

Research



Cite this article: Yeung E, Kim J, Yuan Y, Gonçalves J, Murray RM. 2021 Data-driven network models for genetic circuits from time-series data with incomplete measurements.

J. R. Soc. Interface **18**: 20210413.

<https://doi.org/10.1098/rsif.2021.0413>

Received: 18 May 2021

Accepted: 12 August 2021

Subject Category:

Life Sciences—Engineering interface

Subject Areas:

synthetic biology, systems biology, computational biology

Keywords:

genetic networks, network reconstruction, genetic circuits, data-driven modelling, dynamical structure functions, dynamic networks

Author for correspondence:

Enoch Yeung

e-mail: eyeung@ucsb.edu

Electronic supplementary material is available online at <https://doi.org/10.6084/m9.figshare.c.5606936>.

Data-driven network models for genetic circuits from time-series data with incomplete measurements

Enoch Yeung¹, Jongmin Kim², Ye Yuan³, Jorge Gonçalves⁴ and Richard M. Murray^{5,6}

¹Center for Biological Engineering, Biomolecular Science and Engineering Program, Department of Mechanical Engineering, Center for Control, Dynamical Systems, and Computation, University of California, Santa Barbara, CA, USA

²Department of Life Sciences, POSTECH, Pohang, South Korea

³School of Artificial Intelligence and Automation, Hua Zhong University of Science and Technology, Wuhan, People's Republic of China

⁴Systems Biology Research Group, University of Luxembourg, Belvaux, Luxembourg

⁵Control and Dynamical Systems, and ⁶Biology and Biological Engineering, California Institute of Technology, Pasadena, CA, USA

EY, 0000-0001-7630-7429

Synthetic gene networks are frequently conceptualized and visualized as static graphs. This view of biological programming stands in stark contrast to the transient nature of biomolecular interaction, which is frequently enacted by labile molecules that are often unmeasured. Thus, the network topology and dynamics of synthetic gene networks can be difficult to verify *in vivo* or *in vitro*, due to the presence of unmeasured biological states. Here we introduce the dynamical structure function as a new mesoscopic, data-driven class of models to describe gene networks with incomplete measurements of state dynamics. We develop a network reconstruction algorithm and a code base for reconstructing the dynamical structure function from data, to enable discovery and visualization of graphical relationships in a genetic circuit diagram as *time-dependent functions* rather than static, unknown weights. We prove a theorem, showing that dynamical structure functions can provide a data-driven estimate of the size of crosstalk fluctuations from an idealized model. We illustrate this idea with numerical examples. Finally, we show how data-driven estimation of dynamical structure functions can explain failure modes in two experimentally implemented genetic circuits, a previously reported *in vitro* genetic circuit and a new *E. coli*-based transcriptional event detector.

1. Introduction

Synthetic gene networks fulfil diverse roles in realizing circuit logic [1] and timing in living organisms [2], ranging from single-input inverters [3,4] to combinatorial input logic gates [5,6], reduction in DNA synthesis and sequencing costs have made it possible to build increasingly complex genetic circuits with tens to hundreds of components. However, the ability to build novel biological circuitry often outpaces our ability to revise designs or to verify what has been built behaves as intended. As the fields of synthetic and systems biology continue to build and integrate on successes of circuit and device-level complexity to engineer entire genetic systems or pathways, we are consistently seeing failure modes that arise from a lack of modularity, e.g. retroactivity [7–9], and context effects [10].

Likewise, the expansion of CRISPR-based methods for genome editing [11,12] has led to new network control problems in systems biology, e.g. design of

minimal genomes [13], reprogramming regulatory networks for phenotype control [14], and fine-tuned optimization of metabolic pathways. The dynamical system in these design challenges is often a complex network of interacting genes, mRNA, proteins and metabolites. The expansion in DNA sequencing read depth has made it possible to profile individual genes via the transcriptome [15], which combined with quantitative proteomics [16] or metabolomics [17], enables systems-level analysis of network activity. But prohibitive sampling and library preparation costs make obtaining highly time-resolved omics measurements hard. This makes it difficult to infer dynamic network activity at the scale of whole cell models [18] without extensive experimental investment.

Dynamic network models that describe the intricate interactions between every biomolecular state or species are referred to as state-space models. Two key variables that often determine the behaviour of these network models are network topology [19,20] and parametric realization [21,22]. The structure of a network is generally determined by how states in the system causally affect each other [22]; edges in the network are determined by causal dependence while nodes are determined by the states of the system [23].

Identifying the active, dynamic network structure of a biological network is critical, since the hypothesized network architecture of a genetic circuit may be very different from the realized network architecture using a specific collection of parts, sequences, and composition approach [24]. While network structure alone does not determine dynamical behaviour, parametric information is also important in determining what dynamical behaviours a system can achieve [25]. Rather, network structure, or topology, often defines or narrows the possible behaviours a system can achieve. Without any structural constraints, a dynamical system can have arbitrary input–output behaviour. Once network structure is imposed, the set of realizable input–output trajectories can be reduced [26,27]. If the realized network differs significantly from the intended network design, the dynamics of the system may produce faults or glitches when appropriately excited or interrogated [28,29]. Getting the actual network topology to match the intended network motif is thus a key element to robust synthetic biological design.

In systems and synthetic biology, canonical network motifs are broadly accepted as enabling useful dynamical behaviour [27,30]. For example, an incoherent feedforward loop can be used for fold-change detection or adaptation [31,32]. A cyclic network of repressors is associated with either oscillations [33–35] or multi-stability [36] while a dual negative feedback network of two nodes is used as memory module or toggle switch [37]. Still, the active, dynamic network architecture of most realizations of these network motifs in the form of genetic circuits are not formally characterized or catalogued [38]. Systematic, generalizable tools that can discover and model dynamic network topology from data are valuable [1].

Circuit network discovery is, at its core, a network reconstruction problem. Given a desired network motif and a physical system, we need to use measurements of the system to determine if the actual, active network of the system matches the intended design. There have been many network reconstruction algorithms developed for natural and synthetic biological networks [24,39–45]. Historically, the approach to discovering network interactions has involved direct perturbation of biochemical species or components in a network [41,45,46]. Individual nodes are perturbed and depending on

if nearby nodes positively or negatively correlate, an activating or repressing relationship between two network nodes can be inferred. In [24], this framework was taken a step further, by showing that the behaviour of direct and indirect links in a benchmark circuit is network topology dependent. This provided a means for using steady-state perturbation data [39,45,47] to estimate network models. Furthermore, these steady-state estimation algorithms have been verified using a benchmark synthetic gene circuit [43]. More recently, the authors in [42] and [40] showed that retroactivity in gene networks can paradoxically confound network predictions that are based wholly on correlation measures. The core issue is that even when measurement data for all biological states are available, causality is difficult to determine from steady-state measurement data affected by back-action or retroactivity in genetic networks [40].

At the single-cell level, the reconstruction problem for biological networks introduces challenges of inferring non-linear stochastic models from noisy data [44,48,49]. In [48], the authors show that by comparing average abundances, molecule lifetimes, covariances and magnitude of step, they can map pairwise interaction dynamics, even when the rest of the system is completely unspecified. The key observation is that assembly stoichiometry of new molecules is fixed, so unbalanced production of linked precursor components will exacerbate imbalance further, resulting in empirically observed large fluctuations. Furthermore, Hilfinger *et al.* [49] showed that there are statistical invariants for certain kinds of network interactions, which can be used to evaluate and challenge existing hypotheses of stochastic gene interaction. More recently, Wang *et al.* showed that effective stoichiometric spaces can be used to determine network structure from the covariances of single-cell multiplex data [44]. These studies show that it is possible to infer meaningful structural information about a genetic network, even when only a portion of the network states are observed and the data are fundamentally noisy.

In this paper, we introduce a class of mesoscopic network reconstruction models with adaptable resolution, commensurate with the depth or coverage of the circuit states (or genome) available from fluorimetric, spectrometry-based, or sequencing based measurements. Our method is distinct in that we consider the use of high-resolution time-series data, but where only partial measurement of the network's nodes is feasible. Furthermore, we consider dynamic measurements of bulk culture rather than single cell, where we benefit from the assumptions of high molecular copy number and large reaction volumes [49]. Specifically, we present the dynamical structure function, an abstract model class from linear time-invariant systems theory and show it can be used as a generalized representation of measured interactions between biological or biochemical states. The contributions of this paper are: (1) we show how a dynamical structure function can encode both direct and crosstalk network interactions, by way of theorem and simulated examples, (2) we develop a direct estimation algorithm and code to directly estimate the dynamical structure function, as well as visualization tools to monitor repression and activation in genetic circuits and (3) we demonstrate this theory on two experimental systems: (A) an *in vitro* genelet repressilator from the synthetic biology literature and (B) a novel transcriptional event detector that we build specifically to illustrate dynamical structure reconstruction.

2. Representing network interactions in partially measured biological networks with dynamical structure functions

In both systems and synthetic biology, discovering (or verifying) the network of a (engineered) biological system is an important problem. However, discovering an entire biochemical reaction network is typically ill-posed, since many network dynamics occur simultaneously from host or environmental context [50], loading effects [51], or unanticipated retroactivity effects [7,51–54]. Even without these effects, the reconstruction problem is equivalent to finding a unique realization for the dynamical system from direct measurements of every state of the system. Unique realization problems are difficult, unless the system of interest has specific structure, e.g. measurement functions of the state that are diffeomorphic [55,56]. On the other hand, there are many inputs that can be used to perturb the system of interest, e.g. silencing RNA [57], genetic knockouts [58] and small chemical inducers [59]. Using these inputs, it is straightforward to reconstruct the system transfer function $G(s)$ of the system [60,61], where

$$Y(s) = G(s)U(s).$$

$Y(s)$ is a system's output, $U(s)$ a system's input and s is the Laplace variable. However, the standard system transfer function $G(s)$ only models closed-loop input-to-output dependencies. It has no direct information about how chemical species within the system are interacting with each other.

There are other kinds of transfer function models which have the potential to describe the network of interactions between chemical species, for example, the dynamical structure function [62]. The dynamical structure function is a representation derived from linear systems theory, and thus can be used to model transients of a genetic circuit around an operating point or even unstable network dynamics diverging from an equilibrium point. It is a more detailed description of network structure than the system transfer function $G(s)$ since it models causal interactions between measured outputs, in addition to the causal dependencies of outputs on input variables. Most notably, necessary and sufficient conditions for recovery of dynamical network models have been developed and well-studied [22,62–66], but so far no open-source algorithms, code bases, or applications of this theory have been developed directly for synthetic biology.

2.1. Dynamical structure functions

Here we introduce the mathematical formulation of a dynamical structure function, formulated in the context of biological analysis. The dynamical structure function is formulated on the premise that partial, rather than full, state measurements are accessible. The measured states are denoted as $y \in \mathbb{R}^p$ and the hidden or latent states are denoted as $x_h \in \mathbb{R}^{n-p}$, where n is the dimension of the whole state. We then denote the state of the dynamical system

$$x = [y^T \ x_h^T]^T \in \mathbb{R}^n.$$

We let $u \in \mathbb{R}^m$ denote exogenous inputs that can be introduced to influence the dynamics of the state x . With the exception of oscillators, many biochemical reaction networks converge to a steady state. Moreover, it is generally the case that the parameters of biochemical reaction networks are time-invariant

[67], so long as macroscopic experimental settings of the system such as temperature, growth media and dissolved oxygen content remain fixed. Therefore, while the model of a biochemical reaction network is of the form

$$\left. \begin{aligned} \dot{y} &= f_y(y, x_h, u), & y(0) &= y_0 \\ \dot{x}_h &= f_{x_h}(y, x_h, u), & x_h(0) &= x_{h,0} \end{aligned} \right\} \quad (2.1)$$

and

$$y = \begin{bmatrix} \mathbf{I}_{p \times p} & 0 \end{bmatrix} \begin{bmatrix} y \\ x_h \end{bmatrix},$$

we will suppose that we can linearize the system about either an equilibrium point, a nominal operating point, or even an (unstable or oscillatory) initial condition to extract network dynamics. In biological systems, networks are almost never precisely linear, but we presume to model local fluctuations or perturbations from a target point in the state space. As we will see in the following, this will be enough to extract relevant network information. Proceeding with the linearization, we can write the system in the form

$$\left. \begin{aligned} \begin{bmatrix} \dot{y} \\ \dot{x}_h \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x_h \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \end{aligned} \right\} \quad (2.2)$$

and

$$y = C \begin{bmatrix} y \\ x_h \end{bmatrix},$$

where

$$C = [\mathbf{I}_{p \times p} \ 0].$$

We also assume the system's initial condition of the linearized system is $x(0) = 0$, and the entries in $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are calculated as

$$\begin{aligned} A_{11} &\equiv \left. \frac{\partial f_y(y, x_h, u)}{\partial y} \right|_{x=x_e, u=u_e}, & A_{12} &\equiv \left. \frac{\partial f_y(y, x_h, u)}{\partial x_h} \right|_{x=x_e, u=u_e} \\ A_{21} &\equiv \left. \frac{\partial f_{x_h}(y, x_h, u)}{\partial y} \right|_{x=x_e, u=u_e}, & A_{22} &\equiv \left. \frac{\partial f_{x_h}(y, x_h, u)}{\partial x_h} \right|_{x=x_e, u=u_e} \end{aligned}$$

and

$$B_1 \equiv \left. \frac{\partial f_y(y, x_h, u)}{\partial u} \right|_{x=x_e, u=u_e}, \quad B_2 \equiv \left. \frac{\partial f_{x_h}(y, x_h, u)}{\partial u} \right|_{x=x_e, u=u_e}.$$

Taking Laplace transforms, solving for $X_h(s)$ and replacing it in $Y(s)$ we obtain

$$sY = W(s)Y(s) + V(s)U(s), \quad (2.3)$$

where

$$\left. \begin{aligned} W(s) &= A_{11} + A_{12}(sI - A_{22})^{-1}A_{21} \\ \text{and} \quad V(s) &= B_1 + A_{12}(sI - A_{22})^{-1}B_2. \end{aligned} \right\} \quad (2.4)$$

Defining $D(s) = \text{diag}(W(s))$ and subtracting $D(s)Y(s)$ from both sides of equation (2.3) and solving for $Y(s)$ we obtain the following equation:

$$Y(s) = Q(s)Y(s) + P(s)U(s), \quad (2.5)$$

where

$$Q(s) = (sI - D(s))^{-1}(W(s) - D(s)) \quad (2.6)$$

is a $p \times p$ matrix transfer function and

$$P(s) = (sI - D(s))^{-1}V(s) \quad (2.7)$$

is a $p \times m$ matrix transfer function. In our use of the term transfer function here, we distinguish between the *system transfer function* $G(s)$ that describes the closed loop relationship between inputs and outputs and the matrix transfer functions

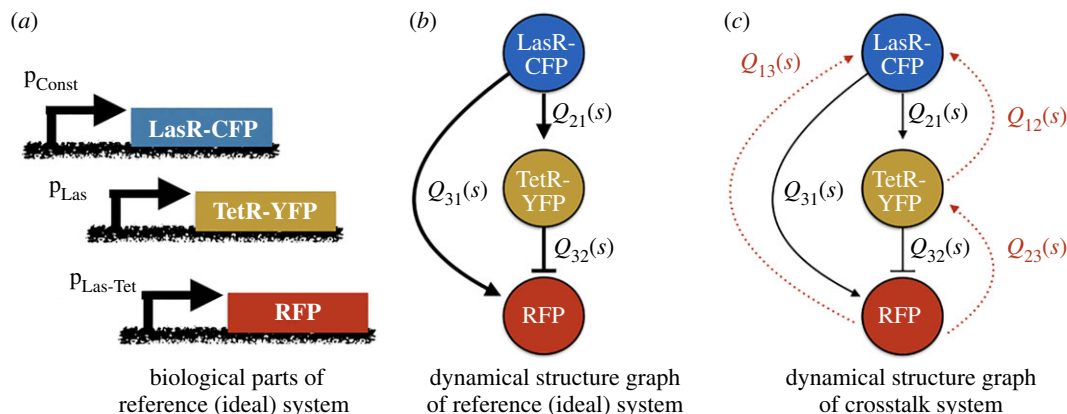


Figure 1. Dynamical structure functions can be used to analyse synthetic gene networks. (a) Synthetic biological parts for an incoherent feedforward loop (IFFL) using the LasR activator, the TetR repressor and reporter proteins CFP, YFP and RFP. (b,c) The dynamical structure graphs of the crosstalk-free IFFL from system (2.8), in (b) and the crosstalk-impacted IFFL from system (2.9), in (c). Nodes represent measured biochemical species, with black edges denoting open-loop causal dependencies stemming from designed interactions, and red edges denoting open-loop causal dependencies arising from crosstalk or loading effects. Note that the dynamical structure captures network model interactions that are not described by the system transfer function $G(s)$.

$Q(s)$ and $P(s)$ that encode *open-loop* causal relationships. Each entry $Q_{ij}(s)$ is a transfer function that describes the open-loop causal dependency of measured state $Y_i(s)$ on measured state $Y_j(s)$. Similarly, the transfer function $P_{ij}(s)$ describes the open-loop causal dependency of measured state $Y_i(s)$ on input $U_j(s)$. The matrix pair $(Q(s), P(s))$ is known as the *dynamical structure function*, where $Q(s)$ is referred to as the network structure and $P(s)$ as the control structure. Finally, we define the time-dependent dynamical structure function $(Q(t), P(t))$ as the inverse Laplace transform of the dynamical structure function, where

$$Q(t) \equiv \mathcal{L}^{-1}(Q(s))$$

and

$$P(t) \equiv \mathcal{L}^{-1}(P(s)).$$

Note that $Q(s)$ is defined as $Q(s) = (sI - D)^{-1}(W - D)$ rather than $Q_{\text{alt}}(s) = 1/sW(s)$. This guarantees that the diagonal entries of $Q(s)$ are 0, which implies that any non-zero terms $Q_{ij}(s)$ are strictly proper transfer functions and thus descriptions of causal interactions among measured Y_i and Y_j . This also means that $Q(s)$, defined in this way, is unique and has p fewer transfer functions to identify on its diagonal. This construction of $Q(s)$ and $P(s)$ ultimately ensures identifiability [62] under reasonable assumptions of independent input perturbation [24,39–45]. Furthermore, if $Q(s) = W/s$, then we would face two simultaneous challenges in estimation: (1) disentangling autoregulatory dynamics (Y_i to Y_i) from pairwise interactions (Y_i to Y_j) and (2) too many unknown parameters in both $Q(s)$ and $P(s)$. Lastly, we find that studying the pairwise interactions $Q_{ij}(s)$ can already elicit important functional information about a genetic network, as illustrated by the next two examples.

