Controlling *E. coli* gene expression noise

Kyung Hyuk Kim, Kiri Choi, Bryan Bartley, Herbert M. Sauro.

Abstract

Intracellular protein copy numbers show significant cell-to-cell variability within an isogenic population due to the random nature of biological reactions. Here we show how the variability in copy number can be controlled by perturbing gene expression. Depending on the genetic network and host, different perturbations can be applied to control variability. To understand more fully how noise propagates and behaves in biochemical networks we developed stochastic control analysis (SCA) which is a sensitivity-based analysis framework for the study of noise control. Here we apply SCA to synthetic gene expression systems encoded on plasmids that are transformed into *Escherichia coli*. We show that (1) dual control of transcription and translation efficiencies provides the most efficient way of noise-vs.-mean control. (2) The expressed proteins follow the gamma distribution function as found in chromosomal proteins. (3) One of the major sources of noise, leading to the cell-to-cell variability in protein copy numbers, is related to bursty translation. (4) By taking into account stochastic fluctuations in autofluorescence, the correct scaling relationship between the noise and mean levels of the protein copy numbers was recovered for the case of weak fluorescence signals.

Keywords

Synthetic biology, gene expression noise, stochasticity, noise control, two state model, stochastic control analysis

Kyung Hyuk Kim, Kiri Choi, Bryan Bartley, and Herbert M. Sauro are with the Department of Bioengineering, University of Washington, Seattle, WA 98195 USA. The e-mail address of Kyung H. Kim is kkim@uw.edu.

Controlling *E. coli* gene expression noise

I. Introduction

CELL-TO-CELL variability in protein copy numbers within isogenic populations are typically observed in various types of cells due to underlying random biochemical reaction processes [1], [2], [3]. The variability can lead to noise-induced cellular phenotypes such as cellular differentiation [3], multiple stability [4], and either sensitivity enhancement or suppression [5], [6]. Here we investigate the ability to differentially control the noise and mean levels of gene expression in *E. coli*.

Such differential control in gene expression has been achieved in different organisms such as yeast [7], [8], [9], soil bacteria [10], and mammalian cell lines [11]. In *E. coli*, systematic noise control have not been performed by perturbing promoter DNA sequences and ribosome binding sites, while most studies have been focused on genome-wide expression without such perturbations [16], [21], [2]. Here we aim to understand differential control of mean and noise levels of protein concentrations at the single cell levels of *E. coli*.

The approach we use is based on stochastic control analysis (SCA)[13], [12], a body of theory we developed and reported in previous publications. SCA is a sensitivity analysis framework, that is a direct extension of metabolic control analysis [14], [15] to the stochastic regime [13]. This approach is based on a *local* sensitivity analysis that can be applied to study first-order effects of finite-size perturbations. SCA can identify which parameters in stochastic systems – here, gene regulatory circuits – need to be varied by how much to achieve a desired control aim. This includes orthogonal control of noise levels with respect to mean levels, and simultaneous changes in noise and mean levels in the same or opposite directions for the same or different protein species. SCA can provide control efficiency and strength to identify the most effective control schemes that are experimentally relevant [12]. Here, we apply SCA experimentally to *E. coli* genetic systems.

In this paper, gene circuits are encoded on plasmid backbones, which are transformed into *E. coli* MG1655. The circuits express green fluorescent proteins (GFP) under the lac-promoter. We perturbed the expression system by inducing the promoter with isopropyl β-D-1-thiogalactopyranoside (IPTG) and using a library of ribosome binding sites (RBS). We found that by taking into account stochastic fluctuations in autofluorescence, scaling relationship between GFP signal noise and mean levels can be extended to weak signal regions, where autofluorescence becomes moderately strong. This implies that when fluorescent signals are not strong enough compared to autofluorescence, stochasticity in autofluorescence can be systematically taken into account to characterize cellular systems. In addition, we aimed to understand what the major sources of GFP signal noise are by investigating the scaling relationship between the GFP signal noise and mean levels via promoter induction and RBS perturbation. We found that one of the major noise sources is bursty translation.

II. Stochastic control analysis: Review

SCA [12], [13] is a local sensitivity analysis based on control coefficients, which are defined approximately as percentage change in a response signal (y) divided by the percentage change in a system parameter (p):

$$C_p^y = \frac{p}{y}\frac{dy}{dp} = \frac{d\log y}{d\log p}.$$

We note that the slope in the log-log plot of y vs. p corresponds to C_p^y. The response signal can be the mean or noise levels of mRNAs or proteins. The parameters can include transcription and translation efficiencies, degradation rates of mRNAs and proteins, dilution rate due to cell growth, and reaction rates of transcription-factor binding and unbinding from promoter regions, etc. Another important quantity in SCA, is the control vector, each element of which corresponds to a control coefficient for a given response signal (y):

$$\mathbf{C}_{\mathbf{p}}^y = (C_{p_1}^y, C_{p_2}^y, \cdots, C_{p_N}^y),$$

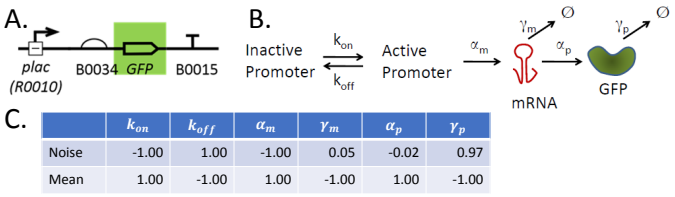
where N defines the number of parameters (dimension of the parameter space) that will be varied to control the value of y. In this paper, we are mostly interested in dual control of transcription and translation efficiencies, i.e., $N = 2$. One of the important properties of the control vector is that its inner-product with a parameter perturbation vector $\delta\mathbf{p}$ becomes the amount of change in the response signal δy,

$$\mathbf{C}_{\mathbf{p}}^y \cdot \frac{\delta\mathbf{p}}{\mathbf{p}} = \frac{\delta y}{y}.$$

We can quantify which parameter value, and by how much it should be controlled to achieve specific control aims. For example, consider a case where the noise level of a protein needs to be reduced by 9%, while its mean level should remain the same. Here, the noise level (n) is defined by the variance divided by the squared mean value, i.e., squared coefficient of variation. Two control vectors for the noise and mean levels, $\mathbf{C}_{\mathbf{p}}^n$ and $\mathbf{C}_{\mathbf{p}}^m$, need to be computed based on a given mathematical model. System parameters need to be perturbed while satisfying

$$\frac{\delta n}{n} = \mathbf{C}_{\mathbf{p}}^n \cdot \frac{\delta\mathbf{p}}{\mathbf{p}} = -0.09 \quad \text{and} \quad \frac{\delta m}{m} = \mathbf{C}_{\mathbf{p}}^m \cdot \frac{\delta\mathbf{p}}{\mathbf{p}} = 0.$$

The perturbation vector $\delta\mathbf{p}/\mathbf{p}$ satisfying these two equations can be solved, but the solutions can be infinite. In that case, it is important to select the optimal control scheme (perturbation vector) among the possible solutions. For this, the control efficiency and strength were introduced [12]. Based on these two quantities, one can choose desired control schemes that are appropriate to systems of interest with the maximum control strength and/or efficiency.



C.

	k_{on}	k_{off}	α_m	γ_m	α_p	γ_p
Noise	-1.00	1.00	-1.00	0.05	-0.02	0.97
Mean	1.00	-1.00	1.00	-1.00	1.00	-1.00

Fig. 1. GFP expression system: (A) GFP is expressed under the *lac*-promoter (BioBrick part BBa_R0010) with a ribosome binding site (BBa_B0034). This expression cassette was placed in a low-medium copy number plasmid backbone (pGA3K3; origin of replication p15A). The 'T' symbol represents a terminator (double terminator used here to ensure transcription termination). (B) Two-promoter-state model. When the promoter is active, mRNA is transcribed with a rate constant α_m. From the transcript, GFP is translated with a rate constant α_p. mRNA and GFP degrade or are diluted with net rate constants γ_m and γ_p, respectively. (C) Control coefficients for noise and mean levels are listed. All control coefficients in the same row add up to zero (up to rounding error), satisfying summation theorems in SCA [13]. Parameters (unit: hr^{-1}): $k_{on} = 50$, $k_{off} = 51000$, $\alpha_m = 160$, $\gamma_m = 30$, $\alpha_p = 1400$, and $\gamma_p = 1$ (refer to the Material and Methods for the detailed description of the mathematical model and its parameters).

