Supplementary Material for
A framework for scalable parameter estimation of gene circuit models using structural information
Hiroyuki Kuwahara\textsuperscript{1a}, Ming Fan\textsuperscript{1a}, Suojin Wang\textsuperscript{2}, and Xin Gao\textsuperscript{1,*}
\textsuperscript{a}Equal contributions.
\textsuperscript{*}All correspondence should be addressed to xin.gao@kaust.edu.sa.
\textsuperscript{1}Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia
\textsuperscript{2}Department of Statistics, Texas A&M University, College Station, Texas, 77840, U.S.A.
Contents

S1 Detailed information of PEDI 3
  S1.1 Objective function ................................................. 3
  S1.2 Initial estimation of the mean protein level ......................... 3
  S1.3 Initial estimation of the mean mRNA level ........................ 4
  S1.4 Computation of protein levels at the refinement step ................ 4
  S1.5 Computation of the objective function at the refinement step .......... 4
  S1.6 Termination condition ............................................. 5
  S1.7 Probability of accepting the current parameters in each iteration .... 5

S2 Configuration of SA 6
  S2.1 Objective function ................................................ 6
  S2.2 Algorithmic parameters of SA .................................. 6
  S2.3 Boundaries for parameter search in the two models ................. 6
  S2.4 Initial guess for the parameters in the two models ............. 7

S3 Comparison between PEDI and SA 7

S4 Configuration of parameter estimation methods 8
  S4.1 Configuration of SRES ............................................ 8
  S4.2 Configuration of TDDM ......................................... 9
  S4.3 Configuration of HEKF+MM .................................... 9
    S4.3.1 Configuration of HEKF .................................. 9
    S4.3.2 Initial guess for the parameters in the two models .......... 9

S5 Comparison using a variant of M3 10

S6 Supporting Tables 11

S7 Supporting Figures 15
S1 Detailed information of PEDI

S1.1 Objective function

We utilize a weighted sum of squared residuals as the objective function of which an optimization method within the PEDI framework attempts to minimize the value. Since PEDI decomposes a gene circuit model into individual rate equations, it optimizes \( \theta_i \) separately and has an objective function for each mRNA \( m_i \). The form of objective function for \( m_i \) is

\[
J_i = \sum_{j=1}^{M} w_{ij} (\hat{m}_i(t_j) - \bar{m}_{ij})^2,
\]

where \( w_{ij} \) is the weight for the squared error for mRNA \( m_i \). Thus, instead of having the contribution of the squared error at each time point equally, the importance of each error is specified by its weight. It is often the case that the mean of a random variable representing the population of a molecular species positively correlates with its variance. Thus, we often want to decrease the contributions of the error terms as their sample means increase. In this study, we set \( w_{ij} \) to be the inverse square root of \( \bar{m}_{ij} + 1 \). This increases the importance of the terms with higher mean values relative to weights proportional to the sample means, and this choice appears to work well in this study by balancing the terms with higher and lower values of sample means. Note that this objective function was also used in simulated annealing.

S1.2 Initial estimation of the mean protein level

We first approximate the time integral of the rate equation of each protein \( p_i \) using the linear interpolation as follows:

\[
\hat{p}_i(t_j) = \int_{t_0}^{t_j} (\alpha_i \hat{m}_i(t) - \beta_i \hat{p}_i(t)) dt
\]

\[
= e^{-\beta_i \Delta t} \left( \hat{p}_i(t_{j-1}) + \int_{t_{j-1}}^{t_j} \alpha_i \hat{m}_i(t)e^{\beta_i(t-t_{j-1})} dt \right)
\]

\[
\approx e^{-\beta_i \Delta t} \left( \hat{p}_i(t_{j-1}) + \frac{\alpha_i \Delta t}{2} \left( \bar{m}_{ij} e^{\beta_i \Delta t} + \bar{m}_{ij-1} \right) \right),
\]

where \( \Delta t \equiv t_j - t_{j-1} \). Here, \( \hat{m}_i \) is also a time-dependent variable whose rate equation often depends on transcription factors, we utilize the trapezoidal rule to approximate \( \int_{t_{j-1}}^{t_j} \alpha_i \hat{m}_i(t)e^{\beta_i(t-t_{j-1})} dt \) and substitute \( \bar{m}_{ij} \) for \( \hat{m}_i(t_j) \) at each discrete time point. By iteratively evaluating Equation (S1) from \( j = 1 \) to \( M \), we are able to obtain the initial estimate of \( \hat{p}_i(t_j) \).
S1.3 Initial estimation of the mean mRNA level

