Principles of Computation by Competitive Protein Dimerization Networks

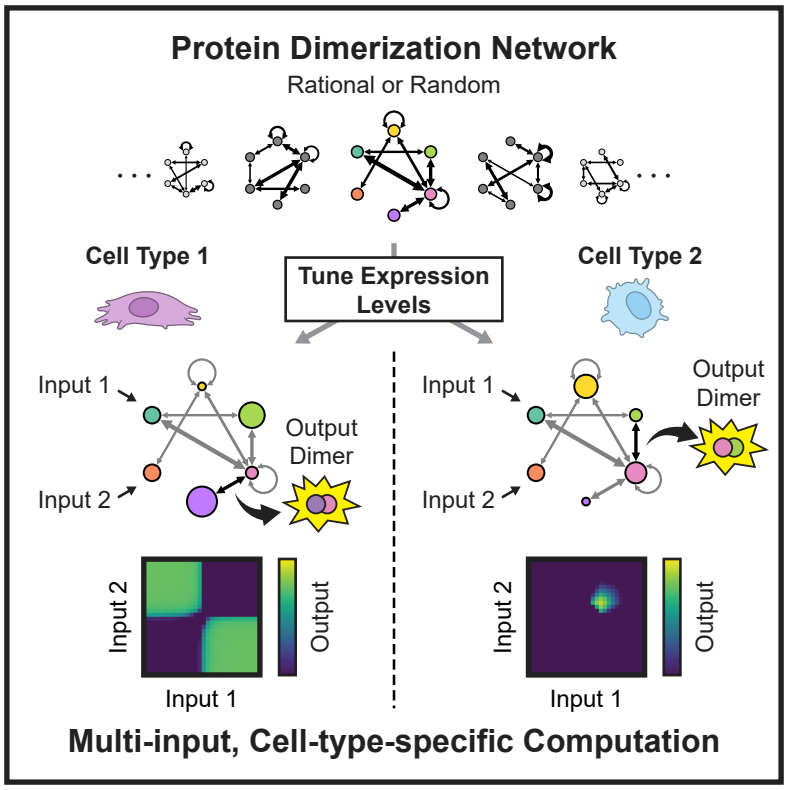
Jacob Parres-Gold,[1,2] Matthew Levine,[3] Benjamin Emert,[1] Andrew Stuart,[4]
Michael B. Elowitz[1,5]*

[1]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA 91125, USA
[2]Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, CA 91125, USA
[3]Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA
[4]Division of Engineering and Applied Sciences, California Institute of Technology, Pasadena, CA 91125, USA
[5]Howard Hughes Medical Institute, Chevy Chase, MD 20815, USA

*correspondence: melowitz@caltech.edu

Keywords: competitive dimerization; biological computation; protein-protein interaction networks; computational modeling; expressivity

Graphical Abstract



Summary

Many biological signaling pathways employ proteins that competitively dimerize in diverse combinations. These dimerization networks can perform biochemical computations, in which the concentrations of monomers (inputs) determine the concentrations of dimers (outputs). Despite their prevalence, little is known about the range of input-output computations that dimerization networks can perform (their "expressivity") and how it depends on network size and connectivity. Using a systematic computational approach, we demonstrate that even small dimerization networks (3-6 monomers) are *expressive*, performing diverse multi-input computations. Further, dimerization networks are *versatile*, performing different computations when their protein components are expressed at different levels, such as in different cell types. Remarkably, individual networks with random interaction affinities, when large enough (≥8 proteins), can perform nearly all (~90%) potential one-input network computations merely by tuning their monomer expression levels. Thus, even the simple process of competitive dimerization provides a powerful architecture for multi-input, cell-type-specific signal processing.

Introduction

Many biochemical signal processing circuits employ families of proteins that competitively dimerize with one another in diverse combinations. For example, the motif of many-to-many dimerization can be found in transcription factor families such as the nuclear receptor (NR)[1] (Figure 1A), basic leucine zipper (bZIP),[2–4] basic helix loop helix (bHLH),[5] and MADS-box proteins,[6] among others.[7] These dimerization networks regulate genes involved in a vast array of biological processes, including cell proliferation,[8–10] differentiation,[11–13] hormone signaling,[14,15] stress responses,[16] and circadian rhythms.[17] Similar dimerization (or higher-order multimerization) networks also occur in ligand-receptor signaling,[18–20] adhesion,[21,22] and other systems.[23] In addition to their many-to-many patterns of dimerization (Figure 1B), these proteins commonly exhibit diverse but overlapping expression profiles across different cell types,[24] calling into question how these cell types might interpret input signals differently (Figure 1C).