III. SCA FOR A SINGLE GENE EXPRESSION CASSETTE

We constructed plasmid expression systems that express green fluorescent protein (GFP) under *lac*-promoters in *E. coli* (Fig. 1A). The plasmid copy number in a single cell fluctuates in time because a set of plasmids are randomly partitioned during cell division and are synthesized in a stochastic fashion. Thus, the copy number of *lac*-promoters per cell fluctuates. For simplicity, we will assume that the plasmid copy number is tightly controlled, i.e., constant at the first level of approximation. The total number of *plac* will be the sum of the number of inactive and active *lac*-promoters (Fig. 1B), which will be set to a constant, N_p. We call this the two-state model. The plasmid backbone that we used is pGA3K3 with the replication origin, p15A ($N_p = 10 - 30$). Based on this two-state model, we computed control vectors for the mean and noise levels of GFP fluorescence as shown in Fig. 1C (refer to the Materials and Methods and [12] for the control vector computation).

The computed control coefficients show that noise can be controlled efficiently by varying k_{on}, k_{off}, α_m or γ_p; $C^n_{\mathrm{k}_{on}} = C^n_{\alpha_m} = -1.00$, $C^n_{\mathrm{k}_{off}} = 1.00$ and $C^n_{\gamma_p} = 0.97$, indicating that, for example, with an increase in α_m by 10%, n will reduce by 10% (this is a first-order approximation, because control coefficients are defined locally). Similarly, with an increase in γ_p by 10%, n will increase by 9.7%. For the mean level (m), any model parameter will efficiently change m, because the absolute values of all the control coefficients for m are equal to one.

IV. MEAN LEVEL CONTROL

From the computed control coefficients, the mean protein levels can be controlled without changing the noise level (with a minor change, $\sim$10 folds less than the change in the mean levels) by varying either α_p or γ_m. To confirm this theoretical prediction, we changed the translation efficiency α_p by using a library of both ribosome binding sites (RBSs) and spacer sequences as shown in Fig. 2. Among them, four different spacers – TACTAG, AAAAAA=$(A)_6$, $(A)_{10}$, and $(A)_{13}$ – that are placed between B0034 and the start codon showed distinct GFP expression levels when *plac* is fully active ([IPTG] = 1 mM). Here, the introduced spacer sequences are presumed to change ribosome binding affinity, in particular, translation initiation – the limiting step for a translation rate [23], [24]. We note that strong RBSs can recruit many ribosomes to mRNAs, causing an implication depending on the availability of ribosomes. This can apply an upper limit in the value of α_p.

Based on our flow cytometry data, the mean level was successfully varied by using different spacer sequences as shown in Fig. 3 and Fig. 4A. We compared three different cases: Points A, B, and C in Fig. 3, corresponding to [IPTG]=1 mM. As shown in Fig. 4A, the rescaled probability density functions (pdfs) were overlapped with a minor discrepancy. This scale invariance confirms that the noise levels of all the points are the same.

Scale invariance in the gamma distribution: Furthermore, the observed invariance implies a special property that we need to consider carefully. This invariance property is satisfied by the gamma distribution function as shown in the Materials and Method section when the burst size is rescaled together. This implies that the difference between the system parameters of Points A, B, and C is only the burst size. For these Points, different spacer sequences were used between B0034 and the start codon, while the *lac*-promoter was fully induced (saturated). Thus, the translation rate constant α_p is expected to be varied for these three Points and the burst size must be closely related to α_p, which is consistent with theoretical prediction based on our model (Eq. (3)). This result supports that the observed distribution functions are the gamma distributions (confer to [25], [26] about claims for other types of distribution functions). To confirm this, we fit the GFP pdfs to the gamma distribution functions as shown in Fig. 5. We confirmed that the pdfs follow the gamma distributions well.

V. NOISE LEVEL CONTROL

As discussed above, the noise level can be efficiently controlled by varying k_{on}, k_{off}, α_m and γ_p. However, when these parameters are changed, the mean level also changes with the same fold difference but in the opposite direction; for example, in Fig. 1C, $C^n_{\alpha_m}$ and $C^m_{\alpha_m}$ are -1.00 and 1.00, meaning that when α_m is increased by x%, the noise level decreases by x%, while the mean level increases by x%. Thus, to change the noise level without changing the mean level, we must vary at least two different parameters simultaneously.

Since the mean level can be controlled almost independently of the noise level by changing α_p, we will vary α_p along with one of the parameters in $\{k_{on}, k_{off}, \alpha_m\}$ to compensate for the change. The reason that we did not choose to vary γ_p is that this parameter is highly dependent on cell growth rate, rather than protein degradation in *E. coli*; GFP lifetime is much longer than the cell doubling time ~ 1 hr in M9 media.

Based on the SCA, an individual change in k_{on}, k_{off}, and α_m and any combination of the individual changes can vary the noise and mean levels while satisfying the same

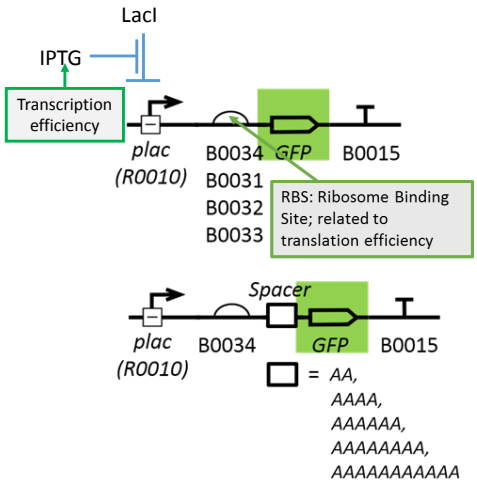


Fig. 2. Perturbations in the GFP expression systems: The GFP expression cassette is placed in the plasmid backbone pGA3K3 in *E.coli* MG1655Z1 that constitutively expresses LacI. IPTG concentrations were varied for a given complex of ribosome binding site and spacer. ~ 10 fold increase in the GFP noise level can be achieved without changing its mean level.

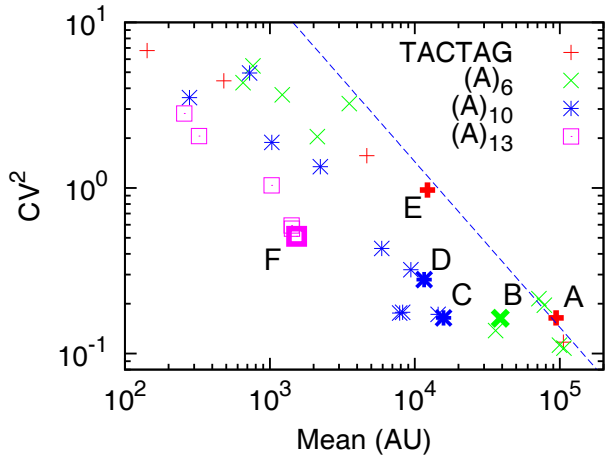


Fig. 3. Scaling relationship between noise and mean levels: Four different cases of spacer sequences between the ribosome binding site BBa_B0034 and the start codon are shown. The same symbol represents the same spacer with different [IPTG]. The noise levels (squared coefficient of variation) are inversely proportional to the mean level. For comparison, a line with a slope -1 is drawn. The contribution to the noise level by background fluorescence signals was removed via the noise level correction method (Materials and Methods). For the IPTG concentration information, we refer to the Supplementary Notes Fig. 2.

scaling law: $n = c/m$ with c a constant (not varied). We note that the ratio of control coefficients for n and m for a given parameter, e.g., k_{on}, has a graphical meaning: In Fig. 3, when [IPTG] is varied, the corresponding data point shifts (e.g., Point C $\rightarrow$ D) and the slope of the shift in the log-log plot corresponds to the ratio of the control coefficients: $\frac{C^n_{k_{on}}}{C^m_{k_{on}}} = \left.\frac{d\log n}{d\log m}\right|_{k_{on}}$. For k_{on}, k_{off}, and α_m, the ratios are

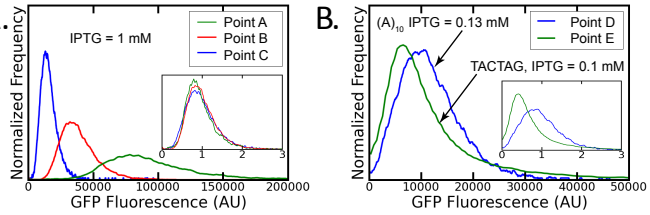


Fig. 4. Probability density functions (pdfs) of GFP fluorescence signals measured from a flow cytometer: (A) Orthogonal mean level control: Points A, B, and C in Fig. 3 correspond to [IPTG]=1 mM. In the inset plot, both the pdfs were re-scaled by the mean values of their respective GFP fluorescence signals, so that the transformed pdfs are centered around one. (B) Orthogonal noise level control: Points D and E. Both the [IPTG] and the spacer sequences were varied. [IPTG]=.13 mM for Point D and .1 mM for Point E. Autofluorescence was removed via the fluorescence histogram correction method (Materials and Methods).