To estimate \( \hat{m}_i(t_j) \) initially, we approximate the time integral of the rate equation of each mRNA \( m_i \) using the linear interpolation as follows:

\[
\hat{m}_i(t_j) = \int_{t_0}^{t_j} (h_i(\hat{p}; \theta_t) - \theta_{1i}\hat{m}_i) \, dt \\
= e^{-\theta_{1i}t} \left( \hat{m}_i(t_{j-1}) + \int_{t_{j-1}}^{t_j} h_i(\hat{p}(t); \theta_t)e^{\theta_{1i}(t-t_{j-1})} \, dt \right) \\
\approx e^{-\theta_{1i}t} \left( \hat{m}_i(t_{j-1}) + \frac{\Delta t}{2} \left( h_i(\hat{p}(t_j); \theta_t)e^{\theta_{1i}\Delta t} + h_i(\hat{p}(t_{j-1}); \theta_t) \right) \right). 
\]  

(S2)

By using the approximate integration shown in Equation (S2), the objective function that we minimize for the initial parameter estimation is a weighted sum of squared residuals as follows:

\[
\sum_{j=1}^{M} w_{ij} \left[ e^{-\theta_{1i}\Delta t} \left( \hat{m}_i(t_{j-1}) + \frac{\Delta t}{2} \left( h_i(\hat{p}(t_j); \theta_t)e^{\theta_{1i}\Delta t} + h_i(\hat{p}(t_{j-1}); \theta_t) \right) \right) - \bar{m}_{ij} \right]^2. 
\]  

(S3)

S1.4 Computation of protein levels at the refinement step

In the refinement step of PEDI, we compute equally-spaced \( L_p + 1 \) protein data points using equally-spaced \( L_m + 1 \) mRNA data points of \( \hat{m}_i \). The level of each protein \( \bar{p}_i \) at time point \( k\Delta t/d_p \) for all \( k \in [1, L_p] \) can be expressed by integrating the rate equation of protein \( p_i \) as follows:

\[
\bar{p}_i(t_U) = e^{\beta_i \Delta t} \bar{p}_i(t_L) + \int_{t_L}^{t_U} \alpha_i \bar{m}_i(t)e^{\beta_i(t-t_L)} \, dt, 
\]  

(S4)

where \( t_U = k\Delta t/d_p \) and \( t_L = (k-1)\Delta t/d_p \). Since \( L_m \) is much greater than \( L_p \), we use \( d_q \) subintervals of mRNA data where \( d_q = L_m/L_p \) to better interpolate the time evolution of \( m_i \) between the two time points \( t_L \) and \( t_U \) and to more accurately integrate \( \int_{t_L}^{t_U} \alpha_i \bar{m}_i(t)e^{\beta_i(t-t_L)} \, dt \) using various numerical integration methods. In this study, unless otherwise specified, we set \( L_m = 3,000 \) and \( L_p = 300 \) and implemented the composite Simpson’s rule as follows:

\[
\int_{t_L}^{t_U} \alpha_i \bar{m}_i(t)e^{\beta_i(t-t_L)} \, dt \approx \frac{\Delta t}{3d_q} \left[ \alpha_i \bar{m}_i(t_L) + 2 \sum_{s=1}^{d_q/2-1} \alpha_i \bar{m}_i(t_L + \frac{2s-1}{d_m} \Delta t)e^{\beta_i \frac{2s-1}{d_m} \Delta t} \right. \\
+4 \sum_{s=1}^{d_q/2} \alpha_i \bar{m}_i(t_L + \frac{2s-1}{d_m} \Delta t)e^{\beta_i \frac{2s-1}{d_m} \Delta t} + \alpha_i \bar{m}_i(t_U)e^{\beta_i \frac{\Delta t}{d_p}} \right]. 
\]  

(S5)

S1.5 Computation of the objective function at the refinement step

To compute \( \hat{m}_i(t_j) \) at the refinement step, we approximate the time integral of the rate equation of each mRNA \( m_i \) for each \( \Delta t \) interval by using \( d_p \) subintervals of protein levels. The expression
of \( \hat{m}_i(t_j) \) is as follows:

\[
\hat{m}_i(t_j) = e^{-\theta_1 \Delta t} \left( \hat{m}_i(t_{j-1}) + \int_{t_{j-1}}^{t_j} h_i(\hat{p}(t); \theta_i) e^{\theta_1 (t - t_{j-1})} \, dt \right).
\]

(S6)

