Supplementary Information for

Robust reconstruction of gene expression profiles from reporter gene data using linear inversion

Valentin Zulkower, Michel Page, Delphine Ropers,
Johannes Geiselmann, and Hidde de Jong*

April 10, 2015

Contents

*To whom correspondence should be addressed

S1 Sensitivity of the estimation results to the value of the parameter ϵ

Section 2.1 of the main text describes the practical implementation of the regularization parameter λ, involving the introduction of a parameter ϵ in the discrete differentiation matrix $\mathbf{L_u}$. When the value of ϵ is not carefully chosen, it introduces an unwanted penalty on the initial values of the variable to estimate. This may modify the range of values of λ for which the matrix is invertible and thus influence the estimation results. In this section we investigate the sensitivity of the estimation results to the value of ϵ and suggest how to proceed in finding an appropriate value.

As an illustration, we solved the growth rate estimation problem presented in Figure S1 for several different values of ϵ. The simulated data (in green in panel A) represent the bacterial population volume obtained from the growth rate (dashed line in the same panel), with added noise having the same properties as in the reporter gene data set in Figure 3. For $\epsilon = 1$, we observe as expected that the estimation of μ_0 is negatively biased. The penalization parameter $\lambda = \lambda_{gcv}$, chosen by generalized cross-validation, minimizes the error ErrReg(λ) associated with the regularized problem, defined as the right-hand side of Equation 8 in the main text (dashed vertical line in panel F). For ϵ between 10^{-2} and 10^{-5}, the estimation is unbiased and corresponds well to the real input (panels B-C). The estimation is not sensible to ϵ in this interval and the value of λ_{gcv} is of the same order of magnitude as for the case $\epsilon = 1$. However, for even lower values of ϵ (panels D-E), the same value of λ_{gcv} makes the problem ill-posed (hence the peaks in ErrReg(λ) in panels I-J). As a consequence, the value of λ_{gcv} is aberrant and the resulting estimations are off the mark.

Throughout the paper we used $\epsilon = 10^{-5}$ and verified in each instance the appropriateness of this choice by a sensitivity analysis of the type shown in Figure S1. This procedure is recommended more generally.

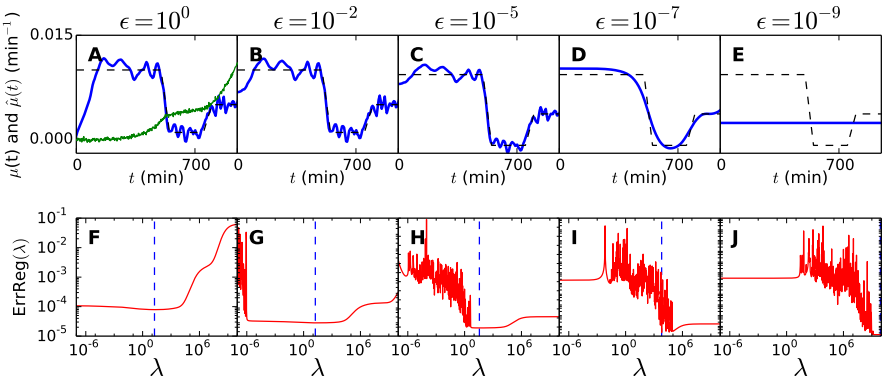

Figure S1: **Effect of parameter ϵ on growth rate estimation from *in-silico* data.** The simulated data (in green) for known input $\mu(t)$ (dashed line) shown in **A**. The estimation results (solid line) for different values of ϵ are shown in **A**-**E**. The corresponding error profile for the regularized problem, ErrReg(λ), and the minimal value λ_{gcv} for different choices of ϵ are shown in **F**-**J**.

S2 Linear inversion problems with linear constraints

It can be useful to impose constraints on the values of the estimated $\mathbf{w}$, to ensure that they are comprised between certain bounds. Equation 6 in the main text can be reformulated as follows, using the definition of the discrete derivation matrix $\mathbf{L_w}$ in the same section:

$$\begin{aligned}
\hat{\mathbf{w}} &= \underset{\mathbf{w}}{\operatorname{argmin}}\ \|\mathbf{H_w w} - \tilde{\mathbf{y}}\|_2^2 + \lambda\|\mathbf{L_w w}\|_2^2 \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ (\mathbf{H_w w} - \tilde{\mathbf{y}})^T(\mathbf{H_w w} - \tilde{\mathbf{y}}) + \lambda(\mathbf{L_w w})^T(\mathbf{L_w w}) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathbf{w}^T\mathbf{H_w}^T\mathbf{H_w w} - 2\tilde{\mathbf{y}}^T\mathbf{H_w w} + \tilde{\mathbf{y}}^T\tilde{\mathbf{y}} + \lambda\mathbf{w}^T\mathbf{L_w}^T\mathbf{L_w w} \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathbf{w}^T(\mathbf{H_w}^T\mathbf{H_w} + \lambda\mathbf{L_w}^T\mathbf{L_w})\mathbf{w} - 2\tilde{\mathbf{y}}^T\mathbf{H_w w}.
\end{aligned}$$

