

Cell, Volume 166

Supplemental Information

**Information Integration and Energy Expenditure
in Gene Regulation**

Javier Estrada, Felix Wong, Angela DePace, and Jeremy Gunawardena

SUPPLEMENTAL INFORMATION

Contents

SUPPLEMENTAL EXPERIMENTAL PROCEDURES	1
1 The linear framework	3
1.1 Laplacian dynamics	3
1.2 Steady states at thermodynamic equilibrium	3
1.3 Independent generators at equilibrium	4
1.4 Steady states away from equilibrium	4
1.5 Gene regulation functions	5
2 The graph G_n and background assumptions	5
3 Equilibrium parameters for G_n	6
3.1 Equilibrium parameters $K_{i,S}$ and $\omega_{i,S}$	6
3.2 Independent generators for the $K_{i,S}$	7
3.3 The exchange formula for $K_{i,S}$	8
3.4 Independent non-dimensional κ_i and $\omega_{i,S}$	8
4 Laplacian steady states of G_n	9
4.1 Steady states of G_n at equilibrium	9
4.2 History dependence away from equilibrium	9
4.3 Steady states of G_n away from equilibrium	10
4.3.1 Parameterisation away from equilibrium	11
5 Gene regulation functions for G_n	11
5.1 The normalised GRF for protein level	11
5.1.1 T acts only as a transcriptional activator	12
5.2 Equilibrium gene regulation functions	12
5.3 Similarity of $f_n(x)$ in the all-or-nothing strategy to $\mathcal{H}_n(x)$	13
5.4 Non-equilibrium gene regulation functions	14
5.5 GRFs are increasing but not always sigmoidal	14
6 Properties of the shape measures γ and ρ	15
6.1 γ and ρ for the Hill function \mathcal{H}_a	15
6.2 γ and ρ do not depend on the scale factor	15

7	Parameter ranges and search algorithms	16
7.1	Equilibrium parameter ranges	16
7.2	Non-equilibrium parameter ranges	16
7.3	Biased sampling algorithm for position-steepness regions	16
7.3.1	Calculating γ and ρ	18
7.4	Testing convergence and numerical accuracy	19
	SUPPLEMENTAL REFERENCES	21

SUPPLEMENTAL EXPERIMENTAL PROCEDURES

Sections §1 to §7 which follow provide supplemental procedures which give full details of how the mathematical results in the Paper were derived. For the convenience of readers, the following is a list, in order of occurrence in the Paper, of each citation to the Supplemental Information (SI), with pointers to the relevant parts of this document.

- **“the linear framework allows steady-state probabilities to be calculated . . . ; see Experimental Procedures and Supplemental Information (SI)”**. See §1.
- **“This is not a fundamental limitation but arises from technical difficulties with the Principle of Detailed Balance, which imposes algebraic constraints on higher-order cooperativities (SI)”**. See §1.2.
- **“the relevant parameters are the association constants $K_{i,S}$ (Figure 2B), of which there are $n2^{n-1}$ (SI)”**. See §3.1 and Eq. 8.
- **Paper Equation 2 and “We prove that, because of detailed balance, higher-order cooperativities must satisfy the “exchange formula” (SI)”**. See Eq. 13 in §3.4.
- **“all the $K_{i,S}$ can be calculated from them using Eq. 2 (SI)”**. See Eq. 9 and §3.2.
- **“we assume that T acts as a transcriptional activator (SI)”**. See §5.1.1.
- **“these three strategies broadly sample the spectrum of possibilities (SI)”**. See §5.1.1.
- **Paper Equation 4 and “The level of protein expression after normalisation . . . has the form, for the all-or-nothing strategy (SI)”**. See Eq. 23 in §5.2.
- **“The coefficients c_k are given (SI) by a sum of products”**. See Eq. 22 in §5.2.
- **“For the other strategies (Figure 3A), only the numerator of Eq. 4 changes (SI)”**. See §5.1.1 and Eqs. 23 to 25 in the equilibrium case.
- **“The GRFs discussed in this paper are strictly increasing functions (SI)”**. See §5.5.
- **“The algebraic form of the Hill function, $\mathcal{H}_a(x) = x^a/(1 + x^a)$, is closest to that of the GRF for the the all-or-nothing strategy in Eq. 4 (SI)”**. See §5.3.
- **“If $a = n$, algebraic resemblance is only possible, and then only approximately, if the parameters in the GRF are given implausible numerical values (SI)”**. See §5.3.
- **“Because of normalisation, γ and ρ do not depend on $K_{1,\emptyset}$ (SI)”**. See §6.2.
- **“We believe this range is generous but most of our results do not depend on it (below and SI)”**. See §7.1.
- **“We therefore used . . . a biased sampling algorithm to identify the boundary of the region (SI)”**. See §7.3.
- **“The non-equilibrium GRF for the all-or-nothing strategy then takes the form (SI)”**. See Eq. 26 in §5.4.
- **“GRFs for the other strategies differ only in the numerator (SI)”**. See §5.1.1 and Eqs. 26 to 28 for the non-equilibrium case.
- **“the largest number of sites that we can feasibly analyse is $n = 3$ (SI)”**. See §4.2.

- **“if the system were at equilibrium, it would be comparable with the previous equilibrium analysis (SI)”**. See §7.2.
- **“the steepness and position of g_n^{ne} are independent of the values of $a_{1,\emptyset}$ and $b_{1,\{1\}}$ (SI)”**. See §6.2.
- **“which leads to a dramatic increase in the complexity of the coefficients in Eq. 7 (SI; see Discussion)”**. See §4.2.
- **“This discrepancy arises from loss of detailed balance (SI)”**. See §4.3 and §5.4.
- **“The mathematical model and the the results presented here are based on the “linear framework” . . . see the SI for full details”**. See §1.
- **“The details are given in the SI, along with the tests”**. See §7.3 for the details of the algorithm and §7.4 for the tests.
- Paper Figure 4 caption, **“obtained by a biased sampling algorithm (SI)”**. See §7.3.
- Paper Figure 5 caption, **“The biased sampling algorithm had to be modified to find the average-binding regions (SI)”**. See Step 2 of the algorithm described in §7.3.

1 The linear framework

The linear framework is a graph-based method for time-scale separation in biochemical systems. It was introduced in previous work [2, 5] and reviewed in [3]. It was applied to gene regulation in [1], which provides the foundation on which the present Paper is based. As these ideas are relatively new, we provide here the background needed to understand the present Paper but [1, 5] should be consulted for more details and proofs.

1.1 Laplacian dynamics

The framework starts from a finite, connected, labelled, directed graph, G , as in Paper Figure 2A. The vertices are microstates, or “snapshots” of chromatin together with bound proteins, the edges are transitions between microstates and the edge labels are infinitesimal transition rates for a Markov process, with units of $(\text{time})^{-1}$. Assume that the vertices are indexed $1, \dots, N$. The stochastic master equation of the Markov process is given by the equation

$$\frac{du}{dt} = \mathcal{L}(G).u, \quad (1)$$

where, for $1 \leq i \leq N$, $u_i(t)$ is the probability of the system being in microstate i at time t and $\mathcal{L}(G)$ is the $N \times N$ Laplacian matrix of G . Eq. 1 is a linear differential equation, hence “linear framework”. The nonlinearities in the biochemistry are absorbed into the edge labels. For any graph and any initial condition, the Laplacian dynamics in Eq. 1 always converges to a steady state $u^* \in \mathbb{R}^N$ where

$$\left. \frac{du}{dt} \right|_{u=u^*} = 0,$$

so that $u^* \in \ker \mathcal{L}(G)$. An important basic result in the linear framework is that if G is strongly connected, then

$$\dim \ker \mathcal{L}(G) = 1.$$

It follows that, if ρ^G is any basis element of $\ker \mathcal{L}(G)$, then the steady-state probabilities, u^* , can be calculated by normalising ρ^G to its total,

$$u^* = \frac{\rho^G}{1.\rho^G}. \quad (2)$$

There are two basic methods for calculating a Laplacian steady state, ρ^G .

1.2 Steady states at thermodynamic equilibrium

If the system can reach thermodynamic equilibrium, then detailed balance must be satisfied. This requires, first, that each edge $i \xrightarrow{a} j$ is accompanied by a reverse edge, $j \xrightarrow{b} i$ and, second, in any cycle of reversible edges, the product of the labels going clockwise around the cycle equals the product going counterclockwise (“cycle condition”). These requirements guarantee that G is strongly connected and that, in any steady state, each pair of reversible edges is independently at steady state, irrespective of any other edges involving the vertices. If vertex 1 is chosen as a reference vertex and j is any other vertex, then there is always a path of reversible edges from 1 to j

$$1 = i_1 \xrightleftharpoons[b_1]{a_1} i_2 \xrightleftharpoons[b_2]{a_2} \dots \xrightleftharpoons[b_{p-1}]{a_{p-1}} i_p \xrightleftharpoons[b_p]{a_p} i_{p+1} = j.$$

A basis element $\xi^G \in \ker \mathcal{L}(G)$ is then given, for each vertex j , by the corresponding product of label ratios

$$\xi_j^G = \left(\frac{a_p}{b_p} \right) \left(\frac{a_{p-1}}{b_{p-1}} \right) \dots \left(\frac{a_2}{b_2} \right) \left(\frac{a_1}{b_1} \right). \quad (3)$$

The cycle condition ensures that the quantity in Eq. 3 does not depend on the choice of path from 1 to j .

It can be seen from van't Hoff's formula that ξ_j^G is the Boltzmann factor of microstate j , as calculated in equilibrium statistical mechanics. The denominator of Eq. 2 is the partition function. The linear framework reduces to equilibrium statistical mechanics for systems which can reach thermodynamic equilibrium.