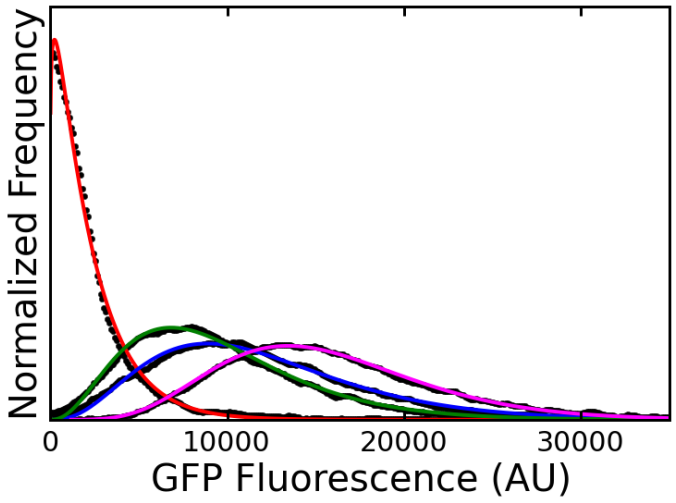


Fig. 5. True GFP signal distribution function for the $(A)_{10}$ cases with different IPTG concentrations: The fluorescence histogram correction was applied to remove autofluorescence effects. The true GFP signal distribution satisfies the Gamma distribution functions.

the same: $\frac{C^n_{k_{on}}}{C^m_{k_{on}}} = \frac{C^n_{k_{off}}}{C^m_{k_{off}}} = \frac{C^n_{\alpha_m}}{C^m_{\alpha_m}} = -1$. This implies that the directions of data point shifts in the log-log plot of n vs. m are identical for each individual perturbation of k_{on}, k_{off}, and α_m, and thus for any combination of these three individual perturbations. Therefore, the shift of data points with the slope of -1, observed when varying IPTG concentrations as shown in Fig. 3, cannot determine which parameters among k_{on}, k_{off}, and α_m were affected by IPTG concentration changes. We note that in [21] promoter perturbations in *E. coli* was claimed to affect k_{off} only.

As shown in Fig. 3, by using a library of RBS as well as different concentrations of IPTG, the noise level was controlled and ~ 10 fold change in the noise level was achieved without changing the mean level. What is the biological reason that noise can be increased in this way? In other words, what causes to increase the value of the Fano factor? In the scaling relationship, $n = c/m$, c is the Fano factor, which is expressed

for the case of *E. coli* (Materials and Methods):

$$c \simeq 1 + b(1 + b_m),$$

where

$$b = \frac{\alpha_p}{\gamma_m}$$

and

$$b_m = \frac{\alpha_m}{k_{on} + k_{off}} \frac{k_{off}}{k_{on} + k_{off}}.$$

b is the translational burst size, quatifying the number of proteins that are synthesized from a single mRNA during the mRNA lifetime ($1/\gamma_m$). b_m is the transcriptional burst size, quantifying the number of mRNA that are synthesized per plasmid during the time-scale ($1/(k_{on} + k_{off})$) of gene switching (refer to the Materials and Methods). The Fano factor depends on both transcriptional and translational bursts. In our case of lacI^q expression of LacI, we can neglect the transcriptional burst (Materials and Methods). Thus, the Fano factor becomes $c = 1 + b$, and n can be expressed as

$$n \simeq \frac{1+b}{m}. \tag{1}$$

The Fano factor can be increased by applying stronger translation efficiencies (from Point C to A) and remains the same by decreasing [IPTG] (from Point A to E), leading to the increase in the noise level without changing the mean level.

The translational bursts lead to longer-tail pdfs, more precisely, higher cutoff values in the pdfs (in the gamma distribution, there is an exponential factor $e^{-x/b}$ and b acts as a cutoff value): Figure 4B shows that a longer tail in the GFP pdf can be generated by using stronger translation efficiency (Point D $\rightarrow$ Point E).

Another interpretation for the observed longer tail in Fig. 4B is that the major source of fluctuations in the protein copy numbers is in mRNA copy numbers, which merely get amplified by the translation rate in a non-bursty way:

$$N_{pr}(t) \sim (\alpha_p/\gamma_p) N_{rna}(t).$$

The variance in protein expression levels becomes

$$\text{Variance}(N_{pr}) \sim (\alpha_p/\gamma_p)^2 \text{Variance}(N_{rna}),$$

resulting in that the noise level does not depend on α_p:

$$n = \frac{\text{Variance}(N_{pr})}{\text{Mean}(N_{pr})^2} \sim \frac{\text{Variance}(N_{rna})}{\text{Mean}(N_{rna})^2}.$$

Since the protein mean level m must be proportional to the translation rate constant α_p (i.e., $m = \beta\alpha_p$ with β a constant), we obtain again similar scaling relationship:

$$n \text{ is independent of } \alpha_p. \Leftrightarrow n \propto \frac{\alpha_p}{\beta\alpha_p} = \frac{\alpha_p}{m}.$$

The Fano factor again increases with α_p. Therefore, based on the scaling relationship alone, it is difficult to differentiate whether the translation is bursty or not.

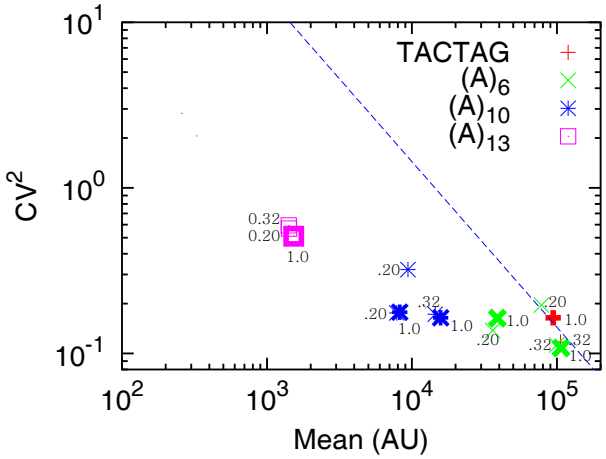


Fig. 6. GFP mean and noise levels for the cases that *lac*-promoters are fully induced ([IPTG] = 0.20, 0.32, 1.0 mM). Two biological replicates were used for $(A)_6$ and $(A)_{10}$.

VI. Translation is bursty.

We claim that translation processes are bursty. The data points in bold in Fig. 6 correspond to different RBS strength but the same level of [IPTG] equal to 1 mM, where the *lac*-promoter becomes constitutively active. The noise values for TACTAG, $(A)_6$, and $(A)_{10}$ were similar, but the noise level for $(A)_{13}$ was higher than the rest. This difference cannot be explained in the non-bursty translation scenario, because n should be independent of α_p, i.e., RBS strength. In the bursty translation scenario, the noise level can be dependent on the value of α_p, especially when the value of α_p is similar to that of γ_m. Since m is proportional to α_p ($m = \beta\alpha_p$), Eq. (1) becomes

$$n = \frac{1 + \alpha_p/\gamma_m}{\beta\alpha_p} = \frac{1}{\beta\alpha_p} + \frac{1}{\beta\gamma_m} \tag{2}$$

For a strong RBS such as the TACTAG case, α_p can be roughly around 1400 hr^{-1} (Materials and Methods). In this case, $\alpha_p/\gamma_m \sim 1400/30 \simeq 47$, i.e. much larger than 1. Thus, Eq. (2) becomes $n \simeq \frac{\alpha_p/\gamma_m}{\beta\alpha_p} = \frac{1}{\gamma_m\beta}$, resuling in that n is independent of α_p, which is what we observed from Point A to C. For the case of $(A)_{13}$ (Point F), the RBS strength is reduced by ~ 60 times (by comparing the mean levels of Point A and F) and $\alpha_p/\gamma_m \simeq 47/60 \simeq 0.8$. Thus, n becomes dependent on α_p (Eq. (2)). As α_p decreases, n increases. This is consistent with our observation.