To this quadratic minimization problem we can add a set of linear constraints of the form

$$\mathbf{G}_1\mathbf{w} = \mathbf{c}, \tag{S1}$$

$$\mathbf{G}_2\mathbf{w} \leq \mathbf{0}, \tag{S2}$$

where $\mathbf{G}_1, \mathbf{G}_2$ are constant matrices and $\mathbf{c}$ a constant vector. In this paper we want to ensure that the initial conditions (which represent quantities of molecules or volumes) and the input variable (which represents a growth rate, promoter activity, or protein concentration in Section 3 of the main text) is positive. This corresponds to setting

$$(\mathbf{G}_1, \mathbf{c}, \mathbf{G}_2) = (\mathbf{0}, \mathbf{0}, \mathbf{I}).$$

Several solvers have been proposed for the general quadratic programming problem, and in particular for the special case of ensuring positive solutions. In this paper we used the solver `cvxopt.solvers.qp` from the Python module `cvxopt`, which is well adapted to large-scale problems (Andersen et al., 2012). Notice that it is not possible to use generalized cross-validation on the constrained problem. Therefore, we first used GCV on the unconstrained problem to select the regularization parameter λ, and then solved the constrained problem for that particular value of λ.

S3 Computation of observation matrices

The computation of the observation matrix $\mathbf{H_w} = \begin{pmatrix}\mathbf{H}_{\mathbf{x}_0} & \mathbf{H_u}\end{pmatrix}$ defined in Section 2.1 of the main text can be achieved in a straightforward way. The jth column of $\mathbf{H}_{\mathbf{x}_0}$ is the vector of values obtained by solving Equation 1 at times $(t_i)_{1\leq i\leq N_y}$, using $u(t) = 0$ and

$$\mathbf{x}_0 = (0, \cdots, 0, \underbrace{1}_{\mathbf{x}_0[j]}, 0, \cdots, 0).$$

The jth column of matrix $\mathbf{H_u}$ is obtained by solving the same system with $\mathbf{x}_0 = \mathbf{0}$ and

$$u(t) = \mathbf{1}_{[\tau_j, \tau_{j+1}[}(t).$$

The computation of $\mathbf{H_w}$ can be performed using a numerical differential equation solver, but this is usually time-consuming because of the large number of ODE integrations required $(n + N_u)$. An alternative is to use the explicit solution of Equation 2 and exploit the specific form that Equation 1 takes when estimating growth rate, promoter activity, and protein concentration. The latter approach will be further developed in the remainder of this section.

S3.1 Explicit formula for the observation matrix for growth rate estimation

In section Section 3.2 we proposed the following model as a basis for the estimation of $\mu(t)$:

$$\frac{d}{dt}(\alpha\, V)(t) = \tilde{V}(t)\, \mu(t). \tag{S3}$$

This model admits the following general solution:

$$\alpha\, V(t) = \alpha\, V(0) + \int_0^t \tilde{V}(\sigma)\, \mu(\sigma)\, d\sigma. \tag{S4}$$

The observation matrix is of the form $\mathbf{H_w} = \begin{pmatrix} \mathbf{H_{x_0}} & \mathbf{H_u} \end{pmatrix}$. $\mathbf{H_{x_0}}$ has dimensions $1 \times N_y$ and its values are obtained by computing Equation S4 at the different observation times $(t_i)_{1 \leq i \leq N_y}$, with $\alpha\, V(0) = 1$ and $\mu(t) = 0$ for all t. Therefore, we have

$$\mathbf{H_{x_0}} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

The element of $\mathbf{H_u}$ at position $[i, j]$ is computed by evaluating Equation S4 at time t_i with $\alpha\, V(0) = 0$ and

$$\mu(t) = \mathbf{1}_{[\tau_j, \tau_{j+1}[}(t).$$

This leads to

$$\mathbf{H_u}[i, j] = \int_0^{t_i} \tilde{V}(\sigma)\, \mathbf{1}_{[\tau_j, \tau_{j+1}[}(\sigma)\, d\sigma = \begin{cases} 0 & \text{if } t_i < \tau_j, \\ \int_{\tau_j}^{\min(t_i, \tau_{j+1})} \tilde{V}(\sigma)\, d\sigma & \text{otherwise.} \end{cases}$$

The size of the intervals $[\tau_i, \tau_{i+1}[$, denoted by $\delta\tau$, can be chosen arbitrarily small, so we will suppose that the volume is constant on each interval. This allows the expression above to be simplified and we obtain the following approximate expression of $\mathbf{H_u}[i, j]$, which is used for the estimation of the growth rate in the figures of the main text, and in the `WellFARE` package:

$$\mathbf{H_u}[i, j] \simeq \tilde{V}(\tau_j) \max\left(0, \min(t_i - \tau_j, \delta\tau)\right).$$