1.3 Independent generators at equilibrium

The cycle condition also implies that the edge labels are not independent quantities. Consider any cycle of reversible edges from vertex i to itself,

$$i = i_1 \xrightleftharpoons[b_1]{a_1} i_2 \xrightleftharpoons[b_2]{a_2} \cdots \xrightleftharpoons[b_{p-1}]{a_{p-1}} i_p \xrightleftharpoons[b_p]{a_p} i_{p+1} = i.$$

According to the cycle condition,

$$a_1 a_2 \cdots a_{p-1} a_p = b_p b_{p-1} \cdots b_2 b_1, \quad (4)$$

which implies an algebraic relationship between the labels. Eq. 3 shows that it is not the labels themselves which are important at equilibrium but the label ratios. If $i \xrightarrow{a} j$ and $j \xrightarrow{b} i$ are a pair of reversible edges, define $K_{i \rightarrow j} = a/b$, so that $K_{j \rightarrow i} = (K_{i \rightarrow j})^{-1}$. Eq. 4 can be rewritten in terms of these ratios as the algebraic relationship

$$\prod_{j=1}^{p-1} K_{i_j \rightarrow i_{j+1}} = 1, \quad (5)$$

The relations among the $K_{i \rightarrow j}$ arise in this way from cycles of reversible edges. For the applications below, it is important to know which sets of $K_{i \rightarrow j}$ constitute independent generators, so that, on the one hand, there are no algebraic relations between these generators ("independent"), while, on the other hand, all the other $K_{i \rightarrow j}$ can be "generated" as rational functions of the generators using Eq. 5. If T is any directed spanning tree of G , the label ratios over the edges of T , $\{K_{i \rightarrow j} \mid i \rightarrow j \in T\}$, constitute a set of independent generators. They are independent because there can be no cycle of reversible edges among edges which belong to a tree. Furthermore, if $i \rightarrow j \notin T$, then adding this edge to T cannot add new vertices to T because T is spanning. It must therefore create a cycle of reversible edges. If $u \rightarrow v$ is an edge on this cycle, which is not the same edge as $i \rightarrow j$, then either $u \rightarrow v$ is an edge in T or $v \rightarrow u$ is an edge in T . If the former, then $K_{u \rightarrow v}$ is a generator. If the latter, then $u \rightleftharpoons v \rightleftharpoons u$ is a cycle of reversible edges, so by Eq. 5, $K_{u \rightarrow v} = (K_{v \rightarrow u})^{-1}$ and $K_{u \rightarrow v}$ is the inverse of a generator. Note that this inverse property can be deduced from the cycle condition without having to know how the association constants are defined in terms of the labels. The $K_{i \rightarrow j}$ can now be written as a rational function of the generators using Eq. 5. Since G is connected, the number of edges in a spanning tree is one less than the number of vertices in G , so the number of independent generators is $N - 1$.

1.4 Steady states away from equilibrium

If the system does not reach thermodynamic equilibrium but the graph G is still strongly connected, then a basis element $\nu^G \in \ker \mathcal{L}(G)$ is given by the Matrix-Tree Theorem. Let $\Theta_j(G)$ be the set of spanning trees of G which are rooted at microstate j . Then, ν_j^G is given by taking $T \in \Theta_j(G)$, multiplying together the labels on the edges of T to form a monomial and adding up these monomials over all spanning trees in $\Theta_j(G)$,

$$\nu_j^G = \sum_{T \in \Theta_j(G)} \left(\prod_{k \rightarrow l \in T} a \right). \quad (6)$$

The structure of ν^G is considerably more complicated than that of ξ^G . A spanning tree contains many paths that reach the root and there are usually many spanning trees rooted at any vertex. In particular, G is strongly

connected if, and only if, it has a rooted spanning tree at any vertex, so that $\nu_j^G > 0$ for $1 \leq j \leq N$. Moreover, each spanning tree has $N - 1$ edges, so that the monomials which appear in ρ^G have degree $N - 1$ in the edge labels. It can be shown that if G satisfies detailed balance then $\nu^G = \lambda \xi^G$, for an appropriate scalar factor λ , which is a polynomial in the edge labels (Wong, Gunawardena, unpublished results). Accordingly, in calculating steady-state probabilities using Eq. 2, identical results are obtained at equilibrium with either $\rho^G = \xi^G$ and Eq. 3 or $\rho^G = \nu^G$ and Eq. 6. The linear framework extends equilibrium statistical mechanics to systems that are far from equilibrium.

1.5 Gene regulation functions

The linear framework can treat transcription as a complex process with multiple irreversible steps [1] but the Paper follows the thermodynamic formalism and assumes that gene expression is a fast process compared to gene regulation, so that the overall rate of gene expression can be treated as an average over the steady-state probabilities of the microstates, as given by u^* in Eq. 2. Let $r(i)$ be the rate of gene expression in microstate i . The overall rate of mRNA synthesis is given by the average

$$\frac{d}{dt}[\text{mRNA}] = \sum_{1 \leq i \leq N} r(i)u_i^* = \left(\sum_{1 \leq i \leq N} r(i)\rho_i^G \right) \left(\frac{1}{1 \cdot \rho^G} \right), \quad (7)$$

where we have used Eq. 2 in the second step. If the system is at equilibrium then Eq. 7 is calculated from $\rho^G = \xi^G$ using Eq. 3 and if it is away from equilibrium then Eq. 7 is calculated from $\rho^G = \nu^G$ using Eq. 6. In either case, the rate of mRNA synthesis can be expressed in terms of the edge labels, which may themselves contain the concentrations of transcription factors (TFs), as in the graph G_n in Paper Figure 2A.

Eq. 7 describes a gene regulation function for the rate of mRNA expression. The Paper uses a GRF for protein level. The additional assumptions required to obtain such a GRF are discussed in §5.1.

2 The graph G_n and background assumptions

We follow the notation introduced in the Paper. A single monomeric TF, T , is assumed to bind to n sites. The microstates are the $N = 2^n$ patterns of TF binding. It will be helpful to use the notation $\langle S \rangle$ for a microstate, where $S = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$, is the subset of sites bound by T . We will work with this notation rather than converting from $\langle S \rangle$ to an index $1, \dots, N$, as in §1. The edges in G_n correspond to binding of T to a site $i \notin S$ with label $a_{i,S}[T]$ or, in reverse, the unbinding of T from $\langle S \cup \{i\} \rangle$ with label $b_{i,S \cup \{i\}}$. To make the text more readable, we will avoid repeating requirements like $i \notin S$ from now on but will use the notation $S \cup \{i\}$ where necessary, as a reminder.

As discussed in the Paper, molecular mechanisms other than binding and unbinding of T are not directly represented in G_n but can exert their influence through the rate constants and through the dependence of the rate constants on the microstates. Such indirect influences could be exerted, for instance, by other transcription factors, co-regulators, nucleosomes or multi-protein complexes like Mediator. The main assumption is that the total levels of these indirect factors do not change with changes in $[T]$ but the factors may participate in the dynamics which gives rise to the edges of G_n and, thereby, to the numerical values of the labels on the edges.

Since edge labels have units of $(\text{time})^{-1}$, the rate constant $a_{i,S}$ has units of $(\text{concentration} \cdot \text{time})^{-1}$, while $b_{i,S}$ is a pure rate constant with units of $(\text{time})^{-1}$. The factor $[T]$ brings the nonlinearity of binding into the edge label, thereby allowing the dynamics to be treated in a linear manner, as in Eq. 1. To avoid additional complexity, it is conventional, in both the thermodynamic formalism and the linear framework to make the ‘‘no depletion’’ assumption, so that the binding of T is presumed not to deplete the free concentration of T . Accordingly, $[T] \approx T_{tot}$ and the labels are time-independent constants in the Laplacian dynamics, provided the total concentration of the TF is constant.

In the non-equilibrium case, a gene regulatory system can be maintained away from thermodynamic equilibrium through continual dissipation of free energy by the background dissipative mechanisms discussed in the Paper. The transitions between microstates due to binding and unbinding of T can become effectively irreversible and the graph G_n can thereby lose some of its edges. For simplicity, we consider only the case where the structure of G_n remains the same, so that each edge, which may represent an irreversible transition, has a reverse edge, which also represents an irreversible transition. For instance, if background phosphorylation of a nucleosome is required for a binding reaction, then dephosphorylation is assumed to be available for the unbinding reaction; phosphorylation and dephosphorylation are each individually irreversible reactions. With that simplification, the non-equilibrium case can be based on the same labelled, directed graph G_n as the equilibrium case and differs only in that detailed balance is no longer satisfied.

The number of vertices in G_n is 2^n . The number of edges can be determined as follows. There are $\binom{n}{i}$ microstates of order $\#S = i$ and each of these gives rise to a binding edge to $n - i$ microstates of order $\#S + 1$. Hence, the total number of binding edges is

$$\sum_{i=0}^n (n-i) \binom{n}{i} = \sum_{i=0}^n (n-i) \frac{n!}{i!(n-i)!} = n \sum_{i=0}^{n-1} \binom{n-1}{i} = n2^{n-1}.$$

The unbinding edges are in one-to-one correspondence with the binding edges. Accordingly, the total number of edges is $n2^n$.

If labels are ignored and the reversible edges of G_n are coalesced into a single undirected edge, the resulting undirected graph is the well-studied hypercube graph [4].

