SI Appendix

Modelling Genetic Networks

Allosteric Model of Transcription Factors

We model transcription factors as allosteric molecules having two states: a DNA-binding state (B) and a non-DNA binding state (N). Following Monod-Wyman-Changeux (1), the presence of sugar causes a shift in the time the transcription factor spends in each state (Fig. 5). Sugar can bind to either the B or the N form of the transcription factor, but does so with a different binding affinity (K_b for the DNA-binding state and K_n for the non-DNA-binding state). Only when unbound by sugar can the transcription factor change between its two states. The reaction describing this change has an equilibrium constant of K_t. If $K_b \gg K_n$, sugar preferentially binds to the B state. By binding to the transcription factor, sugar converts B_0 molecules into B_r molecules, more so than N_0 molecules into N_r molecules (where the subscript r denotes that r sugar molecules are bound). The reaction between B_0 and N_0 is no longer at equilibrium and more N molecules convert to B molecules while this equilibrium is restored. The population of transcription factors as a whole is now more in the stronger sugar-binding B state, and so again more B than N molecules are likely to bind sugar. This positive feedback means that the number of transcription factors in the B state can be a highly non-linear function of the number of sugar molecules (1). If $K_n \gg K_b$ the opposite behaviour occurs, and sugar drives the transcription factors into the non-DNA binding N state.

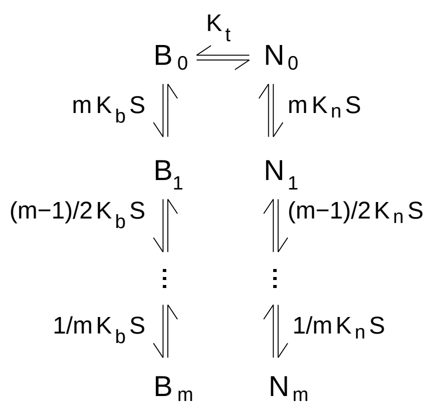

Fig. 5. An allosteric transcription factor that binds sugar S and exists in a DNA-binding state (B) and a non-DNA binding state (N). Each sugar binding site is assumed identical, and the subscripts denote the number of bound sugar molecules. Consequently, the basic equilibrium association constants for sugar binding, K_b and K_n, are altered by the ratio of the number of sites available for binding sugar (which increase the forward rate of the reaction) to the number of bound sugars (which increase the backward rate).

We assume that both the total amount of sugar and the total amount of transcription factors are conserved:

$$
\begin{aligned}
S_{\text{tot}} &= S + \sum_{r=0}^{m} (rN_r + rB_r) && \textbf{[6]} \\
T_{\text{tot}} &= \sum_{r=0}^{m} (N_r + B_r), && \textbf{[7]}
\end{aligned}
$$

where m is the number of sugar binding sites. Following ref. 1, we assume that each reaction in Fig. 5 is at equilibrium:

$$
\begin{aligned}
K_t N_0 &= B_0 \\
mK_b S B_0 &= B_1 \\
(m-1) K_b S B_1 &= 2B_2 \\
&\vdots \\
K_n S N_{m-1} &= mN_m. && \textbf{[8]}
\end{aligned}
$$

Each equilibrium concentration can be solved in terms of N_0, the amount of transcription factor in the non-DNA binding state unbound by sugar:

$$
\begin{aligned}
N_r &= \binom{m}{r} (K_n S)^r N_0 \\
B_r &= \binom{m}{r} (K_b S)^r K_t N_0. && \textbf{[9]}
\end{aligned}
$$

Using these expressions and carrying out the summations in Eqs. **6** and **7** with the binomial theorem gives:

$$
\begin{aligned}
S_{\text{tot}} &= S + N_0 m S \left[K_n (1 + K_n S)^{m-1} + K_t K_b (1 + K_b S)^{m-1} \right] && \textbf{[10]} \\
T_{\text{tot}} &= N_0 \left[(1 + K_n S)^m + K_t (1 + K_b S)^m \right]. && \textbf{[11]}
\end{aligned}
$$

For a given S_{tot} and T_{tot} and the equilibrium association constants K_t, K_b, and K_n, we numerically solve Eqs. **10** and **11** for the amount of free sugar, S, and for N_0. We can therefore calculate the total amount of transcription factor in the non-DNA binding state, $N = N_0(1 + K_n S)^m$, and the total amount in the DNA-binding state, $B = N_0 K_t (1 + K_b S)^m$.