VII. Scaling relationship between the noise and mean levels

Figure 7 shows that the scaling relationship $n = c/m$ can be observed after autofluorescence was systematically removed, even for the small mean value region, where the autofluorescence interferes with the true GFP signals. We took into account the stochasticity in autofluorescence and assumed that the fluctuations in the autofluorescence signals are statistically independent of the true GFP signals. Under this

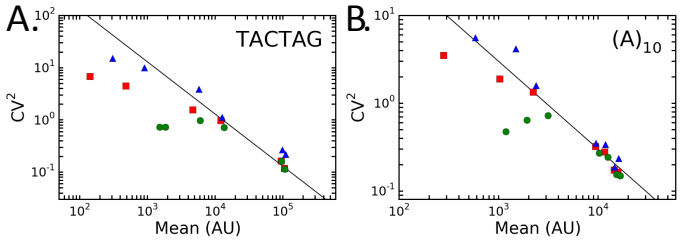


Fig. 7. Effect of autofluorescence on scaling relationship between n and m: Autofluorescence was removed in two different ways via (1) the noise level correction method (red squares) and (2) the fluorescence histogram correction method (blue triangles) (refer to the Materials and Methods). To show the trend clearly, the data corresponding to the same biological replicate are only shown.

assumption, we compensated for the autofluorescence effect in two different ways: (1) direct noise level correction (red squares in Fig. 7) and (2) fluorescence histogram correction (blue triangles in Fig. 7; an example is presented in Fig. 8) (Materials and Methods). This implies that it is important to take into account the stochasticity of the autofluorescence signals when characterizing the systems by using the pdfs of fluorescence signals.

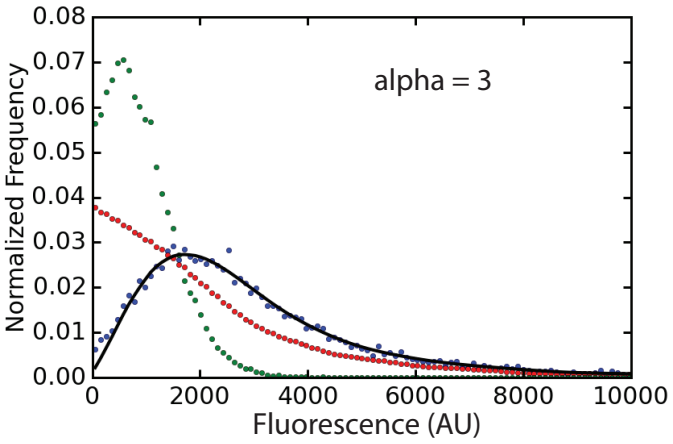


Fig. 8. Autofluorescence compensation: The green dots correspond to autofluorescence (IPTG=0 case), the blue dots to the measured GFP signals, and the red dots to the optimized solution S, i.e., the pdf of the true GFP signals. The black line is to verify the optimized solution S can generate the measured GFP pdfs via convolution (refer to the Materials and Methods).

VIII. Conclusions

In summary, we perturbed the strength of ribosome binding sites and investigated scaling relationship between the mean and noise levels of the expressed proteins. We confirmed that translational bursts are one of the important sources of noise at the protein level by using our numerical sensitivity analysis method, SCA, and the analytical structure of noise propagation. To investigate the scaling relationship further in detail, we compensated the effect of autofluorescence by taking into account stochasticity in the autofluorescence and recovered the expected scaling relationship even when autofluorescence becomes moderately strong. This shows that the autofluorescence can be systematically removed and its compensation can be applied to characterize cellular systems.

Materials and Methods

A. GFP expression circuits and strains

All genetic components used in this manuscript are BioBrick parts, from which genetic circuits were constructed by using the Gibson assembly method [30]. The constructed circuits were integrated into a low-to-medium copy number plasmid pGA3K3 with a Kanamycin resistance gene and *Escherichia coli* MG1655 Z1 was transformed with the plasmids. The strain (lacI^q) constitutively overexpresses LacI from its chromosome.

B. Cell Growth and Flow Cytometry Measurements

E. coli strains were grown to OD600$\sim$0.2 in 2 mL Luria-Bertani (LB) media (Becton Dickinson) with kanamycin 50 μg/mL at 37°C and 300rpm in a shaker. The cultures were diluted 1:200 into 200 μL prewarmed fresh M9 media (Teknova 2M1990) in 96-well plates (Costar 3904) with kanamycin 50 μg/mL. 12 different IPTG concentrations (0 mM, 0.02$\sim$1 mM) were used for each well (refer to the Supplementary Notes for more detailed information on IPTG concentrations) and grown to OD600=0.3-0.4 in a shaker (37°C, 300 rpm). For the flow cytometry measurements, the grown cultures were diluted 1:4 in 1xPBS. A Sony Biotechnology ec800 flow cytometer was used with a 525 nm filter and a 488 nm excitation laser for GFP fluorescence. 100,000 events were collected for each sample and gated by using a 2-D normal distribution (Bioconductor flowCore norm2filter function with scale.factor=1) [31] within the R software environment as well as by using python package FlowCytometryTools (http://gorelab.bitbucket.org/flowcytometrytools/#). To prevent well-well contamination we executed a Medium Flush cycle after each sample well. When computing the mean and noise levels of GFP signals, background fluorescence was removed by using the mean and noise levels of GFP signals, or the signal histogram for the case without IPTG for each different gene circuit.

C. Mathematical Model

A two-state model [27], [16], [28] is introduced to describe active and inactive states of a promoter along with transcription and translation processes:

$$N_i \underset{k_{off}N_a}{\overset{k_{on}N_i}{\rightleftharpoons}} N_a$$
$$\xrightarrow{\alpha_m N_a} N_{rna} \xrightarrow{\gamma_m N_{rna}}$$
$$\xrightarrow{\alpha_p N_{rna}} N_{pr} \xrightarrow{\gamma_p N_{pr}},$$

where N_i denotes the number of inactive promoters, N_a that of active promoters, N_{rna} the RNA copy number, and N_{pr} the protein copy number. All the above reaction events are generated stochastically. The noise level, n, of N_{pr} can be

analytically solved (refer to the Supplementary Notes of [27] for all the detailed derivation):

$$n = \mathrm{CV}^2 = \frac{c}{m},$$

with m denoting the mean value of N_{pr} and the Fano factor c is expressed as

$$c \simeq 1 + \alpha_p \tau_m \left(1 + \alpha_m \frac{k_{off}}{k_{on}+k_{off}} \frac{\tau_p}{1+(k_{on}+k_{off})\tau_p}\right).$$

Here, we assumed that the protein lifetime τ_p, defined by $1/\gamma_p$, is much larger than the mRNA lifetime τ_m. We note that control coefficients for n can be computed from this equation analytically. We assume that the inactive and active promoter states switch back and forth many times during the protein lifetime ($k_{on} + k_{off} \gg \gamma_p$). We believe this is the case of our experiments and refer to the parameter value estimation described below. In this case, the above equation can be further approximated:

$$c \simeq 1 + \alpha_p \tau_m \left(1 + \frac{\alpha_m}{k_{on}+k_{off}} \frac{k_{off}}{k_{on}+k_{off}}\right). \quad (3)$$

Here, $1/(k_{on} + k_{off})$ is the time scale of the promoter state switching and $k_{off}/(k_{on} + k_{off})$ is a suppression weight because the promoter state follows the binomial distribution (due to the fact that the total promoter number is constant) instead of the Poisson distribution. Thus, the second term in the parenthesis can be considered as a transcriptional burst size b_m. In our case, $\alpha_m/(k_{on} + k_{off}) \sim 160/(50 + 51000) \simeq 0.003$. Thus, c can be further simplified:

$$c \simeq 1 + \alpha_p \tau_m,$$

implying that the change in the IPTG concentration has no effect on the Fano factor, c, thus moving along the line of slope -1 as shown in Fig. 3.