3 Equilibrium parameters for G_n

3.1 Equilibrium parameters $K_{i,S}$ and $\omega_{i,S}$

At equilibrium, it follows from the discussion in §1.3 that it is not the individual labels which are relevant but only the label ratios. In the binding direction, these take the form

$$K_{\langle S \rangle \rightarrow \langle S \cup \{i\} \rangle} = \frac{a_{i,S}[T]}{b_{i,S \cup \{i\}}}.$$

The factor $[T]$ has a different role to that of the constants $a_{i,S}$ or $b_{i,S \cup \{i\}}$. It occurs in every label ratio and becomes the independent variable in the gene regulation functions defined in §5.2. It must also cancel out of the relationships that come from the cycle condition in Eq. 5 since, if it did not, this would imply that $[T]$ was determined by the rate constants at equilibrium, which is impossible. It therefore only plays the role of a scale factor and it is easier to work with ratios from which it has been removed, which leads to the association constants defined in Paper Figure 2B,

$$K_{i,S} = \frac{a_{i,S}}{b_{i,S \cup \{i\}}} = \frac{K_{\langle S \rangle \rightarrow \langle S \cup \{i\} \rangle}}{[T]}. \quad (8)$$

The association constants have units of (concentration) $^{-1}$ and, from the previous calculation, there are $n \cdot 2^{n-1}$ such parameters.

For numerical calculations, it is more convenient and meaningful to use non-dimensional parameters. It is also helpful to identify higher-order cooperativities, since these are biologically significant. Both goals are realised by defining the higher-order cooperativities, $\omega_{i,S}$, to be

$$\omega_{i,S} = \frac{K_{i,S}}{K_{i,\emptyset}}.$$

These non-dimensional parameters capture how the binding of T to site i depends on the higher-order influence of the sites in S , as compared to the “bare” affinity in the absence of anything bound. Note that $\omega_{i,\emptyset} = 1$. To non-dimensionalise the bare association constants $K_{i,\emptyset}$, we normalise them to $K_{1,\emptyset}$,

$$\kappa_i = \frac{K_{i,\emptyset}}{K_{1,\emptyset}}.$$

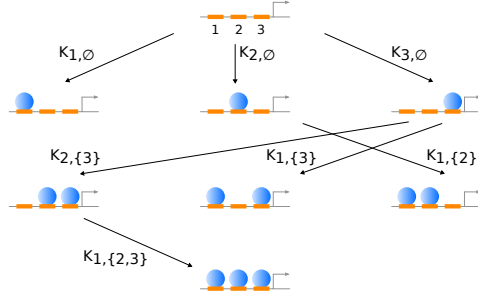
These are the non-dimensional parameters introduced in Paper Figure 2B, leaving $K_{1,\emptyset} = K_1$ as the only dimensioned parameter. The non-dimensional parameters are simply an alternative way of writing the $K_{i,S}$, which may be recovered from them and K_1 by noting that

$$K_{i,S} = \left(\frac{K_{i,S}}{K_{i,\emptyset}} \right) \left(\frac{K_{i,\emptyset}}{K_{1,\emptyset}} \right) K_{1,\emptyset} = \omega_{i,S} \kappa_i K_1. \quad (9)$$

3.2 Independent generators for the $K_{i,S}$

For numerical calculations, it is necessary to identify a set of independent generators. We do this first for the $K_{i,S}$ and then deduce the results for their non-dimensional equivalents, κ_i and $\omega_{i,S}$, which are used in the Paper.

Following the method described in §1.3, consider the directed subgraph P_n of G_n obtained by retaining only those binding edges, $\langle S \rangle \rightarrow \langle S \cup \{i\} \rangle$, such that, if $j \in S$ then $i < j$. As in the Paper, this last condition is abbreviated to $i < S$. The diagram below shows the directed spanning tree P_3 for G_3 , as described in §3.2. The edges are annotated with their corresponding association constants, which constitute a set of independent generators for the $K_{i,S}$. Note that this spanning tree is not rooted in the sense of Eq. 6.



Suppose that $\langle S \rangle$ is any microstate and that $S = \{i_1, \dots, i_k\}$, where the indices in S are ordered so that $1 \leq i_1 < \dots < i_k \leq n$. We can always find a path of reversible edges from the empty microstate to $\langle S \rangle$ of the form

$$\langle \emptyset \rangle \xrightleftharpoons[b_{i_k, \{i_k\}}]{a_{i_k, \emptyset} [T]} \langle \{i_k\} \rangle \xrightleftharpoons[b_{i_{k-1}, \{i_{k-1}, i_k\}}]{a_{i_{k-1}, \{i_k\}} [T]} \langle \{i_{k-1}, i_k\} \rangle \cdots \langle \{i_2, \dots, i_k\} \rangle \xrightleftharpoons[b_{i_1, S}]{a_{i_1, \{i_2, \dots, i_k\}} [T]} \langle S \rangle. \quad (10)$$

The binding edges in this path all satisfy the ordering condition just introduced. Hence P_n contains all microstates and is, therefore, a spanning subgraph. Moreover, P_n has the property that each microstate is the target of exactly one edge: if the microstate is $\langle S \rangle$, where $S = \{i_1, \dots, i_k\}$ with $1 \leq i_1 < \dots < i_k \leq n$, then the unique edge corresponds to the association constant $K_{i_1, \{i_2, \dots, i_k\}}$. Hence, P_n cannot have an undirected cycle. Accordingly, P_n is a directed spanning tree of G_n . It follows from §1.3, ignoring the factor $[T]$ for the reasons explained in §3.1, that the association constants $K_{i,S}$ with $i < S$, which we refer to as “ordered”, constitute a set of independent generators for the $K_{i,S}$.

As a check on this, the number of pairs (i, S) with $i < S \subseteq \{1, \dots, n\}$ can be counted as follows. For any given $1 \leq i \leq n$, there are 2^{n-i} subsets S satisfying the constraint $i < S$, so that the number of generators is

$$2^{n-1} + \dots + 2 + 1 = 2^n - 1,$$

as expected from the discussion in §1.3 since $N = 2^n$. The number of ordered association constants ($2^n - 1$) becomes substantially smaller than the total number of association constants ($n2^{n-1}$) as soon as $n \geq 3$.

3.3 The exchange formula for $K_{i,S}$

The discussion in §1.3 provides a method for expressing any $K_{i,S}$ in terms of the ordered ones. This is not straightforward to apply in practice as it requires locating where the edge corresponding to $K_{i,S}$ occurs in relation to P_n . The following alternative method provides a formula that is independently useful. Suppose that $i, j \notin S \subseteq \{1, \dots, n\}$. There is then a cycle of reversible edges from $\langle S \rangle$ to itself,

$$\langle S \rangle \xrightleftharpoons[b_{i,S \cup \{i\}}]{a_{i,S}[T]} \langle S \cup \{i\} \rangle \xrightleftharpoons[b_{j,S \cup \{i,j\}}]{a_{j,S \cup \{i\}}[T]} \langle S \cup \{i, j\} \rangle \xrightleftharpoons[a_{i,S \cup \{j\}}[T]]{b_{i,S \cup \{i,j\}}} \langle S \cup \{j\} \rangle \xrightleftharpoons[a_{j,S}[T]]{b_{j,S \cup \{j\}}} \langle S \rangle.$$

Applying Eq. 5 to this and cancelling $[T]$ (§3.1), leads to the exchange formula for association constants

$$K_{i,S} K_{j,S \cup \{i\}} = K_{j,S} K_{i,S \cup \{j\}}. \quad (11)$$

Eq. 11 is sufficient to calculate any $K_{i,S}$ in terms of the ordered ones. The proof of this is by induction on $\#S$. If $\#S = 0$ then $i < S$ for all $1 \leq i \leq n$, so the parameters $K_{i,\emptyset}$ are all ordered and there is nothing to prove. Now suppose that $\#S = k$, where $k > 0$, and that all higher-order cooperativities of the form $\omega_{i,U}$ with $\#U < k$ have been expressed in terms of the ordered parameters. Consider the k -th order cooperativity $\omega_{i,S}$ for any $1 \leq i \leq n$. Since $k > 0$, $S = \{i_1, \dots, i_k\}$ with $1 \leq i_1 < \dots < i_k \leq n$, then i could occur anywhere along this sequence. The exchange formula can be used to move i by a succession of swaps to before the left-hand end of the sequence, at which point the association constant becomes one of the ordered ones. This requires a second induction.

Let $\mu(i, S)$ denote the number of elements of S which precede i in the sequence:

$$\mu(i, S) = \#\{i_j \in S \mid i_j < i\}.$$

If $\mu(i, S) = 0$, then $\omega_{i,S}$ is one of the ordered parameters so there is nothing to prove. Suppose, as a second inductive hypothesis, that $\omega_{v,U}$ has been expressed in terms of the ordered parameters for all $\#U = k$ and for all $\mu(v, U) < p$. Suppose that $\mu(i, S) = p$, so that

$$i_1 < \dots < i_p < i < i_{p+1} < \dots < i_k.$$

Let $V = \{i_1, \dots, i_{p-1}, i_{p+1}, \dots, i_k\}$, so that $\#V = k - 1$ and $S = V \cup \{i_p\}$. It follows from the exchange formula in Eq. 11 that

$$K_{i,S} = \left(\frac{K_{i,V}}{K_{i_p,V}} \right) K_{i_p, V \cup \{i\}}.$$

Of the terms on the right, $K_{i,V}$ and $K_{i_p,V}$ have $\#V < k$ so that, by the first inductive hypothesis, they have been expressed in terms of the ordered parameters. Furthermore, $K_{i_p, V \cup \{i\}}$ has $\#(V \cup \{i\}) = k$ with $\mu(i_p, V \cup \{i\}) < p$ so that, by the second inductive hypothesis, it has also been expressed in terms of the ordered parameters. It follows that $K_{i,S}$ can be expressed in terms of the ordered parameters, thereby completing both inductions. The proof offers an algorithm for expressing $K_{i,S}$ in terms of the ordered association constants.