Let \( \hat{H}_i(\hat{p}; \theta_i, j) \) be an approximate numerical solution of \( \int_{t_{j-1}}^{t_j} h_i(\hat{p}(t); \theta_i)e^{\theta_1(t-t_{j-1})} \, dt \), where \( \hat{p} \) as defined in the main text—is a time-dependent variable representing the \( L_p + 1 \) data points of \( p_i \). Then, we perform an optimization of each \( \theta_i \) which attempts to minimize the following objective function:

\[
\sum_{j=1}^{M} w_{ij} \left[ e^{-\theta_1 \Delta t} \left( \hat{m}_i(t_{j-1}) + \hat{H}_i(\hat{p}; \theta_i, j) \right) - \bar{m}_{ij} \right]^2.
\]

(S7)

In this study, we implemented the composite Simpson’s rule to compute \( \hat{H}_i(\hat{p}; \theta_i, j) \) as follows:

\[
\hat{H}_i(\hat{p}; \theta_i, j) = \frac{\Delta t}{3 \hat{d}_p} \left[ h_i(\hat{p}(t_{j-1}); \theta_i) + \sum_{k=1}^{d_p/2-1} h_i(\hat{p}(t_{j-1} + \frac{k \hat{d}_p}{\Delta t}); \theta_i) e^{\theta_1 \frac{k \hat{d}_p}{\Delta t}} \Delta t \\
+ \sum_{k=1}^{d_p/2} h_i(\hat{p}(t_{j-1} + \frac{k - 1}{\Delta t}); \theta_i) e^{\theta_1 \frac{k - 1}{\Delta t} \frac{\hat{d}_p}{\Delta t}} + h_i(\hat{p}(t_j); \theta_i) e^{\theta_1 \Delta t} \right].
\]

(S8)

### S1.6 Termination condition

In this study, we set the termination condition of PEDI so that the relative change of each \( J_i \) from the previous round to the current round is less than five percent for the last five iterations. Thus, let \( J_i(k) \) to be the \( J_i \) in the \( k \)-th round of the refinement step. Then, we compute

\[
\epsilon_i(k) = \begin{cases} 
1 & \text{if } k \leq 1, \\
\frac{|J_i(k) - J_i(k-1)|}{J_i(k-1)} & \text{otherwise,}
\end{cases}
\]

(S9)

for all \( i \). For a given \( k \), if, for all \( i \in [1, N] \), we have, for all \( s \in [k - 4, k] \), \( \epsilon_i(s) < 0.05 \) but \( \epsilon_i(k - 5) \geq 0.05 \), then the termination condition is satisfied at the \( k \)-th round. In other words, the termination condition is satisfied if the relative change is less than 5% for five consecutive refinement rounds. Thus, we perform at least 6 rounds of the PEDI refinement step. To guarantee the termination of the refinement step, we set the maximum number of the iterations to be 100. In this study, however, we did not encounter any instances in which the refinement step exceeded this maximum iteration count.

### S1.7 Probability of accepting the current parameters in each iteration

As described in the main text, we select the current \( \hat{\theta} \) as the seed for the next round at the refinement step when the current sum of \( J_i \) is smaller than the previous one. Otherwise, we
probabilistically decide which one to select between the current one and the previous one. The probability of accepting the current \( \hat{\theta} \) is
\[
p_{\text{max}} \exp \left(1 - \epsilon_c/\epsilon_o\right).
\] (S10)

In this study, we set \( p_{\text{max}} \) to be 0.1. With this setting, when the current sum of \( J_i \) is greater than the previous one, we most likely select the previous estimate of \( \theta \) over the current one.

S2 Configuration of SA

S2.1 Objective function

When we use SA as a standalone parameter estimation method, we set the objective function \( J \) to be the sum of \( J_i \). Similar to the objective function for PEDI, we set \( w_{ij} \) to be the inverse square root of \( \bar{m}_{ij} + 1 \) in this study.

S2.2 Algorithmic parameters of SA

We implemented PEDI in MATLAB. For the SA algorithm, we used \textit{simulannealbnd} provided in MATLAB. To run the function \textit{simulannealbnd} for our models, we set several options to better fit our purpose. These options are described below:

InitialTemperature The default InitialTemperature in Matlab is 100. The higher the value of InitialTemperature is, the longer the algorithm search for a solution in the search space. We set the value of InitialTemperature to be 500.

ReannealInterval ReannealInterval specifies the interval for re-annealing (i.e., the number of steps before re-heating occurs). The default value of ReannealInterval is 100. We set this parameter to be 200.