2.1.1. Example: the dynamical structure function of an idealized incoherent feedforward loop

Consider the following synthetic biology design problem: design and implement an incoherent feedforward loop. Specifically, we consider implementing a feedforward loop using the synthetic parts pLac-LasR-CFP-LVA, pLas-TetR-YFP-LVA, and pLas-Tet-RFP-LVA and IPTG, $C_3O_6H_{12}$ -HSL, and aTc as inputs (figure 1). We model the protein

concentration of LasR-CFP, TetR-YFP, and RFP as x_1 , x_2 and x_3 , respectively. We denote the corresponding mRNA species for each of these proteins as m_1 , m_2 and m_3 . A simple model without any loading effects, describing the dynamics of these states can be written as

$$\left. \begin{aligned} \dot{x}_1 &= \rho_1 m_1 - \frac{C_0 x_1 / k_{1,d}}{1 + x_1 / k_{1,d}} \\ \dot{x}_2 &= \rho_2 m_2 - \frac{C_0 x_2 / k_{2,d}}{1 + x_2 / k_{2,d}} \\ \dot{x}_3 &= \rho_3 m_3 - \frac{C_0 x_3 / k_{3,d}}{1 + x_3 / k_{3,d}} \\ \dot{m}_1 &= \frac{\alpha_1 u_1}{k_{M,u1} + u_1} - \delta_m m_1 \\ \dot{m}_2 &= \frac{\alpha_2 x_1 u_2 / k_{M,u2}}{1 + x_1 u_2 / k_{M,u2}} - \delta_m m_2 \\ \dot{m}_3 &= \frac{\alpha_3 x_1 u_2 / k_{M,u2}}{1 + x_1 u_2 / k_{M,u2} + x_2 / (k_{M,2} + u_3 / k_{M,u3})} - \delta_m m_3 \end{aligned} \right\}$$

and $y = [I_{3 \times 3} \quad 0_{3 \times 3}] [\bar{x}^T \quad \bar{m}^T]^T$,

(2.8)

where protein production rates $\rho_1 = 641.4$, $\rho_2 = 585.1$, $\rho_3 = 652.8 \text{ nM s}^{-1}$ [68], mRNA production rates $\alpha_1 = 7.8$, $\alpha_2 = 7.1$, $\alpha_3 = 7.92 \text{ nM s}^{-1}$ [69], degradation Michaelis constants $k_{1,d} = k_{2,d} = k_{3,d} = 200 \text{ nM}$, input Michaelis constants $k_{M,u1} = k_{M,u2} = k_{M,u3} = 4000 \text{ nM}$ [67], degradation rate constant $C_0 = 1 \text{ nM s}^{-1}$ and mRNA degradation or dilution rate $\delta_m = 10 \text{ nM s}^{-1}$ [70].

Based on the parameters selected above, the system yields a stable equilibrium point (x_e, u_e) which becomes the point about which we linearize this example system. The dynamical structure function for this system is derived following the procedure outlined above: first, we take Laplace transforms; second, we eliminate the hidden mRNA states of x_1 , x_2 and x_3 , namely m_1 , m_2 , m_3 . The network structure $Q^a(s)$ and control structure $P^a(s)$ matrix transfer functions evaluate to the following:

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{0.045}{s^2 + 1.5s + 5.7 \cdot 10^{-4}} & 0 & 0 \\ \frac{1.5 \cdot 10^{-4}}{s^2 + 1.7s + 0.2} & -\frac{1.5 \cdot 10^{-4}}{s^2 + 1.7s + 0.2} & 0 \end{pmatrix}$$

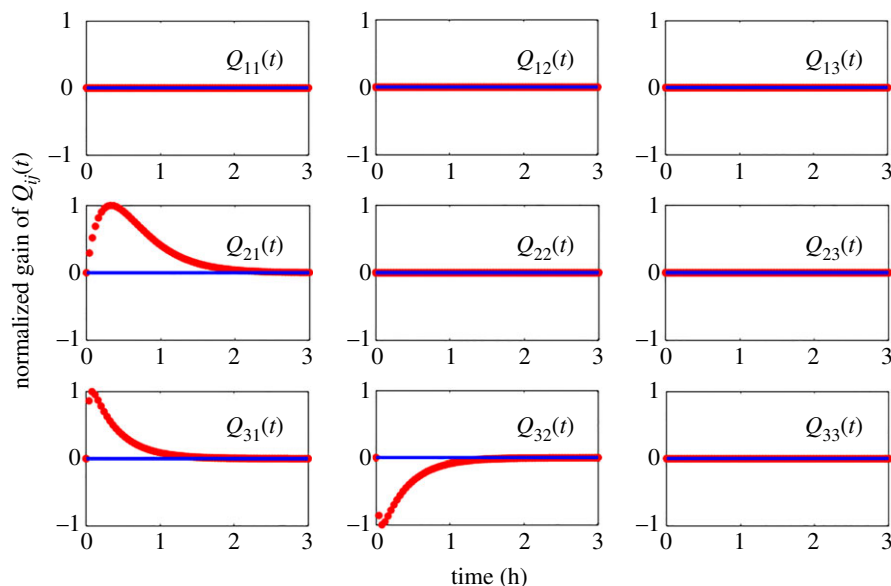


Figure 2. Dynamical structure functions describe how network structure evolves over time (and as a function of frequency). The time-lapse response of the dynamical structure convolution kernel $Q^a(t) = \mathcal{L}^{-1}(Q^a(s))$ for the incoherent feedforward loop in system (2.8). By examining the functional response of each entry in $Q^a(t)$ (or $Q^a(s)$), we see that the network structure of the incoherent feedforward loop in example 2.1.1 is a time-evolving, or *dynamic*, entity.

and $P^a(s)$ is

$$\begin{pmatrix} \frac{6.7 \cdot 10^{-7}}{s^2 + 1.5s + 5.7 \cdot 10^{-4}} & 0 & 0 \\ 0 & \frac{244.0}{s^2 + 1.5s + 5.7 \cdot 10^{-4}} & 0 \\ 0 & \frac{0.78}{s^2 + 1.7s + 0.2} & \frac{7.8}{s^2 + 1.7s + 0.2} \end{pmatrix}.$$

The network, with edge weight functions corresponding to the entries of $Q^a(s)$, is drawn in figure 1*b*. Note that if we take $s \in \mathbb{R}_{>0}$, the sign of the entries in $Q^a(s)$ coincides with the form of transcriptional regulation implemented by TetR and LasR, respectively. In [71], it was shown that the sign definite properties of entries in $Q(\mathbb{R}_{>0})$ are useful for reasoning about the monotonicity of interactions between measured outputs and how fundamental limits in system performance relate to network structure.

Let us now consider $Q^a(t)$ as defined above. We remark that $y(t) = \int_0^t Q^a(t) y(t - \tau) d\tau$ follows from the equation

$$\mathcal{L}^{-1}(Y(s)) = \mathcal{L}^{-1}(Q^a Y(s) + P^a U(s)),$$

whenever $u(t) \equiv 0$ such that $U(s)$ is 0. This argument holds in general for any system of the form (2.2). In particular, the entries $Q^a(t)$ act as convolution kernels, and taken with the integral, define an operator for mapping $y_j(t)$ to $y_i(t)$. $[Q^a(t)]_{ij}$ also models the isolated impulse response of $y_i(t)$ to an impulse $y_j(t)$, while assuming all other elements are off or set to zero. In this way, $[Q^a(t)]_{ij}$ encodes the time-dependent gain of $y_i(t)$ on $y_j(t)$ assuming all other nodes in the network are momentarily off. In the case of our example, we can see that the network structure of this incoherent feedforward loop is *dynamical*, hence our usage of the term *dynamical structure function* to describe the network structure among the measured chemical species $y(t)$. In this particular case, the time-domain analogue of the dynamical structure (or dynamical structure convolution kernel) is given as

$$Q^a(t) \equiv \begin{pmatrix} 0 & 0 & 0 \\ Q_{21}(t) & 0 & 0 \\ Q_{31}(t) & Q_{32}(t) & 0 \end{pmatrix},$$

where

$$Q_{21}(t) = (5.5 \times 10^{-3}) e^{-(8.3 \times 10^{-4})t} \sinh((6.0 \times 10^{-5})t)$$

$$Q_{31}(t) = (4.1 \times 10^{-7}) e^{-(4.9 \times 10^{-3})t} \sinh((4.1 \times 10^{-3})t)$$

$$\text{and } Q_{32}(t) = -(9.7 \times 10^{-8}) e^{-(4.9 \times 10^{-3})t} \sinh((4.1 \times 10^{-3})t).$$

A visualization of each of these impulse kernel functions and their corresponding location in the dynamic adjacency matrix, defined by $Q^a(t)$, is given in figure 2. Note how the activating or repressing nature of genetic regulation is encoded by the positive or negative sign of the corresponding kernel response. In addition to uncovering the Boolean network of interactions between biological states, the dynamical network convolution kernel $Q^a(t)$ reveals the time scales of response of each network edge, as well as the amplitude and the rate of decay of the gain from the time of impulse. Similar response profiles can be generated for step function inputs, though finite impulse inputs are typically more common in biological networks. Interestingly, the transfer function $G^a(s)$ of the system is likewise lower triangular, reflecting the feedforward network topology in the genetic circuit. Specifically, $G^a(s)$ has a sparsity structure of the form

$$G^a(s) \equiv \begin{bmatrix} G_{11}^a(s) & 0 & 0 \\ G_{21}^a(s) & G_{22}^a(s) & 0 \\ G_{31}^a(s) & G_{32}^a(s) & G_{33}^a(s) \end{bmatrix}.$$

2.1.2. Example: the dynamical structure function of an incoherent feedforward loop with crosstalk

In prototyping a feedforward loop, it is important to anticipate *in vivo* context effects. We consider the same biocircuit as described in example 2.1.1, except now we specifically consider loading effects frequently neglected in the design process of synthetic biology. First, we note that each gene may be susceptible to loading effects [7]. For each gene in figure 1*a*, a degradation tag is added, to provide tunability, to the rate of degradation of the protein. Inside the cell, a protease called ClpXP targets these degradation tags and degrades the associated protein. Different tags can be incorporated to modulate

the gain of the degradation process. Furthermore, these degradation tags can be subject to mutagenesis experiments, as a means to modulate tunability.

Tunability of degradation introduces a tradeoff in performance. Since the ClpXP protease is a housekeeping protein expressed to form a common pool of proteases for all genes in the cell, there is a limit to the supply of free ClpXP protein in any instant of the cell's growth cycle. When there are too many degradation-tagged proteins [72], the overloading of the protein degradation queue can trigger unwanted effects such as stress response. More directly, the competition for scarce proteases can induce coupled dynamics or a *virtual* or *indirect* interaction between two genes competing for the same protease pool. Even if the genes were engineered to have no direct transcriptional or translational cross-regulation, the competition for the same protease effectively couples the protein states of both genes. Modifying the above model to account for these type of loading effects yields

$$\left. \begin{aligned} \dot{x}_1 &= \rho_1 m_1 - \frac{C_0 x_1 / k_{1,d}}{1 + x_1 / k_{1,d} + x_2 / k_{2,d} + x_3 / k_{3,d}} \\ \dot{x}_2 &= \rho_2 m_2 - \frac{C_0 x_2 / k_{2,d}}{1 + x_1 / k_{1,d} + x_2 / k_{2,d} + x_3 / k_{3,d}} \\ \dot{x}_3 &= \rho_3 m_3 - \frac{C_0 x_3 / k_{3,d}}{1 + x_1 / k_{1,d} + x_2 / k_{2,d} + x_3 / k_{3,d}} \\ \dot{m}_1 &= \frac{\alpha_1 u_1}{k_{M,u1} + u_1} - \delta_m m_1 \\ \dot{m}_2 &= \frac{\alpha_2 (x_1 u_2 / k_{M,u2})}{1 + x_1 u_2 / k_{M,u2}} - \delta_m m_2 \\ \dot{m}_3 &= \frac{\alpha_3 x_1 u_2 / k_{M,u2}}{1 + x_1 u_2 / k_{M,u2} + x_2 / (k_{M,2} + u_3 / k_{M,u3})} - \delta_m m_3 \end{aligned} \right\} \quad (2.9)$$

and $y = [\mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 3}] [\bar{x}^T \quad \bar{m}^T]^T$.

We use the same parametric values as before. Computing the dynamical structure function, we obtain $Q^c(s)$ as

$$\begin{pmatrix} 0 & \frac{1.6 \cdot 10^{-3}}{s+2.1 \cdot 10^{-3}} & \frac{0.041}{s+2.1 \cdot 10^{-3}} \\ \frac{(1.6 \cdot 10^{-3})s+0.048}{s^2+1.5s+3.3 \cdot 10^{-3}} & 0 & \frac{0.041}{s+2.1 \cdot 10^{-3}} \\ \frac{(3.8 \cdot 10^{-4})s+7.4 \cdot 10^{-4}}{s^2+1.6s+0.13} & \frac{(3.8 \cdot 10^{-4})s+4.4 \cdot 10^{-4}}{s^2+1.6s+0.13} & 0 \end{pmatrix}$$

and $P^c(s)$ as

$$\begin{pmatrix} \frac{6.7 \cdot 10^{-7}}{s^2+1.5s+3.2 \cdot 10^{-3}} & 0 & 0 \\ 0 & \frac{244.0}{s^2+1.5s+3.3 \cdot 10^{-3}} & 0 \\ 0 & \frac{0.78}{s^2+1.6s+0.13} & \frac{7.8}{s^2+1.6s+0.13} \end{pmatrix}.$$

Note that $Q^c(s)$ is no longer lower-triangular, but fully connected. Introducing loading effects creates additional coupling between nodes in the network. If the coupling is significant, the *designed* network interactions of the incoherent feedforward loop are overcome by the *crosstalk* network interactions [8,20,51,53,71,73,74]. Thus, the coupling that is introduced into the biochemical reaction network by loading effects is reflected in the structure of $(Q^c, P^c)(s)$ (see figure 3).

By contrast, the input–output transfer function $G^c(s)$ of the crosstalk system only characterizes how system outputs causally depend on inputs. When calculated, $G_c(s)$ also is a full matrix like $Q^c(s)$ of sixth-order SISO transfer functions

$$G^c \equiv \begin{bmatrix} G_{11}^c(s) & G_{12}^c(s) & G_{13}^c(s) \\ G_{21}^c(s) & G_{22}^c(s) & G_{23}^c(s) \\ G_{31}^c(s) & G_{32}^c(s) & G_{33}^c(s) \end{bmatrix} (s),$$

but all structural information about how loading effects cause interference *among* measured system states $Y(s)$ is mixed with the information about how outputs causally depend on inputs $U(s)$ in $G(s)$. An identification algorithm of entries in $G(s)$ will thus be *unable* to quantify the size of crosstalk or interference among system states $Y(s)$.

To what extent can the entries of $(Q(s), P(s))$ can be used to quantify the size of crosstalk in a synthetic gene network? The following theorem shows that the dynamical structure function can quantify crosstalk-induced deviation from idealized or ‘designed’ system behaviour.

Theorem 2.1. *Let \mathcal{L} denote the two-sided Laplace operator. Suppose we have a system model that incorporates the effect of crosstalk:*

$$\left. \begin{aligned} \dot{y}^c &= f_y^c(y^c, x_h^c, u), \quad y^c(0) = 0 \\ \text{and} \quad \dot{x}_h^c &= f_{x_h}^c(y^c, x_h^c, u), \quad x_h^c(0) = 0, \end{aligned} \right\} \quad (2.10)$$

where $y^c \in \mathbb{R}^p$ is a vector of measured states in the output, $x_h^c \in \mathbb{R}^{n-p}$ are the unmeasured states of the system, $x^c = (y^c, x_h^c)$ is the full system state, and $u \in \mathbb{R}^m$ is a vector of system inputs. Furthermore, suppose we have a idealized system model to simulate system dynamics in the absence of crosstalk:

$$\left. \begin{aligned} \dot{y}^a &= f_y^a(y^a, x_h^a, u), \quad y^a(0) = 0 \\ \text{and} \quad \dot{x}_h^a &= f_{x_h}^a(y^a, x_h^a, u), \quad x_h^a(0) = 0, \end{aligned} \right\} \quad (2.11)$$

where $y^a \in \mathbb{R}^p$ is a vector of the measured states in the output, $x_h^a \in \mathbb{R}^{n-p}$ are the unmeasured states of the system, $x^a = (y^a, x_h^a)$ is the full system state, and $u \in \mathbb{R}^m$ is a vector of system inputs. Let $(Q^c(s), P^c(s))$ and $(Q^a(s), P^a(s))$ denote the respective dynamical structure functions calculated for each linearized system about the origin. Let

$$\zeta(t) = x^a(t) - x^c(t)$$

denote the deviation of the crosstalk state from the idealized state. Then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = Q_{ij}^a(s) - Q_{ij}^c(s) + \frac{\partial}{\partial Y_j} \mathcal{L}(O((x^c)^2)), \quad (2.12)$$

and in particular, if

$$Q_{ij}^a(s) \equiv 0,$$

then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = -Q_{ij}^c(s) + \frac{\partial}{\partial Y_j} \mathcal{L}(O((x^c)^2)),$$

and can be estimated from input output data $(Y(s), U(s))$.

Proof. The proof of this theorem is provided in the electronic supplementary material (theorem C7). ■

There are two ways to apply this result. First, if one has an idealized or reference dynamical structure model of the circuit, then this can be compared to the dynamical structure model estimated from the data. Second, in the absence of a reference model, the dynamical structure model can be estimated directly from data to observe new edge functions or gain mismatch directly from the data-driven dynamical structure model. In the latter scenario, if the crosstalk interactions are mediated on an existing edge, it will not be possible to separate the magnitude of the crosstalk from the existing (intended) dynamics of a given active edge in the network. However, the discovered

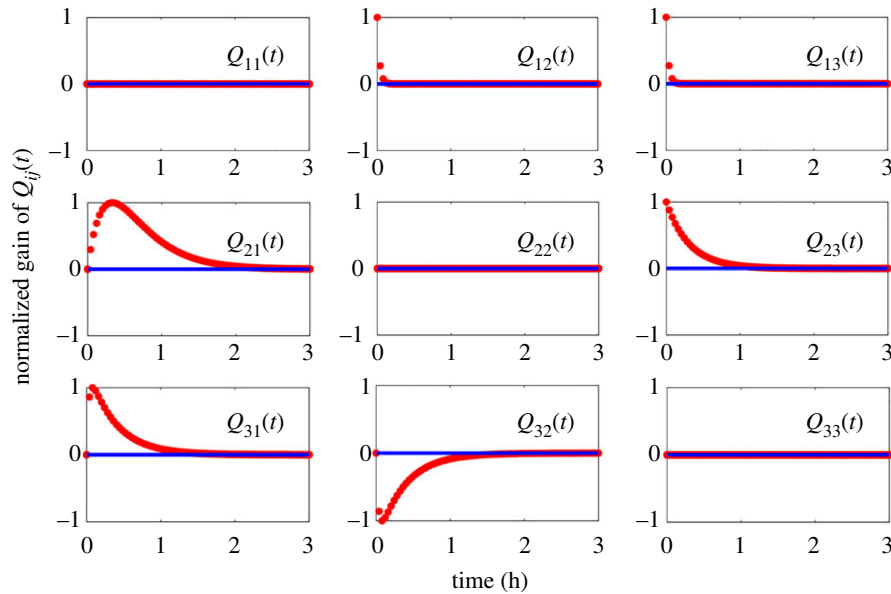


Figure 3. Dynamical structure functions describe how network structure evolves over time (and as a function of frequency). The time-lapse response of the dynamical structure convolution kernel $Q^c(t) = \mathcal{L}^{-1}(Q^c(s))$ for the incoherent feedforward loop in system (2.9). By examining the functional response of each entry in $Q^c(t)$ (or $Q^c(s)$), we see that the network structure of the incoherent feedforward loop in example 2.1.1 is a time-evolving, or *dynamic*, entity.

edge dynamics can be compared to the intended behaviour, e.g. intentional activation or repression over a certain growth phase of the cells.