The process of competitive dimerization can be thought of as a biochemical input-output computation. In this perspective, upstream signals modulate the concentrations or activities of specific monomers (inputs), altering the distribution of biologically active dimers (outputs).[25,26] For example, steroid hormones bind and activate cognate nuclear receptors (NRs), allowing them to dimerize with other NR monomers (Figure 1A). The resulting dimers can bind to sites across the genome to regulate downstream genes. In this work, we use the terms "computation" and "response function" to refer to the quantitative relationship between the concentrations of input monomer(s) and an output dimer.

Experimental work has long demonstrated that natural dimerization networks can exhibit complex and cell-type-specific modes of signal processing. In 1993, Neuhold and Wold suggested that networks of dimerizing bHLH transcription factors could allow changes in just a few monomers (inputs) to "radiate" throughout the network, changing the concentrations of dimers

(outputs) to generate "major changes in cellular phenotype."[27] More recent experimental work studying the bone morphogenetic protein (BMP) signaling pathway has also observed complex and cell-type-specific responses. In the BMP pathway, ligand inputs associate with multiple receptor subunits to form complexes that phosphorylate SMAD proteins, which promote the expression of downstream target genes.[18] When multiple ligand variants are present in combination, they compete for a limited pool of receptor variants, potentially antagonizing or synergizing with each other. For instance, two ligands can activate the BMP pathway independently, but not in combination. Thus, the BMP pathway appears to use many-to-many interactions to perform computations on combinations of ligand inputs. Further, different cell types, which express BMP receptor variants at different abundances, can respond differently to the same ligand combinations.[19,28]

However, dimerization is a relatively limited type of biochemical interaction. It does not consume energy and is stoichiometric rather than catalytic, rendering it incapable of amplifying the magnitude of input signals.[25] Whereas computation by enzymatic networks has been studied by analogy to feed-forward neural networks,[29–31] dimerization networks thus appear to belong to a distinct class of computational systems. Therefore, while the experimental characterization of particular dimerization networks in nature has provided great insights,[11] we lack a fundamental, systems-level understanding of dimerization network computation – including which computations are (and are not) possible, to what extent a single network can perform different computations in different cell contexts, and how parameters such as network size and connectivity influence their computational power. Addressing these questions is essential for understanding the prevalence, architectures, expression patterns, and signal-processing functions of natural dimerization networks.

Here, to study dimerization network computation in general, we constructed a minimal model that captures the key features of natural dimerization networks: competitive, many-to-many dimerization interactions of varying strengths and cell-type-specific component expression levels (Figure 1D). We borrow the term *expressivity* from the field of neural network computation to describe the range of all quantitatively unique functions that may be performed by a class of dimerization networks across all physiologically reasonable parameter values (Figure 1E).[32,33] Further, we use the term *versatility* to describe the ability of the same network of proteins to perform different functions when network proteins are expressed at different abundances (such as in different cell types). Network versatility would allow a single set of proteins to exhibit the cell-type-specific patterns of signal interpretation that are prevalent in nature.[19,34–37]

In this work, we use both random parameter screens and optimization to characterize the expressivity and versatility of competitive dimerization networks. We find that dimerization networks can compute a variety of non-monotonic functions on multiple inputs, including essential information processing tasks such as bandpass filters and logic gates. We investigate how network expressivity and versatility vary with network size and connectivity (Figure 1F) and use our results to contextualize the features of natural dimerization networks. We then demonstrate that dimerization networks can readily perform computations on multiple inputs, including all