MaxFunEvals MaxFunEvals specifies the maximum number of objective function evaluations that are allowed. The default value of MaxFunEvals is 3,000 \( \times \) the number of variables. We set the value of MaxFunEvals to be 10,000 \( \times \) the number of variables.

S2.3 Boundaries for parameter search in the two models

We constrained the SA based optimization so that search space has the lower and upper bounds for each unknown parameter. These boundaries allow us not only to perform search in realistic parameter ranges but also to restrict our search within the biologically relevant range.

For model \( \mathcal{M}1 \), the lower bound of each parameter is set to \( 10^{-10} \) while the upper bound is set to \( 10^3 \). For model \( \mathcal{M}2 \), the boundaries are set differently depending on the types of parameters. The lower and upper bounds of the parameters representing the maximum transcription rates are set to 0.01 and 1, respectively. The lower and upper bounds of the parameters representing the transcription factor-cis-regulatory element binding affinity are set to \( 10^{-10} \) and \( 10^3 \), respectively. Finally, the lower and upper bounds of the parameters representing the Hill coefficients are set to 1 and 8, respectively.
S2.4 Initial guess for the parameters in the two models

The initial guess of the value of each parameter is set to be the lower bound of that parameter. That is, in model $M_1$, for example, the initial guess is $10^{-10}$ for all unknown parameters.

S3 Comparison between PEDI and SA

In this comparison, we generated four synthetic mRNA datasets, each consisting of the same 31 data points, based on the ODE simulation of each model. To generate each data point, we sampled a value from a Gaussian random variable with mean and variance being the simulation output of the given time point. Throughout this comparison, we set the number of intermediate points to be such that $d_m = 100$ and $d_p = 10$.

The results from model $M_1$ showed that PEDI substantially increased the quality of the estimated parameters and the reconstructed models; while SA was often trapped in suboptimal local minima and produced widely different sets of dynamics and estimated parameters, PEDI kept the prediction error small and produced more accurate parameter combinations more consistently (Figures S1 and S2). Next, we compared the results from the best parameter combinations from PEDI and SA in terms of the prediction error. We found that, whereas PEDI and SA both produced a similar level of quality of the reconstructed $m_3$ regulation, PEDI produced more than 1.7 times smaller prediction error than SA for the regulation of $m_1$ and $m_2$ (Figure S3). This makes sense because while $m_3$ is constitutively expressed in this model and the reconstruction of this regulation is simpler, the regulation of the other two mRNAs are more involved and complex. PEDI also generated the estimated parameter sets of $m_1$ and $m_2$ which were much closer to the true parameter sets (Figure S3). Since the linearization involved in the initial estimate of PEDI lowers the quality of numerical integration compared with the full simulation utilized in SA, an increase in the accuracy comes from the refinement process. Indeed, while the estimated parameters from the initial step resulted in a higher prediction error than that of SA, the prediction error of the $m_1$ and $m_2$ regulation from the first iteration of the refinement decreased sharply (Figure S4).

Next, we estimated the 12 unknown parameters of model $M_2$ using PEDI and SA. As with the analysis of model $M_1$, we ran both PEDI and SA for 10 times. In this analysis, we compared the accuracy of reconstructed dynamics based on estimates of the binding affinity, hill coefficient, and the ratio between the maximum transcription rate and mRNA degradation rate constant. Our results show that the estimated parameters from SA produced much higher prediction error and that the variance of the estimates for each parameter is large (Figures S5 and S6). This may be explained by the fact that oscillations require a stringent stability condition at fixed points and that the optimization path to reach parameter combinations satisfying such conditions with an appropriate amplitude, phase, and frequency in a high-dimensional search space based on the gradient of the objective function is not straightforward. Indeed, none of the parameter combinations estimated by SA exhibited oscillations. The results from PEDI, on the other hand, show that the estimated parameters consistently resulted in small prediction error and each of the parameter values is relatively close to the true one (Figures S5 and S6). Furthermore, all of the 10 parameter combinations generated from PEDI exhibit damped oscillations. We next compared the estimates which resulted in the smallest prediction error from the two approaches. We found
that, while SA could not exhibit an oscillation, PEDI led to a damped oscillation which closely resembles the true trajectory and there was a substantial difference between the two in terms of the prediction error; in particular, PEDI resulted in more than 19 times smaller prediction error for the regulation of $m_3$ compared with SA (Figure S7). Furthermore, the parameter set from PEDI estimated the true parameter set more accurately. By looking at the trace of how this best parameter combination was generated, we found that initially the estimated parameters did not lead to oscillatory dynamics and its prediction error is on par with SA (Figure S8). By the second round of refinement, however, the model started to exhibit an oscillation, and as the parameters were refined over iterations, the amplitude and the frequency of the oscillation were changed to better capture the time-series data (Figure S8).