D. Model parameter estimation

Transcription rate constant, $\alpha_m = 160$ hr^{-1}: The *lac*-promoter strength, when fully induced with IPTG, was shown ∼1.5 time stronger than J23101 by directly measuring the transcript levels with our malachite-green aptamer probes (refer to Figure 6.3 of [22]). For J23101, α_m was estimated at 0.03 sec^{-1}=110 hr^{-1} [32]. Thus, α_m for our *lac*-promoter can be estimated at 160 hr^{-1}.

We used the translation rate constant, α_p = 1400 hr^{-1} (Supplementary Notes in [33]), the dilution rate, γ_p = 1 hr^{-1}, and the degradation rate constant of mRNA, γ_m = 30 hr^{-1}.

Gene inactivation, $N_a \xrightarrow{k_{off} N_a} N_i$: The number of the inactive promoters is denoted by N_i and that of the active promoters, N_a. The sum of N_i and N_a is equal to the copy number of the plasmids, N_p (considering that one *lac*-promoter is included per plasmid). Here, we used $N_p \sim 10$ (http://parts.igem.org/Part:pSB3K3; pGA3K3 is a variant of pSB3K3). k_{off} is related to the search time for LacI to find *lac*-promoter. When there exist one LacI molecule and one *lac*-promoter within an *E. coli* cell, the search time is less than 6 min = 0.1 hr [21]. In the case of N_a unoccupied *lac*-promoters and N_{lacI} copies of *LacI*, the search time becomes $0.1/N_{lacI}N_a$, which is equal to the inverse of the inactivation rate ($1/k_{off}N_a$). Therefore, k_{off} is estimated to be $10N_{lacI}$ hr^{-1}. N_{lacI} can be roughly estimated from the fact that the strength of the lacI^q is similar to the promoter J23101 (α_m of J23101 is 110 hr^{-1}) [32]. Thus, the genomic expression level of LacI, N_{lacI}, becomes $\alpha_p[\mathrm{mRNA}]_{lacI}/\gamma_p = \alpha_p\alpha_m/\gamma_p\gamma_m = 1400 \cdot 110/1 \cdot 30 \simeq 5100$. Thus, k_{off} can be roughly estimated as 5.1×10^4 hr^{-1}.

Gene activation, $N_i \xrightarrow{k_{on} N_i} N_a$: The activation rate constant k_{on} is related to how fast the genomically-expressed LacI detached from its specific promoter *plac* (BBa_R0010). Considering that the dissociation constant is in the range of $0.1 - 1$ pM = $10^{-4} - 10^{-3}$ (copy number unit; here we used 1 nM corresponds to roughly 1 molecule number in the volume of *E. coli*) [34], [35], k_{on} can be in the range of $k_{off} \times (10^{-4} - 10^{-3}) = 5.1 - 51$ hr^{-1}.

E. Noise level correction

The mean level was corrected with a simple subtraction. The noise level was corrected by using the property that the observed variance (Variance$_o$) is the sum of the GFP variance (Variance$_g$) and the background signal variance (Variance$_b$) under the assumption that the GFP signals are statistically independent of the background signals. More precisely, the noise level of GFP signals, defined by the square coefficient of variation, can be obtained by

$$\mathrm{CV}^2 = \frac{\mathrm{Variance}_o - \mathrm{Variance}_b}{(\mathrm{Mean}_o - \mathrm{Mean}_b)^2}.$$

where the subscripts o and b denote observed and background signals, respectively.

F. Fluorescence histogram correction

The effect of autofluorescence was removed from the GFP signal histogram, more precisely probability mass function (pmf), by assuming that the autofluorescence is statistically independent of the true GFP signals [36]. Under this assumption, the pmf of the measured GFP signals, $T(\nu)$, is related to both the autofluorescence pmf $C(\nu)$ and the true GFP signal pmf $S(\nu)$ via convolution:

$$T(\nu) = \int_0^\nu C(\nu - \nu')S(\nu')d\nu'.$$

$S(\nu)$ is obtained by minimizing the fitness function:

$$M^\alpha(C,T,S) = \int_0^a \left[\int_0^\nu C(\nu-\nu')S(\nu')d\nu' - T(\nu)\right]^2 d\nu + \alpha \parallel S(\nu) \parallel .$$

Here, a is the value of ν beyond which $T(\nu)$ is essentially zero, and in our study, we used the entire range of pmf. $\parallel S(\nu) \parallel$ is a regularization term, defined as

$$\parallel S(\nu) \parallel = \int_0^a [g_0(S(\nu))^2 + g_1(dS(\nu)/d\nu)^2]d\nu$$

Constants	Values
a	1000
α	0.5
g_0	0.001
g_1	1.0
dS	10^{-6}

TABLE I. CONSTANTS USED IN THE AUTOFLUORESCENCE COMPENSATION ALGORITHM

where g_0 and g_1 are positive regularization constants. The optimized solution of $S(\nu)$ is obtained by following the procedure described below.

1) Remove background noise that is equipment-specific. Fluorescence signals of strength 0 and 1 (Sony Biotechnology ec800) were considered as background noise and removed. Then, $T(\nu)$ (for the induction case of [IPTG] > 0) and $C(\nu)$ (for the case of [IPTG]=0) were computed from the fluorescence signals using 1000 equal-width bins to obtain individual bin-sizes. Here, the bin-size of T is larger than that of C.
2) To compute the convolution, we will set the bin-size of C equal to that of T. Compute C again from the raw data using the bin-size of T, and append an array of zero at the end of C to make the total bin number equal to 1000.
3) Set the initial values of S equal to T.
4) Generate two different random numbers ν_1 and ν_2 in the range of [0, 999]. $S(\nu_1)$ and $S(\nu_2)$ were added and subtracted, respectively, by a constant $dS = 10^{-6}$: $S(\nu_1) \rightarrow S(\nu_1) + dS$ and $S(\nu_2) \rightarrow S(\nu_2) - dS$. When $S(\nu_2) - dS$ is less than zero, set $S(\nu_1)$ equal to $S(\nu_1)$ + $S(\nu_2)$ and then $S(\nu_2)$ equal to 0. In this way, the new S is automatically re-normalized and guaranteed to be non-negative.
5) Compute M^α. If M^α decreases, we accept the change and, otherwise, reject it and revert $S(\nu)$ to the old S values before the update.
6) Repeat the steps 4 and 5 until M^α converges and compare $\int_0^\nu C(\nu - \nu')S_{op}(\nu')d\nu'$ and $T(\nu)$, where S_{op} is the obtained optimized solution of S. If $S_{op}(\nu)$ shows oscillation, reduce the value of α while rebalancing g_0 and g_1 and go back to the step 4. If S_{op} is noisy, increase the value of α while rebalancing g_0 and g_1 and go back to the step 4.

The constants used for the optimization are listed in Table I. The analysis was performed with Python 2.7.9 with Numpy 1.9.2, Scipy 0.15.1, and Spyder 2.3.4. Our python code is provided in the Supplementary Notes.

G. Nonlinear Regression

The gamma distribution function was used to fit our flow cytometry data. Protein copy number N_{pr} can be converted to fluorescence signal intensity x: $N_{pr} = c_s x$ with c_s a scaling constant. The gamma distribution function can be rescaled:

$$\begin{aligned} p(N_{pr}; a, b) &= p(c_s x; a, b) = \frac{(c_s x)^{a-1} e^{-c_s x/b}}{\Gamma(a)\, b^a} \\ &= c_s^{-1} \frac{x^{a-1} e^{-x/(b/c_s)}}{\Gamma(a)\, (b/c_s)^a} \\ &= c_s^{-1} p(x; a, b/c_s). \end{aligned}$$

Here, Γ is a gamma function, and

$$a \equiv \frac{\alpha_m}{\gamma_p} N_a$$

is the number of mRNA produced per cell doubling time, called burst frequency with N_a the number of active promoters, and

$$b \equiv \frac{\alpha_p}{\gamma_m}$$

is the number of proteins produced during the mRNA lifetime, called burst size. Therefore, the fluorescence intensity should also follow the gamma distribution if its corresponding copy number follows the gamma distribution, with the burst size rescaled with c_s. Nonlinear regression was carried by using the Scipy curve_fit function (http://www.scipy.org/), which employs the Levenberg-Marquardt algorithm for the least squares fitting to estimate a and b.

