12	−569.95	1145.91	0.0066
	3 (E)	0.4	139.67	0.16	−570.50	1146.99	0.0038
	6 (E)	1.4	1990.31	0.26	−572.42	1150.85	0.00056
3 (B)	1 (D)	NA	NA	NA	NA	NA	NA
	2 (E)	0	0.97	5.04	−402.32	810.65	0.052
	3 (E)	**0.4**	**1.08**	**5.37**	**−399.37**	**804.74**	**1**
	6 (E)	1.4	1.48	5.18	−402.91	811.82	0.029
6 (C)	1 (D)	NA	NA	NA	NA	NA	NA
	2 (E)	0	1.19	3.60	−421.53	849.05	1.14×10^{-7}
	3 (E)	0.4	1.40	4.00	−412.58	831.16	0.00088
	6 (E)	**1.4**	**1.91**	**4.20**	**−405.54**	**817.08**	**1**

Corresponding model letter is displayed next to the number of states in the first and second columns. $\tilde{c}_1$ and $\tilde{c}_3$ represent MLEs for c_1 and c_3, respectively. The expression $(1-\tilde{c}_3)/\tilde{c}_3$ represents mean burst size for models with transcriptional burst reactions. True values for c_1 and $(1-c_3)/c_3$ are both equal to 5. Relative likelihoods are computed separately for each data trajectory; results in bold represent most probable models. Values of 'NA' for c_2 and $(1-\tilde{c}_3)/\tilde{c}_3$ due to model (D) lacking these parameters; remaining NA values from fitting data of models (B) and (C) due to inability of one-state model to generate trajectories matching data with bursts.

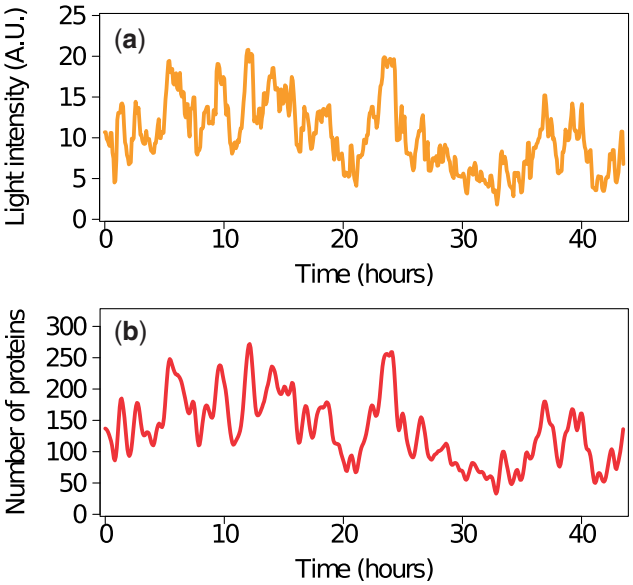


Fig. 2. Glutaminase promoter time-lapse microscopy data from Suter *et al.* (2011). (**a**) before and (**b**) after data smoothing and calibration. Data consist of 539 measurements sampled approximately once every 5 min for 43.5 h. AU, arbitrary units

$$\begin{aligned} &\mathrm{DNA_{ON}} \xrightarrow{c_1\omega(t)^{c_2}} \mathrm{DNA_{ON}} + \beta \times \mathrm{mRNA} \\ &\beta \sim \mathrm{Geometric}(c_3) \end{aligned} \tag{G}$$

$$\begin{aligned} &\mathrm{DNA_{OFF}} \underset{c_1\omega(t)^{c_2}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} \mathrm{DNA_{ON}} \\ &\mathrm{DNA_{ON}} \xrightarrow{k_{\mathrm{m}}} \mathrm{DNA_{ON}} + \mathrm{mRNA} \end{aligned} \tag{H}$$

$$\begin{aligned} &\mathrm{DNA_{OFF}} \underset{k_{\mathrm{off}}}{\overset{c_1\omega(t)^{c_2}}{\rightleftharpoons}} \mathrm{DNA_{ON}} \\ &\mathrm{DNA_{ON}} \xrightarrow{k_{\mathrm{m}}} \mathrm{DNA_{ON}} + \mathrm{mRNA}, \end{aligned} \tag{I}$$