Both SA and PEDI with SA were implemented in MATLAB (Mathworks Inc, Natick, MA). In terms of efficiency, the average runtime from 10 parameter estimation runs of model $M_1$ was 4.1 minutes with SA while it was just 2 minutes with PEDI, making PEDI more than two times faster than SA. As for the parameter estimation of model $M_2$, the average runtime from 10 runs was 28.8 minutes with SA while it was 15 minutes with PEDI, making PEDI 1.92 times faster than SA. The slower runtime of SA in model $M_2$ could be mainly a result of high dimensional parameter search space. Despite the fact that PEDI decomposes a high dimensional parameter estimation problem into subproblems with much smaller search space, the runtime of PEDI for model $M_2$ also significantly slowed down compared with model $M_1$. Such an increase in runtime could be explained by much slower convergence of estimated parameters in model $M_2$ (Figures S4 and S8). Nevertheless, our results indicate that PEDI improves the runtime efficiency of parameter estimation while it also improves the accuracy. Overall, our results show that PEDI is able to improve the consistency and the accuracy of parameter estimation while increasing the runtime efficiency. These improvements make the parameter estimation process more efficient and effective by permitting PEDI to produce high quality estimates in much less computing time.

S4 Configuration of parameter estimation methods

S4.1 Configuration of SRES

We used a Matlab implementation of SRES [1], which we downloaded from https://notendur.hi.is/tpr/index.php?page=software/sres/sres. To configure SRES, we set the following options:

lambda 200;
mu 30;
px 0.45;
varphi 1;

The boundaries were set to be consistent with those of SA.
S4.2 Configuration of TDDM

We implemented TDDM in Matlab. Since the parameters for the regulation of the proteins were all known in our experiments, we were able to skip one of the steps in TDDM. The only options for TDDM that we set are those for the scatter search method (SSM) [2]. We downloaded SSM from http://www.iim.csic.es/gingproc/ssmGO.html. To configure the SSM, we set the following options:

\[-\text{maxeval} \quad 10^4;\]
\[-\text{local.solver} \quad \text{dhc};\]
\[-\text{local.finish} \quad \text{fmincon};\]

The boundaries were set to be consistent with those of SA.

S4.3 Configuration of HEKF+MM

S4.3.1 Configuration of HEKF

Detailed information about HEKF for this hybrid method can be found in [3]. There are a couple of algorithmic parameter that we set for applying HEKF to the two gene circuit models:

**Covariance matrix** $R$  $R$ is an $N \times N$ matrix that represents the measurement noise. Each element $R_{i,j}$ is the covariance of $m_i$ and $m_j$, which is assumed to be known and fixed over time. Since the measurement noise is assumed to be uncorrelated for different mRNAs, off-diagonal elements are set to 0. $R_{i,i}$ is set to be the mean of $\mu_{m_{ij}}$ over all $j$ as in [3]. That is,

$$R_{i,j} = \begin{cases} \frac{1}{M} \sum_{j=0}^{M} \mu_{m_{ij}} & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

**Linear constraint term** Following [3], we specified the constraints for the states and the parameters using the following linear form:

$$D \times \hat{x} \leq d,$$

where $D \equiv diag(-1, -1, \cdots, -1)$ is a negative identity matrix with suitable size, $\hat{x}$ contains the state variables and the unknown parameters, $d$ is a vector specifying the constraint that HEKF must satisfy for each iteration. To have a biologically relevant solution, $d$ must be at least zero. An important things to note here is that depending on the value of $d$, the results from HEKF can change, which can in turn change the results from HEKF+MM. To obtain different results from HEKF+MM for comparison, we had $d \in \{0, 10^{-13}, 10^{-12}, 10^{-11}, 10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}\}$. This allowed us to have 10 different sets of results from HEKF+MM.