ACKNOWLEDGMENT

This work was supported by the National Science Foundations (NSF MCB 1158573).

REFERENCES

[1] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators." *Nature*, vol. 403, no. 6767, pp. 335–338, Jan. 2000.

[2] A. Sanchez, S. Choubey, and J. Kondev, "Regulation of noise in gene expression." *Annu. Rev. Biophys.*, vol. 42, pp. 469–91, Jan. 2013.

[3] G. Balázsi, A. van Oudenaarden, and J. J. Collins, "Cellular decision making and biological noise: from microbes to mammals." *Cell*, vol. 144, no. 6, pp. 910–925, Mar. 2011.

[4] S.-L. To and N. Maheshri, "Noise Can Induce Bimodality in," *Science*, vol. 327, no. February, pp. 1142–1146, 2010.

[5] K. H. Kim, H. Qian, and H. M. Sauro, "Nonlinear biochemical signal processing via noise propagation," *J. Chem. Phys.*, vol. 139, p. 144108, 2013.

[6] J. Paulsson, O. G. Berg, and M. Ehrenberg, "Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation." *Proc. Natl. Acad. Sci. U. S. A.*, vol. 97, no. 13, pp. 7148–7153, Jun. 2000.

[7] W. J. Blake, G. Balàzsi, M. A. Kohanski, F. J. Isaacs, K. F. Murphy, Y. Kuang, C. R. Cantor, D. R. Walt, J. J. Collins, "Phenotypic Consequences of Promoter-Mediated Transcriptional Noise.'" *Mol. Cell*, vol. 24, pp. 853–865, Dec. 2006.

[8] K. F. Murphy, G Balázsi, J. J. Collins, "Combinatorial promoter design for engineering noisy gene expression." *Proc. Natl. Acad. Sci. U. S. A.*, vol 104, pp. 12726–12731, Jul. 2007.

[9] K. F. Murphy, R. M. Adams, X., Wang, G. Balázsi, J. J. Collins, "Tuning and controlling gene expression noise in synthetic gene networks." *Nucleic Acids Res.*, vol 38, pp. 2712-2726, May 2010.

[10] H. Maamar, A. Raj, D. Dubnau, "Noise in gene expression determines cell fate in *Bacillus subtilis*." *Science*, vol 317, pp. 526–529, Jul. 2007.

[11] M. R. Birtwistle, A. von Kriegsheim, M. Dobrzyński, B. N. Kholodenko, W. Kolch, "Mammalian protein expression noise: scaling principles and the implications for knockdown experiments." *Mol. BioSyst.*, vol 8, pp. 3068–3076, 2012.

[12] K. H. Kim and H. M. Sauro, "Adjusting Phenotypes by Noise Control," *PLoS Comput. Biol.*, vol. 8, no. 1, p. e1002344, Jan. 2012.

[13] ——, "Sensitivity summation theorems for stochastic biochemical reaction systems." *Math. Biosci.*, vol. 226, no. 2, pp. 109–119, Aug. 2010.

[14] D. A. Fell, "Metabolic control analysis: a survey of its theoretical and experimental development." *Biochem. J.*, vol. 286, pp. 313–330, 1992.

[15] H. Kacser and J. A. Burns, "The control of flux." *Biochem. Soc. Trans.*, vol. 23, pp. 341—-366, 1995.

[16] Y. Taniguchi, P. J. Choi, G.-W. Li, H. Chen, M. Babu, J. Hearn, A. Emili, and X. S. Xie, "Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells." *Science*, vol. 329, no. 5991, pp. 533–538, Jul. 2010.

[17] E. M. Ozbudak, M. Thattai, I. Kurtser, A. D. Grossman, and A. van Oudenaarden, "Regulation of noise in the expression of a single gene." *Nature Genetics*, vol. 31, pp. 69–73, 2002.

[18] A. Sanchez and I. Golding, "Genetic determinants and cellular constraints in noisy gene expression." *Science*, vol. 342, no. 6163, pp. 1188–1193, Dec. 2013.

[19] J. Elf, G.-W. Li, X. S. Xie, "Probing transcription factor dynamics at the single-molecule level in a living cell." *Science*, vol. 316, no. 5828, pp. 1191–1194, May 2007.

[20] J. R. Kelly, A. J. Rubin, J. H. Davis, C. M. Ajo-Franklin, J. Cumbers, M. J. Czar, K. de Mora, A. L. Glieberman, D. D. Monie, D. Endy, "Measuring the activity of BioBrick promoters using an in vivo reference standard." *J. Biol. Eng.*, vol 3, p. 4, 2009.

[21] L.-H. So, A. Ghosh, C. Zong, L. Sepúlveda, R. Segev, I. Golding, "General properties of transcriptional time series in Escherichia coli." *Nature Genetics*, vol 43, pp. 554–560, 2011.

[22] W. Copeland, "Biological probes to measure transcription dynamics in E. coli. ", Diss. University of Washington, 2015. Print.

[23] H. M. Salis, E. a. Mirsky, and C. a. Voigt, "Automated design of synthetic ribosome binding sites to control protein expression." *Nat. Biotechnol.*, vol. 27, no. 10, pp. 946–50, Oct. 2009.

[24] R. G. Egbert and E. Klavins, "Fine-tuning gene networks using simple sequence repeats." *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 42, pp. 16 817–22, Oct. 2012.

[25] H. Salman, N. Brenner, C.-k. Tung, N. Elyahu, E. Stolovicki, L. Moore, A. Libchaber, and E. Braun, "Universal Protein Fluctuations in Populations of Microorganisms," *Phys. Rev. Lett.*, vol. 108, no. 23, p. 238105, Jun. 2012.

[26] S. Ghosh, K. Sureka, B. Ghosh, I. Bose, J. Basu, and M. Kundu, "Phenotypic heterogeneity in mycobacterial stringent response." *BMC Syst. Biol.*, vol. 5, no. 1, p. 18, Jan. 2011.

[27] A. Bar-Even, J. Paulsson, N. Maheshri, M. Carmi, E. O'Shea, Y. Pilpel, and N. Barkai, "Noise in protein expression scales with natural protein abundance." *Nat. Genet.*, vol. 38, no. 6, pp. 636–643, Jun. 2006.

[28] V. Shahrezaei and P. S. Swain, "Analytical distributions for stochastic gene expression." *Proc Natl Acad Sci U S A*, vol. 105, no. 45, pp. 17 256–17 261, Nov. 2008. [Online]. Available: http://dx.doi.org/10.1073/pnas.0803850105

[29] O. K. Silander, N. Nikolic, A. Zaslaver, A. Bren, I. Kikoin, U. Alon, and M. Ackermann, "A genome-wide analysis of promoter-mediated phenotypic noise in Escherichia coli." *PLoS Genet.*, vol. 8, no. 1, p. e1002443, Jan. 2012.

[30] D. G. Gibson, L. Young, R.-Y. Chuang, J. C. Venter, C. a. Hutchison, and H. O. Smith, "Enzymatic assembly of DNA molecules up to several hundred kilobases." *Nat. Methods*, vol. 6, no. 5, pp. 343–5, May 2009.

[31] F. Hahne, N. LeMeur, R. R. Brinkman, B. Ellis, P. Haaland, D. Sarkar, J. Spidlen, E. Strain, and R. Gentleman, "flowCore: a Bioconductor package for high throughput flow cytometry." *BMC Bioinformatics*, vol. 10, p. 106, Jan. 2009.

[32] S. Zucca, L. Pasotti, G. Mazzini, M. G. C. De Angelis, P. Magni, "Characterization of an inducible promoter in different DNA copy number conditions." *BMC Bioinformatics*, vol. 13, p. S11, 2012.

[33] B. Canton, A. Labno, D. Endy, "Refinement and standardization of synthetic biological parts and devices." *Nature Biotechnology'*, vol. 26.7, pp. 787–793, 2008.

[34] P. Wong, S. Gladney, J. D. Keasling, "Mathematical model of the lac operon: Inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose." *Biotechnol. Progr.* vol. 13, pp. 132-143, 1997.