where the latter three models also include the mRNA degradation, protein translation and protein degradation reactions of model (F) with given rate constants. Models (F) and (G) are similar in structure to models (D) and (E), respectively, with protein translation and degradation reactions added. Models (H) and (I) represent promoters with multi-step ON to OFF and OFF to ON transitions, respectively. Figure 1b represents models (H) and (I) by selecting appropriate values for c_{11}, c_{21}, c_{12} and c_{22}. Like models (B) and (C), these two models exhibit burst-like transcription with the proper parameterization, and they do not assume instantaneous production of bursts. All four models share fixed, identical rates of mRNA degradation (derived from the 45-min glutaminase reporter mRNA half life experimentally determined by Suter *et al.*, 2011), protein degradation (derived from the 21 minute luciferase protein half life experimentally determined by Suter *et al.*, 2011) and protein translation (reported in Molina *et al.*, 2013). We performed model selection over all models, using fixed values of c_2 (when present) ranging from 0 to 11. Altogether, these models approximate the behavior of promoters containing 1–101 states. We set the initial guesses for all other unknown parameters to 1 (except for c_3, which was set to .5). As before, we set the initial value of the reaction clock $\omega(0)$ to 0 for all bursty MCEM^2 model simulations. In addition, we set the unobserved initial state of the promoter for models (H) and (I) to $\mathrm{DNA_{OFF}}$. Finally, for the unobserved initial number of mRNA molecules, we tried values from zero to 30 in increments of five. However, as 20 molecules allowed the simulation of trajectories with the largest observed data likelihood, we used this number for all bursty MCEM^2 model simulations. Supplementary Tables S1 and S2 (Supplementary Section S5) and Table 2 lists the resulting parameter estimates, log likelihoods, AIC values and relative likelihoods (calculated across all models) for versions of models (F), (G), (H) and (I).

As Supplementary Tables S1 and S2 show, no versions of models (F), (G) or (H) provide plausible fits to the data. From these results, we predict that the glutaminase promoter (i) occupies at least one OFF state, (ii) does not exhibit instantaneous bursting of mRNA and (iii) does not undergo a multi-step ON to OFF transition with a single OFF state. Instead, the results in Table 2 suggest that model (I) exhibiting a multi-step OFF to ON transition with a single ON state best fits the data, and we predict that the glutaminase promoter occupies between 10 and 50 OFF states (model versions with relative likelihood $\geq$.368) in the process of transitioning to a single ON state. Once in the ON state, our results predict a mean

Table 2. Bursty MCEM^2 parameter inference and model selection results for model (I) using glutaminase data

No. states	c_2	$\tilde{c}_1$	$\tilde{k}_{off}$	$\tilde{k}_m$	Log likelihood	AIC	Relative likelihood
2	0	2.14	5.52	75.52	−1738.01	3484.02	0.000047
3	0.40	2.78	4.26	72.64	−1734.57	3477.15	0.0015
4	0.75	3.50	3.96	72.62	−1732.58	3473.16	0.011
5	1.10	4.33	3.67	71.15	−1731.10	3470.19	0.047
6	1.40	5.26	3.68	71.89	−1730.33	3468.67	0.10
7	1.65	6.30	3.60	70.85	−1729.75	3467.49	0.18
9	2.10	8.66	3.69	72.10	−1729.08	3466.17	0.35
11	**2.50**	**11.44**	**3.62**	**71.12**	**−1728.60**	**3465.19**	**0.58**
16	**3.40**	**19.36**	**3.52**	**70.23**	**−1728.06**	**3464.13**	**0.98**
21	**4.10**	**28.40**	**3.35**	**68.14**	**−1728.06**	**3464.11**	**0.99**
26	**4.75**	**40.28**	**3.30**	**67.62**	**−1728.05**	**3464.09**	**1**
51	**7.40**	**161.84**	**2.84**	**61.65**	**−1728.77**	**3465.53**	**0.49**
101	11.00	1019.21	2.49	56.74	−1729.18	3466.36	0.32

$\tilde{c}_1$, $\tilde{k}_{off}$ and $\tilde{k}_m$ represent MLEs for c_1, k_{off} and k_m, respectively. Relative likelihoods are computed across all versions of models (F), (G), (H) and (I); results in bold font represent models that best fit the data (relative likelihood $\geq$0.368).

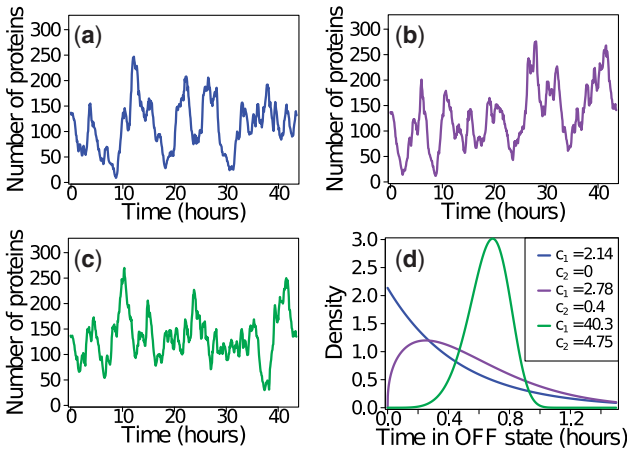


Fig. 3. Differences between versions of model (I) with varying numbers of OFF states. Simulated trajectories using the MLEs from Table 2 for (**a**) two-state, (**b**) three-state and (**c**) most probable 26-state model versions. (**d**) Comparison of OFF state dwell time distributions between model versions [obtained by differentiating the Weibull distribution function (6)]. Two- and three-state promoter models exhibit more variable OFF times than 26-state promoter model. Values of c_1 and c_2 used for each model version are shown