S3.2 Efficient computation of the observation matrix for promoter activity estimation

In the main text we presented the following ODE model for the expression of the reporter gene:

$$\begin{cases} \frac{d}{dt}M(t) = k_M\, a(t)\, V(t) - d_M\, M(t) = k'_M\, a(t)\, \alpha\, V(t) - d_M\, M(t), \\ \frac{d}{dt}R_u(t) = k_U\, M(t) - (d_R + k_R)\, R_u(t), \\ \frac{d}{dt}R(t) = k_R\, R_u(t) - d_R\, R(t), \end{cases}$$

where $k'_M = k_M/\alpha$.

The observation matrix is of the form $\mathbf{H_w} = \begin{pmatrix}\mathbf{H_{x_0}} & \mathbf{H_u}\end{pmatrix}$. The element of $\mathbf{H_u}$ at position $[i,j]$ is computed by solving the ODE system, with $\begin{pmatrix}M(0) & R_u(0) & R(0)\end{pmatrix} = \mathbf{0}$ and

$$a(t) = \mathbf{1}_{[\tau_j,\tau_{j+1}[}(t),$$

and then evaluating $R(t)$ at time-point t_i. We can reformulate this as follows, using the input-output system notation from Section 2.1 of the main text (Chen, 1970):

$$\mathbf{H_u}[i,j] = R(t_i, \mathbf{1}_{[\tau_j,\tau_{j+1}[}, \mathbf{0}).$$

Computing $\mathbf{H_u}$ in this way, however, would require the solution of as many ODE systems as there are intervals $[\tau_j, \tau_{j+1}[$, typically on the order of 1000.

A more efficient procedure for computing $\mathbf{H_u}$ can be obtained by choosing a suitable approximation. Assume that the intervals $[\tau_j, \tau_{j+1}[$ are of equal length $\delta\tau$ and small compared to the characteristic variation time of $V(t)$. As a consequence, $V(t)$ can be supposed constant over the interval $[\tau_j, \tau_{j+1}[$, and we can use the following approximated system to compute the $[i,j]$th element of $\mathbf{H_u}$:

$$\begin{cases} \frac{d}{dt}M(t) \simeq k'_M\, \alpha\, V(\tau_j)\, \mathbf{1}_{[\tau_j,\tau_{j+1}[}(t) - d_M\, M(t), \\ \frac{d}{dt}R_u(t) = k_U\, M(t) - (d_R + k_R)\, R_u(t), \\ \frac{d}{dt}R(t) = k_R\, R_u(t) - d_R\, R(t). \end{cases} \tag{S5}$$

It is easy to see that this system is linear in $\alpha\, V$, so that we can write

$$R(t_i, \mathbf{1}_{[\tau_j,\tau_{j+1}[}, \mathbf{0}) = \alpha\, V(\tau_j)\, R_1(t_i, \mathbf{1}_{[\tau_j,\tau_{j+1}[}, \mathbf{0}),$$

where R_1 is the output of the system of Equation S5 with $\alpha\, V(\tau_j)$ set to 1. Because the coefficients of this system are time-invariant, we also have that

$$R_1(t_i, \mathbf{1}_{[\tau_j,\tau_{j+1}[}, \mathbf{0}) = \begin{cases} 0 & \text{if } t_i < \tau_j, \\ R_1(t_i - \tau_j, \mathbf{1}_{[0,\delta\tau[}, \mathbf{0}) & \text{otherwise.} \end{cases}$$

This leads to the following approximation of $H_{\mathbf{u}}[i,j]$:

$$\mathbf{H_u}[i,j] = R(t_i, \mathbf{1}_{[\tau_j,\tau_{j+1}[}, \mathbf{0}) \simeq \begin{cases} 0 & \text{if } t_i < \tau_j, \\ \tilde{V}(\tau_j)\, R_1(t_i - \tau_j, \mathbf{1}_{[0,\delta\tau[}, \mathbf{0}) & \text{otherwise.} \end{cases}$$

The advantage of this approximate method for computing $\mathbf{H_u}$ is that it requires the ODE system of Equation S5 to be solved only once, replacing the term $\alpha\, V(\tau_j)\, \mathbf{1}_{[\tau_j,\tau_{j+1}[}(t)$ by $\mathbf{1}_{[0,\delta\tau[}(t)$ and evaluating the output at all time-points t_i, instead of solving N_u ODEs.

S3.3 Computation of the observation matrix for promoter activity estimation in a reduced gene expression model

In Section 3.3 of the main text, the production of mature GFP proteins was described as a three-step process (transcription, translation, and maturation). In this section we consider a simplified version of this model, which allows us to explicitly formulate the observation matrix $\mathbf{H_w}$ as a function of the measured signal $\left(\tilde{V}(t_i)\right)_{1\leq i\leq N_y}$ and the degradation constant d_R.