Promoter Models

We consider three different models of the promoter (Fig. 2*B* and *C*). The type A model has just one operator site. The type B model has two operators: a transcription factor at either operator prevents or initiates transcription independently. The final model, type C, has two operators but only one is sufficiently close to the RNA polymerase binding site to directly affect transcription. Nevertheless, a transcription factor bound to the inactive operator can stabilize a transcription factor bound to the active operator. We denote the fraction of time that the promoter is able to initiate transcription at equilibrium as promoter efficacy, P_{eff}.

We follow Shea and Ackers (2) to calculate the occupancy of the promoter at equilibrium. For example, for a negatively controlled type A promoter, which has just one binding site for a repressor, we consider the promoter existing in two states: P_1, bound by repressor, and P_0, not bound by repressor. If K_1 is the association constant for repressor binding and B is the number of repressors that are able to bind DNA, then $P_1 = K_1 B P_0$. The promoter is conserved: $P_0 + P_1 = 1$, if there is only one copy of the promoter. Combining these two equations implies that the promoter efficacy, P_0, obeys $P_0 = 1/(1 + K_1 B)$. We solve for the promoter efficacy for more complicated promoters similarly.

For a negatively controlled system, P_{eff} is the equilibrium fraction of promoter free from repressor. For the different promoter models:

type A,

$$P_{\text{eff}} = \frac{1}{1 + K_1 B} \qquad \textbf{[12]}$$

type B,

$$P_{\text{eff}} = \frac{1}{1 + (K_1 + K_2)B + K_1 K_2 B^2} \qquad \textbf{[13]}$$

type C,

$$P_{\text{eff}} = \frac{1 + K_2 B}{1 + (K_1 + K_2)B + \frac{1}{2}(K_1 K_2 + K_1 K_2 K_c)B^2}, \qquad \textbf{[14]}$$

where B is the total amount of transcription factor in the DNA-binding form, K_1 and K_2 are association constants for transcription factor binding to the two operator sites, and K_c determines the degree of cooperativity between two interacting, DNA bound transcription factors.

For positively controlled systems, P_{eff} is the equilibrium fraction of promoter bound by activator. For

type A,

$$P_{\text{eff}} = \frac{K_1 B}{1 + K_1 B} \qquad \textbf{[15]}$$

type B,

$$P_{\text{eff}} = \frac{(K_1 + K_2)B + K_1 K_2 B^2}{1 + (K_1 + K_2)B + K_1 K_2 B^2} \qquad \textbf{[16]}$$

type C,

$$P_{\text{eff}} = \frac{K_1 B + \frac{1}{2}(K_1 K_2 + K_1 K_2 K_c)B^2}{1 + (K_1 + K_2)B + \frac{1}{2}(K_1 K_2 + K_1 K_2 K_c)B^2}. \qquad \textbf{[17]}$$

Note when $K_c = 1$, that is, no cooperative interaction between the transcription factors, the type C models do not reduce to the type B models because only one operator is active for type C, whereas both are active for type B.

Comparison of the Models as Bayesian Classifiers

Generating the Posterior Probabilities

To generate a set of two-state classification problems, we assumed that each state can be described by a lognormal distribution:

$$P(S|\text{state}_i) = \frac{\mathrm{e}^{\frac{-(\ln S - \mu_i)^2}{2\sigma_i^2}}}{\sqrt{2\pi}\sigma_i S}. \qquad \textbf{[18]}$$

The low state has a sugar distribution with mean μ_1 and standard deviation σ_1; the high state has a mean μ_2 and standard deviation σ_2. We choose μ_1 to be either 1, 3, or 5; μ_2 to be either 5.1, 6.6, 8.1, or 9.6; σ_1 to be either 0.4, 0.5, 0.6, 0.7, 0.8, or 0.9; and σ_2 to be either 1, 1.25, 1.5, 1.75, or 2. All possible combinations of these parameters were considered, and we chose 50 pairs of distributions that best gave a range of different posterior probabilities (Fig. 6).