transcriptional burst size of ~20 ($\tilde{k}_m/\tilde{k}_{off}$) before returning to the first OFF state.

These predictions are in stark contrast to the two- and three-state promoter models used in Suter *et al.* (2011) to model the glutaminase promoter. To illustrate the differences between these models, Figure 3 displays simulated trajectories and OFF state dwell time distributions from versions of model (I) with 2, 3 and most probable 26 states (and the corresponding MLEs from Table 2). As Figure 3d shows, the 26-state version of model (I) exhibits much less variable OFF state dwell times than the two- or three-state versions. This is reflected in the tendency of the trajectory in Figure 3c to stay above a small number of proteins (~25), while the trajectories in Figure 3a and b occasionally decrease to near zero. The glutaminase data trajectory in Figure 2b also consistently remains above 25 proteins, providing visual evidence for the superior fit of a 26-state promoter model to the data.

3 Discussion

In this work, we present a novel model reduction for multi-state promoters along with an efficient computational technique for inferring the number and configuration of promoter states from single-cell gene expression data. Specifically, we first developed a time-dependent reaction propensity function for transcriptional bursting that generates Weibull-distributed ON/OFF promoter state dwell times. Using this function, we demonstrated an ability to closely approximate the behavior of promoters undergoing multi-step transitions between OFF and ON states. Next, we created 'bursty MCEM^2' which, when combined with our model reduction, allows for computationally efficient parameter inference and model selection across a wide range of promoter architectures.

As an example of the computational savings conferred, we note that modeling a promoter transitioning between nine OFF states and one ON state requires specification of 10 switching parameters and simulation of 10 chemical reactions per transcriptional cycle. In contrast, use of our time-dependent propensity function introduces only three parameters and requires the simulation of only two reactions per cycle. These computational and parametric savings increase linearly with the number of promoter states modeled, enabling efficient analysis of arbitrarily complex promoter architectures with bursty MCEM^2.

Through a simulation study, we demonstrated the ability of bursty MCEM^2 to correctly identify the number of promoter states used to generate single-cell time-series trajectories. In addition, we showed that our method accurately infers the values of kinetic parameters within the chosen promoter model. We then applied bursty MCEM^2 to experimental time-lapse microscopy data from a reporter gene driven by the endogenous mouse glutaminase promoter (Suter *et al.*, 2011). Our results suggest that, rather than occupying two or three states as previously described, the glutaminase promoter likely traverses through 10 or more OFF states before transcribing mRNA from an ON state. We hypothesize that a distinct advantage of a promoter architecture with >10 states lies in its potential noise buffering properties. As the number of states increases, promoter OFF times become more deterministic, leading to more consistent rates of mRNA production. As shown in Figure 3c, this leads to less variable protein levels over time (particularly at lower levels), which can confer more robust behavior in response to stochastic perturbations. Because of the intrinsic noise present in the transcriptional and translational machinery, we note that the confidence interval accompanying our estimated number of glutaminase promoter OFF states is somewhat large (10–50). The addition of measurements at later time points would certainly help reduce this uncertainty. However, if protein measurements were replaced with direct quantification of mRNA, we would expect a much more precise estimate, since translational noise would no longer affect our analysis. Results from the simulation study support this, where we obtained more precise estimates of numbers of promoter states by performing inference on mRNA data.

When discussing particular numbers of promoter states within our model reduction, it is important to note that these values represent 'effective' numbers of states rather than distinct biochemical configurations. In particular, since the correspondence between the parameters of the Weibull distribution and the number of promoter states (4) is most accurate when the switching rates are equal, the effective number of states is likely an underestimation of the true number. This follows from the observation that as the switching rates depart from equality, the slowest promoter transitions become rate-limiting and thus mask the presence of faster transitions. Given that

one of our goals in this work was to demonstrate that mammalian promoters occupy more than two or three states, this discrepancy only strengthens the conclusions drawn from our results.