[35] Y. Setty, A. E. Mayo, M. G. Surette, U. Alon, "Detailed map of a cis-regulatory input function." *Proc. Natl. Acad. Sci. U. S. A.* vol. 100, pp. 7702-7707, 2003.

[36] Corsetti, J. P., Sotirchos, S. V., Cox, C., Cowles, J. W., Leary, J. F., Blumburg, N. (1988). Correction of cellular autofluorescence in flow cytometry by mathematical modeling of cellular fluorescence. Cytometry, 9(6), 539-547.

Supplementary Notes

1 DNA sequences of promoter-RBS-insertion-start codon regions

Table 1 shows the DNA sequences that were used for the region of ribosome binding sites along with different kinds of spacer sequences:

Name	Sequence (BBa_R0011_end-spacer-BBa_B0034-spacer-BBa_E0040_begin)	Backbone
pKK16B34	TTTCACACATACTAGAGAAAGAGGAGAAATACTAGATGCGTAAA	pGA3K3
pKK16A6	TTTCACACATACTAGAGAAAGAGGAGAAAAAAAAAATGCGTAAA	pGA3K3
pKK16A10	TTTCACACATACTAGAGAAAGAGGAGAAAAAAAAAAAAATGCGTAAA	pGA3K3
pKK16A13	TTTCACACATACTAGAGAAAGAGGAGAAAAAAAAAAAAAAATGCGTAAA	pGA3K3

Table 1: RBS region DNA sequences

2 Autofluorescence compensation in fluorescence histograms

Depending on the value of the regularization constant α, the optimized solution $S(\nu)$ can show oscillation. In this section, we provide its sample pictures. There is a trade-off between the strength of noise in S and the fitting accuracy (comparison between the black and the blue dots in Fig. 2).

The python code used for the compensation is provided below.

```
import os
import matplotlib.pyplot as plt
import scipy
import scipy.stats
import numpy as np
import FlowCytometryTools as fct

binno = 1000 # Number of bins
a = binno
g0 = .001
g1 = 1.
alphalist = ["list of alpha values"]
ds = 1e-6
repeat = 200000 #N

data = []
listfiles = []
histdata = []
binedgesdata = []
binsize = []

dirnames = ["Directories to the datasets"]

for i in range(len(dirnames)):
    fullpath = []
    for j in range(len(os.listdir(dirnames[i]))):
        files = os.listdir(dirnames[i])
        fullpath.append(os.path.join(dirnames[i],files[j]))
    listfiles.append(fullpath)

#%% Gating
for i in range(len(listfiles)):
    for j in range(len(listfiles[i])):
        filename = listfiles[i][j]
```

```
        print listfiles[i][j]

        sample = fct.FCMeasurement(ID = "KK16A10D7", datafile=listfiles[i][j])

        gate1range = []#List of tuples for vertical interval gates
        gate2range = []#List of tuples for horizontal interval gates

        FSgate =  fct.IntervalGate(gate1range[i][j], "FS-Lin", region="in")
        SSgate = fct.IntervalGate(gate2range[i][j], "SS-Lin", region="in")

        compgate = fct.core.gates.CompositeGate(FSgate, "and", SSgate)
        gated_out = sample.gate(compgate)

        fllin = gated_out.data["FL1-Lin"].values
        fllin = np.array(fllin)

# Raw data preprocessing
        fllin = np.delete(fllin, np.where(fllin == 0))
        fllin = np.delete(fllin, np.where(fllin == 1))
        data.append(fllin)

# Initial Histogram Calculation
        hist, bin_edges = np.histogram(fllin, bins=binno, density=True)
        binsize.append(np.diff(bin_edges)[0])

#%% Setting bin numbers equal to 1000
binnumbers = []
maxbin = np.max(binsize)
for i in range(len(listfiles)):
    for j in range(len(listfiles[i])):
        binnumbers.append(int(round(binsize[j]/maxbin*binno)))
        histf, bin_edgesf = np.histogram(data[j], bins=int(round(binsize[j]/maxbin*
            binno)), density=True)
        numzeros = binno - binnumbers[j]

        print "binsize: " + str(np.diff(bin_edgesf)[0])

        if numzeros != 0.:
            zeros = np.zeros(int(numzeros))
            histf = np.append(histf, zeros)
            bintemp = np.linspace(bin_edgesf[-1] + np.diff(bin_edgesf)[0],
                                  bin_edgesf[-1] + ((numzeros) * np.diff(bin_edgesf)[0])
                                     , numzeros)
            bin_edgesf = np.append(bin_edgesf, bintemp)

        histdata.append(histf)
        binedgesdata.append(bin_edgesf)

#%% Iterations over M
M = []
histdata_f = np.copy(histdata)

def CSint(gfp, auto, raw, a, maxbin): # Integration of CS - T
    tempint2 = np.multiply(maxbin, np.convolve(auto, gfp)[:a])
    return np.sum(np.square(np.subtract(tempint2, raw)))

def Sint(gfp, g0, g1): #Regularization
    return np.multiply(g0, np.square(gfp[:-1])) + np.multiply(g1, np.square(np.diff(
        gfp)))
```

```
def m_alpha(gfp, auto, raw, a, g0, g1, alpha, maxbin):
    return CSint(gfp, auto, raw, a, maxbin) + np.multiply(alpha, scipy.integrate.simps
       (Sint(gfp, g0, g1)[:a]))

for i in range(len(listfiles[0])):
    alpha = alphalist[i]
    maxbin = np.diff(binedgesdata[i])[0]
    if i == 0: # Pass autofluoresence
        M.append([0])
    else:
        M.append([m_alpha(histdata_f[i], histdata[0], histdata[i], a, g0, g1, alpha,
           maxbin)]) #Initial M calculation
        for j in range(repeat):
            ind = int(np.random.uniform(0, binnumbers[i])) #Picking Random number
            ind2 = int(np.random.uniform(0, binnumbers[i]))
            if ind >= a or ind2 >= a:
                pass
            elif ind >= binnumbers[i] or ind2 >= binnumbers[i]:
                pass
            else:
                histdata_temp = np.copy(histdata_f[i])
                #Addtion subtraction
                if histdata_f[i][ind] - ds < 0:
                    histdata_f[i][ind2] = histdata_f[i][ind2] + histdata_f[i][ind]
                    histdata_f[i][ind] = 0.
                else:
                    histdata_f[i][ind] = histdata_f[i][ind] - ds
                    histdata_f[i][ind2] = histdata_f[i][ind2] + ds
                integral_t = m_alpha(histdata_f[i], histdata[0], histdata[i], a, g0,
                   g1, alpha, maxbin) #M recalculation
                #Accept change
                if integral_t < M[i][-1]:
                    M[i].append(integral_t)
                #Reject change
                else:
                    histdata_f[i] = np.copy(histdata_temp)
                    M[i].append(M[i][-1])

#%% Calculate Convolution and Plot against Raw Data
well=7

multfactor = np.diff(binedgesdata[well])[0]
conv1 = np.multiply(multfactor, np.convolve(histdata[0], histdata_f[well])) #
   convolution term

binedgesdatax = binedgesdata[well][:-1] + np.diff(binedgesdata[well])/2

fig = plt.figure(figsize=(12,8))
ax = fig.add_subplot(1,1,1)
plt.plot(binedgesdatax, multfactor*histdata[well], "bo")
plt.plot(binedgesdatax, multfactor*histdata_f[well], "r", lw=4)
plt.plot(binedgesdatax, multfactor*histdata[0], "go")
plt.plot(binedgesdatax, multfactor*conv1[:binno], "k", lw=4)
ax.tick_params('both', length=10, width=3, which='major')
ax.tick_params('both', length=10, width=3, which='minor')
ax.spines['top'].set_linewidth(3)
ax.spines['right'].set_linewidth(3)
ax.spines['left'].set_linewidth(3)
ax.spines['bottom'].set_linewidth(3)
ax.tick_params(axis='x', pad=10)
```

```
ax.tick_params(axis='y', pad=10)
plt.axis([0, 15000, 0, .005])
ax.set_xlabel("Index")
ax.set_ylabel("Normalized Frequency")
plt.xticks(fontsize = 30)
plt.yticks(fontsize = 30)
plt.show()

#%% Plot M Values

fig = plt.figure(figsize=(12,8))
ax = fig.add_subplot(1,1,1)
plt.plot(M[well], lw=4)
plt.rc('font', size=30)
ax.tick_params('both', length=10, width=3, which='major')
ax.tick_params('both', length=10, width=3, which='minor')
ax.spines['top'].set_linewidth(3)
ax.spines['right'].set_linewidth(3)
ax.spines['left'].set_linewidth(3)
ax.spines['bottom'].set_linewidth(3)
ax.tick_params(axis='x', pad=10)
ax.tick_params(axis='y', pad=10)
ax.set_xlabel("Iterations")
ax.set_ylabel(r'$\mathregular{M^{\alpha}}$')
plt.xticks(fontsize = 30)
plt.yticks(fontsize = 30)
plt.show()
```