When the assumptions of theorem 2.1 are satisfied, the entries of $Q_{ij}^c(s)$ can be used to quantify the size of the crosstalk in the system. Example 1 in the electronic supplementary material shows how the \mathcal{H}_2 gain of entries in $Q_{ij}^c(s)$ quantify crosstalk due to enzyme loading. Furthermore, we can compute $Q_{ij}^c(t) = \mathcal{L}^{-1}(Q_{ij}^c(s))$, which gives a simulated response model of how crosstalk gain fluctuates over time to affect $Y_i(t)$ in response to an impulse in $Y_j(t)$. This response model also can be used to estimate the relevant time scales where crosstalk gain is as large as the gain of designed interactions between circuit components.

Knowledge of the active crosstalk in a genetic circuit can be useful for revising closed-loop design or designing biological controllers to compensate for crosstalk effects. As discussed above, discovering the network model of a complex biological network can be useful for design, verification, analysis or control. In the next section, we formally introduce the theoretical conditions under which identification of the dynamical structure function is possible, as well as the formal problem of network reconstruction.

3. Direct estimation algorithm for dynamical structure functions

In this section, we will introduce a direct estimation algorithm for estimating the dynamical structure function. The dynamical structure function is a tuple of matrix transfer functions $(Q(s), P(s))$ and can be directly estimated from experimental or simulation data so long as the data and the conditions of the experiment satisfy the assumptions of the following theorem, originally shown in [62].

Theorem 3.1. *Given a $p \times m$ transfer function $G(s)$, dynamical structure reconstruction is possible from partial structure information if and only if $p - 1$ entries in each column of $[Q(s) P(s)]^*$*

are known, to specify linearly independent elements of the nullspace of the matrix function $[G(s) I]^$.*

Proof. See the proof of theorem 2 in section IIIB of [62]. ■

It follows from this theorem that without additional structural information about the columns of the matrix $[Q(s)P(s)]^*$, it is not possible to identify $(Q(s), P(s))$. In synthetic gene circuits with complex internal network interactions encoded by $Q(s)$, we can still solve for the structure and parameters of $Q(s)$, as long as $p - 1$ elements of $P(s)$ are known. For example, targeted gene knockdowns (CRISPRi) or knockouts (engineered genomic mutations) can be engineered so that $P(s)$ is a diagonal matrix transfer function. This guarantees that $p - 1$ entries of each column of $P(s)$ are known, which satisfies the conditions of theorem 3.1. In this set-up, *all genes or biological states in the network of interest* are (A) monitored by some measurement channel over time and (B) independently perturbed by a diagonal element in $P(s)$. These are necessary and sufficient conditions for reconstruction of $Q(s)$ and $P(s)$ and corroborate the general network reconstruction conditions developed for network reconstruction of full state measurement systems [24,39–45].

Under the above premises, the task is to estimate the diagonal transfer function entries of $P(s)$ and to estimate all off-diagonal entries of $Q(s)$. Recall from the derivation in §2.1 that the diagonal entries of $Q(s)$ are set to 0 by subtracting the diagonal entries of the precursor $W(s)$ transfer function matrix from $W(s)$. Furthermore, by left-multiplying $(sI - D)^{-1}$ with $W(s) - D(s)$, this guarantees that the representation for $Y(s) = Q(s)Y(s) + P(s)U(s)$ yields a unique $Q(s)$ and $P(s)$. The entries of $Q(s)$ and $P(s)$ are all strictly proper rational transfer functions and thus encode causal dynamics.

Problem 3.2 (Network reconstruction and estimation). Given output measurements $Y(s)$ and input measurements $U(s)$ for a system (2.1) satisfying identifiability conditions described in theorem 3.1, the network reconstruction problem

is to find the dynamical structure function ($Q(s)$, $P(s)$), as defined in (2.6) and (2.7) that satisfies the equation

$$Y(s) = Q(s)Y(s) + P(s)U(s).$$

The network estimation problem is to find an estimate dynamical structure function ($\hat{Q}(s)$, $\hat{P}(s)$) to solve the minimization problem

$$\min_{\hat{Q}, \hat{P}} \|Y(s) - \hat{Q}(s)Y(s) - \hat{P}(s)U(s)\|.$$

In practice, there are two approaches to solve for $Q(s)$ and $P(s)$. The first is to estimate the transfer function $G(s)$ using a standard transfer function estimation routine, followed by inversion of $G(s)$ and calculation of the entries of $P_{ii}(s)$ and subsequently the entries of $Q(s)$. This approach has the drawback of relying on inversion of the matrix transfer function $G(s)$, which often requires symbolic inversion and is thus prone to numerical instability and scaling issues for larger networks.

The second approach, which we propose here and develop code for, is to identify the dynamical structure function directly from data by writing the model estimation problem in normal form. First, we can denote the discrete-time approximations of $\hat{Q}(s)$ and $\hat{P}(s)$ as $\hat{Q}(z)$, $\hat{P}(z)$, where z denotes the z -transform. $\hat{Q}(z)$, $\hat{P}(z)$ will follow the same identifiability conditions [62].

Specifically, we have that

$$Y(z) = \hat{Q}(z)Y(z) + \hat{P}(z)U(z), \quad (3.1)$$

which given that $\hat{Q}(z)$ and $\hat{P}(z)$ share the same denominator, we can multiply the characteristic polynomial of $\hat{Q}(z)$ on both sides, to obtain

$$(z^{-n_d} + a_1 z^{-n_d+1} + \dots + a_{n_d})Y(z) = \hat{Q}_{\text{num}}(z)Y(z) + \hat{P}_{\text{num}}(z)U(z). \quad (3.2)$$

Here, the matrices $\hat{Q}_{\text{num}}(z)$, $\hat{P}_{\text{num}}(z)$ are matrices that contain the numerator entries of each transfer function in $Q(z)$ and $P(z)$. Namely

$$\hat{Q}_{ij}(z) = \frac{[\hat{Q}_{\text{num}}(z)]_{ij}}{(z^{-n_d} + a_1 z^{-n_d+1} + \dots + a_{n_d})}$$

and

$$\hat{P}_{ij}(z) = \frac{[\hat{P}_{\text{num}}(z)]_{ij}}{(z^{-n_d} + a_1 z^{-n_d+1} + \dots + a_{n_d})}.$$

We can express the known quantity $z^{-n_d}Y(z)$ after inverse Z -transforming to obtain

$$\begin{aligned} y[t - n_d] &= -a_1 y[t - n_d + 1] - a_2 y[t - n_d + 2] \\ &\quad - \dots - a_{n_d} y[t - n_d] - \Theta_{Q,\text{num}} Y_{t-m_d:t} \\ &\quad - \Theta_{P,\text{num}} U_{t-m_d:t} \end{aligned} \quad (3.3)$$

where $Y_{t-m_d:t}$ denotes a matrix containing time-shifted entries $y[t - m_d]$, $y[t - m_d + 1]$, $y[t - m_d + 2]$, \dots , $y[t]$, of a time series or time trace $y[t]$, $t \in \mathbb{N}_0$. Similarly, $U_{t-m_d:t}$ denotes a matrix containing time-shifted entries $u[t - m_d]$, $u[t - m_d + 1]$, $u[t - m_d + 2]$, \dots , $u[t]$, of a time series or time trace $u[t]$, $t \in \mathbb{N}_0$. The terms $\Theta_{Q,\text{num}}$ and $\Theta_{P,\text{num}}$ contain the coefficients of the numerator polynomials of off-diagonal elements in $Q(s)$ and the diagonal elements in $P(s)$, respectively.

The above equation can be thus written in normal form as

$$b = v^T \Theta. \quad (3.4)$$

Here b and v are dependent on the elements of a single time trace of output measurements $y[t]$ and input measurements $u[t]$. This equation can be also stacked for multiple time

traces $y^1[t]$, $y^2[t]$, \dots , $y^{N_{\text{exp}}}[t]$ collected from N_{exp} different conditions or experimental replicates, or by simply staggering the time horizon, to obtain the stacked normal form equations

$$B = X\Theta, \quad (3.5)$$

where B is a stacked matrix of b^1 , b^2 , \dots , $b^{N_{\text{exp}}}$ and X is a stacked matrix of v^1 , v^2 , \dots , $v^{N_{\text{exp}}}$ of time-series data and Θ contains the coefficients determining the characteristic polynomial and the numerators of the dynamical structure function. Once these estimates are obtained, a standard discrete-to-continuous transformation can be used to estimate $Q(s)$ and $P(s)$. The steps of our algorithm are summarized in algorithm 1 and the full MATLAB code is provided at the Github repository <https://github.com/YeungRepo/NetRecon>.

Algorithm 1. Direct QP estimation algorithm

```

1 Input:  $Y_{Data}, U_{Data} \in \mathbb{R}^{N \times p \times N_{Exp}}$ 
2 Result: Dynamical structure function ( $Q(s), P(s)$ )
3 Initialize:
4  $n_{d_{min}}$  (min. num. of poles),
5  $n_{d_{max}}$  (max. num. of poles),
6  $h_{max}$  (max. horizon of subsampled time-series trace)
7 for  $n_{ord} = n_{d_{min}}, \dots, n_{d_{max}}$  do
8   while  $1 < h_{max} < N - n_{ord} - 1$  do
9     Initialize:
10     $Y_{LH} = []$  (stacked snapshots of time-series data
      corresponding to char. polynomial of  $Q(s)$ )
11     $Y_{RH} = []$  (stacked snapshots of time-series data
      corresponding to numerator terms of  $Q$ )
12     $U_{RH} = []$  (stacked snapshots of time-series data
      corresponding to numerator terms of  $P$ )
13    for  $expid = 1, \dots, N_{Exp}$  do
14      for  $c = 1, \dots, p$  do
15        for  $r = 1, \dots, h_{max}$  do
16           $t_s = N - (r - 1) - n_{ord}$ 
17           $t_e = N - (r - 1)$ 
18           $Y_{LH}.append(Y[t_s : t_e, c, expid])$ 
19          Initialize:  $Y_{alt}$  as empty array.
20          for  $y_{alt} = 1, \dots, p - 1$  do
21            if  $y_{alt} + c > p$  then
22               $c_{alt} = \text{mod}(y_{alt} + c), p)$ 
23            else
24               $c_{alt} = y_{alt} + c$ 
25            end
26             $t_{s,alt} = N - (r - 1) - n_{ord}$ 
27             $t_{e,alt} = N - (r - 1)$ 
28             $Y_{alt}.append(Y[t_{s,alt} : t_{e,alt}, c_{alt}, expid])$ 
29          end
30           $z_{pre} = n_{ord}(c - 1)(p - 1)$ 
31           $z_{post} = n_{ord}(y - c)(p - 1)$ 
32           $Y_{RH}.append([\mathbf{1}_{c \equiv 1} \mathbf{0}_{1 \times z_{pre}} | Y_{alt} | \mathbf{1}_{c \equiv p} \mathbf{0}_{1 \times z_{post}}])$ 
33          Initialize:  $U_{temp}$  as empty array.
34           $t_s^u = N - r + 1 - n_{ord} + 1$ 
35           $t_e^u = N - r + 1$ 
36          for  $u_{ind} = 1, \dots, m$  (or  $p$  since  $m = p$ ) do
37             $U_{temp}.append(\mathbf{1}_{u_{ind} \equiv c} U_{Data}[t_s^u : t_e^u])$ 
38          end
39           $U_{RH}.append(U_{temp})$ 
40        end
41      end
42     $X = [-Y_{LH}[:, 2 : n_{ord} + 1] | Y_{RH} | U_{RH}]$ 
43     $B = [Y_{LH}[:, 1]]$ 
44     $\Theta(h_{max}) = X^\dagger B$ 
45  end
46 end

```

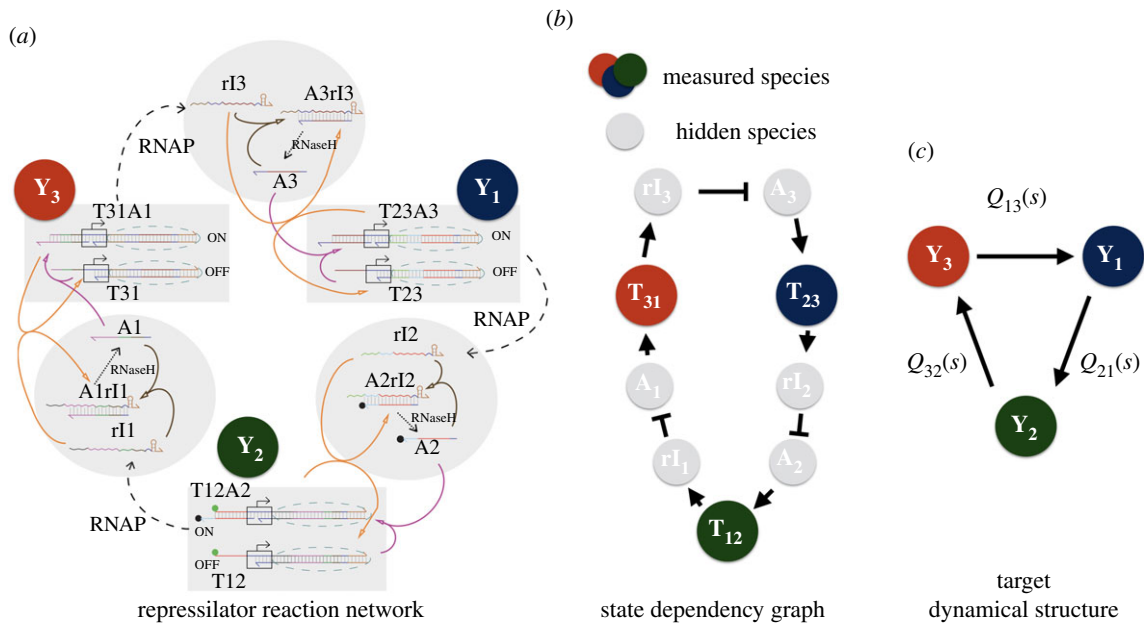


Figure 4. Network representations of a synthetic genelet repressilator. (a) A reaction network using push-arrow reaction notation of the synthetic genelet repressilator. (b) A diagram representing the reaction dynamics in (a) as state dependencies from a nonlinear ODE model in [75]. (c) The dynamical structure of the repressilator (without inputs), with nodes representing measured chemical species and edge weights corresponding to entries in $Q(s)$.

A couple remarks are in order. First, there are two distinct hyperparameters that require optimization in generating the estimates for $Q(s)$ and $P(s)$: (1) n_d , the order of the characteristic polynomial and (2) the selection of the number of subsampled timepoints h_{\max} in given time-series traces. The optimal value of h_{\max} will depend on the dataset, to ensure the condition number of the matrix X must be small. In high-resolution time-series measurements, a small or short subsampled horizon h_{\max} may produce virtually identical data if the transient has a slow rate of change, which can result in ill-conditioning of X . We optimize n_d, h_{\max} to minimize the n -step L1 prediction error across all N_{exp} experimental samples. In practice, we observe that this criterion, by necessity, guarantees an appropriate selection of h_{\max} as well as a lower optimal n_d value. In general, optimization of n_d is non-convex, as this corresponds to an estimate of the model order. In the examples below, we leverage knowledge about the system to inform an initial estimate for n_d and perform a local optimization. A generalized specification of n_d , following the standard calculation of the rank of a Hankel matrix, but generalized for dynamical structure functions, is the subject of future work.

Secondly, at face value, the routine described above may appear similar to a classic transfer function estimation routine, analogous to the tools developed in [60] for MATLAB. The primary difference in algorithm 1 from a standard transfer function estimation procedure for

$$[Q(z) \ P(z)]$$

is that we must impose structural information about $Q(z)$ and $P(z)$ on the estimation process. This results in a structured system identification problem, which typically is non-convex and difficult to initialize properly. Specifically, the diagonal entries of the estimated $Q(z)$ and the off-diagonal entries of $P(z)$ must be exactly 0, as per the premises of [62]. Again, these structural constraints guarantee a unique representation of the dynamical structure function and identifiability of the model. When using standard transfer function estimation tools (in MATLAB's System Identification Toolbox), we found repeatedly over thousands of numerical trials that imposing

structural constraints as model priors sometimes resulted in (A) models with extremely poor n -step prediction capacity (forecasting) or (B) stable transfer function models that were unable to capture transient (divergent) dynamics.

If the dynamics of the system appear to behave like a linear, stable system, MATLAB system identification routines can be adapted to perform reconstruction (e.g. the experimental data for the *in vitro* transcriptional event detector). However, for many *in vivo* gene networks, dynamics are nonlinear or respond to perturbations with transients that are unlike asymptotically stable linear dynamics. The latter observation violates the premise of many transfer function estimation routines in the MATLAB System Identification Toolbox. This motivated the development of a direct estimation algorithm, that mirrors the standard estimation of a discrete-time transfer function model, but where *structural constraints* of $Q(z)$ and $P(z)$ are directly encoded in the formulation of the normal form of equations (lines 19–28, 32 and 36 of algorithm 1).

4. The dynamical structure of an *in vitro* genelet repressilator

We now illustrate the process of data-driven estimation of dynamical structure functions, using experimental data. In this section, we take as a first test case the synthetic genelet repressilator developed by Kim & Winfree [75]. The genelet repressilator consists of three DNA switches that repress one another through indirect sequestration. Specifically, each DNA switch transcribes its mRNA product only when its activator strand binds to complete its T7 RNA polymerase promoter sequence. The RNA product produced from each DNA switch, in turn, acts as an inhibitor to the downstream switch by binding to the downstream switch's DNA activator molecule. Thus, by sequestering the DNA activator from completing the T7 RNA polymerase promoter region, the mRNA product of the upstream switch inhibits activation of the downstream switch. Figure 4a shows the mechanistic design of the genelet switch.

The genelet switch relies heavily on RNase H to degrade any activator–mRNA inhibitor complexes. Without degradation, the binding of activator to mRNA inhibitor is much faster than unbinding and so sequestration is effectively irreversible. Thus, in order for the repressilator to function properly, RNase H must degrade its target substrates sufficiently fast. If RNase H is saturated with high levels of a particular substrate, this slows the degradation of other substrates, creating a crosstalk interaction between competing DNA–RNA complexes.

By performing network reconstruction on the genelet repressilator, we can determine how much crosstalk exists in the biocircuit. Furthermore, we can validate our dynamical structure reconstruction algorithm in an *in vitro* setting, by *deliberately attenuating one of the components to create a gain imbalance*. We can see if the reconstruction process recovers the deliberate imbalance we introduce into the genelet repressilator, even when simply measuring local perturbations of an operating point for a normally oscillatory circuit.