3.4 Independent non-dimensional κ_i and $\omega_{i,S}$

Paper Equation 3 introduces the following set of $2^n - 1$ parameters

$$K_1, \quad \kappa_i (i > 1), \quad \omega_{i,S} (i < S \neq \emptyset), \quad (12)$$

in which only K_1 is dimensioned and where the higher-order cooperativities $\omega_{i,S}$ are restricted to those which are ordered, as above for the $K_{i,S}$. It follows from Eq. 9 that the parameters in Eq. 12 are merely a rewriting of the

ordered $K_{i,S}$ and are therefore also a set of independent generators from which any $K_{i,S}$ can be recovered. The equilibrium GRFs described in §5.2 will be expressed in terms of the parameters in Eq. 12.

For calculations with higher-order cooperativities, a non-dimensional version of Eq. 11 can be obtained by dividing through by $K_{i,\emptyset}K_{j,\emptyset}$ to give

$$\omega_{i,S}\omega_{j,S\cup\{i\}} = \omega_{j,S}\omega_{i,S\cup\{j\}}, \quad (13)$$

Eq. 13 is the exchange formula for higher-order cooperativities given in **Paper Equation 2**. Eq. 13 provides an effective method for calculating any higher-order cooperativity in terms of the ordered ones by following the same algorithm as described above for the $K_{i,S}$.

4 Laplacian steady states of G_n

4.1 Steady states of G_n at equilibrium

At equilibrium, $\rho^{G_n} = \xi^{G_n}$ and the prescription in §1.2 can be used to calculate the latter. Eq. 10 provides a path of reversible edges from $\langle\emptyset\rangle$ to $\langle S\rangle$, where $S = \{i_1, \dots, i_k\}$ with the indices ordered so that $1 \leq i_1 < \dots < i_k \leq n$. It follows from Eq. 3 that, at thermodynamic equilibrium, the Laplacian steady state of $\langle S\rangle$ is given by

$$\xi_{\langle S\rangle}^{G_n} = K_{i_k,\emptyset}K_{i_{k-1},\{i_k\}} \cdots K_{i_1,\{i_2,\dots,i_k\}}[T]^k. \quad (14)$$

Eq. 14 can be non-dimensionalised using Eq. 9 to yield

$$\xi_{\langle S\rangle}^{G_n} = \kappa_{i_k}(\kappa_{i_{k-1}}\omega_{i_{k-1},\{i_k\}}) \cdots (\kappa_{i_1}\omega_{i_1,\{i_2,\dots,i_k\}})(K_1[T])^k.$$

Noting that $\omega_{i_k,\emptyset} = 1$, this may be rewritten more concisely as

$$\xi_{\langle S\rangle}^{G_n} = \left(\prod_{j=1}^k \kappa_{i_j} \omega_{i_j, \{i_{j+1}, \dots, i_k\}} \right) (K_1[T])^k, \quad (15)$$

which is expressed in the independent parameters of Paper Equation 3 (Eq. 12). As an illustration of Eq. 15, the Laplacian steady states of the 8 microstates of G_3 in Paper Figure 2A are listed in the table below. Note that K_1 in Eq. 15 plays the role of a scale factor (§6.2).

microstate	Laplacian steady state
$\langle\emptyset\rangle$	1
$\langle\{1\}\rangle$	$K_1[T]$
$\langle\{2\}\rangle$	$\kappa_2 K_1[T]$
$\langle\{3\}\rangle$	$\kappa_3 K_1[T]$
$\langle\{1, 2\}\rangle$	$\kappa_2 \kappa_1 \omega_{1,\{2\}} (K_1[T])^2$
$\langle\{1, 3\}\rangle$	$\kappa_3 \kappa_1 \omega_{1,\{3\}} (K_1[T])^2$
$\langle\{2, 3\}\rangle$	$\kappa_3 \kappa_2 \omega_{2,\{3\}} (K_1[T])^2$
$\langle\{1, 2, 3\}\rangle$	$\kappa_3 \kappa_2 \kappa_1 \omega_{2,\{3\}} \omega_{1,\{2,3\}} (K_1[T])^3$

4.2 History dependence away from equilibrium

At thermodynamic equilibrium, the Laplacian steady state can be calculated from a single path to each microstate, as in Eq. 14. All paths give the same answer because of detailed balance, which guarantees that the steady state does not depend on the history through which that state was reached. Equilibrium is fundamentally reversible. In marked contrast, away from thermodynamic equilibrium, Eq. 6 shows that all paths to a microstate must be

used to calculate the Laplacian steady state. The rooted spanning trees of the Matrix-Tree Theorem provide the bookkeeping for this calculation. Non-equilibrium systems are history dependent [1].

The number of spanning trees increases worse than exponentially in the size of a graph and the resulting combinatorial explosion is one of the fundamental challenges in analysing systems away from equilibrium [1]. In the case of G_n , the number of spanning trees rooted at any vertex (the number is the same for any vertex because G_n is reversible), is equal to the number of spanning trees in the undirected hypercube graph (§2), which is known to be [4],

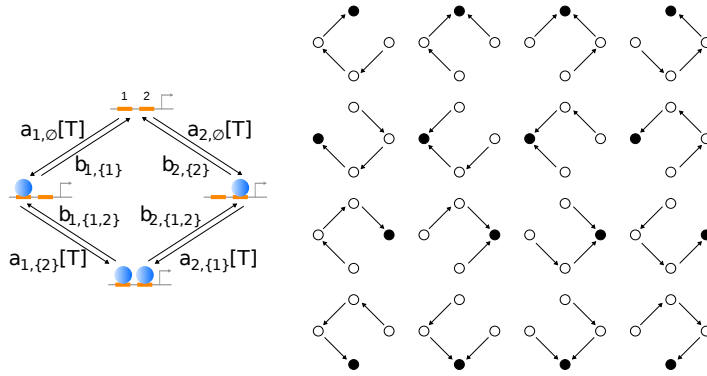
$$2^{2^n - n - 1} \prod_{k=2}^n k \binom{n}{k}.$$

For $n = 2$, this formula gives 4 spanning trees rooted at each vertex but for $n = 3$, it gives 384. The increase with n is extraordinarily rapid: G_4 has no less than 42,467,328 spanning trees rooted at each vertex. Accordingly, the calculation of position-steepness regions had to be restricted to G_3 , the complexity of whose GRF is already substantial.

It is important to appreciate that the size of a graph is not the dominant factor in the complexity of the Laplacian steady state. If a graph is forced to satisfy detailed balance, by making every edge reversible (which only increases the complexity of the graph) and imposing the cycle condition on the edge labels, then its Laplacian steady state is given by Eq. 22, which can be readily calculated even for very large graphs. Complexity arises primarily from non-equilibrium history-dependence.

4.3 Steady states of G_n away from equilibrium

Away from equilibrium, $\rho^{G_n} = \nu^{G_n}$, with the attendant history-dependent complexity described in §4.2. The algebraic structure also becomes different to that at equilibrium. The diagram below shows the graph G_2 on the left, annotated with its edge labels, and shows on the right the rooted spanning trees, with each row giving the spanning trees rooted at the vertex in black. Each tree has 3 edges, as expected for a graph with $2^n - 1$ vertices for $n = 2$ (§1.4). Each vertex has the same number of rooted spanning trees, a property that holds for any labelled, directed graph in which each edge is accompanied by a reverse edge.



Several properties can be inferred from this diagram which hold for general n . According to Eq. 6, the spanning trees rooted at a vertex give rise to monomials of the form

$$a_{i_1, S_1} \cdots a_{i_p, S_p} b_{j_1, S_{p+1} \cup \{j_1\}} \cdots b_{j_q, T_{p+q} \cup \{j_q\}} [T]^p, \quad (16)$$

where p is the number of binding edges and q is the number of unbinding edges and $p + q = 2^n - 1$. Notice that the power of $x = [T]$ can vary depending on the spanning tree, so that the Laplacian steady-state $\nu_{\langle S \rangle}^{G_n}$ is a polynomial in x , in contrast to a monomial at equilibrium.

It is not difficult to show that $\nu_{\langle S \rangle}^{G_n}$ has the following structure as a polynomial in x ,

$$\nu_{\langle S \rangle}^{G_n} = a_1 x^{q_1} + \cdots + a_k x^{q_k}, \quad (17)$$

where $q_1 < \cdots < q_k$ and the minimum and maximum degrees are given by

$$q_1 = \#S \quad \text{and} \quad q_k = 2^n - (n + 1) + \#S.$$

Note, in particular, that $\nu_{\emptyset}^{G_n}$ has a constant term which, unlike the equilibrium case, is different from zero, while $\nu_{\langle S \rangle}^{G_n}$ has the highest degree of $2^n - 1$ only when $S = \{1, \dots, n\}$.

4.3.1 Parameterisation away from equilibrium

The monomial structure in Eq. 16 makes it clear that the ratios of rate constants are no longer the appropriate parameters for the steady states away from equilibrium. However, non-dimensionalisation is still helpful for numerical calculation. We chose the non-dimensional parameters

$$\epsilon_{i,S} = \frac{a_{i,S}}{a_{1,\emptyset}} \quad \text{and} \quad \delta_{i,S \cup \{i\}} = \frac{b_{i,S \cup \{i\}}}{b_{1,\{1\}}}, \quad (18)$$

so that $a_{1,\emptyset}$ and $b_{1,\{1\}}$ are the only two dimensional parameters. It follows from Eq. 16 that the monomial arising from a rooted spanning tree can be rewritten

$$\epsilon_{i_1, S_1} \cdots \epsilon_{i_p, S_p} \delta_{j_1, T_1 \cup \{j_1\}} \cdots \delta_{j_q, T_q \cup \{j_q\}} (a_{1,\emptyset})^p (b_{1,\{1\}})^q [T]^p,$$

where $p + q = 2^n - 1$. We can now divide by the same factor $(b_{1,\{1\}})^{2^n - 1}$ in each monomial, to rewrite it as

$$\epsilon_{i_1, S_1} \cdots \epsilon_{i_p, S_p} \delta_{j_1, T_1 \cup \{j_1\}} \cdots \delta_{j_q, T_q \cup \{j_q\}} \left(\frac{a_{1,\emptyset}}{b_{1,\{1\}}} [T] \right)^p. \quad (19)$$

This division does not change any calculated GRF because the factor $(b_{1,\{1\}})^{2^n - 1}$ cancels between the numerator and the denominator. The quantities $a_{1,\emptyset}$ and $b_{1,\{1\}}$ act as scale factors (§6.2).