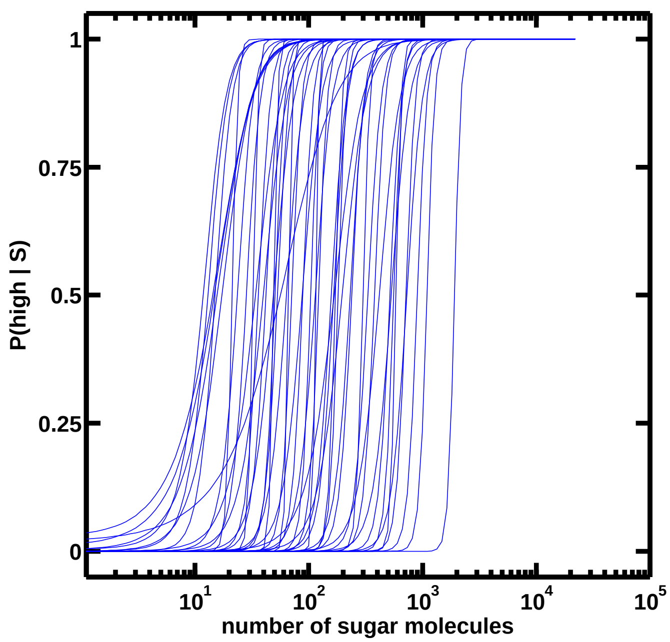

Fig. 6. The collection of posterior probabilities that were generated as solutions of lognormal two-state classification problems and used to compare the different genetic models of Fig. 2 as Bayesian classifiers.

Fitting the Models to the Posterior Probabilities

We used a least-square fit to score how well a model matches the posterior probability of the high state. The residuals plotted in Fig. 2 are the minimum value of the sum of squares:

$$\sum_i^n \Big[P(S_i|\text{high}) - P_{\text{eff}}(S_i, \lambda) \Big]^2, \qquad \textbf{[19]}$$

where we have n sugar levels S_i leading to n points on the posterior probability curve, $P(S|\text{high})$, we are trying to fit, and $P_{\text{eff}}(S, \lambda)$ is the model prediction for the promoter efficacy. This prediction is a function of the set of parameters λ: K_r, K_n, K_b, K_1, and K_2 and K_c depending on the promoter type. The minimum value of Eq. **19** occurs at the best-fit set of parameters λ. To ensure that the fitting algorithm considers only non-negative parameters, we define new variables for each parameter in log space. For example, $\kappa_1 = \log(K_1)$, and therefore can range over positive and negative values (3).

To correctly compare the ability of different models to fit a data set, models with more parameters should be penalized because they have more freedom to match the data. Typical methods are the Bayes Information Criterion (BIC) (4) and the Laplace method for model selection (5). With both of these techniques to compare the different models, the results of Fig. 2*D*–*F* were qualitatively unchanged. For the Laplace method, we need the maximum likelihood of the data (the 100 posterior probability points in our case) given the model. For each parameter, the maximum likelihood is penalized by a term that is determined by the error in the best fit value of the parameter and by its prior (5). We use:

$$\left(\sum_{i}^{n}\Big[P(S_i|\text{high}) - P_{\text{eff}}(S_i, \lambda)\Big]^2\right)^{-\frac{(n-1)}{2}} \qquad \textbf{[20]}$$

for the likelihood. This distribution results from assuming that the data have normally distributed errors with zero mean and any non-negative standard deviation (5). It is maximized when the sum of squares residual, Eq. **19**, is minimized.

Parameter Sensitivity

The sensitivities of the parameters were calculated as the mean log gain sensitivities (6) of the promoter efficacy. For parameter p_j, the sensitivity, χ_j, is

$$\chi_j = \left\langle \frac{\partial \log P_{\text{eff}}}{\partial \log p_j} \right\rangle, \qquad \textbf{[21]}$$

where the angled brackets denote an average over all sugar concentrations. We analytically calculated the $\partial \log P_{\text{eff}} / \partial \log p_j$ derivative as an implicit function of $\partial N_0/\partial p_j$ and $\partial S/\partial p_j$ by differentiating the promoter efficacy, such as Eq. **15** for example. We calculated these last two derivatives by differentiating Eqs. **10** and **11** with respect to p_j and numerically solving the resulting equations. Sensitivity values are given in Table 1.