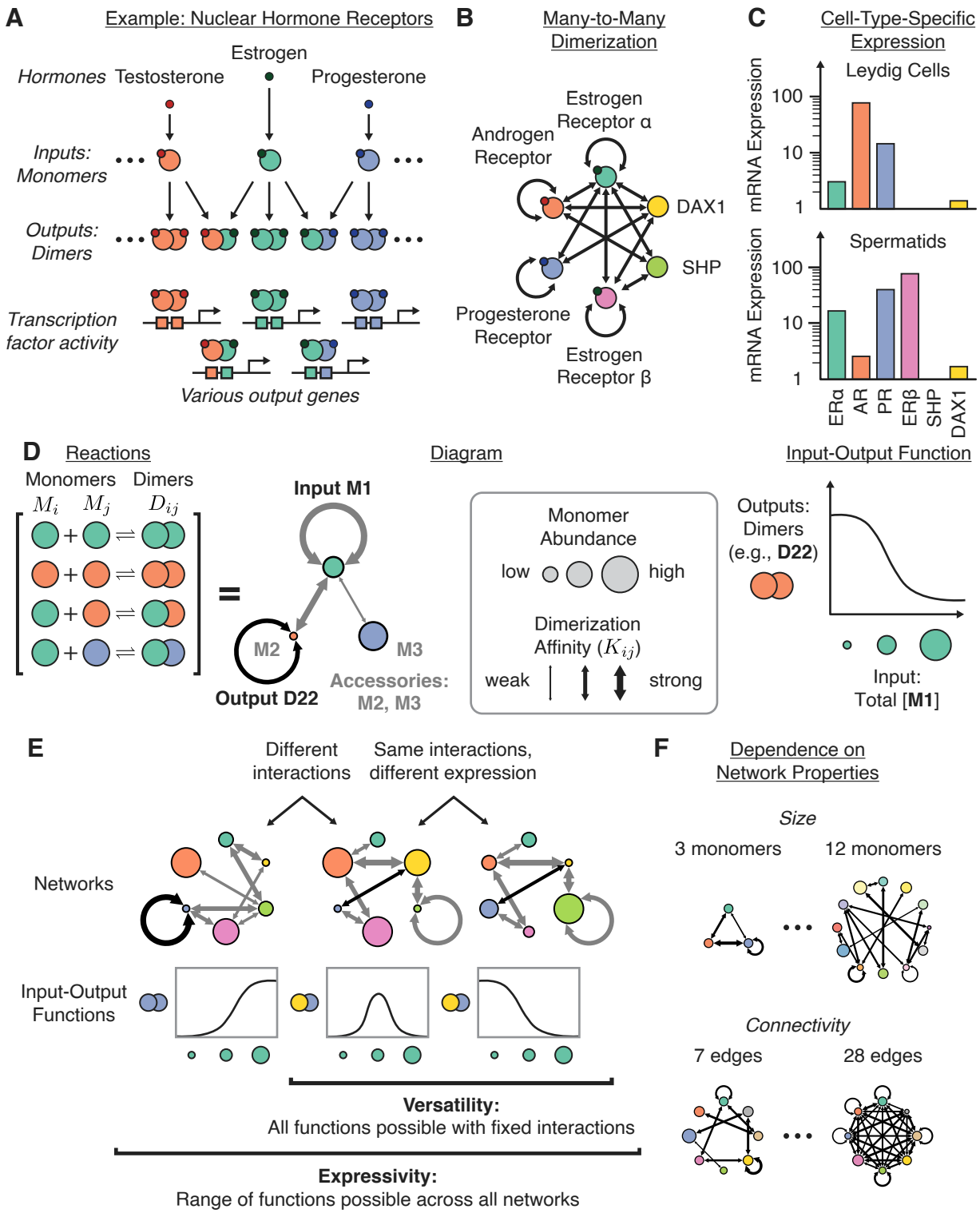

A
Example: Nuclear Hormone Receptors
Hormones
Testosterone
Estrogen
Progesterone
Inputs: Monomers
Outputs: Dimers
Transcription factor activity
Various output genes
B
Many-to-Many Dimerization
Estrogen Receptor α
Androgen Receptor
DAX1
SHP
Progesterone Receptor
Estrogen Receptor β
C
Cell-Type-Specific Expression
Leydig Cells
Spermatids
mRNA Expression
100
10
1
ERα
AR
PR
ERβ
SHP
DAX1
D
Reactions
Monomers
Dimers
M_i
M_j
D_{ij}
Diagram
Input M1
M2
M3
Output D22
Accessories: M2, M3
Monomer Abundance
low
high
Dimerization Affinity (K_{ij})
weak
strong
Input-Output Function
Outputs: Dimers (e.g., D22)
Input: Total [M1]
E
Different interactions
Same interactions, different expression
Networks
Input-Output Functions
Versatility:
All functions possible with fixed interactions
Expressivity:
Range of functions possible across all networks
F
Dependence on Network Properties
Size
3 monomers
12 monomers
Connectivity
7 edges
28 edges