When both transcription and maturation are fast as compared to the other processes involved in the expression of the reporter gene, it can be assumed that their effect on the dynamics of the folded reporter is negligible. In this case, the entire process of synthesizing mature GFP can be lumped into a single step and Equation 15 in the main text becomes:

$$\frac{d}{dt}R(t) = k'_R\, \alpha\, V(t)\, a(t) - d_R\, R(t), \tag{S6}$$

where k'_R denotes a lumped protein synthesis parameter.

The quantity of reporter protein $R(t)$ can be explicitly formulated as a function of $a(t)$ by solving Equation S6:

$$R(t) = R(0)\, e^{-d_R t} + e^{-d_R t} \int_0^t e^{d_R \sigma} k'_R\, \alpha\, V(\sigma)\, a(\sigma)\, d\sigma. \tag{S7}$$

In other words, the promoter activity is linearly related to the amount of reporter protein. In what follows, we set $k'_R = 1$, which allows the promoter activity to be estimated up to an unknown proportionality constant (Section S6).

The observation matrix $\mathbf{H_w}$ for the corresponding linear inversion problem is of the general form $\mathbf{H_w} = \begin{pmatrix}\mathbf{H_{x_0}} & \mathbf{H_u}\end{pmatrix}$, where $\mathbf{H_{x_0}}$ is given by:

$$\mathbf{H_{x_0}} = \begin{pmatrix} e^{-d_R t_0} \\ e^{-d_R t_1} \\ \vdots \\ e^{-d_R t_{N_y}} \end{pmatrix}.$$

The element of $\mathbf{H_u}$ at position $[i,j]$ is computed by evaluating Equation S7 at time t_i for $R(0) = 0$ and $a(t) = \mathbf{1}_{[\tau_j,\tau_{j+1}[}(t)$. This leads to

$$\mathbf{H_u}[i,j] = \begin{cases} 0 & \text{if } t_i < \tau_j, \\ e^{-d_R t_i} \int_{\tau_j}^{\min(\tau_j+\delta\tau,\, t_i)} e^{d_R \sigma} \alpha\, V(\sigma)\, d\sigma & \text{otherwise,} \end{cases} \tag{S8}$$

where $\delta\tau$ denotes the length of the time-interval $[\tau_j, \tau_{j+1}[$. We can exploit the fact that $\delta\tau$ can be chosen arbitrarily small to simplify the integral by assuming

that the volume is approximately constant over the time-interval considered:

$$\begin{aligned}\int_{\tau_j}^{\min(\tau_j+\delta\tau,t_i)} e^{d_R\sigma}\alpha\,V(\sigma)\,d\sigma &\simeq \alpha\,V(\tau_j)\int_{\tau_j}^{\min(\tau_j+\delta\tau,t_i)} e^{d_R\sigma}d\sigma\\ &= \frac{1}{d_R}\,\alpha\,V(\tau_j)\left(e^{d_R\,\min(\tau_j+\delta\tau,t_i)}-e^{d_R\tau_j}\right)\\ &\simeq \frac{1}{d_R}\,\tilde{V}(\tau_j)\left(e^{d_R\,\min(\tau_j+\delta\tau,t_i)}-e^{d_R\tau_j}\right).\end{aligned}$$

As a consequence,

$$\mathbf{H_u}[i,j] \simeq \begin{cases} 0 & \text{if } t_i < \tau_j,\\ \frac{1}{d_R}\,\tilde{V}(\tau_j)\left(e^{d_R\,\min(\tau_j+\delta\tau,t_i)}-e^{d_R\tau_j}\right) & \text{otherwise.}\end{cases} \tag{S9}$$

The latter expression is used to compute promoter activities in the `WellFARE` package.

S3.4 Explicit formula for the observation matrix for protein concentration estimation in a reduced gene expression model

In the main text we have presented the production of a protein of interest and its reporter as multistep processes. The observation matrix allowing the estimation of the protein concentration $p(t)$ from the absorbance and fluorescence data can be computed by means of the procedure in Section 2.2. In this section, like in Section S3.3, we will simplify the problem by considering single-step gene expression models, enabling an explicit formulation of the observation matrix.

Using the same notation as in Section S3.3, $P'(t)$ (defined in Section 3.4) and $R(t)$ are driven by the following one-step gene expression models:

$$\frac{d}{dt}P'(t) = k'_P\,\alpha\,V(t)\,a(t) - d_P\,P'(t), \tag{S10}$$

$$\frac{d}{dt}R(t) = k'_R\,\alpha\,V(t)\,a(t) - d_R\,R(t), \tag{S11}$$

$$p(t) = P'(t)/\left(\alpha V(t)\right) \tag{S12}$$

where k'_P and k'_R denote lumped protein synthesis parameters. Notice that the degradation constants of the protein of interest (d_P) and the reporter protein (d_R) are different *a priori*.