Robustness of the Best-Fit Parameters

The fits of the promoter efficacy to the posterior probability curves are robust to changes in all but two of the parameters specifying each model. To investigate this robustness, we considered the model that best fit the posterior probability curves of Fig. 6. This model is transcriptionally controlled by a repressor that has four sugar binding sites. We varied each parameter individually and calculated the average change in the sum of squares residual, Eq.

Table 1. Parameter sensitivities for repressor and activator models

	Repressor model	Activator model
K_t	0.07	0.10
K_n	0.21	0.004
K_b	0.004	0.20
K_1	0.07	0.07
K_2	0.02	0.04
K_c	0.06	0.04

Parameters are defined in Fig. 2.

19, over all the posterior curves. The results shown in Fig. 7 reflect Table 1: the fit is only significantly sensitive to K_n, the sugar binding affinity for the non-DNA binding form of the repressor, and to a much lesser extent to K_t, the affinity describing transitions between the DNA- and non-DNA binding forms. Nevertheless, the sum of squares residual is so small for this model that the promoter efficacy curves behave like the posterior probability of Fig. 1*C* even if the residual is increased 5,000-fold (see Fig. 7 inset). We comment on possible implications of the high sensitivity to K_n in the text.

Stochastic Simulation

Table 2. Parameter values for the simulation shown in Fig. 3.

	Repressor model	Activator model
K_r	1.27 (10 s)	1.61×10^6 (10 s)
K_n	9.45×10^5 (10 s)	3.04×10^4 (10 s)
K_b	233 (10 s)	1.33×10^6 (10 s)
K_1	3.41×10^6 (0.1 s)	3.62×10^6 (0.1 s)
K_2	3.34×10^{10} (0.1 s)	1.51×10^9 (0.1 s)
K_c	88.3 (10 s)	219 (10 s)

These values are association affinities and are the best-fit values of the networks to the posterior probability of Fig. 1*C*. Each association affinity is dimensionless because we simulate with numbers of molecules rather than concentrations. Shown in brackets is the corresponding dissociation rate. These rates, which are not given by a fit to $P(\text{high}|S)$, were chosen so that the network would respond in a reasonable time to changes in sugar levels.

To confirm that genetic networks can perform inference in real time with a noisy sugar source, we simulated both a repressor and an activator model with fluctuating sugar levels. We chose the posterior of Fig. 1*C* and the repressor and activator model that fit it best (parameters are given in Table 2).

To generate a relatively smooth time series of sugar levels, we used a Markov chain Monte Carlo method (5) to sample from the distributions in Fig. 1*C* (the Metropolis algorithm with a Gaussian trial distribution). We sample from the low distribution for 10^4 s, then from