Figure 1. Competitive dimerization networks compute. (A) In natural dimerization networks, such as the nuclear receptors (NRs), upstream signals regulate the activities of individual monomers, which dimerize to perform biochemical outputs. (B) Network monomers dimerize in a many-to-many fashion; shown are the putative dimerization patterns of several NRs. (C) Different cell types, such as spermatids and Leydig cells, express NR monomers at different abundances, reported in normalized transcripts per million (TPM) from the Human Protein Atlas.[78] (D) Dimerization networks are modeled here with pairwise affinities (shown as arrow widths) and monomer expression levels (shown as circle sizes) as parameters. Titrating input monomer(s), with a single dimer as an output, yields an input-output function. (E) Here, expressivity refers to the collection of input-output functions that may be performed by all networks of a given class, while versatility refers to the functions that may be performed by a single set of proteins with fixed interactions but variable expression levels. (F) Here, we investigate how network expressivity and versatility scale with both network size and connectivity.

---

three-input logic gates. Finally, we show that even networks with random protein-protein interaction affinities, when large enough, can perform a wide variety of functions solely by adjusting the expression levels of their monomer components.

Results

Modeling competitive dimerization networks

We first sought to establish a minimal modeling framework that captures the key features of natural dimerization networks described above: competitive, many-to-many dimerization interactions of varying strengths and variation among cell types in component expression levels. We consider networks of *m* interacting monomers M_1, M_2, … M_m. Each pair of monomers M_i and M_j may reversibly bind, with equilibrium constant K_{ij}, to form the dimer D_{ij} (Figure 1D). The total concentration of each species is the sum of its free form and all of its dimers:

$$[\mathrm{M}_i]_{\mathrm{total}} = [\mathrm{M}_i] + \sum_j [\mathrm{D}_{ij}]$$

To consider these networks as feed-forward computational systems, we designate a subset of monomers as *inputs.* The expression level of an input monomer depends on a corresponding input signal. We term non-input monomers *accessories* and assume that each accessory protein has a fixed total concentration in a given cell type. We consider dimer concentrations as the outputs directly, assuming that cells could use various dimerization-dependent biochemical activities to carry out downstream functions.

In this framework, a given network in a given cell type is completely specified by its set of pairwise affinities, K_{ij}, and the total concentrations of each of the accessory monomers, $[M_i]_{total}$. To simulate the input-output function of a network, we determine the equilibrium concentrations of all network species across a titration of the input monomer(s) (see Methods). This framework does not consider the formation of higher-order oligomers, such as trimers, and also neglects the

potential impacts of DNA binding or subcellular localization on dimerization propensities.[7] These features could further expand the computational potential of these systems beyond what is described below.

Networks of dimerizing proteins can compute a wide variety of functions

What types of functions can dimerization networks compute? To address this question, we first analyzed the input-output behaviors of minimal, elementary networks. The simplest non-trivial dimerization network comprises just two monomers (Figure 2A). In this network, an input monomer, M_1, can heterodimerize with an accessory monomer, M_2, preventing it from forming an output dimer, D_{22}. Increasing the total concentration of M_1 induces the formation of D_{12} heterodimers, which in turn sequesters M_2 and reduces the D_{22} homodimer. Thus, this network computes a *switch-off* function with output dimer D_{22}.

Adding an additional species can invert this circuit to compute a *switch-on* function (Figure 2B). In such a network, the input monomer M_1 can heterodimerize with an accessory monomer, M_2, which can heterodimerize with another accessory monomer, M_3. Increasing the input monomer M_1 increases D_{12} dimers, reducing D_{23} dimers, ultimately increasing the D_{33} dimer. Thus, this network computes a switch-on function with the D_{33} homodimer as the output (Figure 2B). This simple example illustrates the way in which concentration changes can propagate through a network.[25,38]