5 Gene regulation functions for G_n

5.1 The normalised GRF for protein level

The discussion the follows applies to both the equilibrium and non-equilibrium contexts. Eq. 7 shows how the rate of mRNA synthesis can be calculated in terms of the edge labels of G_n and, thereby, in terms of $[T]$. Call this function $s([T])$. Paper Equation 1 describes how Hb protein concentration level, $[\text{Hb}]$, depends on $[T]$, after normalisation. To go from mRNA synthesis rate to protein level, we implicitly assume that post-transcriptional mechanisms do not contribute to sharpness. Explicitly, we assume, first, that mRNA is linearly degraded at rate δ , so that the overall rate of mRNA production is given by

$$\frac{d}{dt}[\text{mRNA}] = s([T]) - \delta[\text{mRNA}].$$

Hence, the steady-state concentration of mRNA, $[\text{mRNA}]^*$, is

$$[\text{mRNA}]^* = \frac{1}{\delta} s([T]).$$

Second, we assume that the steady-state protein level, $[\text{protein}]^*$, is proportional to the steady-state level of mRNA, so that

$$[\text{protein}]^* = \frac{\alpha}{\delta} s([T]).$$

It follows that the steady-state protein level is proportional to $s([T])$. The gene regulation function for protein can now be calculated using Eq. 7,

$$[\text{protein}]^* = \left(\frac{\alpha}{\delta}\right) \left(\sum_{S \subseteq \{1, \dots, n\}} r(S) \rho_{\langle S \rangle}^{G_n} \right) \left(\frac{1}{1. \rho^{G_n}} \right), \quad (20)$$

where, to avoid excessive notation, we use the abbreviation $r(S)$ to denote the rate of gene expression in microstate $\langle S \rangle$.

5.1.1 T acts only as a transcriptional activator

In the Paper, it is assumed that T acts only as a transcriptional activator. This means that the binding of T to a microstate cannot decrease its rate of expression: if $\langle U \rangle$ and $\langle V \rangle$ are microstates and V includes all the sites of S , so that $U \subseteq V$, then the rate of expression in $\langle V \rangle$ cannot be smaller than that in $\langle U \rangle$, so that $r(U) \leq r(V)$. Note that this is the case for each of the expression strategies in Paper Figure 3A. It follows that the maximal rate of expression is $r(\{1, \dots, n\})$. This maximal rate is taken to be 1 and $r(\emptyset) = 0$, so that there is no baseline expression. Hence, $0 \leq r(S) \leq 1$. The maximum steady-state level of protein expression is then given in Eq. 20 by α/δ . For the gene expression strategies described in Paper Figure 3A, this maximum value is attained asymptotically as $[T] \rightarrow \infty$ (§5.2). If we normalise the steady-state protein concentration to this asymptotic maximum, the normalised gene regulation function, $f_n(x)$, is given by

$$f_n(x) = \left(\sum_{S \subseteq \{1, \dots, n\}} r(S) \rho_{\langle S \rangle}^{G_n} \right) \left(\frac{1}{1. \rho^{G_n}} \right). \quad (21)$$

Here, $x = [T]$ and $[T]$ appears through the expressions for ρ^{G_n} either at equilibrium using Eq. 3, as discussed in §5.2, or away from equilibrium using Eq. 6, as discussed in §5.4.

Because the components of both the equilibrium ρ^{G_n} in Eq. 15 and the non-equilibrium ρ^{G_n} in Eq. 17 are polynomials in $x = [T]$, $f_n(x)$ is a rational function of x (ie: a fraction in which the numerator and denominator are each polynomials in x). Furthermore, the denominator in Eq. 21, $1. \rho^{G_n}$, does not depend on the expression rate $r(S)$. Hence, the denominator of $f_n(x)$ is the same for different expression strategies at equilibrium and away from equilibrium.

Very little is known about transcription rates in individual microstates, making it difficult to assign values to $r(S)$ in Eq. 21, even after assuming that T is a transcriptional activator. Numerical exploration is challenging because, for each set of choices, $\{0 \leq r(S) \leq 1 \mid S \subseteq \{1, \dots, n\}\}$, a position-steepness region similar to Paper Figure 4 would have to be calculated by randomly varying the parameters. Instead, we chose three expression strategies (Paper Figure 3A)—two extreme (all-or-nothing, one-or-more) and one middle-of-the-road (average-binding)—which sample the range of possibilities. Our expectation is that the position-steepness regions for other sets of values of $r(S)$ would resemble that of average-binding (Paper Figure 4D) in lying on both sides of the Hill line but would approach \mathcal{H}_n at the tip of a cusp, as is the case for each of the three strategies analysed in the Paper.

5.2 Equilibrium gene regulation functions

At equilibrium, $\rho^{G_n} = \xi^{G_n}$, and the denominator, $1. \xi^{G_n}$, in the expression for $f_n(x)$ in Eq. 21 is the partition function of equilibrium statistical mechanics. The partition function is the same for all equilibrium GRFs, which

differ only in the numerator of the expression. According to Eq. 15, a microstate S with $\#S = k$ gives rise to a monomial in x^k . Hence, the partition function is given by the polynomial

$$1.\xi^{G_n} = 1 + c_1x + \cdots + c_nx^n,$$

where the coefficient c_k of x^k collects together, using Eq. 15, all microstates of order k ,

$$c_k = \left(\sum_{1 \leq i_1 < \cdots < i_k \leq n} \left(\prod_{j=1}^k \kappa_{i_j} \omega_{i_j, \{i_{j+1}, \dots, i_k\}} \right) \right) (K_1)^k. \quad (22)$$

This gives **Paper Equation 5**. The three gene expression strategies described in Paper Figure 3A then have the following GRFs, where c_k is defined in Eq. 22,

all-or-nothing $r(S) = 1$ if $S = \{1, \dots, n\}$ and $r(S) = 0$ otherwise. This gives **Paper Equation 4**,

$$f_n(x) = \frac{c_n x^n}{1 + c_1 x + \cdots + c_n x^n}. \quad (23)$$

one-or-more $r(S) = 1$ if $S \neq \emptyset$ and $r(\emptyset) = 0$. Hence,

$$f_n(x) = \frac{\sum_{k=1}^n c_k x^k}{1 + c_1 x + \cdots + c_n x^n}. \quad (24)$$

average-binding $r(S) = (\#S)/n$. Hence,

$$f_n(x) = \frac{(\sum_{k=1}^n k c_k x^k)/n}{1 + c_1 x + \cdots + c_n x^n}. \quad (25)$$

5.3 Similarity of $f_n(x)$ in the all-or-nothing strategy to $\mathcal{H}_n(x)$

The Hill function $\mathcal{H}_a(x) = x^a/(1 + x^a)$ has a numerator, x^a , which is identical to the highest-degree monomial in the denominator. Its algebraic form therefore corresponds most closely to a GRF in the all-or-nothing strategy (Eq. 23), at least when the Hill coefficient a is an integer. In this case, we can ask if a GRF $f_n(x)$ can be found which coincides in algebraic form with $\mathcal{H}_a(x)$. For obvious reasons, it is necessary that $a = n$. It is also clear from Eq. 23 that c_1, \dots, c_{n-1} must be made 0, while c_n must remain at 1. This cannot be done exactly because it follows from Eq. 22 that if $c_1 = 0$, then $K_1 = 0$, so that $c_i = 0$ for $1 \leq i \leq n$. However, it can be done approximately. If we set $\kappa_i = 1$ for $i > 1$, so that $K_{i,\emptyset} = K_1$, and $\omega_{i,S} = 1$ for $S \neq \{2, \dots, n\}$, then it follows from Eq. 22 that

$$c_k = \begin{cases} \binom{n}{k} (K_1)^k & \text{if } k < n \\ \omega_{1, \{2, \dots, n\}} (K_1)^n & \text{if } k = n. \end{cases}$$

Hence, if we choose K_1 in units of (concentration) $^{-1}$ to be very small compared to $\binom{n}{\lceil n/2 \rceil}^{-1}$, then, for $1 \leq k \leq n$,

$$(K_1)^k < K_1 \ll \frac{1}{\binom{n}{\lceil n/2 \rceil}} \leq \frac{1}{\binom{n}{k}} \leq 1,$$

so that c_k remains very small compared to 1 for $k < n$. However, we must then set

$$\omega_{1, \{2, \dots, n\}} = \frac{1}{(K_1)^n},$$

to achieve $c_n = 1$, which means that $\omega_{1, \{2, \dots, n\}} \gg 1$. The combination of the bare association constants, $K_{i,\emptyset}$, being very small and the highest-order cooperativity, $\omega_{1, \{2, \dots, n\}}$, being very large is biochemically implausible. A similar difficulty is encountered no matter what scheme is used to make c_k small for $k < n$. The Hill function is, therefore, not similar in algebraic form to a GRF for realistic parameter values.