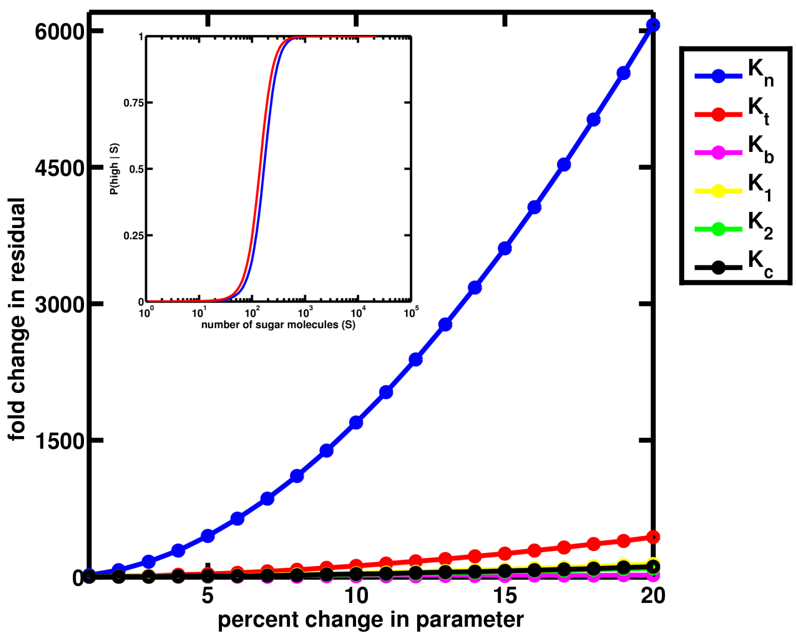

Fig. 7. Robustness of the sum of squares fit to systematic perturbations in individual model parameters away from their best-fit values. The model that best fits the posterior probabilities of Fig. 6 is shown: this model has promoter type C and is negatively regulated by a repressor with four sugar binding sites. Parameters are defined in Fig. 2. (Inset) An example of the promoter efficacy curves where K_n is changed by 20%. The curves are very similar despite the residual for the upper red curve being ≈ 5,000-fold larger than the residual of the original blue curve.

the high distribution for 10^4 s, and the again from the low distribution for another 10^4 s. For each sugar sample, the cytosolic sugar levels in the simulation are changed to the new sampled value. A stochastic simulation of the genetic network is then run for a fixed time interval (either 5, 10, 25, 50, or 100 seconds) by using the Gibson-Bruck version (7) of the Gillespie algorithm (8). The probability of a given reaction per unit time is equal to the product of the kinetic rate for the reaction and the number of potential reactants present. The time steps between reactions obey a Markov process. The cytosolic sugar level is then resampled by using the Markov chain Monte Carlo method and another Gillespie simulation run for this new level of sugar. The promoter efficacy plotted in Fig. 3 is the average promoter efficacy generated during each run of the Gillespie algorithm. Simulations start with one DNA molecule, 25 transcription factors in the DNA binding state and 25 transcription factors in the non DNA binding state.

For each choice of sugar sampling interval, we compared the performance of the two networks (Fig. 3*C* and *D*) to the instantaneous posterior probability (Fig. 3*B*). The comparison was scored by measuring the mean over time of the absolute difference between promoter efficacy and the instantaneous posterior probability. The results are shown in Table 3. Both networks perform better as the sugar sampling interval increases. As the time period grows over which the promoter efficacy is averaged, the average more closely matches the posterior probability of Fig. 2*C* (for a long sampling period, the promoter efficacy will match the posterior probability almost perfectly because we use the best fit parameters for the simulation).

Negatively controlled networks consistently performed better because the network is better able to use its cooperativity (see the argument given in the text).

Table 3. Comparison scores of the mean absolute difference between the promoter efficacy and the instantaneous posterior probability of the high sugar state

Sampling interval, s	**Repressor model**	**Activator model**
5	7.0×10^{-2}	12.5×10^{-2}
10	3.3×10^{-2}	6.3×10^{-2}
25	3.4×10^{-2}	7.4×10^{-2}
50	2.8×10^{-2}	5.0×10^{-2}
100	2.5×10^{-2}	4.4×10^{-2}

Each score is the average from five simulation runs. A score of zero implies the the promoter efficacy exactly follows the instantaneous posterior probability. The sampling interval is the time between the samples of sugar used to generate the sugar time series.

Fitting a Posterior Surface to the Transcription Rate of the *lac* Operon

The Inverse Gaussian Classification Problem

A bivariate, or two dimensional, Gaussian distribution is a function of a vector (s_1, s_2) and is specified by a mean vector (μ_1, μ_2) and a 2×2 covariance matrix $\boldsymbol{\sigma}$. For example, μ_1 is the mean of the s_1 variable and σ_{11} its variance. $P(s_1, s_2)$ obeys:

$$P(s_1, s_2) \sim \frac{1}{\sqrt{\det(\boldsymbol{\sigma})}} \exp\left(-\frac{1}{2}\sum_{i,j}(s_i - \mu_i)\sigma^{-1}_{ij}(s_j - \mu_j)\right), \qquad \textbf{[22]}$$

where $\boldsymbol{\sigma}^{-1}$ is the matrix inverse of $\boldsymbol{\sigma}$.