To reconstruct $Q(s)$ and $P(s)$, we performed a single experiment with three perturbations applied in series [24,39–45,62]. To perturb each switch, we pipetted a small perturbative concentration of DNA inhibitor (a DNA analogue of RNA inhibitor). Since DNA is not degradable in a T7 expression system by RNase H, it effectively acts as a step input since it binds to DNA activator and does not degrade. In this way, our perturbation design ensures sufficiency of excitation and independent perturbation of each activator (and downstream switch), thereby satisfying the identifiability conditions in [62] and the persistence of excitation conditions described in [60]. Furthermore, we attenuated the concentration of the third switch T_{31} by 20%, to *create a deliberate gain imbalance for evaluating our reconstruction algorithm*.

A detailed model of the repressilator can be found in the supplement of [75]. Since the derivation is lengthy, it suffices to write the idealized dynamical structure function $Q^a(s)$ of this system, corresponding to the detailed model provided in [75, supplementary §1.6]. The structure is obtained by linearizing the system, transforming into the Laplace domain, eliminating hidden variables to obtain the following:

$$\begin{bmatrix} 0 & 0 & Q_{13}^a(s) \\ Q_{21}^a(s) & 0 & 0 \\ 0 & Q_{32}^a(s) & 0 \end{bmatrix}$$

reflecting the cyclic structure of the system. This represents an idealized model of the system. As stated in theorem 2.1 every entry where $Q_{ij}^a(s) \equiv 0$, the corresponding entry in $Q(s)$ estimated directly from experimental data will be a crosstalk interaction present in the network. Here $Q(s)$ is used to denote $Q^c(s)$, the dynamical structure function estimated directly from data.

The experimental data used to fit $Q(s)$ and $P(s)$ are plotted in figure 5, along with their respective fits. For each row i of $Q(s)$, we use Y_j , $j \neq i$ and U_i as inputs and Y_i as the output for a direct MIMO $p \times 1$ transfer function estimation problem. The impulse response for the convolution kernel $Q(t)$ of the reconstructed $Q(s)$ is plotted in figure 6.

If we compute the corresponding \mathcal{H}_∞ gain of each entry in $Q_{ij}(s)$ and scale by the maximum gain, we obtain

$$\|Q(s)\|_\infty = \begin{pmatrix} 0 & 0.07 & \mathbf{0.73} \\ \mathbf{1.0} & 0 & 0.3 \\ 0.053 & \mathbf{0.17} & 0 \end{pmatrix}.$$

We see significant crosstalk on the edge $Q_{23}(s)$ and minor crosstalk from entries $Q_{31}(s)$ and $Q_{12}(s)$. This crosstalk need not occur simultaneously, since the \mathcal{H}_∞ gain calculates the worst-case or maximum gain over all possible frequencies. With the exception of $Q_{23}(s)$, all other crosstalk entries have strictly smaller \mathcal{H}_∞ gain than the designed edge. Examining the impulse response of the convolution kernel confirms these observations; the crosstalk edge $Q_{23}(t)$ has a larger impulse response than *designed* edge $Q_{32}(t)$. The response of output is normalized per the maximum signal gain achieved, using the technique described in [75], in arbitrary fluorescent units (a.f.u.).

As intended in the design of the experiment, our estimated network model shows a gain imbalance between the *designed* edges $Q_{32}(s)$, $Q_{13}(s)$ and $Q_{21}(s)$. It is well known that in order for a repressilator to stably oscillate [75], it needs to have approximately the same gain along each edge in the network. This example verifies that our reconstruction algorithm can identify important functional dynamics of a genetic circuit; especially for debugging purposes. The linearization scheme is valid, so long as we model fluctuations in dynamics from a nominal initial condition, even if the initial condition is not stable or leads to oscillatory dynamics. Our results here illustrate how linearized models can provide insight into local dynamics. In this simple, controlled dataset, we know we can increase the gain of the edge in $Q_{32}(s)$ by adjusting the binding affinity of the activator DNA with its inhibitor RNA, or by increasing the concentration of the corresponding downstream switch T_{31} . Note that this design insight may not be obvious by direct examination of experimental trajectories of each switch in figure 5. As long as we have an idealized network model, we can measure the deviation from that model $Q^a(s)$ in the network model identified from the data $Q^c = Q(s)$ and identify edges or nodes in our network that need tuning.

5. The dynamical structure of an *in vivo* transcriptional event detector

We now introduce a new transcriptional event detector circuit, one that is designed, built, and constructed for illustrating the use of our dynamical structure estimation algorithm in *in vivo* circuit design. Event detectors are useful because of their ability to perform temporal logic. Making temporal logic decisions enable applications such as programmed differentiation, where the goal is to perform some operation based on combinatorial and temporal sequences of events that dictate cell fate.

So far there are two demonstrations of temporal logic gates: (1) a temporal logic gate that differentiates start times of two chemical outputs [76] and (2) a molecular counter that counts the number of sequential pulses of inducers [77]. Both event detectors use serine integrases to perform irreversible recombination, while [77] demonstrates the use of transcription-based event detecting to perform event counting. The advantage of an integrase-based approach is the persistent nature of DNA-based memory. At the same time, the drawback of integrase-based event detection is that it is limited to one-time use.

By contrast, transcription based event detectors use proteins instead of DNA to encode a memory state [77,78]. The advantage of a transcription-based event detector is that

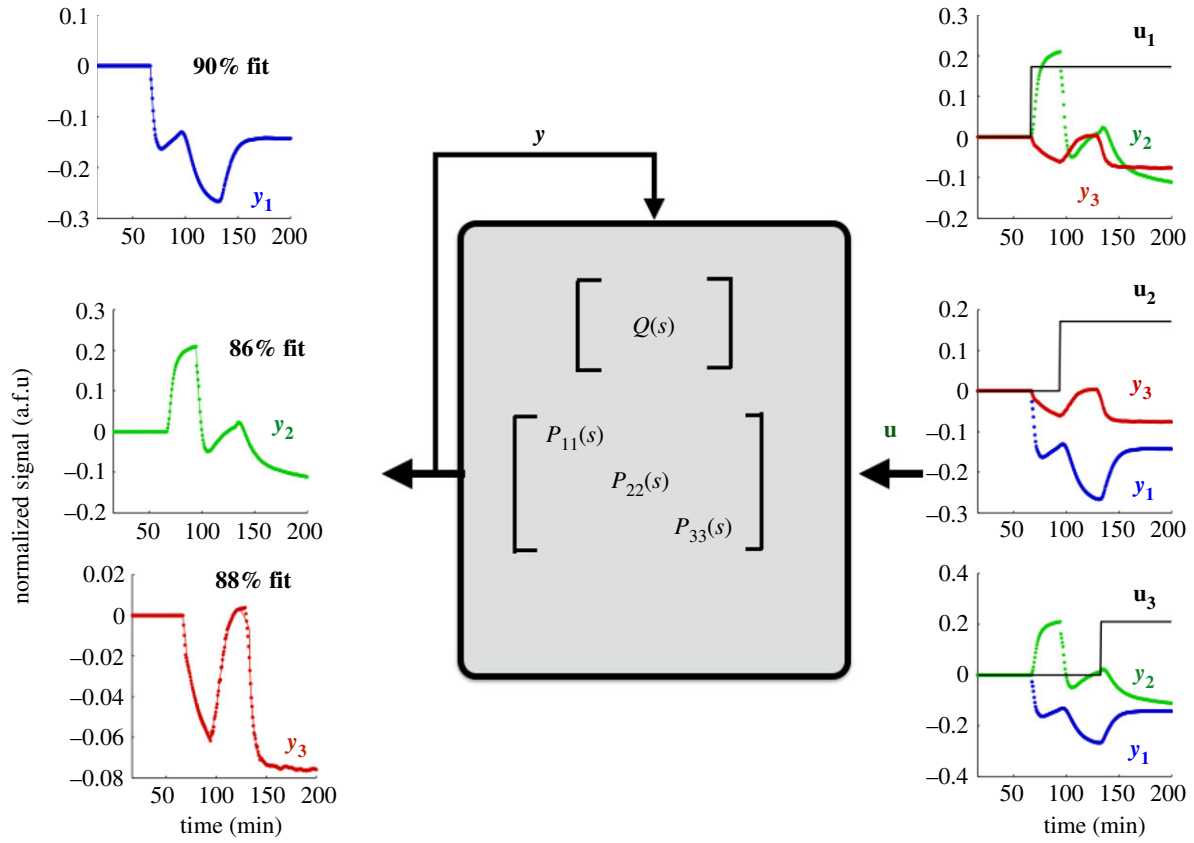


Figure 5. Time-series experimental data from *in vitro* network perturbation experiments of a T7 RNAP genelet repressor. Three outputs are measured simultaneously, y_1 , y_2 and y_3 , corresponding to DNA switches T_{31} , T_{12} and T_{23} . DNA homologues of the RNA inhibitors r_l , $l = 1, 2, 3$ are injected at small concentrations to provide a step input perturbation to the corresponding component Y_j in the genelet circuit.

proteins are labile, since they are diluted through cell growth or can be tagged for degradation. Thus, a transcriptional event detector's memory state can be reset after some period of time. On the other hand, maintaining protein state over multiple generations is metabolically expensive [51] and the dynamics of the circuit can become sensitive to production and growth phase of the cells. Therefore, a transcription based event detector biocircuit must be designed with precise timing, balance of production rates, and carefully tuned gain of each transcriptional regulator. This presents a suitable application for our network reconstruction algorithm.

5.1. Designing a transcriptional event detector

We designed our transcriptional event detector to be made of two constitutively expressed relay genes, AraC and LasR, and an internal toggle switch. The two relay genes transmit the arrival of two distinct induction events (arabinose and HSL) to relay output promoters pBAD and pLas, respectively, which drive production of a fluorescent response in two relay promoters. To record these induction events historically, the output of each relay gene is coupled to one of two combinatorial promoters (pBAD-Lac or pLas-Tet) in a toggle switch. Each combinatorial promoter implements NIMPLY logic, e.g. pBAD-Lac (pLas-Tet) expresses TetR (LacI) only when arabinose (HSL) and AraC (LasR) are present and LacI (TetR) is absent. Thus, when one analyte (e.g. arabinose) arrives, it triggers latching of the toggle switch only if the toggle switch is unlatched to begin with or the prior latching protein state has been diluted out. The relay outputs thus transmit the *current* or *recent* induction event state while the toggle switch maintains the *historical* induction event state.

Depending on the order of arrival of each inducer, we obtain different biocircuit states. Figure 7 details the genetic elements in the event detector biocircuit and the designed component interaction network.

We can write down an idealized model for the event detector (assuming no crosstalk), assuming first-order degradation and production, with Hill functions encoding the NIMPLY logic of each promoter in the memory module:

$$\begin{aligned}
 \dot{x}_1 &= \rho_1 m_1 - \delta_p x_1, \\
 \dot{x}_2 &= \rho_2 m_2 - \delta_p x_2, \\
 \dot{x}_3 &= \rho_3 m_3 - \delta_p x_3, \\
 \dot{x}_4 &= \rho_4 m_4 - \delta_p x_4, \\
 \dot{m}_1 &= \frac{k_1(k_1 + u_1/k_{M,u1})}{(1 + u_1/k_{M,u1})} - \delta_m m_1, \\
 \dot{m}_2 &= \frac{k_2(k_1 + u_1/k_{M,u1})}{(1 + x_3/k_{M,3} + u_1/k_{M,u1})} - \delta_m m_2, \\
 \dot{m}_3 &= \frac{k_3(k_1 + u_2/k_{M,u2})}{(1 + x_2/k_{M,2} + u_2/k_{M,u2})} - \delta_m m_3, \\
 \dot{m}_4 &= \frac{k_4(k_1 + u_2/k_{M,u2})}{(1 + u_2/k_{M,u2})} - \delta_m m_4
 \end{aligned} \quad (5.1)$$

and $y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} [x^T \ m^T]^T$,

where the measured outputs of the system are $y_i = x_i$, $i = 2, 3$, ρ_i is the translation rate of m_i into x_i , δ_p is the effective dilution rate of x_i , $i = 1, \dots, 4$, δ_m is the combined dilution and degradation rate of m_i , $i = 1, \dots, 4$, k_M , u_i is the Michaelis constant

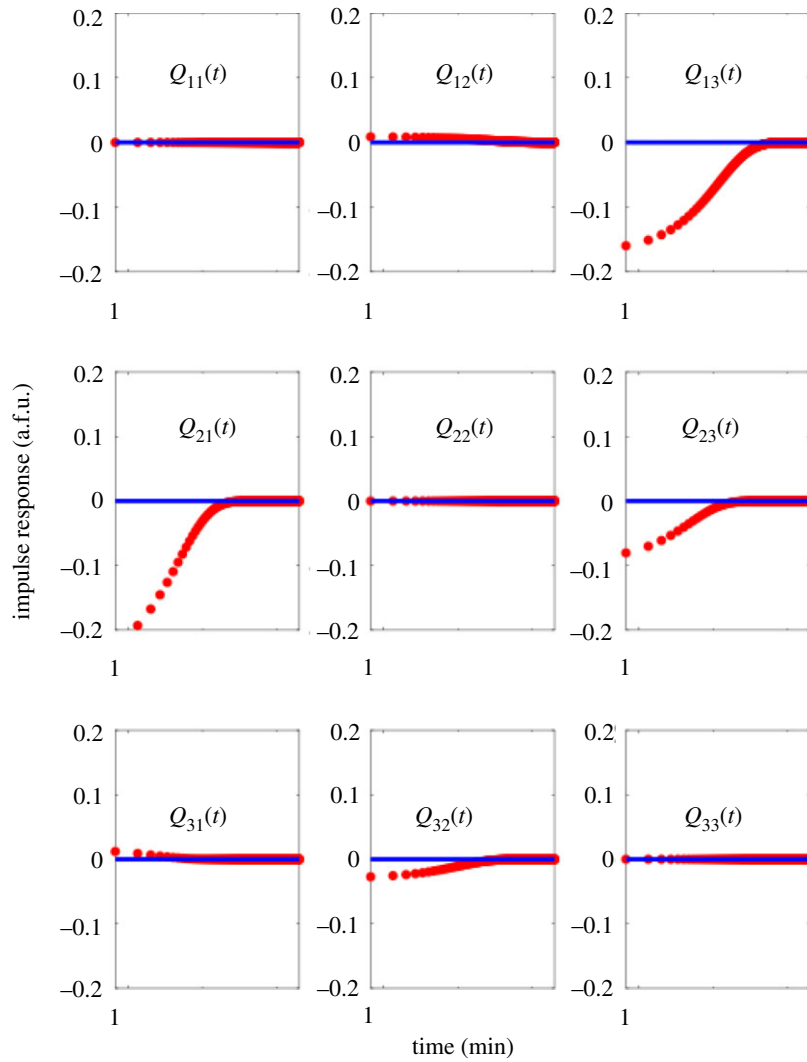


Figure 6. Impulse response of the estimated convolution kernel $Q(t)$ matrix. $Q(s)$ is estimated directly from experimental data, transformed into the frequency domain, and simulated in time for $t = 0$ to $t = 300$ min. The x -axis is plotted in log scale, to visualize fast, transient edge dynamics that happen upon impulse stimulation of a given edge.

for u_i , k_i is the leaky catalytic transcription rate, k_i is the catalytic transcription rate for m_i , and u_1 , u_2 are arabinose and HSL, respectively.

Again, the dynamical structure function for this system is calculated by linearizing the system about a nominal initial condition, (x_0, m_0) , taking a Laplace transform and solving out the hidden variables m_1, \dots, m_4 . We present a simplified case here, assuming algebraic symmetry of the parameters $k_i = k$, $\rho_i = \rho$, $k_{M,i} = k_M$ as it does not qualitatively change the structure of $(Q(s), P(s))$. We obtain

$$Q^a(s) = \begin{bmatrix} 0 & Q_{12}(s) \\ Q_{21}(s) & 0 \end{bmatrix}$$

and

$$P^a(s) = \begin{bmatrix} P_{11}(s) & 0 \\ 0 & P_{22}(s) \end{bmatrix},$$

where $P_{ii}(s) = \rho / (\delta_m + s)(\delta_p + s)$ for $i = 1, 2$ and

$$Q_{12}(s) = \frac{-k\rho(k_i + u_1/k_M)}{k_M(\delta_m + s)(\delta_p + s)(u_1/k_M + x_3/k_M + 1)^2}$$

and

$$Q_{21}(s) = \frac{-k\rho(k_i + u_2/k_M)}{k_M(\delta_m + s)(\delta_p + s)(u_2/k_M + x_2/k_M + 1)^2}.$$

In the absence of protein degradation, $Q_{12}(s)$ and $Q_{21}(s)$ can be approximated with first-order SISO transfer functions. These expressions for $Q(s)$ and $P(s)$ are for the idealized dynamical structure function of the *alternative* system. Notice that $Q_{12}(s)$ and $Q_{21}(s)$ are strictly negative transfer functions, indicating the repression present in an idealized simulation of the event detector circuit. This is the intended *dynamical network* structure of the event detector, in the absence of all genetic crosstalk or context effects.

Depending on the abundance of transcription factors such as LacI, TetR and AraC, as well as commonly shared transcriptional and translational proteins, the *actual* dynamical structure function $Q^c(s)$ may not exhibit monotonic repression or may even unveil unwanted interactions. We can investigate these interactions under a range of conditions with dynamical structure estimation.

We constructed a biological implementation of the event detector, using the design specified in figure 7. The logical components containing the relays and the memory module

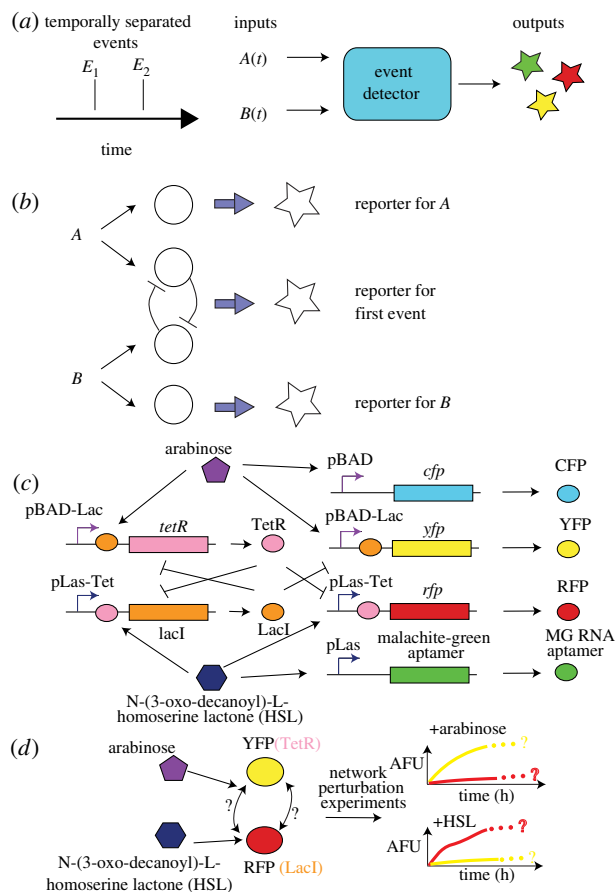


Figure 7. (a) Left: We design an event detector to determine the identity and relative ordering of two events E_1 and E_2 occurring within a finite time horizon. (b) A schematic showing the logic of the circuit for the event detector. Arrival of event type A triggers transient reporter for A (top) and latching of the toggle in an A-dominant state as a memory state. Similarly, arrival of event type B triggers transient reporter for B (bottom) and latching of the toggle in a B-dominant state as a memory state. (c) A diagram showing the synthetic biocircuit parts used to implement the network architecture in (b). (d) The arabinose and HSL inducers independently perturb distinct elements of the memory module in the event detector; a network model of the dynamic graph of the event detector can be reconstructed using dynamical structure function reconstruction experiments.

were encoded on to a plasmid vector with a kanamycin resistance marker and a ColE1 (high copy) replication origin. The fluorescent reporter elements with the relay promoters and readouts for the toggle switch were encoded on a plasmid vector with chloramphenicol resistance and the p15 replication origin.