S4.3.2 Initial guess for the parameters in the two models

The initial guess that is fed into HEKF is set to be the same as PEDI and SA.
S5 Comparison using a variant of $\mathcal{M}^3$

To test if our overall conclusions about the performance of PEDI still hold with a variant of $\mathcal{M}^3$, we have modified the 7-gene repressilator model to add a repression connection from gene $g_3$ to gene $g_6$. This modification changes only the ODE for the regulation of $m_6$ from $\mathcal{M}^3$ as follows:

$$\frac{dm_6}{dt} = \frac{k_{p6}}{1 + (k_{e61}p_6)^{n_{61}} + (k_{e62}p_3)^{n_{62}}} - k_{m6}m_6.$$ 

As with the analysis of $\mathcal{M}^3$, we set $k_{m6}$ to be known. In total, this makes the number of unknown parameters in this modified repressilator model 23. The parameters in the ODE for the regulation of $m_6$ were valued as: $k_{p6} = 16; k_{e61} = 1/20; n_{61} = 4; k_{e62} = 1/40; n_{63} = 4;$ and $k_{m6} = 0.2$.

The results from the comparison of the four methods using this modified 7-gene repressilator model are summarized in Table S5. Similar to the results from $\mathcal{M}^3$, PEDI was the second fastest method, closely trailing TDDM. In terms of the quality of reconstructed dynamics, PEDI outperformed the other three methods on the criteria of the average, the best, and the worst prediction errors. Remarkably, the average prediction error from PEDI was lower than the best prediction error from any of the other three methods. In addition, the estimated parameters from PEDI were at least an order-of-magnitude closer to the true parameter values. Furthermore, PEDI was also able to extrapolate the mRNA levels at next few time points substantially more accurately than the other three. These results show that PEDI was able to produce much higher quality parameter solutions very efficiently.

References


**S6 Supporting Tables**

**Table S1.** The parameter setting for model $M_1$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Setup$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>0.02</td>
<td>U</td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.05</td>
<td>U</td>
</tr>
<tr>
<td>$k_3$</td>
<td>6.25</td>
<td>U</td>
</tr>
<tr>
<td>$k_4$</td>
<td>0.01</td>
<td>U</td>
</tr>
<tr>
<td>$k_5$</td>
<td>2</td>
<td>U</td>
</tr>
<tr>
<td>$k_6$</td>
<td>1</td>
<td>U</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.01</td>
<td>K</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>0.01</td>
<td>K</td>
</tr>
<tr>
<td>$\gamma_3$</td>
<td>0.01</td>
<td>K</td>
</tr>
<tr>
<td>$n_1$</td>
<td>2</td>
<td>K</td>
</tr>
<tr>
<td>$n_2$</td>
<td>2</td>
<td>K</td>
</tr>
<tr>
<td>$n_3$</td>
<td>2</td>
<td>K</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.01</td>
<td>K</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.01</td>
<td>K</td>
</tr>
</tbody>
</table>

$^a$Setup indicates which parameters are treated as unknown and are to be estimated in each model. Each ‘U’ represents an unknown parameter while ‘K’ represents a known parameter.
Table S2. The parameter setting for model $M_2$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Setup$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{e1}$</td>
<td>0.0025</td>
<td>U</td>
</tr>
<tr>
<td>$k_{p1}$</td>
<td>6</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m1}$</td>
<td>0.1</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e2}$</td>
<td>1/300</td>
<td>U</td>
</tr>
<tr>
<td>$k_{p2}$</td>
<td>10</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m2}$</td>
<td>0.1</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e3}$</td>
<td>0.002</td>
<td>U</td>
</tr>
<tr>
<td>$k_{p3}$</td>
<td>8</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m3}$</td>
<td>0.1</td>
<td>U</td>
</tr>
<tr>
<td>$n_1$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$n_2$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$n_3$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.1</td>
<td>K</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.01</td>
<td>K</td>
</tr>
</tbody>
</table>

$^a$Setup indicates which parameters are treated as unknown and are to be estimated in each model. Each ‘U’ represents an unknown parameter while ‘K’ represents a known parameter.
Table S3. The parameter setting for model $\mathcal{M}3$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Setup$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{p1}$</td>
<td>20</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e1}$</td>
<td>0.025</td>
<td>U</td>
</tr>
<tr>
<td>$n_1$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m1}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$k_{p2}$</td>
<td>14</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e2}$</td>
<td>1/60</td>
<td>U</td>
</tr>
<tr>
<td>$n_2$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m2}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$k_{p3}$</td>
<td>16</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e3}$</td>
<td>1/60</td>
<td>U</td>
</tr>
<tr>
<td>$n_3$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m3}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$k_{p4}$</td>
<td>16</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e4}$</td>
<td>0.02</td>
<td>U</td>
</tr>
<tr>
<td>$n_4$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m4}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$k_{p5}$</td>
<td>15</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e5}$</td>
<td>0.02</td>
<td>U</td>
</tr>
<tr>
<td>$n_5$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m5}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$k_{p6}$</td>
<td>18</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e6}$</td>
<td>1/70</td>
<td>U</td>
</tr>
<tr>
<td>$n_6$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m6}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$k_{p7}$</td>
<td>18</td>
<td>U</td>
</tr>
<tr>
<td>$k_{e7}$</td>
<td>0.025</td>
<td>U</td>
</tr>
<tr>
<td>$n_7$</td>
<td>4</td>
<td>U</td>
</tr>
<tr>
<td>$k_{m7}$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.2</td>
<td>K</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.2</td>
<td>K</td>
</tr>
</tbody>
</table>