A two-state, bivariate Gaussian classification problem is described by the prior probabilities of the two states, $P(\mathrm{II})$ and $P(\mathrm{I}) = 1 - P(\mathrm{II})$; the mean $\boldsymbol{\mu}^{\mathrm{I}}$ and covariance matrix $\boldsymbol{\sigma}^{\mathrm{I}}$ for s_1 and s_2 for state I; and the mean $\boldsymbol{\mu}^{\mathrm{II}}$ and covariance matrix $\boldsymbol{\sigma}^{\mathrm{II}}$ for s_1 and s_2 for state II. Given an observation of s_1 and of s_2, the posterior probability of state II is:

$$\begin{aligned} P(\mathrm{II}|s_1, s_2) &= \frac{P(s_1, s_2|\mathrm{II})P(\mathrm{II})}{P(s_1, s_2|\mathrm{I})P(\mathrm{I}) + P(s_1, s_2|\mathrm{II})P(\mathrm{II})} \\ &= \left(1 + \frac{P(s_1, s_2|\mathrm{I})P(\mathrm{I})}{P(s_1, s_2|\mathrm{II})P(\mathrm{II})}\right)^{-1}. \end{aligned} \qquad \textbf{[23]}$$

Inserting Eq. **22** in Eq. **23** gives:

$$P(\mathrm{II}|s_1, s_2) = \left(1 + \sqrt{\frac{\det(\boldsymbol{\sigma}^{\mathrm{II}})}{\det(\boldsymbol{\sigma}^{\mathrm{I}})}} \times \frac{\exp(-\frac{1}{2}\sum_{i,j}(s_i - \mu^{\mathrm{I}}_i)(\sigma^{\mathrm{I}})^{-1}_{ij}(s_j - \mu^{\mathrm{I}}_j))}{\exp(-\frac{1}{2}\sum_{i,j}(s_i - \mu^{\mathrm{II}}_i)(\sigma^{\mathrm{II}})^{-1}_{ij}(s_j - \mu^{\mathrm{II}}_j))} \times \frac{1 - P(\mathrm{II})}{P(\mathrm{II})}\right)^{-1}. \qquad \textbf{[24]}$$

From the posterior surface $P(\mathrm{II}|s_1, s_2)$, we would like to recover the parameters of the classification problem: $P(\mathrm{II})$, $\boldsymbol{\mu}^{\mathrm{I}}$, $\boldsymbol{\sigma}^{\mathrm{I}}$, $\boldsymbol{\mu}^{\mathrm{II}}$, and $\boldsymbol{\sigma}^{\mathrm{II}}$. This recovery is degenerate: different sets of parameters can result in the same posterior surface. With a little algebra, Eq. **24** can be reduced to the general form:

$$P(\mathrm{II}|s_1, s_2) = \left[1 + \exp(c_0 + c_1 s_1 + c_2 s_2 + c_3 s_1 s_2 + c_4 s_1^2 + c_5 s_2^2)\right]^{-1}, \qquad \textbf{[25]}$$

where the c_i depend on the parameters $P(\mathrm{II})$, $\boldsymbol{\mu}^{\mathrm{I}}$, $\boldsymbol{\sigma}^{\mathrm{I}}$, $\boldsymbol{\mu}^{\mathrm{II}}$, and $\boldsymbol{\sigma}^{\mathrm{II}}$. Although these parameters have 11 degrees of freedom [$P(\mathrm{II})$, two each for vectors $\boldsymbol{\mu}^{\mathrm{I}}$ and $\boldsymbol{\mu}^{\mathrm{II}}$, and three each for the covariance matrices $\boldsymbol{\sigma}^{\mathrm{I}}$ and $\boldsymbol{\sigma}^{\mathrm{II}}$], the posterior surface only has six degrees of freedom. The parameters therefore have five unrecoverable degrees of freedom.