5.2. Event detector latching experiments

We evaluated the performance of our transcriptional event detector circuit using a temporal logic test. A standard temporal logic experiment for any two-input event detector is to evaluate the effect of varying the order of presentation of two input signals. In one test, we present the first input, arabinose, for 7.5 h, followed by induction of the second input, a homoserine lactone (HSL) quorum sensing molecule to activate the pLas-Tet promoter. In the second test, we swap the order of the inputs, presenting HSL quorum sensing molecule to the event detector for 7.5 h, then present arabinose inducer as a second input. Both tests evaluate the ability of the memory

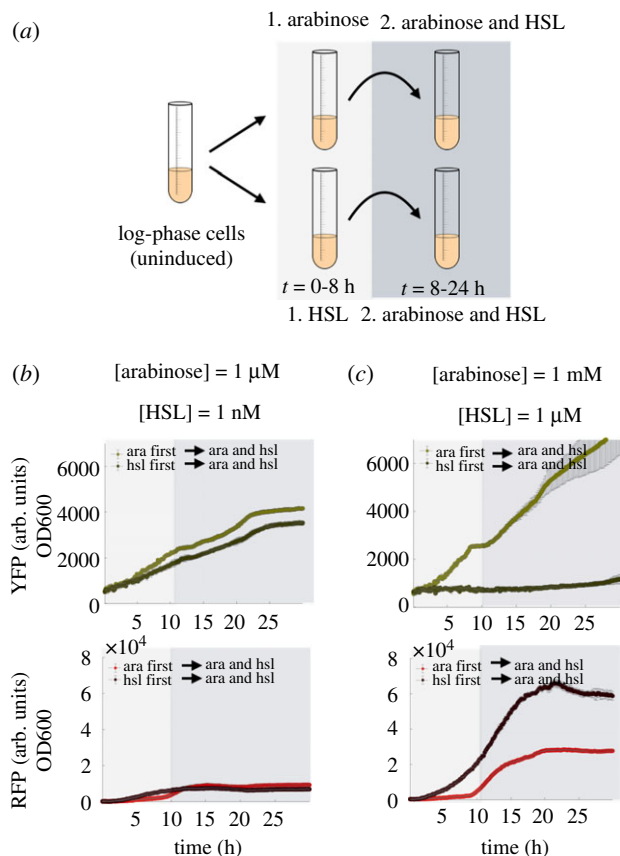


Figure 8. A plot of data from *in vivo* plate reader experiments, testing the temporal logic properties of the event detector diagrammed in figure 7. Note that at $1 \mu\text{M}$ HSL and 1mM arabinose, the event detector functions properly, expressing different levels of YFP and RFP depending on the order of arrival of arabinose and HSL. At 1nM HSL and $1 \mu\text{M}$ arabinose induction concentrations, the temporal logic properties of the event detector are completely abolished.

module of the event detector to latch in the correct state in response to the first input, followed by a challenge to ignore the second input signal while the relays detect and read out the second input signal. The data for both of these *in vivo* tests are plotted in figure 8*b,c*.

The event detector showed the correct latching response in all tests at standard maximum induction concentrations of arabinose (1mM) and working induction concentrations of $1 \mu\text{M}$ HSL. For example, figure 8*c* shows that when the event detector is given arabinose followed by HSL, it generates the correct fluorescent response of YFP, with lower expressions level of RFP. Conversely, when we add HSL first, followed by arabinose, RFP signal ramps up immediately beginning as early as 1–2 h after induction while YFP expression is abolished to background levels.

We tested a variety of combinations of high and low concentrations for arabinose and HSL. When the concentration of HSL was decreased to 1nM , we observed consistent leaks in the memory module in either the YFP channel or the RFP channel. Decreasing arabinose down to $1 \mu\text{M}$ still allows for latching of high YFP expression, but in the presence of $1 \mu\text{M}$ HSL, any arabinose latching is reversed by HSL induction (data not plotted). Conversely, when we attenuate HSL induction to 1nM , HSL does not prevent arabinose from reversing a HSL latch on the memory module; see figure 8*b*. This leak is significant enough in the 1nM HSL induction level that the difference in signal between the arabinose–HSL induction

scenario versus the HSL–arabinose induction scenario vanished. This temporal logic response profile is evident of a glitch in the event detector circuit that occurs at lower HSL and arabinose concentrations.

5.3. Network reconstruction experiments to debug circuit failure

We conducted 4 *in vivo* network reconstruction experiments (2 inducers versus 2 concentrations), recording time-series data of the memory module relay elements, YFP and RFP. The memory module is designed using two hybrid promoters, so from a design standpoint, verification of the memory module was most critical. The arabinose inducer targets the pAra-Lac promoter, while the HSL inducer targets the pLas-Tet promoter (see electronic supplementary material for sequences).

As shown in the model (5.1) of the event detector, the actual event detector we constructed exhibits nonlinear response. However, for any one parametric concentration regime, e.g. at a fixed arabinose or HSL concentration, the response of the system behaves similar to that of a linear system. Thus, we estimated a dynamical structure function for both conditions of the reconstruction experiment. The one-step accuracies in fitting dynamical structure models to the low gain condition (1 μ M arabinose, 1 nM HSL) and high gain condition (1 mM arabinose and 1 μ M HSL) were 99.996% and 99.995%, respectively.

As in the case of the genelet repressilator, we can plot a dynamical network graph for the *in vivo* event detector to understand how the memory module components labelled by YFP and RFP, representing TetR and LacI, respectively, interact with each other. A movie visualizing the dynamics of the edges of the graph is available for download (see electronic supplementary material). Each edge represents the convolution kernel response of the edge to an impulse applied to that input. All responses are superimposed to form a dynamical graph. Snapshots of the graph are plotted in figure 10, while time-lapse responses of the weights of each edge are plotted in figure 9. Again as with the repressilator, we can see that the regulatory nature of edges in the event detector's memory module manifests as two edges with negative or positive values indicating repression or activation, respectively.

The reconstructed network of our transcriptional event detector reveals the functional relationship between states in the circuit at different concentration regimes. At lower concentrations of arabinose and HSL, the reconstructed transcriptional event detector network reveals functional cause of failed circuit latching. Both edges in the memory module did not repress their target promoters as intended, while the pLas-Tet promoter appears to enact a much higher gain of activated expression from HSL induction than does the activated expression of the pAra-Lac promoter in response to arabinose.

In the high gain setting, where arabinose is induced at 1 mM and HSL is induced at 1 μ M, we see that the memory module exhibits the proper mutually repressing motif characteristic of the genetic toggle switch up after the arrival of the HSL inducer. The repression in both edges steps up their gain as t approaches 4 h, which is roughly the time when we see a plateauing of production in the RFP signal in figure 8c. From our reconstruction model, we can see that the edges are well balanced at the higher concentration of inducers. At the low gain of inducers, the network is completely inactive, even though the genetic sequence of the circuit is the same. This example shows that our network verification algorithm can

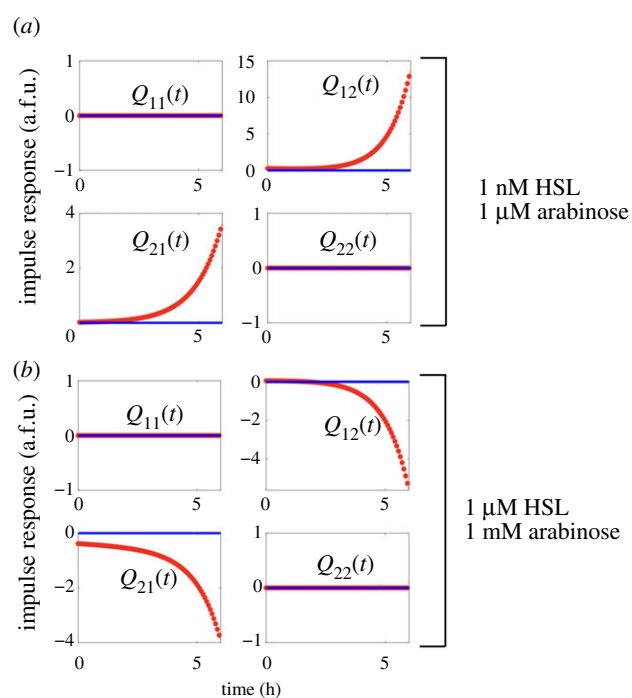


Figure 9. Impulse response of the estimated convolution kernel $Q(t)$ matrix when the event detector biocircuit is induced with (a) 1 nM HSL and 1 μ M arabinose or (b) 1 μ M HSL and 1 mM arabinose. $Q(s)$ is estimated directly from experimental data, transformed into the frequency domain, and simulated in time as a function of hours from arrival time of an inducer input.

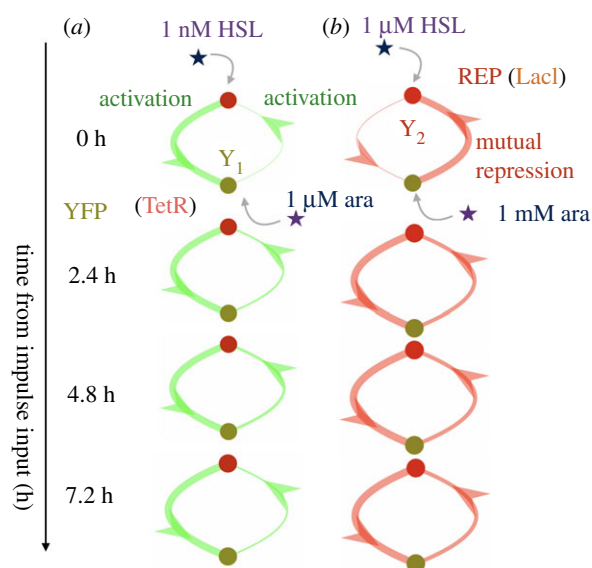


Figure 10. A visualization of the impulse response of the estimated convolution kernel $Q(t)$ matrix when the event detector biocircuit is induced with low (a) versus high (b) concentrations of arabinose and HSL inducer. The width of edges in this graph coincide with the magnitude of the impulse response, while colouring is red if the sign of the impulse response for a given edge is negative (repression) and green if the given edge is positive (activation).

be used to determine the conditions, or the performance envelope, under which the circuit is functioning properly. Even though the underlying model of our system is a linear approximation to a nonlinear system, we can test the system at multiple initial conditions, operating points, or equilibria, to quantify network behaviour of the system locally. Taken in aggregate, these can provide a

parameterized view of how the network behaves over a range of experimental conditions (figure 10).

6. Conclusion

The dynamical structure function models the dependencies among measured states. It is a flexible representation of network structure that naturally adapts to the constraints imposed by experimental measurement. Since identifiability conditions of the dynamical structure function have been well characterized, appropriate experimental design can ensure that the process of network reconstruction produces a sensible answer.

In this work, we introduced a network reconstruction algorithm and a code base for reconstructing the dynamical structure function from data, to enable discovery and visualization of graphical relationships in a genetic circuit diagram as *time-dependent functions* rather than static, unknown weights. We proved a theorem, showing that dynamical structure functions can provide a data-driven estimate of the size of crosstalk fluctuations from an idealized model. We then illustrated these findings with numerical examples. Next, we used an *in vitro* genetic circuit, deliberately tuned with gain imbalance, to validate our algorithm on experimental data. Finally, we built a new *E. coli* based transcriptional event detector and showed how estimation of the dynamical structure reveals active and inactive network states, depending on inducer concentration. These results show how the dynamical structure function characterizes the operational or active network. They also provide a route for future study of relationships between environmental parameters, active network dynamics, and biocircuit performance.

7. Experimental methods

All plasmids were constructed using either Golden Gate assembly [79] or Gibson isothermal assembly [80] in *E. coli*. Plasmids were sequence verified in JM109 cloning strains and transformed into the strain MG1655 Δ Lacl, provided as a courtesy by R. J. Krom and J. J. Collins. The event detector was transformed as a two-plasmid system with kanamycin and chloramphenicol selection. All *in vivo* experiments were carried out with $n = 2$ replicates using MatriPlates (Brook Life Science Systems MGB096-1-2-LG-L) 96 square-well glass bottom plates at 29°C in a H1 Synergy Biotek plate reader using 505/535 nm and 580/610 nm excitation/emission wavelengths. Cell density was quantified with optical density at 600 nm.

For *in vitro* experiments, all genelet repressilator reconstruction experiments were carried out at 37°C in a Horiba spectrofluorometer with 1 min readout times, using Rhodamine Green, TYE 563 and Texas Red fluorophores with 10 nm monochromator excitation and emission bands centred at 502/527, 549/563 and 585/615 nm, respectively. All event detector network reconstruction reactions were performed using 500 μ l reaction volumes in transformed *E. coli*, grown in square well glass-bottom plates using MatriPlates (Brook Life Science Systems MGB095-1-2-LG-L) with Luria-Bertain rich media broth at 29°C.

Data accessibility. All data files, network reconstruction code and visualization scripts can be obtained from the GitHub repository <https://github.com/YeungRepo/NetworkRecon>.

Authors' contributions. E.Y. wrote the paper. E.Y., J.G., Y.Y., J.K. and R.M.M. edited drafts of the paper. E.Y. and J.K. designed and carried out experiments and processed experimental data. E.Y. performed analysis and modelling. J.G. and R.M.M. secured research funding. R.M.M. supervised the research process.

Competing interests. We declare we have no competing interests.

Funding. This work was supported by the Engineering and Physical Sciences Research Council, the Luxembourg National Research Foundation, Air Force Office of Scientific Research, grant no. FA9550-14-1-0060, the Defense Advanced Research Projects Agency, grant nos. HR0011-12-C-0065 and FA8750-19-2-0502, the Army Research Office Young Investigator Program, grant no. W911NF-20-1-0165, the National Science Foundation, grant no. 1317291, and the John and Ursula Kanel Charitable Foundation.

Acknowledgements. We would like to acknowledge Sean Warnick, Shara Balakrishnan, Vipul Singhal and Anandh Swaminathan for insightful conversations on network reconstruction algorithms. We would like to especially thank Shara Balakrishnan for editorial comments through the writing process. We would like to thank and acknowledge Zachary Sun, Victoria Hsiao, Ophelia Venturelli, Clarmyra Hayes, Emmanuel de los Santos and Joe Meyerowitz for guidance with experimental techniques.

Appendix A. Supplementary information

A.1. Experimental methods for circuit preparation, assembly and testing

A.1.1. The repressilator genelet circuit

The DNA sequences for the T31, T12, T23 switch were obtained as a gift from the Winfree lab, mirroring the design identically of the repressilator genelet circuit used in [75]. Oligonucleotides were ordered with functionalized fluorophores or quenchers, corresponding to the original design of the genetic repressilator. DNA sequences were suspended in Tris-EDTA buffer for primary stock storage, while all genelet switches T12, T31, T23 added at concentrations of 75 nM, 75 nM and 60 nM, respectively to match previous tuning experiments to balance the repressilator, with 7.5 mM working concentration of mono-NTP solution, 24 mM MgCl₂, and 1 \times T7 expression system buffer.

DNA analogues of RNA inhibitors were added to sequester DNA activator signal from the switches as an effective step input perturbation to each node. The switches produced a RNA signal that was designed to interfere with formation of a complete promoter region of the next downstream switch in the repressilator circuit. Adding DNA served as a step perturbation to the corresponding switch. Each DNA moiety added thus had the effect of an activator. Activator DNA molecules A1, A2 and A3, each containing Iowa Black quencher were added at 75 nM, 80 nM and 75 nM working concentration at 20 min from the onset of the reaction, to determine the maximum range of quenching. At 58 min, we added 0.7 μ l of pyrophosphatase, 3 μ l of T7 RNA Polymerase and 2.2 μ l of RNase H to achieve identical working concentrations to those described in [75].

A.2. The transcriptional event detector circuit

The transcriptional event detector circuit, as illustrated in figure 7 in the main text, is composed of four distinct gene expression cassettes that define the regulatory logic of the circuit and four distinct gene expression cassettes that generate the fluorescent reporter elements of the circuit. Each gene cassette defines a transcriptional unit, with a promoter element, an

RBS, a coding sequence, and a terminator sequence. Each gene cassette was cloned using a 5 part Golden Gate assembly, with a type II BsaI restriction enzyme and overhang sequences from [81,82]. Each assembled gene cassette was cloned in JM109 *E. coli* cloning strains and sequence verified at Eurofins Genomic, by Sanger sequencing. Assembled plasmids were engineered to enable a second stage Golden Gate assembly, using the BbsI type II restriction enzyme, and assembled to either (1) form a master regulatory logic plasmid (pEY15K), comprised of four distinct gene expression cassettes driving transcription factor or allosteric response or (2) form a master reporter plasmid comprised of four distinct reporter elements (pEY14C). Both stage 2 assembled regulatory logic and reporter plasmids were sequence verified using Sanger sequencing (Eurofin Genomics) and transformed into MG1655ΔLacI (a gift from the Collins laboratory). The sequences for all individual plasmids and the circuit plasmids are listed in table 1.

Appendix B. Sequences of genetic circuit components

The sequences for all genetic components and circuits for the event detector circuit are listed in table 1. All genelet repressor sequences are identical to the sequences used and listed in [75]. All ribosome binding site (RBS) sequences were derived from the bicistronic design (BCD) ribosome binding site library [83], while all terminator sequences were drawn from the synthetic terminator library characterized in [84].

Appendix C. Quantifying crosstalk in biochemical reaction networks

A common way that crosstalk arises in biochemical reaction networks is when species compete for commonly shared enzymes. When this occurs, the sequestration of an enzyme by one competing species makes the enzyme less accessible to other competing species. For example, when two mRNA are competing for a single ribosome, the binding of one mRNA to the ribosome during translation makes it less accessible to other mRNA. At the core of any such crosstalk is a sudden increase in the dependency of one biochemical state on another. Though enzyme loading may be a common source of crosstalk, such interactions can be modelled at a higher level of abstraction, namely how the dynamics of a given state are affected by the concentration fluctuations of other states.