$^a$Setup indicates which parameters are treated as unknown and are to be estimated in each model. Each ‘U’ represents an unknown parameter while ‘K’ represents a known parameter.
Table S4. The best parameter solutions for the feedforward loop model.

<table>
<thead>
<tr>
<th>Method</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_{XY}$</th>
<th>$k_{XZ}$</th>
<th>$k_{YZ}$</th>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEDI</td>
<td>1.06</td>
<td>1.3983</td>
<td>0.4423</td>
<td>0.5226</td>
<td>0.0549</td>
<td>3.6083</td>
<td>9.9276</td>
<td>9.927</td>
<td>0.0213</td>
</tr>
<tr>
<td>SRES</td>
<td>0.2635</td>
<td>0.1</td>
<td>1.4998</td>
<td>413.8736</td>
<td>799.1846</td>
<td>1.6520</td>
<td>2.7076</td>
<td>5.7072</td>
<td>0.20403</td>
</tr>
<tr>
<td>TDDM</td>
<td>0.0419</td>
<td>0.01</td>
<td>16.1359</td>
<td>23.8544</td>
<td>0.1079</td>
<td>1.0</td>
<td>1.000</td>
<td>8.000</td>
<td>0.3754</td>
</tr>
</tbody>
</table>

Table S5. Comparison of the results from the variant of $\mathcal{M}3$. The comparison criteria are: the average runtime; the average, best, and worst prediction errors; the average relative error of the best parameter set; and the quality of data extrapolation.

<table>
<thead>
<tr>
<th></th>
<th>PEDI</th>
<th>SRES(^a)</th>
<th>HEKF+MM</th>
<th>TDDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime</td>
<td>33.2min</td>
<td>312.0min</td>
<td>78.2min</td>
<td>27.6min</td>
</tr>
<tr>
<td>Average PE</td>
<td>2651.3</td>
<td>5336.1</td>
<td>N/A(^b)</td>
<td>4821.03</td>
</tr>
<tr>
<td>Best PE</td>
<td>578.6</td>
<td>3766.8</td>
<td>N/A(^b)</td>
<td>4820.7</td>
</tr>
<tr>
<td>Worst PE</td>
<td>4116.3</td>
<td>5955.2</td>
<td>N/A(^b)</td>
<td>4821.3</td>
</tr>
<tr>
<td>Best param(^c)</td>
<td>0.02</td>
<td>212</td>
<td>N/A(^b)</td>
<td>0.26184</td>
</tr>
<tr>
<td>Pred(1)(^d)</td>
<td>31.2</td>
<td>268.0</td>
<td>N/A(^b)</td>
<td>315.0</td>
</tr>
<tr>
<td>Pred(3)(^d)</td>
<td>33.8</td>
<td>316.6</td>
<td>N/A(^b)</td>
<td>473.7</td>
</tr>
<tr>
<td>Pred(5)(^d)</td>
<td>40.8</td>
<td>422.0</td>
<td>N/A(^b)</td>
<td>680.6</td>
</tr>
</tbody>
</table>