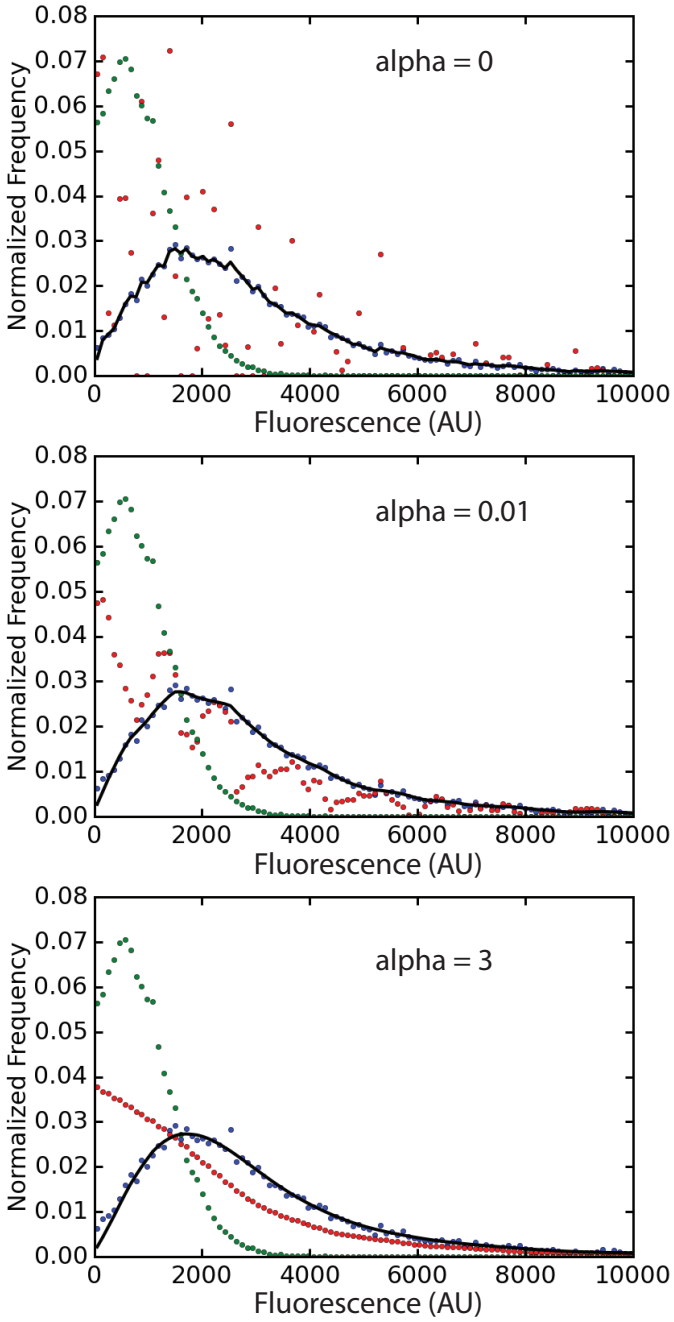


Figure 1: Autofluorescence compensation: The green dots correspond to autofluorescence (IPTG=0 case), the blue dots to the measured GFP signals, and the red dots to the optimized solution S, i.e., the true GFP signal probability mass function (normalized frequency). The black line is to verify the optimized solution S can generate T via convolution (refer to the Materials and Methods in the main manuscript).

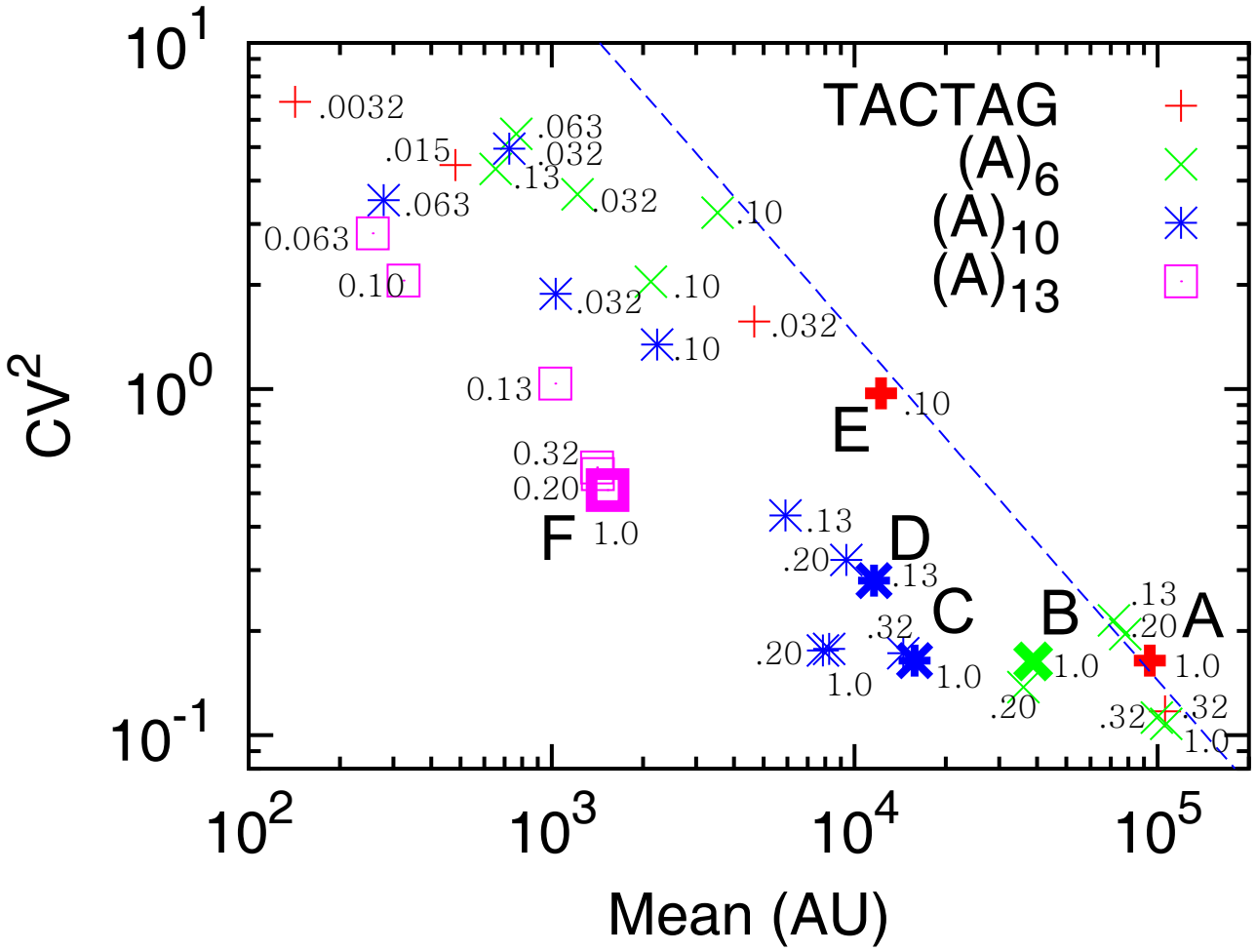


Figure 2: This figure is exactly the same as Figure 3 in the main manuscript. IPTG concentrations are included in the unit of mM. For $(A)_6$ and $(A)_10$, two biological replicates were used.

References

[1] Corsetti, J. P., Sotirchos, S. V., Cox, C., Cowles, J. W., Leary, J. F., Blumburg, N. (1988). Correction of cellular autofluorescence in flow cytometry by mathematical modeling of cellular fluorescence. Cytometry, 9(6), 539-547.