Fitting the Transcription Rate Surface from the *lac* Operon

To fit the Setty *et al.* data (9), we used Eq. **25**, with s_1 corresponding to the logarithm of the IPTG concentration and s_2 corresponding to the logarithm of the cAMP concentration. As the base of the logarithm and a constant offset can be absorbed by the coefficients c_i, we chose to let $s_1 \in \{0, 1, \ldots, 5\}$ correspond to the six sample levels of IPTG and $s_2 \in \{0, 1, \ldots, 9\}$ correspond to the 10 sample levels of cAMP. We used a simplex search method (`fminsearch` in Matlab, Mathworks) to optimize the six parameters $c_0, c_1, \ldots, c_5$ so that the sum-squared error between Eq. **25** and the *lac* transcription data was minimized. We used multiple optimization runs and experimented with different initial conditions, but these factors seems to have little influence on the outcome of optimization. All or nearly all runs converged to essentially the same solution, which we therefore take to be close to optimal. The final parameters found were:

c_0	c_1	c_2	c_3	c_4	c_5
4.09	−1.88	0.15	−0.11	0.32	−0.06

which define the surface shown in Fig. 4*B*.

There is not a unique two state, bivariate Gaussian discrimination problem corresponding to these parameters (as described above). Of the many discrimination problem parameter sets consistent with the optimized c_i, we chose one by making the following assumptions:

$$(\boldsymbol{\sigma}^{\mathrm{I}})^{-1} = \begin{bmatrix} 0.4 & -c_3 \\ -c_3 & 0.3 \end{bmatrix} \qquad \textbf{[26]}$$

$$(\boldsymbol{\sigma}^{\mathrm{II}})^{-1} = (\boldsymbol{\sigma}^{\mathrm{I}})^{-1} + \begin{bmatrix} 2c_4 & 0 \\ 0 & 2c_5 \end{bmatrix} \qquad \textbf{[27]}$$

$$\boldsymbol{\mu}^{\mathrm{I}} = \begin{bmatrix} 1.5 \\ 3.5 \end{bmatrix} \qquad \textbf{[28]}$$

$$\boldsymbol{\mu}^{\mathrm{II}} = \left(\begin{bmatrix} -c_1 \\ -c_2 \end{bmatrix} + (\boldsymbol{\sigma}^{\mathrm{I}})^{-1}\boldsymbol{\mu}^{\mathrm{I}}\right)\boldsymbol{\sigma}^{\mathrm{II}} \qquad \textbf{[29]}$$

$$P(II) = (1 + \mathrm{e}^{z})^{-1} \qquad \textbf{[30]}$$

where

$$z = c_0 + \frac{1}{2}(\boldsymbol{\mu}^{\mathrm{I}})^T(\boldsymbol{\sigma}^{\mathrm{I}})^{-1}\boldsymbol{\mu}^{\mathrm{I}} - \frac{1}{2}(\boldsymbol{\mu}^{\mathrm{II}})^T(\boldsymbol{\sigma}^{\mathrm{II}})^{-1}\boldsymbol{\mu}^{\mathrm{II}} + \frac{1}{2}\log\left[\frac{\det(\boldsymbol{\sigma}^{\mathrm{I}})}{\det(\boldsymbol{\sigma}^{\mathrm{II}})}\right], \qquad \textbf{[31]}$$

which results in the distinct lognormal distributions in Fig. 4*C*. The five parameters unrecoverable from the posterior surface can be seen in our arbitrary choices for $\boldsymbol{\mu}^{\mathrm{I}}$, the diagonal elements of $(\boldsymbol{\sigma}^{\mathrm{I}})^{-1}$, and the off-diagonal elements (zero) added to $(\boldsymbol{\sigma}^{\mathrm{I}})^{-1}$ to make $(\boldsymbol{\sigma}^{\mathrm{II}})^{-1}$.

References

1. Monod J, Wyman J, Changeux JP (1965) *J Mol Biol* 12:88–118.

2. Shea MA, Ackers GK (1985) *J Mol Biol* 181:211–230.

3. Brown KS, Sethna JP (2003) *Phys Rev E* 68:21904.

4. Schwartz G (1978) *Ann Stat* 6:461–464.

5. Mackay DJC (2003) *Information theory, inference, and learning algorithms* (Cambridge University Press, New York, NY).

6. Stelling J, Gilles ED, Doyle FJ (2004) *Proc Natl Acad Sci USA* 101:13210–13215.

7. Gibson MA, Bruck J (2000) *J Phys Chem* 104:1876–1889.

8. Gillespie DT (1977) *J Phys Chem* 81:2340–2361.

9. Setty Y, Mayo AE, Surette MG, Alon U (2003) *Proc Natl Acad Sci USA* 100:7702–7707.