5.4 Non-equilibrium gene regulation functions

Away from equilibrium, $\rho^{G_n} = \nu^{G_n}$. As in the equilibrium case, the denominator in Eq. 21 is the same in all strategies. Using Eq. 17, it can be seen to take the form,

$$1.\nu^{G_n} = e_0 + e_1x + \dots + e_{2^n-1}x^{2^n-1}.$$

Note the exponential increase in the degree from the partition function at equilibrium. Also, unlike the equilibrium case, $e_0 \neq 0$, as is seen from the table below, which gives the non-equilibrium steady states for the graph G_2 .

microstate	Laplacian steady state
$\langle \emptyset \rangle$	$b_{1,\{1,2\}}b_{1,\{1\}}b_{2,\{2\}} + b_{1,\{1\}}b_{2,\{2\}}b_{2,\{1,2\}} + (a_{2,\{1\}}b_{1,\{1,2\}}b_{1,\{1\}} + b_{1,\{1\}}b_{2,\{2\}}a_{1,\{2\}})[T]$
$\langle \{1\} \rangle$	$(b_{1,\{1,2\}}b_{2,\{2\}}a_{1,\emptyset} + b_{2,\{1,2\}}b_{2,\{2\}}a_{1,\emptyset})[T] + (b_{1,\{1,2\}}a_{2,\{1\}}a_{2,\emptyset} + b_{1,\{1,2\}}b_{2,\{2\}}a_{1,\emptyset})[T]^2$
$\langle \{2\} \rangle$	$(b_{1,\{1,2\}}b_{1,\{1\}}a_{2,\emptyset} + b_{2,\{1,2\}}b_{1,\{1\}}a_{2,\emptyset})[T] + (b_{2,\{1,2\}}a_{1,\{2\}}a_{2,\emptyset} + b_{2,\{1,2\}}a_{1,\{2\}}a_{1,\emptyset})[T]^2$
$\langle \{1,2\} \rangle$	$(b_{2,\{2\}}a_{1,\{2\}}a_{1,\emptyset} + b_{1,\{1\}}a_{2,\{1\}}a_{1,\emptyset})[T]^2 + (a_{2,\{1\}}a_{1,\{2\}}a_{1,\emptyset} + a_{2,\{1\}}a_{1,\{2\}}a_{2,\emptyset})[T]^3$

It is difficult to give an expression for the coefficients e_i similar to the equilibrium one in Eq. 22 because each coefficient has contributions from microstates of different orders.

all-or-nothing $r(S) = 1$ if $S = \{1, \dots, n\}$ and $r(S) = 0$ otherwise. Using the information about the structure of $\nu_{\langle \{1, \dots, n\} \rangle}^{G_n}$ in Eq. 17, it can be seen that the the GRF has the form given in **Paper Equation 7**,

$$f_n^{ne}(x) = \frac{d_n x^n + \dots + d_{2^n-1} x^{2^n-1}}{e_0 + e_1 x + \dots + e_{2^n-1} x^{2^n-1}}, \quad (26)$$

where $d_{2^n-1} = e_{2^n-1}$.

one-or-more $r(S) = 1$ if $S \neq \emptyset$ and $r(\emptyset) = 0$. Hence, using Eq. 17 again,

$$f_n^{ne}(x) = \frac{g_1 x + \dots + g_{2^n-1} x^{2^n-1}}{e_0 + e_1 x + \dots + e_{2^n-1} x^{2^n-1}}, \quad (27)$$

where, again, $g_{2^n-1} = e_{2^n-1}$.

average-binding $r(S) = (\#S)/n$. Hence, using Eq. 17 again,

$$f_n^{ne}(x) = \frac{h_1 x + \dots + h_{2^n-1} x^{2^n-1}}{e_0 + e_1 x + \dots + e_{2^n-1} x^{2^n-1}}, \quad (28)$$

where, again, $h_{2^n-1} = e_{2^n-1}$. Note that $h_i \neq g_i$.

For all these cases, history-dependent complexity and the overlapping contributions of different microstates make it hard to calculate the coefficients, d_i , g_i and h_i , in the numerator polynomials. However, for all expression strategies, the GRF $f_3^{ne}(x)$, which is the case discussed in the Paper, can be calculated using the software described in [1].

5.5 GRFs are increasing but not always sigmoidal

Gene regulation functions for a transcriptional activator T are expected to be increasing functions of $x = [T]$. More precisely, under the assumptions that we have made, we expect the following.

$$f_n(0) = 0, \quad f_n(x) \rightarrow 1 \text{ as } x \rightarrow \infty, \quad f_n(x) < f_n(y) \text{ if } 0 \leq x < y.$$

The first two parts of this are readily proved for all 6 GRFs (ie: for the all-or-nothing, one-or-more and average-binding strategies at equilibrium and away from equilibrium). The third part, that $f_n(x)$ is strictly increasing, is also readily proved for equilibrium GRFs by differentiating the expressions in Eqs. 23 to 25. However, strict increase is not so readily proved for the non-equilibrium GRFs in Eqs. 26 to 28. We therefore tested it numerically as follows. If f_n ceases to be increasing in $(0, \infty)$, then df_n/dx has a positive zero. This condition can be checked by considering only the numerator polynomial of df_n/dx . For each position-steepness plot, once a final converged boundary had been numerically found (§7.3), the corresponding parameters were passed to Mathematica, the coefficients of the relevant GRF were converted to numbers and the numerator polynomial of df_n/dx was calculated. Mathematica's native *CountRoots* function was used, with the polynomial variable specified to lie in the interval $(0, \infty)$, to determine the number of positive roots of the numerator polynomial. We found this to be zero in all cases, confirming that the non-equilibrium GRFs were all strictly increasing. It would be interesting from a mathematical point of view to find a rigorous proof of strict increase for any GRF arising from a transcriptional activator.

GRFs for a transcriptional activator are often assumed to be sigmoidal (“S-shaped”), by which is meant that df_n/dx increases to a maximum and then decreases steadily towards zero: $df_n/dx \rightarrow 0$ as $x \rightarrow \infty$ and df_n/dx has a single maximum for $x \in [0, \infty)$. The Hill functions exhibit this behaviour. Sigmoidality can be numerically checked by showing that df_n^2/dx^2 has a single zero for $x \in [0, \infty)$, which we did in a similar way to the previous paragraph. We found that GRFs for the equilibrium all-or-nothing strategy were always sigmoidal but that equilibrium GRFs for the one-or-more and average-binding strategies and non-equilibrium GRFs for any strategy were not always sigmoidal. Paper Figure 3B shows a typical GRF which is increasing but not sigmoidal.

6 Properties of the shape measures γ and ρ

6.1 γ and ρ for the Hill function \mathcal{H}_a

The position, γ , and the steepness, ρ , of a GRF are defined in Paper Equation 6 and these definitions apply equally well to the Hill functions $\mathcal{H}_a(x)$. Note that the Hill coefficient can be any real number $a > 0$. A straightforward calculation shows that

$$\frac{d\mathcal{H}_a}{dx} = \frac{ax^{a-1}}{(1+x^a)^2}, \quad \frac{d^2\mathcal{H}_a}{dx^2} = \frac{ax^{a-2}(a-1-(a+1)x^a)}{(1+x^a)^3}.$$

It follows that,

$$\gamma(\mathcal{H}_a) = \left(\frac{a-1}{a+1}\right)^{1/a} \quad \text{and} \quad \rho(\mathcal{H}_a) = \frac{a^2-1}{4a\gamma(\mathcal{H}_a)}.$$

6.2 γ and ρ do not depend on the scale factor

Recall from the Paper that the normalised GRF, $g_n(y)$, is defined by $g_n(y) = f_n(y.x_{0.5})$ where $y = x/x_{0.5}$ and $f_n(x_{0.5}) = 0.5$. It follows that

$$\frac{dg_n}{dy} = \frac{df_n}{dx} \frac{dx}{dy} = x_{0.5} \frac{df_n}{dx}. \quad (29)$$

Suppose that we scale the x value so that we replace x by cx in f_n . Let us call the resulting scaled GRF $f_n^c(x) = f_n(cx)$ and use superscript c in a similar way to denote other entities obtained from f_n^c . It follows that $0.5 = f_n^c(x_{0.5}^c) = f_n(cx_{0.5}^c)$, so that $x_{0.5}^c = x_{0.5}/c$. Then, using Eq. 29 twice,

$$\frac{dg_n^c}{dy} = x_{0.5}^c \frac{df_n^c}{dx} = \left(\frac{x_{0.5}}{c}\right) \left(\frac{df_n(cx)}{dx}\right) = x_{0.5} \frac{df_n}{dx} = \frac{dg_n}{dy}.$$

Hence, both the maximum derivative of dg_n/dy , which is the steepness $\rho(g_n)$, and its position, which is $\gamma(g_n)$, do not depend on the scale factor c .

This argument applies equally well to the scale factor K_1 at equilibrium (Eq. 15) and to the scale factors $a_{1,\emptyset}$ and $b_{1,\{1\}}$ away from equilibrium (Eq. 19).

7 Parameter ranges and search algorithms

7.1 Equilibrium parameter ranges

For numerical calculations at equilibrium, we chose non-dimensional parameters $\kappa_i, \omega_{i,S} \in [10^{-a}, 10^a]$ by logarithmic sampling. The default choice of $a = 3$ which was used for many of the regions shown in the Paper allows the bare association constants $K_{i,\emptyset}$ to vary by a factor of 10^3 and the higher-order association constants $K_{i,S}$ to vary by a factor 10^6 , both in comparison to K_1 . We felt this was sufficiently broad to reveal most properties while giving good numerical stability and computation time. We explored the parameter range as far as $a = 5$ for the all-or-nothing strategy at equilibrium but found no new features (Paper Figure 4C and Figure S1A).