This model enables $R(t)$ to be directly expressed as a function of $P'(t)$. We first derive the following expression of $\alpha\,V(t)\,a(t)$ from Equation S10:

$$\alpha\,V(t)\,a(t) = \frac{1}{k'_P}\left(d_P\,P'(t) + \frac{d}{dt}P'(t)\right),$$

and then inject this expression into Equation S11:

$$\frac{d}{dt}R(t) = K\left(d_P\,P'(t) + \frac{d}{dt}P'(t)\right) - d_R\,R(t),$$

where $K = k'_R/k'_P$. The above differential equation for $R(t)$, with input $P'(t)$, can be solved exactly, yielding

$$R(t) = \underbrace{K\,P'(t)}_{A(t)} + \underbrace{K\,(d_R - d_P)\,e^{-d_R t}\int_0^t e^{d_R\sigma}\,P'(\sigma)\,d\sigma}_{B(t)} + \underbrace{(R(0) - K\,P'(0))\,e^{-d_R t}}_{C(t)}. \tag{S13}$$

The terms $A(t), B(t), C(t)$ in Equation S13 admit a simple interpretation. $A(t)$ shows that $R(t)$ will, at least partly, follow the variations of $P'(t)$. This is to be expected, as $P'(t)$ and $R(t)$ are driven by the same promoter activity $a(t)$. $B(t)$ is a correction term accounting for the difference in degradation constants of the reporter protein and the protein of interest. $C(t)$ accounts for the differences in initial conditions $P'(0)$ and $R(0)$. Equation S13 can be rewritten as

$$\begin{aligned} R(t) = R(0)\,e^{-d_R t} + K\Bigg(&\alpha\,V(t)\,p(t) - \alpha\,V(0)\,p(0)e^{-d_R t} \\ &+(d_R - d_P)\,e^{-d_R t}\int_0^t e^{d_R\sigma}\,p(\sigma)\,\alpha\,V(\sigma)\,d\sigma\Bigg). \end{aligned} \tag{S14}$$

Note that this formulation shows the linear relationship between the output $R(t)$, the input $p(t)$, and initial condition $R(0)$, which correspond respectively to $y_2(t)$, $u(t)$, and $\mathbf{x}_{0,2}$ in Section 2.2 of the main text. Like in Section S3.3, we assume that $k'_R = k'_P = 1$ to simplify computations and obtain a proportional estimator for $p(t)$ (Section S6).

The observation matrix for the estimation problem is of the general form $\mathbf{H_w} = \begin{pmatrix}\mathbf{H}_{\mathbf{x}_0} & \mathbf{H_u}\end{pmatrix}$, where

$$\mathbf{H}_{\mathbf{x}_0} = \begin{pmatrix} e^{-d_R t_0} \\ e^{-d_R t_1} \\ \vdots \\ e^{-d_R t_{N_y}} \end{pmatrix},$$

obtained by setting $p(t) = 0$ for all t and $R(0) = 1$. The element of $\mathbf{H_u}$ at position $[i,j]$ is computed by evaluating Equation S14 at time t_i for $R(0) = 0$ and $p(t) = \mathbf{1}_{[\tau_j,\tau_{j+1}[}(t)$. This leads to

$$\mathbf{H_u}[i,j] = \begin{cases} 0 & \text{if } t_i < \tau_j, \\ \alpha\,V(t_i) + (d_R - d_P)\,e^{-d_R t_i}\int_{\tau_j}^{t_i} e^{d_R\sigma}\,\alpha\,V(\sigma)\,d\sigma & \text{if } \tau_j \le t_i < \tau_{j+1}, \\ (d_R - d_P)\,e^{-d_R t_i}\int_{\tau_j}^{\tau_{j+1}} e^{d_R\sigma}\,\alpha\,V(\sigma)\,d\sigma & \text{if } t_i \ge \tau_{j+1}. \end{cases} \tag{S15}$$

In this expression we can use $\tilde{V}$ instead of $\alpha\,V$, and approximate the integral, like in Section S3.3. This results in an approximate but practical formula for $\mathbf{H_u}[i,j]$:

$$\mathbf{H_u}[i,j] = \begin{cases} 0 & \text{if } t_i < \tau_j, \\ \tilde{V}(\tau_j)\left(1 + \dfrac{d_R - d_P}{d_R}\left(1 - e^{d_R(\tau_j - t_i)}\right)\right) & \text{if } \tau_j \le t_i < \tau_{j+1}, \\ \tilde{V}(\tau_j)\,\dfrac{d_R - d_P}{d_R}\left(e^{d_R(\tau_{j+1} - t_i)} - e^{d_R(\tau_j - t_i)}\right) & \text{if } t_i \ge \tau_{j+1}. \end{cases} \tag{S16}$$

The latter expression is used to compute protein concentrations in the `WellFARE` package.