When the switch-on and switch-off networks are combined, a *band-pass* filter emerges (Figure 2C). In this network, the input M_1 dimerizes strongly with M_3 and weakly with M_2. M_2 and M_3 strongly heterodimerize, and M_3 homodimerizes to form the output dimer, D_{33}. As total M_1 levels increase, they initially form D_{12} heterodimers, thereby decreasing D_{23} heterodimers and allowing the formation of the D_{33} output dimer. However, as M_1 increases further it begins to form dimer D_{13} as well, thereby suppressing the formation of the D_{33} output dimer. Thus, the output dimer D_{33} can form only when M_1 is present at intermediate concentrations. As with the switch-on and switch-off functions, this band-pass response can be inverted by adding an additional monomer to the network, creating a *band-stop* filter (Figure 2D). Taken together, these results provide an intuitive picture of how dimerization networks of increasing size can compute functions of increasing complexity.

Many natural signaling networks respond to multiple input signals,[18,34] provoking the question of whether dimerization networks can similarly compute functions of multiple inputs. Indeed, by considering two monomers as distinct inputs, combinatorial logic can emerge. For example, Figure 2E shows a network that computes a NIMPLY ("M_1 AND NOT M_2") logic gate. Here, the output dimer D_{13} forms in the presence of input M_1 alone, but not in the presence of input M_2, which competes to dimerize with monomer M_3. Similarly, Figure 2F shows a network computing a NAND logic gate, in which the output dimer D_{34} forms in the absence of inputs or the presence of a single input, but not in the combined presence of both inputs M_1 and M_2.