Nearly every synthetic gene network implements causal dependencies among states. Often, these ‘designed’ interactions take the form of transcription factor binding, sense-anti-sense mRNA regulation, and sequestration events. In practice, every physical system exhibits trajectories that are a mixture of the consequences of both interaction types: designed and crosstalk interactions. Throughout the course of this paper, we will denote the physical system of interest in our models as

$$\left. \begin{aligned} \dot{y} &= f_y^c(y, x_h, u), & y(0) &= y_0 \\ \dot{x}_h &= f_{x_h}^c(y, x_h, u), & x_h(0) &= x_{h,0} \end{aligned} \right\} \quad (C1)$$

and

$$y = [\mathbf{I}_{p \times p} \quad 0] \begin{bmatrix} y \\ x_h \end{bmatrix}.$$

To quantify crosstalk in such systems, we can compare the dynamics of system (C1) against the dynamics of a reference or alternative system that is free of crosstalk. Such a reference system will still retain the *desired* interaction dynamics and reflects the idealized model often used to design a synthetic gene network, e.g. the feed forward loop model in example 2.1.1. Moreover, it can represent the desired behaviour of the system in a regime where the magnitude of crosstalk effects are supposed to be minimal or engineered in such a way that they are suppressed [7]. We write the reference system as

$$\left. \begin{aligned} \dot{y} &= f_y^a(y, x_h, u), & y(0) &= y_0 \\ \dot{x}_h &= f_{x_h}^a(y, x_h, u), & x_h(0) &= x_{h,0} \end{aligned} \right\} \quad (C2)$$

and

$$y = [\mathbf{I}_{p \times p} \quad 0] \begin{bmatrix} y \\ x_h \end{bmatrix}.$$

Remark C.1. For the comparison between the alternative and crosstalk system to be fair, it is important that (C4) satisfies *internal equivalence* [85]. Specifically, we will suppose that any parameters or dynamics unassociated with crosstalk, e.g. interaction dynamics, catalytic reactions, or anabolic reactions with no loading effects, are held fixed. Thus, as we compare the behaviour of both systems, any differences in the hidden state x_h or output y dynamics are purely due to effects of crosstalk.

With the definition of an alternative system in place, it becomes possible to reason about the size of crosstalk, by comparing the dynamics of both systems. In particular, we can develop a rigorous notion for describing the amount of crosstalk arising from the difference of trajectories in both systems.

Definition C.2 (Crosstalk trajectory). Consider two systems, a crosstalk system and an alternative or reference system, initialized from the same initial condition $x(0)$. For each initial condition $x(0) = (y(0), x_h(0)) \in \mathbb{R}^n$ and input trajectory $u(t)$, we define the *crosstalk trajectory* $\zeta(t)$ as

$$\zeta(t) = x^a(t) - x^c(t).$$

The crosstalk trajectory is a time-evolving vector that describes the deviation of the physical system (subject to crosstalk) from the reference system’s trajectory. With this notion of crosstalk, we can also make precise the concept of crosstalk between states. We note that in writing the following quantity of interest $(\partial/\partial x_j)\zeta_i$, it is with a slight abuse of notation, since $\zeta_i(x^a(t), x^c(t))$. Mathematically, we are computing the j th partial derivative of each term in $\zeta_i = x_i^a(t) - x_i^c(t)$. Thus, to be clear, when we write $(\partial/\partial x_j)\zeta_i$, it will be implicit that we mean $(\partial/\partial x_j^a)x_i^a(t) - (\partial/\partial x_j^c)x_i^c(t)$.

Definition C.3 (Directed crosstalk). Given an initial condition of $(x(0), y(0))$ and input trajectory $u(t)$ we say that a chemical species x_j exerts a crosstalk effect on chemical species x_i if the i th component of the crosstalk trajectory $\zeta(t)$ has non-zero partial derivative

$$\frac{\partial}{\partial x_j} \zeta_i(t) \neq 0,$$

for some initial condition of $(x(0), y(0))$ and input trajectory $u(t)$. In general, we will refer to $(\partial/\partial x_j)\zeta_i(t)$ as the *crosstalk sensitivity* of x_i to x_j .

Table 1. Table of genetic sequences for all parts used to make the transcriptional event detector circuit.

sequence ID	sequence description	DNA sequence
pAra-Lac	hybrid promoter	CATAGCATTTTATCCATAAGATTAGCGGATCCTAAGCTTTACAA TTGTGAGCGCTCACAATTATGATAGATCAATTGTGAGCGGATA ACAATTTACACA
BCD2	RBS	GGGCCCAAGTTCACCTAAAAGGAGATCAACAATGAAAGCAATT TTCGTAAGTAAACATCTTAATCATGCAGGGGAGGGTTCTAATG
TetR	transcription factor	ATGCTAGATTAGATAAAAAGTAAAGTGATTAACAGCGCATTAGAG CTGCTTAATGAGGTCGGAATCGAAGGTTTAAACAACCCGTAACCT CGCCAGAAAGCTAGGGTAGAGCAGCCTACATTGTATTGGCATG TAAAAAATAAGCGGGCTTGTCTGACGCCTTAGCCATTGAGATGT TAGATAGGCACCATACTCACTTTTGCCTTTAGAAGGGGAAAGCT GGCAAGATTTTTACGTAATAACGCTAAAAGTTTTAGATGTGCTTTA CTAAGTCATCGCGATGGAGCAAAAGTACATTTAGGTACACGGCCTA CAGAAAAACAGTATGAAACTCTGAAAATCAATTAGCCTTTTTATGC CAACAAGGTTTTTCACTAGAGAATGCATTATATGCACTCAGCGCTGT GGGGCATTTTACTTTAGGTTGCGTATTGGAAGATCAAGAGCATCAAG TCGTAAGAAGAAAGGGAAACCTACTACTGATAGTATGCCGCCA TTATTACGACAAGCTATCGAATTATTTGATCACCAAGGTGCAGAGCCA GCCTTCTATTGCGCCTTGAATTGATCATATGCGGATTAGAAAAACAA CTAAATGTGAAAGTGGGCTCTGCAGCAAACGACGAAAACCTACGCTTT AGCAGCTTAA
ECK120033736	terminator	AACGCATGAGAAAAGCCCGGAAGATCACCTCCGGGGGCTTTTT ATTGCCG
BCD9	RBS	GGGCCCAAGTTCACCTAAAAGGAGATCAACAATGAAAGCAATTTTC GTAAGTAAACATCTTAATCATGCAGAGGAGTCTTTCT
AraC	transcription factor	ATGCAATATGGACAATTGGTTTCTCTCTGAATGGCGGGAGTATGAA AAGTATGGCTGAAGCGCAAAATGATCCCTGCTGCCGGGATACTCG TTAATGCCATCTGGTGGCGGTTTAAAGCCGATTGAGGCCAACG GTTATCTGATTTTTTATCGACCGACCGCTGGGAATGAAAGGTTATA TTCTCAATCTCACCATTGCGGTCAGGGGGTGGTAAAATCAGGG ACGAGAATTTGTTGCCGACCGGGTATATTTGCTGTTCCCGCCAG GAGAGATTCACACTACGGTCTCATCCGAGGCTCGCAATGGTAT CACCAGTGGGTTACTTTCGTCGCGCGCTACTGGCATGAATGGCT TAACTGGCCGTCAATATTGCCAATACGGGGTCTTTCCGCCGGATGA AGCGCACCAGCCGATTTACGCGACTGTTGGGCAATCATTAAACG CCGGGCAAGGGGAAGGGCGTATTCGGAGCTGCTGGCGATAAATCT GCTTGAGCAATTGTTACTGCGGCGCATGGAAGCGATTAACGAGTCCG TCCATCCACCGATGGATAATCGGGTACGCGAGGCTTGTGAGTACATCA GCGATCACCTGGCAGACAGCAATTTGATATGCCAGCGTCCGACAGC ATGTTTGCTTGTCCGCTCGCTGTGCATCTTTCCGCCAGCAGTT AGGGATTAGCGTCTTAAGCTGGCGGAGGACCAACGTATCAGCCAGGC GAAGCTGCTTTGAGCACCACCGGATGCCTATCGCCACCGTCCGGTCCG CAATGTTGGTTTTGACGATCAACTCTATTCTCGGGGATTTAAAAAATG CACCAGGGCCAGCCGAGCGAGTTCCGTGCCGGTTGTGAAGAAAAAGT GAATGATGTAGCCGCAAGTTGTCATAA
ECK120029600	terminator	TTACGCCAAAAAAGTAAAGACCGCCGGTCTTGTCCACTACCTTGAGTA ATGCGGTGGACAGGATCGCGGTTTT CTTTCTCTCTCAA

(Continued.)

Table 1. (Continued.)

sequence ID	sequence description	DNA sequence
pLas-Tet	hybrid promoter	TTCTTCGAGCTAGCAAGGGTCCGGGTTACCCGAAATCTA TCTCATTGCTAGTTATAAAATTATGAAATTTGCGTAAATTC CTATCAGTGATAGAGATTGAGAAGC
BCD10	RBS	GGGCCCAAGTTCACCTAAAAGGAGATCAACAATGAAAGCA ATTTTCGTAAGTAAACATCTAATCATGCGGAGGATCGTTTCTA
Lacl	transcription factor	ATGAAACCAAGTAAACGTTATACGATGTCGAGAGTATGCCGGTG TCTTTATCAGACGTTTCCCGCTGGTGAACCAGGCCAGCC ACGTTTCTGCGAAAACGCGGAAAAAGTGAAGCGCGATG GCGGAGCTGAATTACATCCCAACCGCTGGCACAACAATG GCGGGCAAACAGTCTGTTGCTGATTGGCGTTGCCACCTCCAGT CTGGCCCTGCACGCGCTCGCAAATTGTCGCGCGATATAA TCTCGCGCCGATCAACTGGGTGCCAGCTGGTGGTGTGATG GTAGAACGAAGCGCGCTGAAGCCTGTAAGCGGCGGTGCAC AATCTTCTCGCGCAACGCGTCACTGGGCTGATCATAACTATCC GCTGGATGACCAAGGATGCCATTGCTGTGAAGCTGCCTGCACT AATGTTCCGCGTTATTTCTTGTCTGTCTGACCAAGACCCATC AACAGTATTATTTCTCCATGAGGACGGTACGCGACTGGGCGT GGAGCATCTGGTCGATTGGGTCACAGCAAATCGCGCTGTTAG CGGGCCATTAAGTTCTGTCTCGGCGCTGCTGCTGGCTGGC TGGCATAAATATCTCACTCGCAATCAAATTCAGCCGATAGCGGAAC GGGAAGGCGACTGGAGTGCCATGTCCGTTTTCAACAACCATG CAAATGCTGAATGAGGGCATCGTCCCACTGCGATGCTGGTTGC CAACGATCAGATGGCGTGGGCGCAATGCGCGCATTACCGAGT CCGGGCTGCGCGTTGGTGGGATATCTCGTAGTGGGATACGAC GATACCGAGGACAGCTCATGTTATATCCCGCCTAACCCATCA AACAGGATTTTCGCTGCTGGGCAAACAGCGTGGACCGCTTG CTGCAACTCTCAGGGCCAGGCGTGAAGGGCAATCAGCTGTTG CCCGTCACTGGTGAAGAAAAACCCCTGGCGCCCAATACGC AAACCGCTCTCCCGCGGTTGGCGATTGTTAATGAGCTGGC ACGACAGGTTCCCGACTGGAAGCGGGCAGGCAACGACGA AAACTACGCTTAGCAGTTGA
ECK120015440	terminator	TCCGCAATTAAGAAAGCGGCTAACACCGCCGCTTTTTTACGCTGCA
pLas	LasR promoter	GCATTGCTGTTCTGATGGCTAGCTCAGTCTAGTACAATGCAAGC
BCD1	RBS	GGGCCCAAGTTCACCTAAAAGGAGATCAACAATGAAAGCAATTTTCG TACTGAAACATCTAATCATGCACAGGAGACTTTCTAATG
LasR	transcription factor	ATGGCCTTGGTTGACGTTTCTTGTGAGCTGGAACGCTCA AGTGGAAAATTGGAGTGGAGCCATCCTCCAGAAGATG GCGAGCGACCTTGGATTCTCGAAGATCCTGTTCCGCTG TTGCCTAAGGACAGCCAGGACTACGAGAAGCCTTCATC GTCGGCAACTACCGCCGCTGGCGGAGCATTACGA CCGGGCTGGCTACGCGGGTGCACCCGACGGTCAGTC ACTGTACCCAGAGCGTACTGCCGATTTCTGGGAACCGTC CATCTACCAGACGGAAGCAGCAGGTTCTTCGAGGAA GCCTCGGCCGCGCTGGTGTATGGGCTGACCATGCCG

(Continued.)

Table 1. (Continued.)

sequence ID	sequence description	DNA sequence
		CTGCATGGTGTCTCGCGGCAACTCGGCGCTGAGCCTC AGCGTGGAAGCGGAAAACCGGCCGAGCCAACCGTTTC ATAGAGTCGGTCTGCCGACCTGTGGATGCTCAAGGACT ACGCACTGCAAAGCGGTGCCGACTGGCCTTCAACATC CGGTGAGCAAACCGGTGGTCTGACCAGCCGGGAGAAGG AAGTGTTCAGTGGTGCCTATCGGCAAGACCAAGTTGGGA GATATCGGTTATCTGCAACTGCTCGGAAGCCAATGTGAATT CCATATGGGAAATATTCGGCGGAAGTTCGGTGTGACCTCCC GCCGCGTAGCGGCCATTATGGCCGTAATTGGGTCTTATT ACTCTCTAATAA
ECK120010799	terminator	GTTATGAGTCAGGAAAAAGGCGACAGAGTAATCTGTCGCC TTTTTCTTGTCTTCTT
CFP	CDS	ATGAGTAAAGGAGAAGAACCTTCTACTGGAGTTGTCCTCAATC TTGTTGAATTAGATGGTGTGTAATGGGCACAATTTCTGTG AGTGGAGAGGGTGAAGGTGATGCAACATACGAAAACTTACC CTTAAATTTATTTGCACTACTGGAAAACCTGTTCCATGGCC AACACTTGTCACTACTTGTACTGGGGTGTCAATGCTTTGCTA GATACCCAGATCATATGAAACAGCATGACTTTTCAAGAGTGCC TGCCCCAAGGTTATGTACAGGAAAGAACTATTTTTCAAAGAT GACGGGAACACTACAAGACACGTGCTGAAGTCAAGTTGAAGGT GATACCTTGTAAATAGAATCGAGTTAAAAGGTATTGATTTAAA GAAGATGGAAACATTCTTGGACACAATTTGAATACAACGCTAT TTCAGATAATGTATACATCACTGCAGACAAAAGAAATGGAAT CAAAGCTAATTTCAAATTAAGACACAATTGAAGATGGAAGCGT TCAACTAGCAGACCATTATCAACAAAATACTCAATTGGCGATGGC CCTGTCCTTTTACAGACAACATTACCTGTCCACACAATCTGCC TTTGAAAGATCCCAACGAAAAGAGATCACATGGTCTTCTTGG TTTGTAAACAGCTGCTGGATTACTAGGCATGGATGAATAACA ATGCTAAAAGGTGAAGAATTACTGCTGGTGTGCCAATTTGGTT GAATTAGATGGTGTGTAATGGTCACAATTTCTGTCTCCGGTGAA GGTGAAGGTGATGCTACTACGGTAAATTGACCTTAAATTTATTGTA CTACTGGTAAATTGCCAGTTCATGGCCAACCTTAGTCACTCTTAG GTTATGGTTTGTGTTTGTGCTAGATACCCAGATCATATGAAACA TGACTTTTTCAAGTCTGCCATGCCAGAAAGTTATGTTCAAGAAAGAAC TATTTTTTCAAAGATGACGGTAACTACAAGACCAGGCTGAAGTCAAG TTTGAAGGTGATACCTTAGTTAATAGAATCGAATAAAAGGTATTGATTTA AAGAAGATGGTAACATTTAGGTCACAATTTGAATACAATACTCTC ACAATGTTTACATCATGGCTGACAAACAAAAGAAATGGTATCAAAGTTAACT TCAAATTAGACACAACATTGAAGATGGTCTGTTCAATTAGCTGACCATT ATCAACAAAATACTCAATTGGTGTGTTCCAGTCTTGTACAGACAAC CATTACTATCTATCAATCTAGATTATCAAAGATCCAAACGAAAAGAGAG ATCAGATGGTCTTGTAGAATTTGTTACTGCTGCTGTTTACCATGGTAT GGATGAATTGTACAAA
citrine	CDS	ATGCTAAAAGGTGAAGAATTACTGCTGGTGTGCCAATTTGGTT GAATTAGATGGTGTGTAATGGTCACAATTTCTGTCTCCGGTGAA GGTGAAGGTGATGCTACTACGGTAAATTGACCTTAAATTTATTGTA CTACTGGTAAATTGCCAGTTCATGGCCAACCTTAGTCACTCTTAG GTTATGGTTTGTGTTTGTGCTAGATACCCAGATCATATGAAACA TGACTTTTTCAAGTCTGCCATGCCAGAAAGTTATGTTCAAGAAAGAAC TATTTTTTCAAAGATGACGGTAACTACAAGACCAGGCTGAAGTCAAG TTTGAAGGTGATACCTTAGTTAATAGAATCGAATAAAAGGTATTGATTTA AAGAAGATGGTAACATTTAGGTCACAATTTGAATACAATACTCTC ACAATGTTTACATCATGGCTGACAAACAAAAGAAATGGTATCAAAGTTAACT TCAAATTAGACACAACATTGAAGATGGTCTGTTCAATTAGCTGACCATT ATCAACAAAATACTCAATTGGTGTGTTCCAGTCTTGTACAGACAAC CATTACTATCTATCAATCTAGATTATCAAAGATCCAAACGAAAAGAGAG ATCAGATGGTCTTGTAGAATTTGTTACTGCTGCTGTTTACCATGGTAT GGATGAATTGTACAAA
mRFP	CDS	ATGGCTTCTCCGAAGATGTTATCAAAGAGTTTCATGCGTTTCAAAGTTCGT ATGGAAGGTTCCGTTAACGGTCACGAGTTCGAAATCGAAGGTGAAGGTG

(Continued.)

Table 1. (Continued.)

sequence ID	sequence description	DNA sequence
		AAGGTCGTCGGTACGAAGGTACCCAGACCGCTAAACTGAAAGTTACCAA
		AGGTGGTCCGCTGCCGTTTCGTTGGGACATCTGTCCCGCAGTTCCA
		GTACGGTTCCAAAGCTTACGTTAAACACCCGGCTGACATCCCGGACTAC
		CTGAAACTGTCCTTCCCGGAAGGTTTCAAATGGGAACGTGTTATGAACT
		TCGAGGACGGTGGTGTGTACCGTTACCCAGGACTCCTCCCTGCAAG
		ACGGTGAGTTTCATACAAAGTTAAACTGCGTGGTACCAACTCCCGTC
		CGACGGTCCGGTTATGCAGAAAAAACCATGGGTTGGGAAGCTCCAC
		CGAACGTATGTACCCGGAAGATGGTCTCTGAAAGGTGAAATCAAATG
		CGTCTGAAACTGAAAGACGGTGGTCACTACGACGCTGAAGTTAAACC
		ACCTACATGGCTAAAAAACCGTTGAGCTGCCGGGTCTTACAAAACCG
		ACATCAAAGTGGACATCACCTCCACAACGAGGACTACCATCGTTGA
		ACAGTACGAACGTGCTGAAGGTCGTCACTCCACCGGTGCTTAA

Note that the mathematical definition of crosstalk sensitivity $(\partial/\partial x_j)\zeta_i(t)$ depends on the initial condition $x_0(t)$ and the input $u(t)$. This dependency is consistent with the parametric sensitivity of biological function. Many genetic circuits in bacteria behave acceptably in one initial condition and for one input condition, e.g. in log-phase with an attenuated amount of a small molecule or sugar compound, but exhibit significantly different behaviour when input concentrations are increased by an order of magnitude or subject to an alternative preparation method prior to the experiment. The latter imposes a state history that defines a distinct initial condition, which can drive a biological network to a highly coupled or decoupled state.