S4 Reporter gene experiments: materials and methods

The *E. coli* wild-type strain used in this study is the BW25113 strain (Baba et al., 2006). The reporter strains were obtained by transforming the wild-type strain with a reporter plasmid, bearing a transcriptional fusion of the *crp*, *fis*, *gyrA* and *acs* promoter regions with the *gfp* reporter gene, and a promoterless vector for background correction (Table 1). The reporter gene codes either for a stable and fast-folding version of the GFP reporter (GFPmut2) or for a less stable allele (GFPmut3). More information on the half-live and maturation time can be found in (Berthoumieux et al., 2013).

Plasmid	Characteristics	Reference or source
pZE*gfp*	Amp^r, colE1 *ori*, *gfpmut3*	(de Jong et al., 2010)
pZE*fis-gfp*	Amp^r, colE1 *ori*, p*fis-gfpmut3*	(de Jong et al., 2010)
pZE*gyrA-gfp*	Amp^r, colE1 *ori*, p*gyrA-gfpmut3*	(Boyer et al., 2010)
pUA66*gfp*	Kan^r, pSC101 *ori*, *gfpmut2*	(Zaslaver et al., 2006)
pUA66*crp-gfp*	Kan^r, pSC101 *ori*, p*crp-gfpmut2*	(Zaslaver et al., 2006)
pUA66*acs-gfp*	Amp^r, pSC101 *ori*, p*acs-gfpmut2*	(Baptist et al., 2013)

Table 1: **Reporter plasmids used in this study.**

Glycerol stocks (-80°C) of the above-mentioned reporter strains were grown overnight (about 15 h) at 37°C, with shaking at 200 rpm, in M9 minimal medium (Miller, 1972) supplemented with 0.3% glucose and mineral trace elements. For plasmid-carrying strains, the growth medium was supplemented with 100 μg ml^{-1} ampicilin or kanamycin. The overnight cultures were diluted into a 96-well microplate, so as to obtain an adjusted initial OD_{600} of 0.1 for *fis* and *gyrA*, 0.001 for *crp* and *acs*. The wells of the microplate contain M9 minimal medium supplemented with 0.3% glucose, mineral trace elements, and 1.2% of the buffering agent HEPES (4-(2-hydroxyethyl)-1-piperazineethanesulfonic acid) for maintaining physiological pH levels in the growth medium. No antibiotics were added at this stage. The wells were covered with 60 μl of mineral oil to avoid evaporation. The microplate cultures were then grown for about 24 h at 37°C, with agitation at regular intervals, in the Fusion microplate reader (Perkin Elmer).

During a typical experimental run, we acquire about 110 readings each of absorbance (600 nm) and fluorescence (485/520 nm). From the measured signal we remove the background signals of absorbance and fluorescence measured on wells containing growth medium only and strains carrying a promoterless reporter plasmid, respectively (Figure S2).

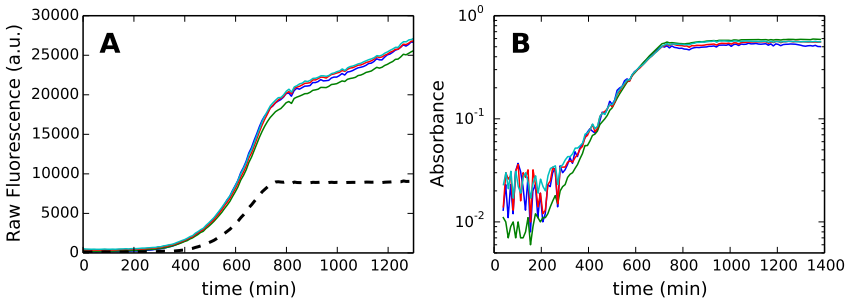

Figure S2: **Experimental data for the reporter gene experiment with the strain carrying the pUA66*crp-gfp* reporter plasmid, shown in Figure 3 of the main text. A.** Primary fluorescence data. The dotted line indicates the auto-fluorescence curve measured on a strain carrying no reporter gene, and is subtracted from the solid lines to obtain the fluorescence profiles shown in Figure 3. **B.** Absorbance corrected by subtracting the (constant) absorbance measured on a well containing only growth medium.

S5 Numerical evaluation of the linear inversion methods

In this section we test the ability of the proposed linear inversion methods to correctly estimate different shapes of growth rate, promoter activity, and protein concentration profiles. We generated 100 absorbance and fluorescence data sets for defined growth rate and promoter activity profiles, similar to those observed for the gene *acs* in the reporter gene experiments in Section 3.1 of the main text (Figure 3). In panel A the ability of the method to reconstruct different growth rate profiles is tested, whereas panels B and C consider different promoter activity and protein concentration profiles (with absorbance data from panel A1), respectively. In every case considered, the methods succeed in providing an almost unbiased estimate of the gene expression quantities.