7.2 Non-equilibrium parameter ranges

For numerical calculations away from equilibrium, we chose the non-dimensional parameters $\epsilon_{i,S}, \delta_{i,S \cup \{i\}} \in [10^{-2}, 10^2]$, for $i > 1$ or $S \neq \emptyset$, by logarithmic sampling (Paper Figure 6). If the gene regulatory system were at equilibrium, so that the association constants $K_{i,S} = a_{i,S}/b_{i,S \cup \{i\}}$ became meaningful, the chosen parameter range would allow $K_{i,S}$ to vary by a factor of 10^4 with respect to $K_{1,\emptyset}$. This allows the bare association constants, $K_{i,\emptyset}$, more leeway than the default factor of 10^3 used at equilibrium (§7.1) but the higher-order association constants, $K_{i,S}$, are now more constrained than the factor of 10^6 used at equilibrium. We felt that this permits a reasonable comparison between the equilibrium and non-equilibrium regions.

When the parameter range is increased, the non-equilibrium region for the all-or-nothing strategy behaves similarly to the equilibrium region and approaches \mathcal{H}_a at the tip of a cusp (Figure S2). However, in the non-equilibrium case, $a = 5$ when $n = 3$, unlike the equilibrium case, in which $a = 3$ when $n = 3$.

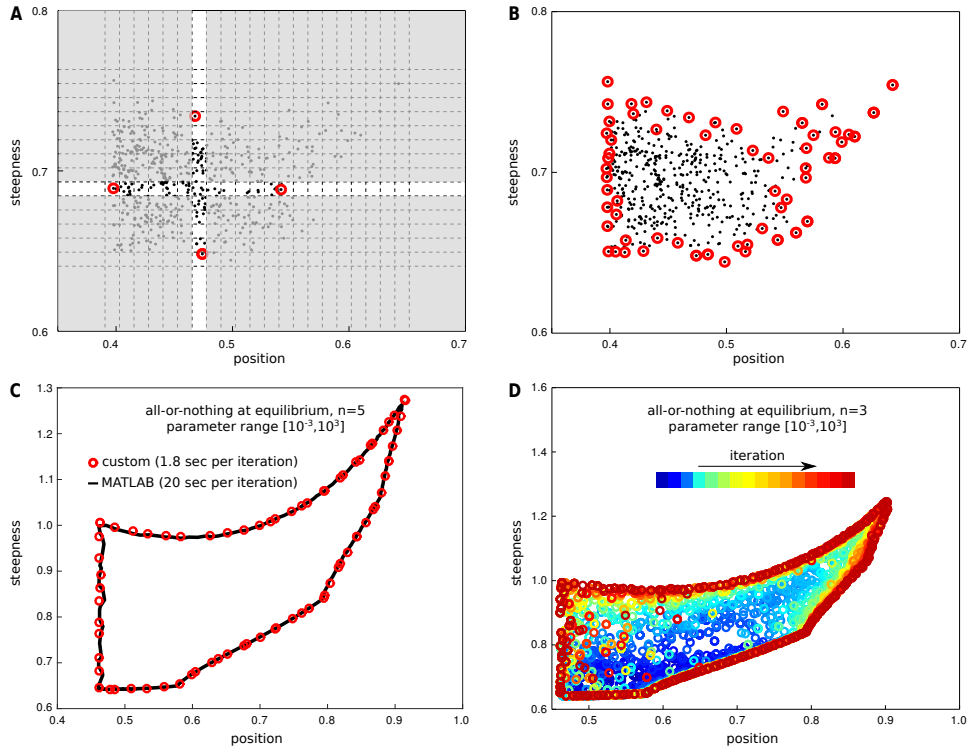
7.3 Biased sampling algorithm for position-steepness regions

The random sampling which yielded the probability density function in Paper Figure 4A does not identify the shape of the sampled region in (γ, ρ) space. We therefore developed an algorithm which biases the sampling towards the boundary of the region and used this to find the position-steepness regions shown in Paper Figures 4, 5 and 6. The algorithm has the following steps. The non-dimensional parameters are assumed to be sampled from the range $[10^{-a}, 10^a]$ for some a , while the scale factors are kept at 1 in their respective units.

1. **Generate the symbolic GRF.** We used previously-developed software¹ described in [1] for calculating ξ^{G_n} at equilibrium or ν^{G_n} away from equilibrium and assembled the resulting GRF, $f_n(x)$, in symbolic form in Mathematica. This symbolic GRF was passed to Matlab.
2. **Create an initial set of GRFs, Φ_{ran} .** An initial set of numerically-specified GRFs, Φ_{ran} , was generated by randomly sampling the non-dimensional parameters from the uniform distribution on $[10^{-a}, 10^a]$ and instantiating the symbolic GRF at these parameter values. For each such numerically-specified GRF, $\gamma(g_n)$ and $\rho(g_n)$ were computed as described in §7.3.1 and those GRFs which failed the position filter $\gamma(g_n) \notin [0.5\mathcal{H}_n, 1.5\mathcal{H}_n]$ were rejected. Typically, 100 GRFs were chosen to create an initial set, Φ_{ran} , of (γ, ρ) values in position-steepness space. A different method was used for the average-binding strategy with pairwise cooperativity only (Paper Figure 5B, bottom) because we found no GRFs which passed the position filter. They appear to very rare. We therefore ran the present algorithm until we found 20 GRFs which passed the position filter and then used these to form the initial set, Φ_{ran} . The variable $\Phi_{\mathcal{X}}$ is set equal to Φ_{ran} .

¹The software is freely and publicly available from <http://vcp.med.harvard.edu/software.html>.

3. **Identify the boundary of the set Φ_X , $\Phi_{ext} = \mathbf{extreme}[\Phi_X]$.** We did this in two ways using a fast, custom subroutine, `extreme.m`, for regions without re-entrant cusps and a slower Matlab subroutine, `boundary.m`, for regions with re-entrant cusps (as in Paper Figure 5A). In the custom subroutine, the γ and ρ axes were divided into small slices and the GRFs with minimum and maximum values of γ and ρ were found within each slice, as shown in part **A** of the diagram below. Typically, 20 equally spaced slices across the point cloud in each axis were sufficient to find enough extreme points for boundary finding, as in part **B** of the diagram. The advantage of this method is that it is not sensitive to differences in the density of points over the boundary. The disadvantage is that any re-entrant cusp on the boundary may only be partially revealed by horizontal and vertical slicing. In contrast, Matlab’s `boundary.m` subroutine finds the entire boundary, including those parts with re-entrant cusps. However, the non-uniform distribution of points in the steepness-position plots causes this subroutine to oversample the high-density (and low steepness) regions and to slow down significantly. We tested `extreme.m` against `boundary.m` and found no difference between the boundaries for regions without re-entrant cusps, as shown in part **C** of the diagram, but with a 10X difference in performance. We therefore used our custom subroutine `extreme.m` as the default and switched to Matlab’s `boundary.m` only when we encountered re-entrant cusps.



4. **Generate “mutated” GRFs from the extremes, $\Phi_{new} = \mathbf{mutate}[\Phi_{ext}]$.** Once the boundary is identified, for each GRF in the boundary, a new “mutated” GRF is constructed by sampling each non-dimensional parameter from the uniform distribution on $[0.5, 1.5]$ times its original value, while ensuring that the sampled value also lies in $[10^{-a}, 10^a]$. $\gamma(g_n)$ and $\rho(g_n)$ are then computed for the mutated GRF as described in §7.3.1 and GRFs which fail the position filter are rejected. Because there are many GRFs on the boundary, it was usually sufficient to generate just one mutated GRF for each GRF on the boundary. However, for pairwise-cooperativity only with the all-or-nothing strategy (Paper Figure 5A), we found that the regions grew very

slowly in discrete jumps, indicating that GRFs beyond the boundary were rare and were being found very slowly. We therefore generated 30 mutated GRFs for each GRF on the boundary and this substantially improved the convergence rate.

5. **Create an enlarged set of extreme GRFs, $\Phi_{all} = \Phi_{ext} \cup \Phi_{new}$.** The mutated GRFs could be less extremal than their parents so they are adjoined to the previous ones to form an enlarged set of extremal GRFs. The variable $\Phi_{\mathcal{X}}$ is then set equal to Φ_{all} , control is returned to Step 3 and the the loop is repeated a fixed number of times.
6. **Generate a smooth boundary for plotting.** Once the algorithm had converged (see below), we used Matlab's `boundary.m` algorithm to re-compute the final boundary, which Matlab provides in a data structure that is more convenient for plotting. The boundaries shown in Paper Figures 4, 5 and 6 were generated in this way.

With each iteration of the algorithm, the extreme points found in Step 3 approach closer to the boundary and eventually condense at the boundary, as shown in part **D** of the diagram above. Because the algorithm chooses random GRFs by mutation in Step 4, we found that the smoothness of the boundary was a good test of whether the algorithm had converged. However, we also undertook further tests as described in §7.4

7.3.1 Calculating γ and ρ

The position, γ , and steepness, ρ , have to be calculated for each GRF that is generated by the algorithm above. This entails multiple evaluations of the derivative and we found it necessary for performance reasons to optimise this calculation. A numerically-specified GRF, $f_n(x)$, is given as a rational function of x with numerical coefficients. Its derivative, df_n/dx , was evaluated at $x = x_1$ as follows.

- At equilibrium, if $n \leq 7$, df_n/dx was calculated by symbolically differentiating f_n with respect to x and then evaluating this expression at $x = x_1$.
- At equilibrium, if $n > 7$, $f_n(x)$ was expressed as a fraction, $f_n(x) = p(x)/q(x)$, where $p(x)$ and $q(x)$ are polynomials in x , and the expression for the derivative of a fraction,

$$\frac{p'(x_1)q(x_1) - p(x_1)q'(x_1)}{q(x_1)^2},$$

was evaluated, where $p'(x)$ and $q'(x)$ are the symbolic derivatives of the polynomials.

- Away from equilibrium, the approximation for the derivative

$$\left. \frac{df_n}{dx} \right|_{x=x_1} \approx \frac{f_n(x_1 + \Delta) - f_n(x_1 - \Delta)}{2\Delta}$$

was calculated with $\Delta = 10^{-8}$.