Example C.4. Consider two mRNA species m_1 and m_2 competing for the same degradation enzyme D in a physical system. For simplicity of exposition, suppose their production dynamics do not depend on each other and can be modelled as $P_1(t)$ and $P_2(t)$, respectively. The crosstalk system is given as

$$\dot{m}_1 = P_1(t) - \frac{D_0 m_1 / k_{M,1}}{1 + m_1 / k_{M,1} + m_2 / k_{M,2}}$$

and

$$\dot{m}_2 = P_2(t) - \frac{D_0 m_2 / k_{M,2}}{1 + m_1 / k_{M,1} + m_2 / k_{M,2}},$$

while the reference system is given as

$$\dot{m}_1 = P_1(t) - \frac{D_0 m_1 / k_{M,1}}{1 + m_1 / k_{M,1}}$$

and

$$\dot{m}_2 = P_2(t) - \frac{D_0 m_2 / k_{M,2}}{1 + m_2 / k_{M,2}}.$$

In both systems, we have supposed that time has been rescaled so that the customary parameter k_{cat} for degradation is unity. The crosstalk sensitivities of m_1 and m_2 (with respect to each other) are given as

$$\frac{\partial \zeta_1}{\partial m_2} = \frac{\partial}{\partial m_2} \int_0^t \frac{-D_0 m_1 / k_{M,1} m_2 / k_{M,2}}{(1 + m_1 / k_{M,1})(1 + m_1 / k_{M,1} + m_2 / k_{M,2})}$$

and

$$\frac{\partial \zeta_2}{\partial m_1} = \frac{\partial}{\partial m_1} \int_0^t \frac{-D_0 m_1 / k_{M,1} m_2 / k_{M,2}}{(1 + m_2 / k_{M,2})(1 + m_1 / k_{M,1} + m_2 / k_{M,2})},$$

respectively. The crosstalk sensitivity between m_1 and m_2 is non-zero whenever m_1 or m_2 have non-zero initial condition.

Remark C.5. In synthetic biocircuit design, two chemical species x_i and x_j are often declared orthogonal when there is no designed interaction between them. Mathematically, in the crosstalk free system, this corresponds to

$$\frac{\partial}{\partial x_j} x_i^a(t) \equiv 0,$$

for all $x(0)$ and $u(t)$. In such a situation, $\zeta(x_i, x_j) \neq 0$ if and only if

$$\frac{\partial}{\partial x_j} x_i^c(t) = \int_0^t f_i^c(y, x_h, u) d\tau \neq 0.$$

This condition is interesting in experimental settings since a computational estimate of $(\partial/\partial x_j)\int_0^t f_i^c(t)$ from perturbation experiments coincides with a direct estimate of the sensitivity of the crosstalk $(\partial/\partial x_j)\zeta_i$. More specifically, when x_i and x_j are measured outputs of the system, we will show in the following that quantifying $\|Q_{i,j}^c(s)\|$ is directly related to an estimate of the crosstalk sensitivity $(\partial/\partial x_j)\zeta_i(t)$ near the equilibrium point x_e^c .

Remark C.6. In general, estimating the crosstalk sensitivity for the nonlinear systems (C 1) and (C 4) can be challenging if either x_i and x_j are not measured directly. Firstly, if experimental data are available, they will often consist of data for the measured species y in the crosstalk system, but not the reference system. Second, if only one of the species x_i (or none) is available for measurement, even if perturbation of x_j is possible, a nonlinear observer is required to estimate the trajectory of $x_j(t)$. Unless the parameters of $f_i(x, u)$ are known *a priori* (which is generally not the case), this then also requires system identification of the parameters of $f_c(x, u)$ and $f_o(x, u)$ which often results in a non-convex optimization problem.

Thus, our goal is to estimate the observed crosstalk between measured species Y_i and Y_j . This crosstalk estimate will invariably include the dynamics of unmeasured chemical species (such as ATP, RNAP, untagged mRNA and protein species, DNA–protein complexes etc.). From a synthetic biology design standpoint, this is not a disadvantage, since the goal is to design a synthetic gene network with an *abstracted* circuit architecture operating reliably in the context of many unmeasured species. In any genetic circuit, there are always additional biochemical compounds that are unmeasured. As stated in the main text, theorem 2.1 shows that under certain conditions, the dynamical structure function is able to estimate the crosstalk in a nonlinear system. We present the proof of this theorem now here.

Theorem C.7. Let \mathcal{L} denote the two-sided Laplace operator. Suppose we have a system model that incorporates the effect of crosstalk

$$\text{and } \left. \begin{aligned} \dot{y}^c &= f_y^c(y^c, x_h^c, u), \quad y^c(0) = 0 \\ \dot{x}_h^c &= f_{x_h}^c(y^c, x_h^c, u), \quad x_h^c(0) = 0, \end{aligned} \right\} \quad (\text{C3})$$

where $y^c \in \mathbb{R}^p$ is a vector of measured states in the output, $x_h^c \in \mathbb{R}^{n-p}$ are the unmeasured states of the system, $x^c = (y^c, x_h^c)$ is the full system state, and $u \in \mathbb{R}^m$ is a vector of system inputs. Furthermore, suppose we have an idealized system model to simulate system dynamics in the absence of crosstalk

$$\text{and } \left. \begin{aligned} \dot{y}^a &= f_y^a(y^a, x_h^a, u), \quad y^a(0) = 0 \\ \dot{x}_h^a &= f_{x_h}^a(y^a, x_h^a, u), \quad x_h^a(0) = 0, \end{aligned} \right\} \quad (\text{C4})$$

where $y^a \in \mathbb{R}^p$ is a vector of the measured states in the output, $x_h^a \in \mathbb{R}^{n-p}$ are the unmeasured states of the system, $x^a = (y^a, x_h^a)$ is the full system state, and $u \in \mathbb{R}^m$ is a vector of system inputs. Let $(Q^c(s), P^c(s))$ and $(Q^a(s), P^a(s))$ denote the respective dynamical structure functions calculated for each linearized system about the origin. Let

$$\zeta(t) = x^a(t) - x^c(t)$$

denote the deviation of the crosstalk state from the idealized state. Then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = Q_{ij}^a(s) - Q_{ij}^c(s) + \frac{\partial}{\partial Y_j} \mathcal{L}(O((x^c)^2)), \quad (\text{C5})$$

and in particular, if

$$Q_{ij}^a(s) \equiv 0$$

then

$$\frac{\partial \mathcal{L}(\zeta_i)}{\partial Y_j} = -Q_{ij}^c(s) + \frac{\partial}{\partial Y_j} \mathcal{L}(O((x^c)^2)),$$

and can be estimated from input output data $(Y(s), U(s))$.

Proof. First, note that the Laplace transform of $\mathcal{L}(\zeta(t)) = \mathcal{L}(x^a - x^c) \triangleq X^a(s) - X^c(s)$, which can be decomposed into its measured and unmeasured states

$$\begin{aligned} & \begin{bmatrix} Y^a \\ X_h^a \end{bmatrix} (s) - \begin{bmatrix} Y^c \\ X_h^c \end{bmatrix} (s) \\ &= \begin{bmatrix} Q^a Y^a(s) - Q^c Y^c(s) + (P^a - P^c)U(s) + \mathcal{L}(O(x^2)) \\ X_h^a(s) - X_h^c(s) + \mathcal{L}(O(x^2)) \end{bmatrix}. \end{aligned}$$

Examining the i th component equation and taking partials along $Y_j(s)$ yields equation (C5). ■

This result is important, since it tells us when estimating $Q^c(s)$ from experimental data will correspond to estimating crosstalk between measured states in $Y(s)$. Since necessary and sufficient conditions for identifying $Q(s)$ and $P(s)$ have been already characterized [62], this provides conditions for inferring crosstalk from input–output data. For example, a sufficient condition required is that there is an input variable available to excite each measured output of the genetic network attempting to be reconstructed. This allows for the possibility that some biological states are unmeasured and unexcited, but these will be viewed as hidden states that play a role in defining the edge dynamics in $Q^c(s)$.

More generally, even if parameters for $f^a(x, u)(t)$ are unknown, the structure of $Q^a(s)$ can be analytically calculated (using a symbolic algebra package). For every zero entry in $Q^a(s)$ (coinciding with designed orthogonality between measured states), we can then estimate $Q^c(s)$ directly.

In practice, estimation of $Q^c(s)$ is also confounded by noise. In our analysis in this paper, we suppose that a series of filters can be applied to eliminate the noise in the data. This may not be the case for biological systems that have been characterized as inherently stochastic, e.g. single cell gene expression dynamics. In such settings, the estimated dynamical structure $Q^c(s)$ is a mixture of the process noise in the system and the crosstalk. From the standpoint of synthetic biocircuit prototyping, both are undesirable in the ultimate iteration of the biocircuit and thus need to be quantified. In this paper, we will demonstrate our theoretical and computational framework with experimental results derived from *in vitro* systems, where signal-to-noise ratios are high and the only sources of noise are measurement noise and pipetting error. For a theoretical treatment of how to reverse engineer $Q^c(s)$ in the presence of process noise or system perturbation, see [86].

An advantage of using $Q^c(s)$ to estimate the crosstalk is that we can use the \mathcal{H}_∞ norm of $Q_{ij}^c(s)$ to calculate the worst-case crosstalk magnitude and \mathcal{H}_2 of $Q_{ij}^c(s)$ to calculate the average crosstalk across all frequencies.

C.1.1. Quantifying crosstalk with $Q^c(s)$

Recall the incoherent feedforward loop in §§2.1.1 and 2.1.2. In particular, comparing $Q^a(s)$ and $Q^c(s)$, we see that $Q^c(s)$ is a full transfer function matrix

$$\begin{pmatrix} 0 & \frac{1.6 \cdot 10^{-3}}{s+2.1 \cdot 10^{-3}} & \frac{0.041}{s+2.1 \cdot 10^{-3}} \\ \frac{(1.6 \cdot 10^{-3})s+0.048}{s^2+1.5s+3.3 \cdot 10^{-3}} & 0 & \frac{0.041}{s+2.1 \cdot 10^{-3}} \\ \frac{(3.8 \cdot 10^{-4})s+7.4 \cdot 10^{-4}}{s^2+1.6s+0.13} & \frac{(3.8 \cdot 10^{-4})s+4.4 \cdot 10^{-4}}{s^2+1.6s+0.13} & 0 \end{pmatrix}$$

and $Q^a(s)$ is lower-triangular, reflecting the network structure of the intended IFFL. By examining the upper triangular entries in $Q^c(s)$, we can directly examine the effects of degradation crosstalk. In the lower entries of $Q^c(s)$, these crosstalk effects are confounded with the direct interactions modelled in $Q^a(s)$. Although the gains of the entries in $Q^c(s)$ are small, they nonetheless can have a significant effect on the dynamics of the IFFL.

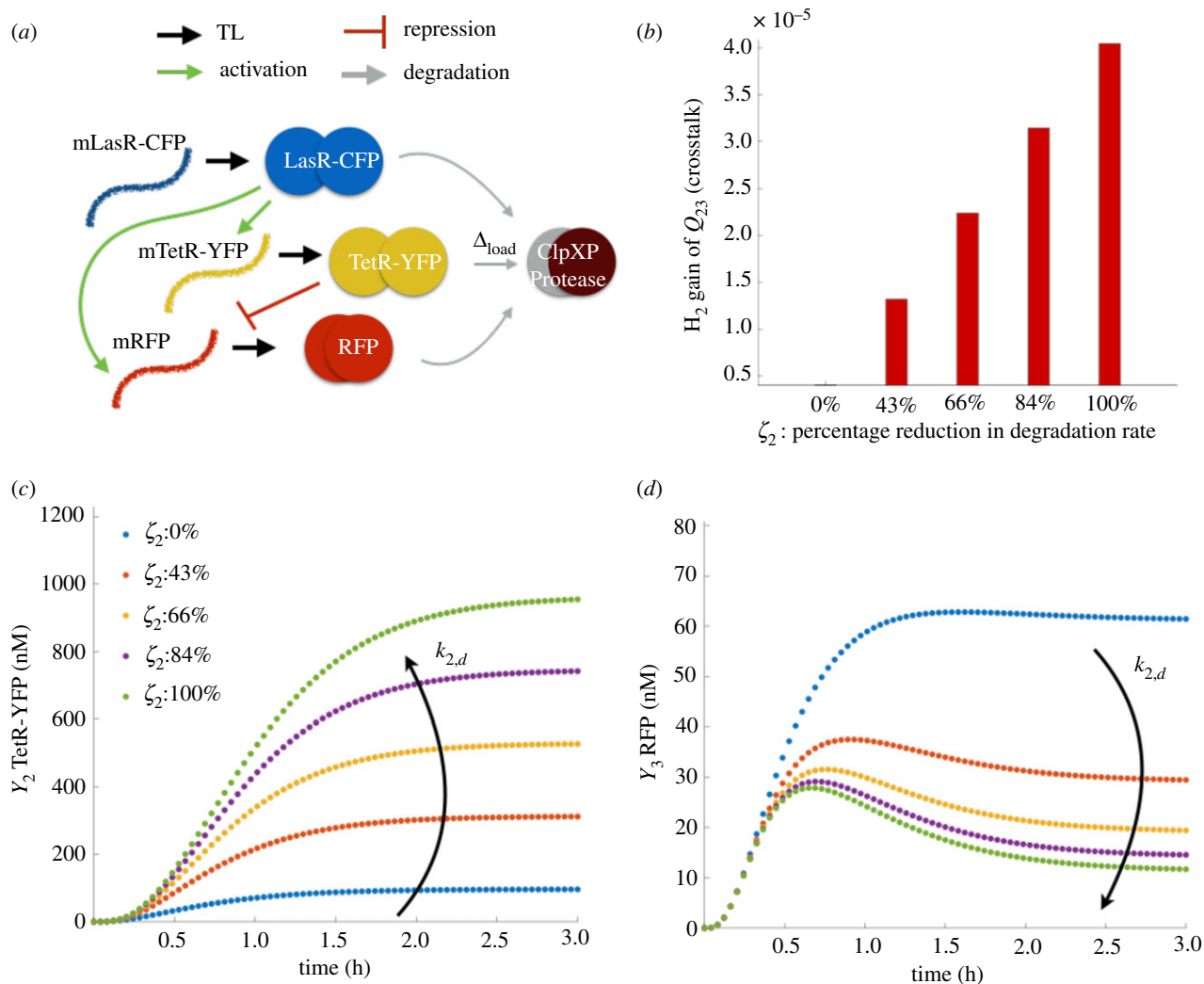


Figure 11. Dynamical structure functions quantify biomolecular crosstalk. (a) A schematic illustrating the design of this simulation example. The crosstalk and reference model of the incoherent feedforward loop from examples 2.1.1 and 2.1.2 are simulated accordingly to satisfy internal equivalence, for varying values of $k_{2,d}$. Standard parameters from the literature [87] were used to generate the simulation. As the size of the load Δ_{load} increases, the ability of the IFFL to respond with a pulse decreases. (b) The \mathcal{H}_2 gain of $Q_{23}^c(s)$ is plotted as a function of ζ . Note that $Q_{23}^c(s)$ is a pure crosstalk term, since $Q_{23}^a(s) \equiv 0$. As the effective crosstalk in ζ_2 increases, $Q_{23}^c(s)$ mirrors that increase, as shown in proposition 1. (c,d) Time lapse responses of the incoherent feedforward loop: for each value of $k_{2,d}$, the value of ζ_2 at $t = 3$ h is calculated and used to label curves (as percentage of maximum load). Note the monotonic relationship between $k_{2,d}$, ζ and the output responses of Y_2 and Y_3 (negatively monotonic).

In figure 11, we plot the time-lapse response of $y_2(t)$ and $y_3(t)$ for varying parameter values of $k_{2,d}$ in equation (2.9). The $k_{2,d}$ parameter is a Michaelis constant that determines the effective affinity of substrate x_2 in binding with C_0 . As $k_{2,d}$ increases, the affinity of substrate x_2 is diminished, relative to the affinity of x_1 and x_3 . Attenuating $k_{2,d}$ can be viewed as similar to swapping out a strong degradation marker for protease degradation with a weaker degradation marker on the species x_2 . In the experimental literature, there are multiple degradation markers for proteins that confer varying binding affinities to an associated protease [88]. In our simulation, we consider five potential values for $k_{2,d}$: 500, 1625, 2750, 3875 and 5000 μ M corresponding to five artificial LVA markers of varying strengths for the protease ClpXP frequently used in *E. coli*.

Note that as we decrease the affinity of y_2 for ClpXP, this also coincides with an increased ζ_2 crosstalk magnitude. Here, we have computed $\zeta_2 = y_2^c(t) - y_2^a(t)$. We find that $|\zeta_2|$ increases as $k_{2,d}$ increases. In figure 11b–d, ζ_2 is plotted as a percentage of maximum absolute change across all values of $k_{2,d}$.

We see that the time-lapse response of $y_2(t)$ increases monotonically for all t as the crosstalk $\zeta_2(t)$ increases. This is consistent with biological intuition, since an increase in competition for resource loading (an increase in $k_{2,d}$) results in prolonged lifetimes of each individual y_2 (TetR-YFP) protein. This in turn results in higher repression levels of y_3 in the incoherent feedforward loop. Increased competition for ClpXP from substrates y_3 and y_1 have the effect of damping y_3 dynamics and reinforcing the pulsatile response of the IFFL. The crosstalk in this circuit thus has the effect of effectively strengthening the negative regulation of y_2 on y_3 , encouraging the downward transient after 0.75 h. Our network analysis shows we can improve the robustness of an IFFL's pulse by attenuating the relative binding affinity of the repressor to its protease.

In general, crosstalk effects do not necessarily reinforce the feedback architecture of a biocircuit. This underscores the importance of having techniques for quantifying crosstalk in a synthetic gene network and validating that designed interactions are dominant over crosstalk interactions.