We also compared the linear inversion methods with other methods, in particular indirect approaches that plug empirically smoothed versions of the data into the measurement models (Figure 4 in the main text). Below we extend this analysis, for the estimation of the growth rate from absorbance measurements, by showing that increasing the smoothing parameter to reduce the variance of the estimates introduces a strong bias (Figure S4). The growth rate is shown as the dotted curve and the absorbance data are the same as in Figure 4 in the main text.

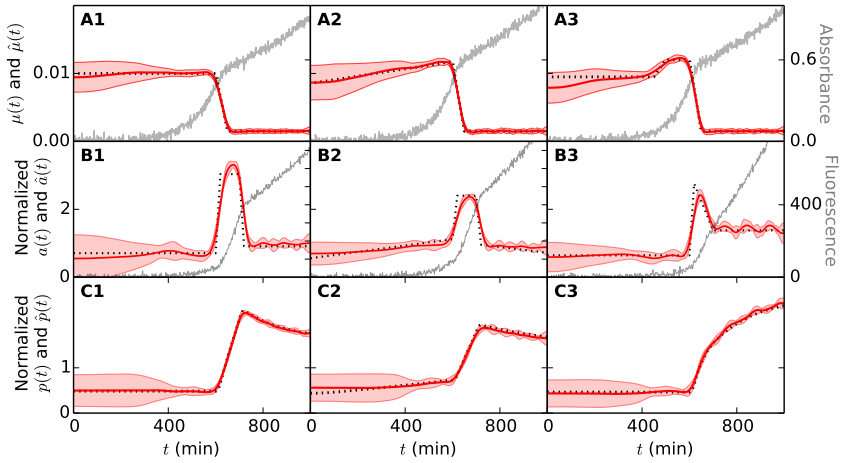

Figure S3: ***In-silico* experiments for testing the ability of the linear inversion methods to correctly estimate different growth rate, promoter activity, and protein concentration profiles.** The estimation results for growth rate, promoter activity, and protein concentration are shown in panels **A**-**C**, respectively. The dotted lines show the profiles used for generating the 100 data sets, the grey solid lines example absorbance and fluorescence time-series data, and the red solid line and the shaded area the mean ± one standard deviation of the 100 estimations, respectively.

S6 Linear inversion when parameters in the gene expression model are unknown

Equation 15 in the main text describes the gene expression model on which the estimation methods are based:

$$\begin{cases} \dfrac{d}{dt}M(t) = k'_M a(t)\widehat{\alpha V}(t) - d_M M(t), \\ \dfrac{d}{dt}R_u(t) = k_U M(t) - (d_R + k_R)R_u(t), \\ \dfrac{d}{dt}R(t) = k_R R_u(t) - d_R R(t), \end{cases}$$

and we remind that $\tilde{R}(t_i) = \beta R(t) + \nu$ (Equation 10 in the main text). The constants k'_M, k_U, and β are generally unknown. In this section we show that the profile of the promoter activity can still be estimated, up to an (unknown) proportionality constant, using a linear inversion.

We consider the following transformed variables:

$$R^*(t) = \beta R(t), \quad R^*_u(t) = \beta R_u(t), \quad M^*(t) = k_U \beta M(t), \quad a^*(t) = k_U k'_M \beta a(t).$$

Replacing the variables M, R_u, R, a in the ODE system above by their starred

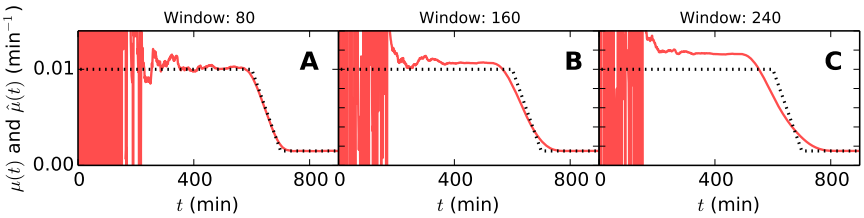

Figure S4: **Growth rate estimation by smoothing of the absorbance measurements.** The growth rate was computed from smoothed absorbance measurements of the volume by means of Equation 12 in the main text. The plots show the true value of the growth rate (dotted line), used to generate 100 absorbance data sets, and the mean (solid red line) ± one standard deviation (shaded area) of the growth rate estimations. Different levels of smoothing of the absorbance data were considered, using sliding windows of different length (80, 160, and 240 data points), corresponding to the panels **A**-**C**, respectively.

counterpart we obtain the following system:

$$\begin{cases} \dfrac{d}{dt}M^*(t) = a^*(t)\widehat{\alpha V}(t) - d_M M^*(t), \\ \dfrac{d}{dt}R_u^*(t) = M^*(t) - (d_R + k_R)R_u^*(t), \\ \dfrac{d}{dt}R^*(t) = k_R R_u^*(t) - d_R R^*(t). \end{cases}$$

Moreover, $\tilde{R}(t_i) = R^*(t) + \nu$. This system is equivalent to Equation 15 in the main text when $\beta = k_U = k'_M = 1$ and can thus be used to estimate the profile $a^*(t)$, which is proportional to $a(t)$ (the proportionality constant being unknown).