In conclusion, we anticipate that our novel approach to modeling promoter fluctuations together with bursty MCEM^2 provides powerful tools for characterizing transcriptional bursting across genes under different environmental conditions. Future work will focus on discovering general transcriptional regulatory principles by applying these methods to single-cell expression data from a wide range of promoters.

Funding

A.S. was supported by the National Science Foundation Grant [DMS-1312926], University of Delaware Research Foundation (UDRF) and Oak Ridge Associated Universities (ORAU). B.J.D. and L.R.P. were supported by NIH RO1-EB014877, DOE DE-SC0008975 and the Institute for Collaborative Biotechnologies through grant [W911NF-09-0001] from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. We also acknowledge computing support from the UCSB Center for Scientific Computing from the CNSI, MRL: an NSF MRSEC (DMR-1121053) and NSF CNS-0960316.

Conflict of Interest: none declared.

References

Anderson,D.F. (2007) A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *J. Chem. Phys.,* **127**, 214107.

Cai,L. *et al.* (2006) Stochastic protein expression in individual cells at the single molecule level. *Nature,* **440**, 358–362.

Choi,P.J. *et al.* (2008) A stochastic single-molecule event triggers phenotype switching of a bacterial cell. *Science,* **322**, 442–446.

Daigle,B.J. Jr. *et al.* (2012) Accelerated maximum likelihood parameter estimation for stochastic biochemical systems. *BMC Bioinformatics,* **13**, 68.

Dar,R.D. *et al.* (2012) Transcriptional burst frequency and burst size are equally modulated across the human genome. *Proc. Natl Acad. Sci. USA,* **109**, 17454–17459.

Evans,M. *et al.* (2000) *Statistical Distributions*. 3rd edn. Wiley, New York, NY.

Gillespie,D.T. (2007) Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.,* **58**, 35–55.

Golding,I. *et al.* (2005) Real-time kinetics of gene activity in individual bacteria. *Cell,* **123**, 1025–1036.

Harper,C.V. *et al.* (2011) Dynamic analysis of stochastic transcription cycles. *PLoS Biol.,* **9**, e1000607.

Lee,T.I. and Young,R.A. (2013) Transcriptional regulation and its misregulation in disease. *Cell,* **152**, 1237–1251.

Malhotra,M. and Reibman,A. (1993) Selecting and implementing phase approximations for semi-Markov models. *Commun. Statist.-Stochastic Models,* **9**, 473–506.

Molina,N. *et al.* (2013) Stimulus-induced modulation of transcriptional bursting in a single mammalian gene. *Proc. Natl Acad. Sci. USA*, **110**, 20563–20568.

Munsky,B. *et al.* (2009) Listening to the noise: random fluctuations reveal gene network parameters. *Mol. Syst. Biol.,* **5**, 318.

Raj,A. *et al.* (2006) Stochastic mRNA synthesis in mammalian cells. *PLoS Biol.,* **4**, e309.

Raser,J.M. and O'Shea,E.K. (2005) Noise in gene expression: origins, consequences, and control. *Science,* **309**, 2010–2013.

Sanchez,A. *et al.* (2013) Stochastic models of transcription: From single molecules to single cells. *Methods*, **62**, 13–25.

Singh,A. (2014) Transient changes in intercellular protein variability identify sources of noise in gene expression. *Biophys. J.,* **107**, 2214–2220.

Singh,A. *et al.* (2010) Transcriptional bursting from the HIV-1 promoter is a significant source of stochastic noise in HIV-1 gene expression. *Biophys. J.,* **98**, L32–L34.

So,L.-H. *et al.* (2011) General properties of transcriptional time series in *Escherichia coli*. *Nat. Genet.,* **43**, 554–60.

Suter,D.M. *et al.* (2011) Mammalian genes are transcribed with widely different bursting kinetics. *Science,* **332**, 472–4.

Toni,T. *et al.* (2009) Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface,* **6**, 187–202.

Wilkinson,D.J. (2006) *Stochastic Modelling for Systems Biology*. Chapman and Hall/CRC Mathematical and Computational Biology Series. Taylor and Francis Group, Boca Raton, FL.

Yu,J. *et al.* (2006) Probing gene expression in live cells, one protein molecule at a time. *Science,* **311**, 1600–1603.

Yuh,C.H. *et al.* (1998) Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science,* **279**, 1896–902.

Zhang,J. *et al.* (2012) Analytical distribution and tunability of noise in a model of promoter progress. *Biophys. J.,* **102**, 1247–57.