- Brophy JA, Voigt CA. 2014 Principles of genetic circuit design. *Nat. Methods* **11**, 508–520. (doi:10.1038/nmeth.2926)
- Franco E, Friedrichs E, Kim J, Jungmann R, Murray R, Winfree E, Simmel FC. 2011 Timing molecular motion and production with a synthetic transcriptional clock. *Proc. Natl Acad. Sci. USA* **108**, E784–E793. (doi:10.1073/pnas.1100060108)
- Endo K, Hayashi K, Inoue T, Saito H. 2013 A versatile cis-acting inverter module for synthetic translational switches. *Nat. Commun.* **4**, 2393. (doi:10.1038/ncomms3393)
- Hsia J, Holtz WJ, Maharbiz MM, Arcak M, Keasling JD. 2016 Modular synthetic inverters from zinc finger proteins and small RNAs. *PLoS ONE* **11**, e0149483. (doi:10.1371/journal.pone.0149483)
- Weiss R, Basu S, Hooshangi S, Kalmbach A, Karig D, Mehreja R, Netravali I. 2003 Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Comput.* **2**, 47–84. (doi:10.1023/A:1023307812034)
- Tamsir A, Tabor JJ, Voigt CA. 2011 Robust multicellular computing using genetically encoded nor gates and chemical ‘wires’. *Nature* **469**, 212–215. (doi:10.1038/nature09565)
- Del Vecchio D, Ninfa AJ, Sontag ED. 2008 Modular cell biology: retroactivity and insulation. *Mol. Syst. Biol.* **4**, 161. (doi:10.1038/msb4100204)
- Gyorgy A, Del Vecchio D. 2012 Retroactivity to the input in complex gene transcription networks. In *51st IEEE Conf. on Decision and Control, Maui, HI, USA, 10–13 December 2012*, pp. 3595–3601. (doi:10.1109/CDC.2012.6426160)
- Jayanthi S, Del Vecchio D. 2009 On the compromise between retroactivity attenuation and noise amplification in gene regulatory networks. In *Proc. 2009 IEEE Conf. on Decision and Control, Shanghai, China, 15–18 December 2009*. (doi:10.1109/CDC.2009.5400631)
- Zhang S, Voigt CA. 2018 Engineered dCas9 with reduced toxicity in bacteria: implications for genetic circuit design. *Nucleic Acids Res.* **46**, 11115–11125.
- Zhao Y, Li L, Zheng G, Jiang W, Deng Z, Wang Z, Lu Y. 2018 Crispr/dCas9-mediated multiplex gene repression in streptomyces. *Biotechnol. J.* **13**, 1800121. (doi:10.1002/biot.201800121)
- Zhang X, Wang J, Cheng Q, Zheng X, Zhao G, Wang J. 2017 Multiplex gene regulation by CRISPR-ddCpf1. *Cell Discov.* **3**, 1–9. (doi:10.1038/celldisc.2017.18)
- Hutchison CA *et al.* 2016 Design and synthesis of a minimal bacterial genome. *Science* **351**, 351. (doi:10.1126/science.aad6253)
- Nielsen AA, Voigt CA. 2014 Multi-input CRISPR/C as genetic circuits that interface host regulatory networks. *Mol. Syst. Biol.* **10**, 763. (doi:10.15252/msb.20145735)
- Hou Z, Jiang P, Swanson SA, Elwell AL, Nguyen BKS, Bolin JM, Stewart R, Thomson JA. 2015 A cost-effective RNA sequencing protocol for large-scale gene expression studies. *Sci. Rep.* **5**, 9570. (doi:10.1038/srep09570)
- van der Kamp MW *et al.* 2010 Dynameomics: a comprehensive database of protein dynamics. *Structure* **18**, 423–435. (doi:10.1016/j.str.2010.01.012)
- Link H, Fuhrer T, Gerosa L, Zamboni N, Sauer U. 2015 Real-time metabolome profiling of the metabolic switch between starvation and growth. *Nat. Methods* **12**, 1091–1097. (doi:10.1038/nmeth.3584)
- Karr JR, Sanghvi JC, Macklin DN, Gutschow MV, Jacobs JM, Bolival Jr B, Assad-Garcia N, Glass JI, Covert MW. 2012 A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401. (doi:10.1016/j.cell.2012.05.044)
- Schöllig A, Münz U, Allgöwer F. 2007 Topology-dependent stability of a network of dynamical systems with communication delays. In *2007 European Control Conf. (ECC), Kos, Greece, 2–5 July 2007*, pp. 1197–1202. (doi:10.23919/ECC/2007.7068977)
- Gyorgy A. 2018 Sharing resources can lead to monostability in a network of bistable toggle switches. *IEEE Control Syst. Lett.* **3**, 308–313. (doi:10.1109/LCSYS.2018.2871128)
- Dullerud GE, Paganini F. 2013 *A course in robust control theory: a convex approach*, vol. 36. Springer Science & Business Media.
- Yeung E, Goncalves J, Sandberg H, Warnick S. 2010 Representing structure in linear interconnected dynamical systems. In *49th IEEE Conf. on Decision and Control, Atlanta, GA, USA, 15–17 December 2010*. (doi:10.1109/CDC.2010.5718109)
- Sommerlade L, Eichler M, Jachan M, Henschel K, Timmer J, Schelter B. 2009 Estimating causal dependencies in networks of nonlinear stochastic dynamical systems. *Phys. Rev. E* **80**, 051128. (doi:10.1103/PhysRevE.80.051128)
- Kang T, Moore R, Li Y, Sontag E, Bleris L. 2015 Discriminating direct and indirect connectivities in biological networks. *Proc. Natl Acad. Sci. USA* **112**, 12893–12898. (doi:10.1073/pnas.1507168112)
- Gates AJ, Rocha LM. 2015 Control of complex networks requires both structure and dynamics. (<http://arxiv.org/abs/1509.08409>)
- Wolf DM, Arkin AP. 2003 Motifs, modules and games in bacteria. *Curr. Opin. Microbiol.* **6**, 125–134. (doi:10.1016/S1369-5274(03)00033-X)
- Mangan S, Alon U. 2003 Structure and function of the feed-forward loop network motif. *Proc. Natl Acad. Sci. USA* **100**, 11 980–11 985. (doi:10.1073/pnas.2133841100)
- Cardinale S, Arkin A. 2012 Contextualizing context for synthetic biology—identifying causes of failure of synthetic biological systems. *Biotechnol. J.* **7**, 856–866. (doi:10.1002/biot.201200085)
- Fontanarrosa P, Doosthosseini H, Espah Borujeni A, Dorfan Y, Voigt CA, Myers CJ. 2020 Genetic circuit dynamics: hazard and glitch analysis. *ACS Synth. Biol.* **9**, 2324–2338. (doi:10.1021/acssynbio.0c00055)
- Tyson JJ, Chen KC, Novak B. 2003 Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.* **15**, 221–231. (doi:10.1016/S0955-0674(03)00017-6)
- Goentoro L, Shoval O, Kirschner MW, Alon U. 2009 The incoherent feedforward loop can provide fold-change detection in gene regulation. *Mol. Cell* **36**, 894–899. (doi:10.1016/j.molcel.2009.11.018)
- Goentoro L, Kirschner MW. 2009 Evidence that fold-change, and not absolute level, of β -catenin dictates wnt signaling. *Mol. Cell* **36**, 872–884. (doi:10.1016/j.molcel.2009.11.017)
- Elowitz MB, Leibler S. 2000 A synthetic oscillatory network of transcriptional regulators. *Nature* **403**, 335–338. (doi:10.1038/35002125)
- El Samad H, Del Vecchio D, Khammash M. 2005 Repressilators and promotilators: loop dynamics in synthetic gene networks. In *Proc. 2005 American Control Conf., Portland, OR, USA, 8–10 June 2005*, pp. 4405–4410. (doi:10.1109/ACC.2005.1470689)
- Garcia-Ojalvo J, Elowitz MB, Strogatz SH. 2004 Modeling a synthetic multicellular clock: repressilators coupled by quorum sensing. *Proc. Natl Acad. Sci. USA* **101**, 10955–10960. (doi:10.1073/pnas.0307095101)
- Müller S, Hofbauer J, Endler L, Flamm C, Widder S, Schuster P. 2006 A generalized model of the repressilator. *J. Math. Biol.* **53**, 905–937. (doi:10.1007/s00285-006-0035-9)
- Gardner TS, Cantor CR, Collins JJ. 2000 Construction of a genetic toggle switch in *Escherichia coli*. *lett. Nat.* **403**, 339–342. (doi:10.1038/35002131)
- Marucci L, Barton DA, Cantone I, Ricci MA, Cosma MP, Santini S, di Bernardo D, di Bernardo M. 2009 How to turn a genetic circuit into a synthetic tunable oscillator, or a bistable switch. *PLoS ONE* **4**, e8083. (doi:10.1371/journal.pone.0008083)
- Wolkenhauer O, Wellstead P, Cho K-H, Sontag ED. 2008 Network reconstruction based on steady-state data. *Essays Biochem.* **45**, 161–176. (doi:10.1042/bse0450161)
- Prabakaran S, Gunawardena J, Sontag E. 2014 Paradoxical results in perturbation-based signaling network reconstruction. *Biophys. J.* **106**, 2720–2728. (doi:10.1016/j.bpj.2014.04.031)
- Sontag E, Kiyatkin A, Kholodenko BN. 2004 Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data. *Bioinformatics* **20**, 1877–1886. (doi:10.1093/bioinformatics/bth173)
- Quarton T, Kang T, Sontag ED, Bleris L. 2016 Exploring the impact of resource limitations on gene network reconstruction. In *2016 IEEE 55th Conf. on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016*, pp. 3350–3355. (doi:10.1109/CDC.2016.7798773)
- Kang T, White JT, Xie Z, Benenson Y, Sontag E, Bleris L. 2013 Reverse engineering validation using a benchmark synthetic gene circuit in human cells. *ACS Synth. Biol.* **2**, 255–262. (doi:10.1021/sb300093y)
- Wang S, Lin J-R, Sontag ED, Sorger PK. 2019 Inferring reaction network structure from single-cell,

- multiplex data, using toric systems theory. *PLoS Comput. Biol.* **15**, e1007311. (doi:10.1371/journal.pcbi.1007311)
45. Andreç M, Kholodenko B, Levy R, Sontag E. 2005 Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy. *J. Theoret. Biol.* **232**, 427–441. (doi:10.1016/j.jtbi.2004.08.022)
 46. Kholodenko BN, Kiyatkin A, Bruggeman FJ, Sontag E, Westerhoff HV, Hoek JB. 2002 Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proc. Natl Acad. Sci. USA* **99**, 12841–12846. (doi:10.1073/pnas.192442699)
 47. Schuetz R, Kuepfer L, Sauer U. 2007 Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. *Mol. Syst. Biol.* **3**, 119. (doi:10.1038/msb4100162)
 48. Hilfinger A, Norman TM, Vinnicombe G, Paulsson J. 2016 Constraints on fluctuations in sparsely characterized biological systems. *Phys. Rev. Lett.* **116**, 058101. (doi:10.1103/PhysRevLett.116.058101)
 49. Hilfinger A, Norman TM, Paulsson J. 2016 Exploiting natural fluctuations to identify kinetic mechanisms in sparsely characterized systems. *Cell Syst.* **2**, 251–259. (doi:10.1016/j.cels.2016.04.002)
 50. Yeung E, Beck JL, Murray RM. 2013 Modeling environmental disturbances with the chemical master equation. In *52nd IEEE Conf. on Decision and Control*, pp. 1384–1391. (doi:10.1109/CDC.2013.676007)
 51. Gyorgy A, Jiménez JI, Yazbek J, Huang H-H, Chung H, Weiss R, Del Vecchio D. 2015 Isocost lines describe the cellular economy of genetic circuits. *Biophys. J.* **109**, 639–646. (doi:10.1016/j.bpj.2015.06.034)
 52. Gyorgy A, Del Vecchio D. 2014 Limitations and trade-offs in gene expression due to competition for shared cellular resources. In *53rd IEEE Conf. on Decision and Control*, pp. 5431–5436. (doi:10.1109/CDC.2014.7040238)
 53. Qian Y, Huang H-H, Jiménez JI, Del Vecchio D. 2017 Resource competition shapes the response of genetic circuits. *ACS Synth. Biol.* **6**, 1263–1272. (doi:10.1021/acssynbio.6b00361)
 54. Yeung E, Kim J, Murray RM. 2013 Resource competition as a source of non-minimum phase behavior in transcription-translation systems. In *52nd IEEE Conf. on Decision and Control, Firenze, Italy, 10–13 December 2013*, pp. 4060–4067. (doi:10.1109/CDC.2013.6760511)
 55. Takens F. 1981 Detecting strange attractors in turbulence. In *Dynamical systems and turbulence*, pp. 366–381. Berlin, Germany: Springer.
 56. Sauer T, Yorke JA, Casdagli M. 1991 Embedology. *J. Stat. Phys.* **65**, 579–616. (doi:10.1007/BF01053745)
 57. Reynolds A, Leake D, Boese Q, Scaringe S, Marshall WS, Khvorova A. 2004 Rational siRNA design for RNA interference. *Nat. Biotechnol.* **22**, 326–330. (doi:10.1038/nbt936)
 58. Elmore JR, Dexter GN, Francis R, Riley L, Huenemann J, Baldino H, Guss AM, Egbert R. 2020 The sage genetic toolkit enables highly efficient, iterative site-specific genome engineering in bacteria. *bioRxiv*. (doi:10.1101/2020.06.28.176339)
 59. Lutz R, Bujard H. 1997 Independent and tight regulation of transcriptional units in *Escherichia coli* via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements. *Nucleic Acids Res.* **25**, 1203–1210. (doi:10.1093/nar/25.6.1203)
 60. Ljung L. 1999 *System identification: theory for the user*. Hoboken, NJ: Prentice Hall.
 61. Tangirala AK. 2018 *Principles of system identification: theory and practice*. Boca Raton, FL: CRC Press.
 62. Gonçalves J, Warnick S. 2008 Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks. *IEEE Trans. Automat. Control* **53**, 1670–1674. (doi:10.1109/TAC.2008.928114)
 63. Chetty V, Warnick S. 2015 Network semantics of dynamical systems. In *54th IEEE Conf. on Decision and Control, Osaka, Japan, 15–18 December 2015*, pp. 1557–1562. (doi:10.1109/CDC.2015.7402432)
 64. Yeung E, Gonçalves J, Sandberg H, Warnick S. 2011 Mathematical relationships between representations of structure in linear interconnected dynamical systems. In *Proc. 2011 American Control Conf., San Francisco, CA, USA, 29 June–1 July 2011*, pp. 4348–4353. (doi:10.1109/ACC.2011.5991314)
 65. Johnson CA, Woodbury N, Warnick S. 2020 Graph theoretic foundations of cyclic and acyclic linear dynamic networks. *IFAC-PapersOnLine* **53**, 26–33. (doi:10.1016/j.ifacol.2020.12.037)
 66. Adebayo J, Southwick T, Chetty V, Yeung E, Yuan Y, Gonçalves J, Grose J, Prince J, Stan G-B, Warnick S. 2012 Dynamical structure function identifiability conditions enabling signal structure reconstruction. In *51st IEEE Conf. on Decision and Control, Maui, HI, USA, 10–13 December 2012*, pp. 4635–4641. (doi:10.1109/CDC.2012.6426183)
 67. Berg JM, Tymoczko JL, Stryer L. 2002 *Biochemistry*. New York, NY: W. H. Freeman.
 68. Bremer H, Dennis PP. 1996 *Modulation of chemical composition and other parameters of the cell by growth rate*, ch. 97. New York, NY: Springer.
 69. Vogel U, Jensen KF. 1994 The rna chain elongation rate in *Escherichia coli* depends on the growth rate. *J. Bacteriol.* **176**, 2807–2813. (doi:10.1128/jb.176.10.2807-2813.1994)
 70. Bernstein JA, Khodursky AB, Lin P-H, Lin-Chao S, Cohen SN. 2002 Global analysis of mRNA decay and abundance in *Escherichia coli* at single-gene resolution using two-color fluorescent DNA microarrays. *Proc. Natl Acad. Sci. USA* **99**, 9697–9702. (doi:10.1073/pnas.112318199)
 71. Yeung E, Kim J, Yuan Y, Gonçalves J, Murray R. 2012 Quantifying crosstalk in biochemical systems. In *51st IEEE Conf. on Decision and Control, Maui, HI, USA, 10–13 December 2012*, pp. 5528–5535. (doi:10.1109/CDC.2012.6425854)
 72. Cookson NA, Mather WH, Danino T, Mondragón-Palomino O, Williams RJ, Tsimring LS, Hasty J. 2011 Queueing up for enzymatic processing: correlated signaling through coupled degradation. *Mol. Syst. Biol.* **7**, 561. (doi:10.1038/msb.2011.94)
 73. Ceroni F *et al.* 2018 Burden-driven feedback control of gene expression. *Nat. Methods* **15**, 387–393. (doi:10.1038/nmeth.4635)
 74. Thomaseth C, Fey D, Santra T, Rukhlenko OS, Radde NE, Kholodenko BN. 2018 Impact of measurement noise, experimental design, and estimation methods on modular response analysis based network reconstruction. *Sci. Rep.* **8**, 16217. (doi:10.1038/s41598-018-34353-3)
 75. Kim J, Winfree E. 2011 Synthetic in vitro transcriptional oscillators. *Mol. Syst. Biol.* **7**, 465. (doi:10.1038/msb.2010.119)
 76. Hsiao V, Hori Y, Rothermund PW, Murray RM. 2016 A population-based temporal logic gate for timing and recording chemical events. *Mol. Syst. Biol.* **12**, 869. (doi:10.15252/msb.20156663)
 77. Friedland AE, Lu TK, Wang X, Shi D, Church G, Collins JJ. 2009 Synthetic gene networks that count. *Science* **324**, 1199–1202. (doi:10.1126/science.1172005)
 78. Andrews LB, Nielsen AA, Voigt CA. 2018 Cellular checkpoint control using programmable sequential logic. *Science* **361**, eaap8987. (doi:10.1126/science.aap8987)
 79. Engler C, Gruetzner R, Kandzia R, Marillonnet S. 2009 Golden gate shuffling: a one-pot DNA shuffling method based on type IIs restriction enzymes. *PLoS ONE* **4**, e5553. (doi:10.1371/journal.pone.0005553)
 80. Gibson DG *et al.* 2010 Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* **329**, 52–56. (doi:10.1126/science.1190719)
 81. Engler C, Kandzia R, Marillonnet S. 2008 A one pot, one step, precision cloning method with high throughput capability. *PLoS ONE* **3**, e3647. (doi:10.1371/journal.pone.0003647)
 82. Weber E, Engler C, Gruetzner R, Werner S, Marillonnet S. 2011 A modular cloning system for standardized assembly of multigene constructs. *PLoS ONE* **6**, e16765. (doi:10.1371/journal.pone.0016765)
 83. Mutalik VK *et al.* 2013 Precise and reliable gene expression via standard transcription and translation initiation elements. *Nat. Methods* **10**, 354–360. (doi:10.1038/nmeth.2404)
 84. Chen Y-J, Liu P, Nielsen AA, Brophy JA, Clancy K, Peterson T, Voigt CA. 2013 Characterization of 582 natural and synthetic terminators and quantification of their design constraints. *Nat. Methods* **10**, 659–664. (doi:10.1038/nmeth.2515)
 85. Savageau MA. 2001 Design principles for elementary gene circuits: elements, methods, and examples. *Chaos* **11**, 142–159. (doi:10.1063/1.1349892)
 86. Yuan Y, Stan G-B, Warnick S, Gonçalves J. 2011 Robust dynamical network structure reconstruction. *Automatica* **47**, 1230–1235. (doi:10.1016/j.automatica.2011.03.008)
 87. Milo R, Jorgensen P, Moran U, Weber G, Springer M. 2010 Bionumbers: the database of key numbers in molecular and cell biology. *Nucleic Acids Res.* **38**(Suppl. 1), D750–D753. (doi:10.1093/nar/gkp889)
 88. Andersen JB, Sternberg C, Poulsen LK, Bjørn SP, Givskov M, Molin S. 1998 New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria. *Appl. Environ. Microbiol.* **64**, 2240–2246. (doi:10.1128/AEM.64.6.2240-2246.1998)