The same approach can be applied to the gene expression system of Equation 17 in the main text to show that it is possible to obtain an estimator or the profile of the protein concentration $p(t)$ (up to an unknown proportionality constant) in the absence of reliable values for the parameters k_N and k_P.

S7 Software implementation of the linear inversion methods

The linear inversion methods discussed in this article have been implemented in the Python library `WellFARE` and are available online through the web application `WellInverter`. In this section we briefly describe `WellFARE` and `WellInverter`, and we refer to the dedicated web pages for more information.

S7.1 The `WellFARE` Python Package

`WellFARE` (well Fluorescence Analysis for Reporter Experiments) is a Python library released under an LGPL licence, implementing the methods for growth rate, promoter activity and protein concentration estimation developed in the main text. In addition, the library provides practical tools for the treatment of

data from reporter experiments, such as automated outlier removal, data synchronization, and parsing of Excel files generated by the TECAN Infinite Pro microplate reader. `WellFARE` uses extensively the Python core scientific library `SciPy` (Jones et al., 2001). The treated data can be exported in the form of JSON objects (Crockford, 2006). Source code, documentation and installation instructions for the `WellFARE` library and its command-line and JSON interfaces are available at the following address:

`https://github.com/ibis-inria/wellfare`

The code used to generate the figures of the main text is available in the `examples` folder of the library.

S7.2 The `WellInverter` web application

The `WellInverter` web application provides online access to the linear inversion methods without having to install the software locally. The server part of `WellInverter` is based on the Python library `WellFARE`, the computational core of the application. It also provides methods for managing experimental and user data as well as storing analysis parameters in JavaScript Object Notation (JSON) format (Crockford, 2006). The client part of `WellInverter` is the graphical user interface of the application, accessible through a web browser. It allows the user to upload, analyze, and visualize the results of a reporter gene experiment as well as downloading the results for further treatment. The client part is written in Javascript, and communicates with the server using Ajax (Asynchronous JavaScript and XML) calls (Garrett, 2006). More information on access to `WellInverter` and a tutorial are available at the following address:

`https://team.inria.fr/ibis/wellinverter`

A test account has been opened with username `guest` and password `guest2015`. The reporter gene data set on the server has been used in the main text.

References

Andersen, M. S., Dahl, J., Liu, Z., and Vandenberghe, L. (2012). Interior-point methods for large-scale cone programming. In Sra, S., Nowozin, S., and Wright, S. J., editors, *Optimization for Machine Learning*, pages 55–83. MIT Press.

Baba, T., Ara, T., Hasegawa, M., Takai, Y., Okumura, Y., Baba, M., Datsenko, K., Tomita, M., Wanner, B., and Mori, H. (2006). Construction of *Escherichia coli* K-12 in-frame, single-gene knockout mutants: the Keio collection. *Mol. Syst. Biol.*, 2:2006.0008.

Baptist, G., Pinel, C., Ranquet, C., Izard, J., Ropers, D., de Jong, H., and Geiselmann, J. (2013). A genome-wide screen for identifying all regulators of a target gene. *Nucleic Acids Res.*, 41(17):e164.

Berthoumieux, S., de Jong, H., Baptist, G., Pinel, C., Ranquet, C., Ropers, D., and Geiselmann, J. (2013). Shared control of gene expression in bacteria by transcription factors and global physiology of the cell. *Mol. Syst. Biol.*, 9:634.

Boyer, F., Besson, B., Baptist, G., Izard, J., Pinel, C., Ropers, D., Geiselmann, J., and de Jong, H. (2010). WellReader: a MATLAB program for the analysis of fluorescence and luminescence reporter gene data. *Bioinformatics (Oxford, England)*, 26(9):1262–3.

Chen, C. (1970). *Introduction to Linear System Theory*. Holt, Rinehart and Winston, New York.

Crockford, D. (2006). The application/json media type for JavaScript Object Notation (JSON). http://tools.ietf.org/html/rfc4627.

de Jong, H., Ranquet, C., Ropers, D., Pinel, C., and Geiselmann, J. (2010). Experimental and computational validation of models of fluorescent and luminescent reporter genes in bacteria. *BMC Syst. Biol.*, 4:55.

Garrett, J. (2006). Ajax: A new approach to web applications. http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. http://www.scipy.org/.

Miller, J. (1972). *Experiments in Molecular Genetics*. CSHL Press, Cold Spring Harbor, NY.

Zaslaver, A., Bren, A., Ronen, M., Itzkovitz, S., Kikoin, I., Shavit, S., Liebermeister, W., Surette, M., and Alon, U. (2006). A comprehensive library of fluorescent transcriptional reporters for *Escherichia coli*. *Nat. Methods*, 3(8):623–8.