Inferring Single-Cell Gene Expression Mechanisms using Stochastic Simulation
Supplementary Information

Bernie J. Daigle, Jr., Mohammad Soltani, Linda R. Petzold, and Abhyudai Singh

November 4, 2014

S1 Monte Carlo Expectation-Maximization with Modified Cross-Entropy Method (MCEM^2)

MCEM^2 computes maximum likelihood parameter estimates (MLEs) and associated uncertainties in three consecutive phases—cross-entropy (CE), Monte Carlo expectation-maximization (MCEM), and uncertainty quantification (UQ) [1]. Given an initial guess for the unknown parameters, the CE phase begins by simulating (using the "direct method" implementation of Gillespie's stochastic simulation algorithm (SSA) [2]) K model trajectories from a completely observed system state at the initial time until the time of the first observed data point. To increase sampling diversity, we slightly perturb the parameter values used to generate each trajectory with a random multiplicative perturbation drawn uniformly from the interval $[-\delta, \delta]$. Next, we compute the distance from each trajectory to the observed data and retain the $K' = \rho \times K$ closest trajectories, whose final states are sampled with replacement to generate initial conditions for the next round of simulations. We then continue simulating K trajectories from the current time until the time of the next data point (with newly perturbed parameters), repeating the above steps until finally reaching the final time. After one final selection of the K' closest trajectories, we join together the trajectory segments leading to each final trajectory, which leaves us with K' full-length, partially correlated trajectories that can be used to update each mass action reaction parameter estimate $\hat{\theta}_j$:

$$\hat{\theta}_j^{(1)} = \hat{\theta}_j^{(0)} \times \frac{\sum_{k=1}^{K'} r_{jk}}{\sum_{k=1}^{K'} \left(\sum_{i=0}^{r_k} a_{jk}^i \times \tau_{ik} \right)} \quad , \tag{1}$$

where $\hat{\theta}_j^{(0)}$ and $\hat{\theta}_j^{(1)}$ represent the initial guess and first update, respectively, for parameter θ_j, k indexes the K' full-length trajectories, i indexes the events including the start of the simulation ($i=0$), the total number of overall reaction firings r_k, and arrival at the final time ($i=r_k+1$), r_{jk} is the number of times the j^{th} reaction fires, a_{jk}^i is the value of the propensity function for the j^{th} reaction immediately after the i^{th} event, and τ_{ik} is the time interval between the i^{th} and $(i+1)^{\text{th}}$ events.

The CE phase repeats the above procedure for C iterations, each time computing the geometric mean of the proportion of simulated trajectories that hit each observed data point ($\bar{\pi}$). If the maximum proportion observed ($\bar{\pi}_{\max}$) exceeds the user-defined cutoff $\bar{\pi}_c$, the CE phase terminates and returns the parameter values that achieved $\bar{\pi}_{\max}$. Otherwise, it continues for another C iterations and repeats the above evaluation.

The parameters returned upon successful completion of the CE phase are used as input to the MCEM phase. This second phase functions almost identically to the first, except that it simulates exactly K_0''

trajectories hitting each consecutive data point by discarding all trajectories that miss. Upon reaching the final time, the MCEM phase computes the update shown in equation (1) (with K' replaced by K_0''). Next, it computes an updated number of trajectories $K_1''(\geq K_0'')$ based on an estimate of the current Monte Carlo error and iterates the above procedure until the estimated increase in observed data likelihood between iterations is sufficiently small (see [3, 1] for more details). Upon completion (after the final MCEM iteration, symbolized by n, with number of trajectories K_n''), the MCEM phase returns the MLE for each parameter, denoted $\tilde{\theta}_j$.

The third and final phase of MCEM^2 estimates the uncertainty associated with each MLE. As described in [1], the negative inverse of the covariance matrix of the log-transformed MLEs is estimated by computing:

$$\begin{aligned} -\left(\hat{\Sigma}\right)^{-1} &= \left\{\frac{1}{K'''}\sum_{k=1}^{K'''} H_{jk}\right\}_j + \frac{1}{K'''}\sum_{k=1}^{K'''}\left(S_{jk}\right)_j\left(S_{jk}\right)_j^{\mathrm{T}} \\ &\quad - \left(\frac{1}{K'''}\sum_{k=1}^{K'''} S_{jk}\right)_j\left(\frac{1}{K'''}\sum_{k=1}^{K'''} S_{jk}\right)_j^{\mathrm{T}} \\ \text{with} \quad H_{jk} &= -\sum_{i=0}^{r_k} a_{jk}^i \times \tau_{ik} \quad \text{and} \quad S_{jk} = r_{jk} + H_{jk}\ , \end{aligned} \tag{2}$$

after simulating K''' full-length, partially correlated trajectories as in the previous phase using the MLEs as parameters. In equation (2), $\{\cdot\}_j$ represents a diagonal matrix with j ranging from 1 to the number of reactions M along the diagonal and $(\cdot)_j$ represents a column vector with j ranging from 1 at the top-most element to M at the bottom. Given $\hat{\Sigma}$, confidence intervals for each MLE can be easily computed.