\(^a\)With 2,000 generations.
\(^b\)Because all the 10 runs resulted in negative parameter values.
\(^c\)The average relative error of the best parameter solution.
\(^d\)Pred(k) indicates the mean squared distance of the next k time points.
Figure S1. Comparison between PEDI and SA for the estimated parameter combinations of model $\mathcal{M}_1$. The hexagon with the red broken line indicates the true parameter combination. The other hexagons, each with different a color, represent 10 different estimated parameter combinations. (A) The results from PEDI. (B) The results from SA.
Figure S2. Comparison between PEDI and SA for variance of estimates for model $M_1$. (A) Comparison of prediction error for each estimated parameter combination. Each bar with different color indicates the parameter combination indicated by that color in Figure S1. The X-axis is the ID of each parameter estimation run. The Y-axis is the prediction error. (B) A boxplot showing statistics of relative differences for each unknown parameter. The X-axis indicates each of the unknown parameters. The Y-axis indicates the relative difference of parameter values with respect to their sample means.
Figure S3. Comparison of PEDI and SA based on the estimated parameter set with the smallest prediction error from model $M_1$. From the top row, we have the results of $m_1$, $m_2$, and $m_3$. The left-hand-side panels show the comparison of the reconstructed mRNA trajectories between PEDI and SA, while the right-hand-side panels show the comparison of the parameter combination generating the estimated trajectory for the two approaches. Here, PE indicates the prediction error of each mRNA species, and the red point in the middle of each side for the parameter comparison indicates the true value of each parameter.
Figure S4. Refinement of parameters for model $\mathcal{M}_1$, showing how estimated curves were refined at various refinement steps for the best estimate of the parameter combination for model $\mathcal{M}_1$. Here, PE stands for prediction error.
Figure S5. Comparison between PEDI and SA for the estimated parameter combinations of model $M_2$. The left column shows the results from PEDI, while the right one shows the results from SA. The top row shows the parameter set for the regulation of $m_1$. The middle row shows the parameter set for the regulation of $m_2$. The bottom row shows the parameter set for the regulation of $m_3$. In each parameter set, the triangle with the red broken line indicates the true parameter combination. Each of the 10 estimated parameter combinations is color-coded differently.
Figure S6. Comparison between PEDI and SA for variance of estimates for model $\mathcal{M}_2$. (A) Comparison of prediction error for each estimated parameter combination. Each bar with different color indicates the parameter combination indicated by that color in Figure S7. The X-axis is the ID of each parameter estimation run. The Y-axis is the prediction error. (B) A boxplot showing statistics of relative differences for each unknown parameter. The X-axis indicates each of the unknown parameters. The Y-axis indicates the relative difference of parameter values with respect to their sample means.
Figure S7. Comparison of PEDI and SA based on the estimated parameter set with the smallest prediction error from model $M_2$. From the top row, we have the results of $m_1$, $m_2$, and $m_3$. The left-hand-side panels show the comparison of the reconstructed mRNA trajectories between PEDI and SA, while the right-hand-side panels show the comparison of the parameter combination generating the estimated trajectory for the two approaches. Here, PE indicates the prediction error of each mRNA species, and the red point in the middle of each side for the parameter comparison indicates the true value of each parameter.
**Figure S8.** Refinement of parameters for model $M_2$, showing how estimated curves were refined at various refinement steps for the best estimate of the parameter combination for model $M_2$. Here, PE stands for prediction error.
Figure S9. Comparison of the four methods based on the estimated parameter set with the smallest prediction error from model $M_1$. From the top row, we have the results of $m_1$, $m_2$, and $m_3$. The left-hand-side panels show the comparison of the reconstructed mRNA trajectories. Here, each value within parentheses next to each method indicates the prediction error for a given mRNA, the red square points indicate the observed data points, and the dotted red lines indicate the true average trajectories. The right-hand-side panels show the comparison of the best parameter combination from each of the methods. The red point in the middle of each side for the parameter comparison indicates the true value of each parameter.
Figure S10. Comparison of the average relative error of each of the unknown parameter in model $M_1$. 
Figure S11. Comparison of the four methods based on the estimated parameter set with the smallest prediction error from model $M_2$. From the top row, we have the results of $m_1$, $m_2$, and $m_3$. The left-hand-side panels show the comparison of the reconstructed mRNA trajectories. Here, each value within parentheses next to each method indicates the prediction error for a given mRNA, the red square points indicate the observed data points, and the dotted red lines indicate the true average trajectories. The right-hand-side panels show the comparison of the best parameter combination from each of the methods. The red point in the middle of each side for the parameter comparison indicates the true value of each parameter.
Figure S12. Comparison of the average relative error of each of the unknown parameter in model $M_2$. 
Figure S13. Comparison of the four methods based on the estimated parameter set with the smallest prediction error from model $\mathcal{M}3$. Here, we have the results of the seven mRNAs. The left-hand-side panels of each column show the comparison of the reconstructed mRNA trajectories. Here, each value within parentheses next to each method indicates the prediction error for a given mRNA, the red square points indicate the observed data points, and the dotted red lines indicate the true average trajectories. The right-hand-side panels show the comparison of the best parameter combination from each of the methods. The red point in the middle of each side for the parameter comparison indicates the true value of each parameter.
Figure S14. Comparison of the average relative error of each of the unknown parameter in model $M_3$. 