To then determine γ and ρ , $x_{0.5}$ was first estimated by using the Matlab subroutine `fzero` to solve the equation $f_n(x_{0.5}) = 0.5$. The derivative df_n/dx , calculated as described above, was then evaluated at the points

$$x_i = i \left(\frac{x_{0.5}}{10^3} \right),$$

for i running from $i = 0$ to 10^4 , thereby covering the interval $[0, 10x_{0.5}]$. The maximum derivative in this list was taken to be $\rho(f_n)$ and the corresponding x at which the maximum was found was taken to be $\gamma(f_n)$. Finally, the position and steepness of the normalised GRF was calculated by using Eq. 29, so that

$$\gamma(g_n) = \frac{\gamma(f_n)}{x_{0.5}} \quad \text{and} \quad \rho(g_n) = x_{0.5}\rho(f_n).$$

The choices made here were determined after several numerical experiments but the overall accuracy of the algorithm was assessed by the tests described next.

7.4 Testing convergence and numerical accuracy

We tested convergence of the biased sampling algorithm as follows. Starting from the converged boundary to be tested, we iterated the algorithm in §7.3 10 times but now generated 10^3 mutated GRFs for each boundary GRF in Step 4, thereby oversampling the boundary. For each region, $0 \leq i \leq 10$, where 0 is the starting region, we computed its area a_i using Matlab's `boundary .m` subroutine and then calculated the MAI, or “Mean Area Increase” over the iterations, as given by

$$\frac{1}{10} \left(\sum_{0 \leq i < 10} \frac{a_{i+1} - a_i}{a_i} \right). \quad (30)$$

Since mutated GRFs are added to the existing GRFs in Step 5, the area can only increase, so that $a_i \leq a_{i+1}$. The MAI scores for each of the boundaries shown in the Paper are show in the table below.

case	#sites	#GRFs	#iterations	MAI	MDI
Paper Figure 4C					
$p = 1$	5	1099	1000	4.8×10^{-4}	1.6×10^{-4}
$p = 2$	5	1995	1000	1.9×10^{-4}	1.8×10^{-4}
$p = 3$	5	22418	1000	8.4×10^{-4}	$6.6 \times 10^{-3*}$
$p = 5$	5	4836	1000	5.1×10^{-4}	3.7×10^{-3}
Paper Figure 4D					
AN	5	22418	1000	8.4×10^{-4}	$6.6 \times 10^{-3*}$
OM	5	15060	1000	2.7×10^{-4}	1.5×10^{-3}
AB	5	15044	1000	1.1×10^{-3}	9.9×10^{-4}
Paper Figure 5A					
AN	3	4704	1000	9.3×10^{-7}	2.5×10^{-5}
AN	4	4886	1000	3.7×10^{-5}	1.5×10^{-5}
AN	5	4773	1000	4.0×10^{-6}	3.1×10^{-5}
AN	6	2878	2000	1.6×10^{-4}	4.9×10^{-5}
AN	7	2157	2000	8.4×10^{-5}	5.4×10^{-5}
AN	8	1338	6000	7.8×10^{-5}	2.7×10^{-3}
AN	9	782	6000	9.3×10^{-4}	6.8×10^{-3}
AN	10	833	6000	1.3×10^{-4}	2.7×10^{-3}
AN	11	553	6000	2.8×10^{-5}	3.6×10^{-3}
AN	12	644	6000	3.1×10^{-4}	3.3×10^{-3}
Paper Figure 5B, top					
OM	3	3748	1000	7.3×10^{-3}	1.0×10^{-4}
OM	4	3692	1000	1.4×10^{-2}	4.7×10^{-5}
OM	5	7926	1000	1.1×10^{-3}	1.0×10^{-4}
OM	6	7502	1000	1.1×10^{-2}	6.3×10^{-4}
OM	7	23068	1000	9.0×10^{-3}	1.1×10^{-4}
OM	8	1818	1000	2.7×10^{-2}	1.0×10^{-4}
OM	9	11014	1000	1.1×10^{-2}	1.1×10^{-4}
OM	10	5528	1000	2.0×10^{-2}	1.0×10^{-4}
OM	11	6516	1000	8.1×10^{-2}	1.0×10^{-4}
OM	12	3041	1000	2.8×10^{-2}	1.1×10^{-4}
Paper Figure 5B, bottom					
AB	3	37376	1000#	6.3×10^{-6}	2.3×10^{-2}
AB	4	4065	1000#	4.0×10^{-6}	4.9×10^{-4}
AB	5	37405	1000#	0 [†]	1.7×10^{-2}
AB	6	37401	1000#	0 [†]	1.0×10^{-2}
AB	7	37362	1000#	0 [†]	8.6×10^{-3}
AB	8	37384	1000#	0 [†]	1.2×10^{-2}
AB	9	2893	1000#	0 [†]	8.9×10^{-3}
AB	10	130	1000#	0 [†]	1.2×10^{-3}
Paper Figure 6					
AN	3	292	3000	4.3×10^{-3}	1.0×10^{-3}
OM	3	293	6000	7.0×10^{-3}	5.0×10^{-3}
AB	3	355	6000	6.2×10^{-3}	4.8×10^{-3}

The abbreviations AN for all-or-nothing, OM for one-or-more and AB for average-binding have been used in this table. The column for #iterations gives the number of iterations of the algorithm after the initial set was found. For the average-binding strategy with pairwise cooperativity only, we used a different strategy to generate the initial set of GRFs (see Step 2 of the algorithm in §7.3) and this is indicated by the symbol #; also in this case, when the number of sites (#sites) was 5 or more, the convergence test created no new GRFs on the boundary, so that the MAI score was identically zero and this is indicated by the symbol †. Where possible, we checked all GRFs generated by the algorithm in §7.3 but when there were too many because of high numbers of iterations, we restricted attention to the GRFs on the final converged boundary. The symbol * in the table signifies a duplicated dataset.

The table shows that the MAI scores are never worse than 0.1—these worst cases occur for the one-or-more strategy with pairwise cooperativity only, as in Paper Figure 5B, top, for which the regions are very small—and are usually less than 0.001, which we considered to be good evidence that the boundary had converged. We note, however, that boundary finding depends on randomly generating GRFs in Step 4 of the algorithm and we cannot rule out the possibility that increased iterations would occasionally find extremely rare GRFs outside the boundaries that we have shown.

As a further test of the algorithm and its numerical accuracy, we undertook an independent calculation of the region using the same parameter values. We recorded all the parameter values generated by the algorithm in §7.3 and passed them to Mathematica. We instantiated the symbolic GRF at each set of parameter values to generate a GRF, f_n , with numerical coefficients. We used the native `Solve[]` capability in Mathematica to normalize x to the half-maximal concentration $x_{0.5}$ and defined $g_n(y) := f_n(x_{0.5}y)$. Like f_n , g_n has numerical, not symbolic, coefficients. We calculated d^2g_n/dy^2 as a function of y and used the `Solve[]` function again to obtain values of y where $d^2g_n/dy^2 = 0$. This process sometimes returned very large values of y , corresponding to where $g_n(y)$ is nearly at its asymptotic limit and $dg_n/dy \rightarrow 0$. To avoid this numerical artifact, we imposed the constraint that $y < 10^3$. We note that y is non-dimensional and the position of the maximum derivative was never found beyond $y = 2$, so the limit of 10^3 seems safe. Having found the values of y at which dg_n/dy has a turning point (ie: either a local maximum or a local minimum), we substituted these values into dg_n/dy and took their maximum, thereby obtaining $\rho(g_n)$, and took the corresponding value of y to be $\gamma(g_n)$. To compare this with the values obtained by the biased sampling algorithm, we took, for each GRF, the Euclidean distance between the position-steepness coordinates found by the algorithm in Matlab, (γ^A, ρ^A) , and those found by the test just described in Mathematica, (γ^T, ρ^T) . We defined the MDI, or Maximum Distance, to be the maximum of this value over all the GRFs generated in calculating a particular boundary:

$$\max_{g \in \text{GRFs}} \sqrt{(\gamma^A(g) - \gamma^T(g))^2 + (\rho^A(g) - \rho^T(g))^2}. \quad (31)$$

The MDI scores for each of the boundaries shown in the Paper are given in the table below, with the column for #GRFs showing the number of GRFs that were checked to evaluate the MDI.

The MDI scores are never worse than 1.0×10^{-2} —these worst cases occur for the average-binding strategy with pairwise cooperativity only, as in Paper Figure 5B, bottom—and are usually an order of magnitude better. We considered this to be evidence of good numerical accuracy.

The MDI test not only assessed numerical accuracy, it was also an effective method for identifying mistakes in our overall work flow, including in file handling, software coding, numerical implementation and algebraic manipulation, which we then corrected. The algorithm calculations in Matlab were carried out by JE and the independent tests in Mathematica were carried out by FW and this cross-checking also helped with the detection of errors.

SUPPLEMENTAL REFERENCES

- [1] T. Ahsendorf, F. Wong, R. Eils, and J. Gunawardena. A framework for modelling gene regulation which accommodates non-equilibrium mechanisms. *BMC Biol.*, 12:102, 2014.
- [2] J. Gunawardena. A linear framework for time-scale separation in nonlinear biochemical systems. *PLoS ONE*, 7:e36321, 2012.
- [3] J. Gunawardena. Time-scale separation: Michaelis and Menten’s old idea, still bearing fruit. *FEBS J.*, 281:473–88, 2014.
- [4] F. Harary, J. P. Hayes, and H.-J. Wu. A survey of the theory of hypercube graphs. *Comput. Math. Applic.*, 15:277–89, 1988.
- [5] I. Mirzaev and J. Gunawardena. Laplacian dynamics on general graphs. *Bull. Math. Biol.*, 75:2118–49, 2013.