For the experiments presented in this paper, we used $K = 10^4$, $\delta = .25$, $\rho = .001$ (and thus $K' = 10$), a distance function defined as the absolute difference in number of molecules of the current endpoint of each trajectory and the observed data at the same time point (assuming a single observed species), $\bar{\pi}_c = .01$, and $K''' = \max(K_n'', 10^4)$. In the Simulation study (Section 2.3 of the main text), we used $C = 10$ and $K_0'' = 2000$ for all data-model combinations except when inferring parameters from model (E) using data from model (A) (where using $C = 300$ dramatically accelerated the MCEM phase) and when inferring parameters from model (D) (where $K_0'' = 10$ was sufficient due to the absence of c_3). In performing Glutaminase data inference (Section 2.4 of the main text), we used $C = 100$ and $K_0'' = 2500$.

S2 Bursty MCEM^2

Unlike in the original MCEM^2, simulation of trajectories for bursty MCEM^2 cannot be achieved using the direct method implementation of the SSA. This is due to the intractability of sampling τ from the density given in equation (13) in the main text. Instead, we can use the first reaction method [2] whereby we independently sample the M time intervals until each reaction's next firing and select the reaction with the smallest interval. After firing this reaction, we recalculate all reaction time intervals for the subsequent firing. The relevant distribution function for sampling an individual reaction time interval τ_j is the following:

$$F_j(\tau_j, t) = 1 - \exp\left(-\int_t^{t+\tau_j} c_{1j} h_j(X(t)) \times \omega_j(s)^{c_{2j}}\,\mathrm{d}s\right). \tag{3}$$

using the same notation as in Section 2.2 of the main text. By simplifying equation (3) and using the method of inversion sampling, we reach a transformation that enables the simulation of τ_j given a uniformly distributed random variate u_j and the system at the current time t:

$$\tau_j = \left(-\frac{c_{2j}+1}{c_{1j} h_j(X(t))} \times \log(u_j) + \omega_j(t)^{c_{2j}+1} \right)^{\frac{1}{c_{2j}+1}} - \omega_j(t) \; . \tag{4}$$

Uncertainty quantification of $\tilde{c}_{1j}$ can be performed using a modified version of equation (2) where S_{jk} and H_{jk} have been replaced with the first and second derivatives, respectively, of equation (15) in the main text with respect to $\log(c_{1j})$ (see [1] for details):

$$\begin{aligned} H_{jk} &= -\sum_{i=0}^{r_k} \tilde{c}_{1j} h_j(X_k(t_{ik})) \times \frac{(\omega_{jk}(t_{ik}) + \tau_{ik})^{c_{2j}+1} - \omega_{jk}(t_{ik})^{c_{2j}+1}}{c_{2j}+1} \\ S_{jk} &= r_{jk} + H_{jk} \end{aligned} \; . \tag{5}$$

Similarly, uncertainty quantification for $\tilde{c}_{3j}$ can be performed by computing the first and second derivatives of equation (17) in the main text with respect to $\log(c_{3j})$ and substituting these values for S_{jk} and H_{jk}, respectively, into equation (2):

$$\begin{aligned} H_{jk} &= -\sum_{i=1}^{r_k} \frac{\tilde{c}_{3j}}{(1-\tilde{c}_{3j})^2} \times \mathbb{1}_j(j'_{ik}) \times \beta_{ik} \\ S_{jk} &= r_{jk} + H_{jk} \times (1-\tilde{c}_{3j}) \end{aligned} \; . \tag{6}$$

where $\mathbb{1}_j(j')$ is an indicator function that takes a value of 1 if $j' = j$ (0 otherwise).

S3 Model selection

As mentioned in the main text, the value of c_2 cannot be directly inferred using bursty MCEM^2. Instead, we consider a range of c_2 values corresponding to different numbers of promoter states and introduce model selection functionality into MCEM^2 to help select the most probable value. Given an initial fixed value for c_2, we first run bursty MCEM^2 to identify MLEs for all other unknown system parameters (e.g., c_1, c_3). Upon completion, we then simulate K''' full-length, partially correlated trajectories using the MLEs and fixed value of c_2 as described above. During this process, we repeatedly compute the probability of simulating a trajectory segment from the previous data point at time t_{l-1} (given the previous ensemble of trajectory segments) to the current data point at time t_l using the following unbiased estimator:

$$\hat{p}_l = \frac{K''' - 1}{K_l - 1} \; , \tag{7}$$

where K_l is the total number of simulated trajectory segments required to give K''' segments that connect the data points at times t_{l-1} and t_l. We then compute the product of these probabilities to yield an overall likelihood estimate of the MLEs with the current value of c_2 given the observed data:

$$\hat{L} = \prod_{l=1}^{D} \hat{p}_l \; , \tag{8}$$

where D is the total number of observed data points. Using the Akaike information criterion (AIC) [4], we can transform the likelihood estimate in equation (8) into a model selection score:

$$AIC = 2m - 2\log(\hat{L}) \; , \tag{9}$$

where m is the total number of unknown parameters in the model. This score rewards goodness of fit while penalizing model complexity, since increasingly complex models typically fit observed data increasingly well.

If we repeat the above procedure for all fixed values of c_2 under consideration, we obtain a list of AIC scores that can be used to select the preferred c_2 value (and the associated MLEs of the remaining parameters). The collection of parameters ("model") with the minimum AIC is most probable, and the relative likelihood that any other model is preferable is given by [5]:

$$\exp((AIC_{min} - AIC_i)/2) \,, \tag{10}$$

where AIC_{min} is the minimum score observed and AIC_i is the score of the model under consideration. For the experiments in this paper, we considered models with relative likelihoods $\geq .368$ (maximum difference of 2 AIC units from best fitting model) to constitute plausible fits to the data.

S4 Glutaminase trajectory preprocessing

We removed measurement noise from the glutaminase expression data by smoothing it using LOESS, a locally weighted linear regression method [6]. LOESS works locally and does not fit a function to the entire data set. For each data point (t, x), where t is time and x is the raw glutaminase light intensity level at time t, we formulate a weighted least squares expression using a second order polynomial for a fixed neighborhood around the point (t, x):

$$S(a_0, a_1, a_2) = \sum_l w_l \left(x_l - \left(a_0 + a_1 t_l + a_2 t_l^2\right)\right)^2 \,. \tag{11}$$

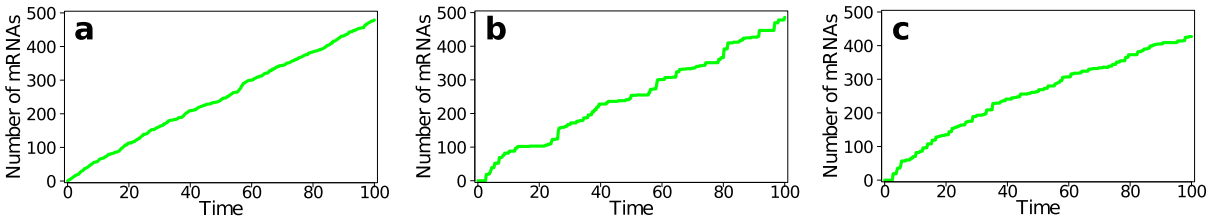
The l^{th} point in the neighborhood of (t, x) is denoted (t_l, x_l). In this work, we use a neighborhood size of 5 data points (and thus 2 points on either side of (t, x) are considered). Note that for the first data point in time, the neighborhood lies exclusively on the right hand side of (t, x); the opposite is true for the last data point. We used the following weighting function for the least squares method:

$$w_l = \left(1 - \left|\frac{t - t_l}{d(t)}\right|^3\right)^3 \,, \tag{12}$$

where $d(t)$ is the length of the span around the point (t, x). For the glutaminase data, where the measurements are taken every 5 minutes, $d(t) = 20$ minutes. Note that for any data point outside of the span, the weight is zero, and points near the data sample of interest have higher weights. By finding values of a_0, a_1, and a_2 that minimize the function S, we can use the expression $x_{smooth} = a_0 + a_1 t + a_2 t^2$ to compute a smoothed measurement value at time t. After repeating this process for all data points, we obtain a smoothed version of the glutaminase data trajectory.

In the final preprocessing step, we converted smoothed glutaminase light intensity values to numbers of proteins. Suter et al. performed calibration experiments suggesting that each reporter protein molecule provides roughly 0.0763 units of light (arbitrary units) [7]. Thus, we divided the smoothed light intensity values by this factor to convert them to protein numbers.

S5 Supplementary Results



Supplementary Figure S1: Simulated mRNA trajectories for (**a**) one-, (**b**) three-, and (**c**) six-state promoter models. Each model produces an average of 5 mRNA molecules per time unit.

Supplementary Table S1: Bursty MCEM^2 parameter inference and model selection results for models (F) (1 state) and (G) (2-101 states) using glutaminase data. $\tilde{c}_1$, $\tilde{c}_3$ represent maximum likelihood parameter estimates for c_1, c_3, respectively. Relative likelihoods are computed across all versions of models (F), (G), (H), and (I).

# States	$\mathbf{c_2}$	$\mathbf{\tilde{c}_1}$	$\mathbf{\frac{1-\tilde{c}_3}{\tilde{c}_3}}$	**Log Likelihood**	**AIC**	**Relative Likelihood**
1	n/a	20.53	n/a	-1840.73	3683.47	2.31×10^{-48}
2	0	4.11	5.02	-1754.80	3515.60	6.55×10^{-12}
3	.40	6.00	6.57	-1758.24	3522.49	2.09×10^{-13}
4	.75	8.47	7.28	-1761.38	3528.76	9.06×10^{-15}
5	1.10	9.81	8.46	-1764.59	3535.19	3.65×10^{-16}
6	1.40	10.58	9.40	-1767.13	3540.26	2.89×10^{-17}
7	1.65	39.03	6.57	-1767.40	3540.80	2.21×10^{-17}
9	2.10	7.24	13.07	-1771.82	3549.64	2.65×10^{-19}
11	2.50	6.71	14.25	-1773.24	3552.48	6.42×10^{-20}
16	3.40	6.46	15.78	-1775.48	3556.95	6.85×10^{-21}
21	4.10	6.60	16.40	-1776.89	3559.78	1.67×10^{-21}
26	4.75	7.07	16.67	-1777.99	3561.99	5.53×10^{-22}
51	7.40	9.63	17.00	-1784.77	3575.54	6.31×10^{-25}
101	11.00	12.41	17.13	-1795.78	3597.56	1.04×10^{-29}

Supplementary Table S2: Bursty MCEM^2 parameter inference and model selection results for model (H) using glutaminase data. $\tilde{c}_1$, $\tilde{k}_{\text{on}}$, $\tilde{k}_{\text{m}}$ represent maximum likelihood parameter estimates for c_1, k_{on}, k_{m}, respectively. Relative likelihoods are computed across all versions of models (F), (G), (H), and (I).

# States	$\mathbf{c_2}$	$\tilde{\mathbf{c}}_\mathbf{1}$	$\tilde{\mathbf{k}}_\mathbf{on}$	$\tilde{\mathbf{k}}_\mathbf{m}$	Log Likelihood	AIC	Relative Likelihood
3	.40	10.24	1.80	73.93	-1735.79	3479.57	4.36×10^{-4}
4	.75	17.60	1.65	74.04	-1734.80	3477.59	1.17×10^{-3}
5	1.10	26.98	1.58	71.42	-1734.24	3476.48	2.05×10^{-3}
6	1.40	44.54	1.54	72.73	-1734.10	3476.20	2.35×10^{-3}
7	1.65	52.73	1.52	69.23	-1733.83	3475.66	3.09×10^{-3}
9	2.10	102.56	1.50	69.62	-1733.74	3475.48	3.36×10^{-3}
11	2.50	164.26	1.49	68.15	-1733.84	3475.69	3.04×10^{-3}
16	3.40	498.88	1.47	67.00	-1734.12	3476.23	2.31×10^{-3}
21	4.10	1149.06	1.47	66.08	-1734.46	3476.92	1.64×10^{-3}
26	4.75	2688.22	1.48	65.74	-1734.77	3477.53	1.21×10^{-3}
51	7.40	41687.03	1.49	62.35	-1735.62	3479.25	5.12×10^{-4}
101	11.00	1480377.77	1.52	60.03	-1736.64	3481.29	1.85×10^{-4}

Supplementary References

[1] Daigle BJ Jr, Roh MK, Petzold LR, Niemi J: **Accelerated maximum likelihood parameter estimation for stochastic biochemical systems**. *BMC Bioinformatics* 2012, **13**:68.

[2] Gillespie DT: **Exact Stochastic Simulation of Coupled Chemical Reactions**. *The Journal of Physical Chemistry* 1977, **81**(25):2340–2361.

[3] Caffo BS, Jank W, Jones GL: **Ascent-based Monte Carlo expectation-maximization**. *Journal Of The Royal Statistical Society Series B* 2005, **67**(2):235–251.

[4] Akaike H: **A new look at the statistical model identification**. *Automatic Control, IEEE Transactions on* 1974, **19**(6):716–723.

[5] Burnham KP, Anderson DR, Burnham KP: *Model selection and multimodel inference: a practical information-theoretic approach*. New York: Springer, 2nd edition 2002.

[6] Cleveland WS, Devlin SJ: **Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting**. *Journal of the American Statistical Association* 1988, **83**(403):596–610.

[7] Suter DM, Molina N, Gatfield D, Schneider K, Schibler U, Naef F: **Mammalian genes are transcribed with widely different bursting kinetics**. *Science* 2011, **332**(6028):472–4.