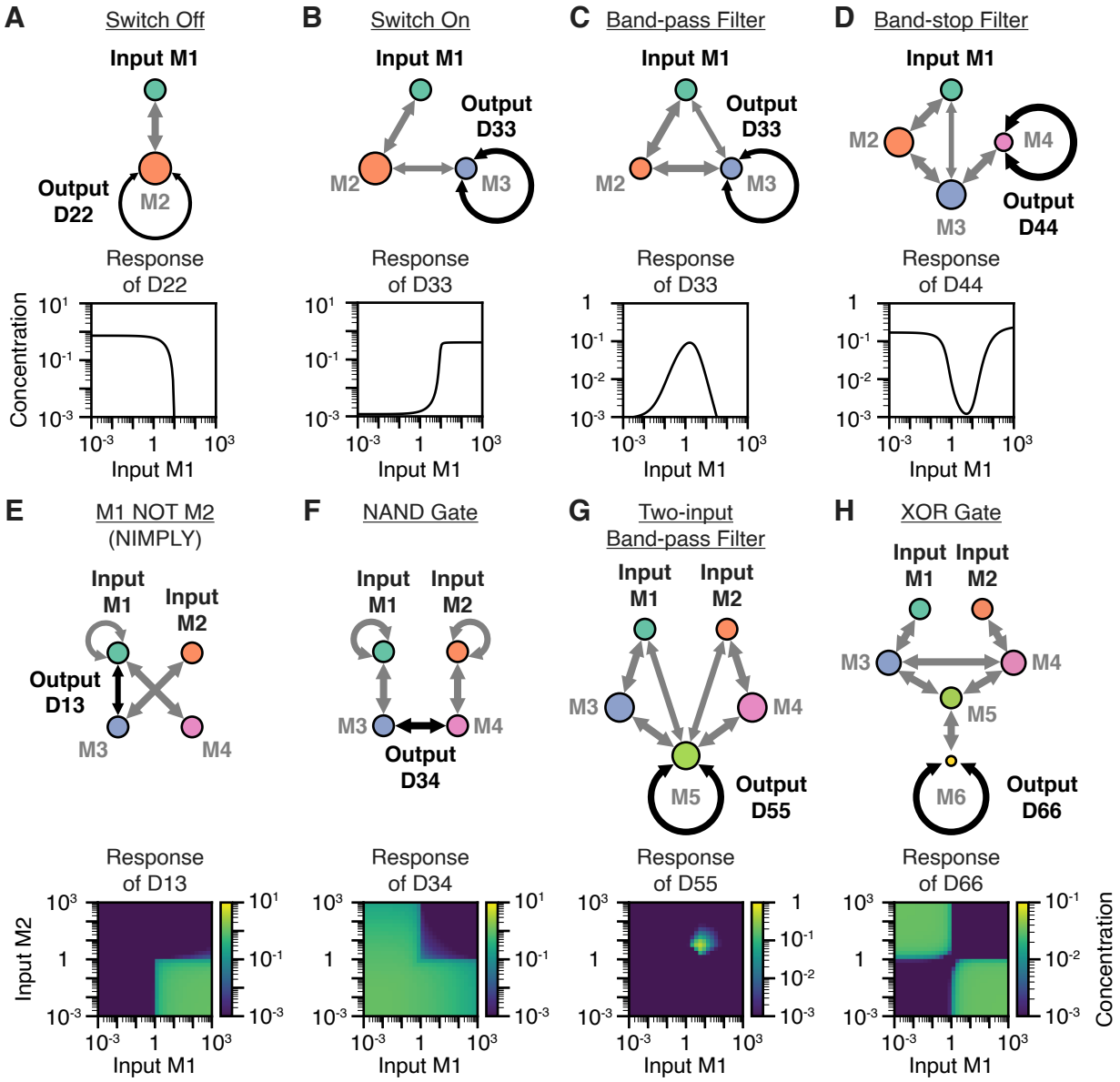

A
Switch Off
Input M1
Output D22
M2
Response of D22
Concentration
Input M1
B
Switch On
Input M1
Output D33
M2
M3
Response of D33
Input M1
C
Band-pass Filter
Input M1
Output D33
M2
M3
Response of D33
Input M1
D
Band-stop Filter
Input M1
M2
M4
M3
Output D44
Response of D44
Input M1
E
M1 NOT M2 (NIMPLY)
Input M1
Input M2
Output D13
M3
M4
Response of D13
Input M2
Input M1
F
NAND Gate
Input M1
Input M2
M3
M4
Output D34
Response of D34
Input M1
G
Two-input Band-pass Filter
Input M1
Input M2
M3
M4
M5
Output D55
Response of D55
Input M1
H
XOR Gate
Input M1
Input M2
M3
M4
M5
M6
Output D66
Response of D66
Input M1
Concentration

Figure 2. Competitive dimerization networks can compute diverse functions on one and two inputs. (A-D) Examples of one-input, one-output functions. From left to right, simulations of networks performing a switch-off function (A), a switch-on function (B), a band-pass filter (C), and a band-stop filter (D) are shown. (E-H) Examples of two-input, one-output functions. From left to right, simulations of networks performing an M1 NOT M2 (NIMPLY) gate (E), an M1 NAND M2 gate (F), a two-input band-pass filter (G), and an M1 XOR M2 gate (H) are shown.

Dimerization networks can also generate analog (non-Boolean) combinatorial responses. For example, the five-monomer network shown in Figure 2G combines two band-pass filter motifs (Figure 2C) to create a two-input band-pass filter. In this network, the output dimer D_{55} only forms in the presence of intermediate amounts of both inputs M_1 and M_2. Finally, larger networks can perform functions with even greater complexity. For example, a six-monomer network can compute an XOR logic gate (Figure 2H). In the absence of either input, heterodimerization of M_3 and M_4 allows M_5 to heterodimerize with M_6, limiting the formation of the D_{66} output dimer. Adding either input alone sequesters either M_3 or M_4, allowing the other of the two to dimerize with M_5, freeing M_6 to form the D_{66} output dimer. When both inputs are present, though, sequestration of both M_3 and M_4 allows M_5 to again dimerize with M_6, reducing the D_{66} output. Other functions beyond those described here are possible and can be rationally understood; atlases of networks performing one- and two-input computations can be found in Figure S1 and Figure S2.

This exploration of elementary dimerization networks demonstrates three key features of computation by dimerization. First, even relatively small networks can perform non-monotonic computations on one or multiple inputs. Second, many dimerization networks can be intuitively understood by analyzing the paths by which input perturbations propagate to affect output dimers. Third, networks of increasing size appear capable of performing increasingly complex input-output responses.

Monomer expression levels can modulate network computations

A key feature of competitive dimerization networks is their computational versatility, defined as the ability for a single network of proteins to perform distinct input-output computations depending on the expression levels of its protein components. For example, the band-pass response shown previously in Figure 2C can be tuned by modulating the expression levels of the accessory monomers M_2 and M_3 (Figure 3A). In this case, the total abundance of M_2 tunes the center of the band-pass filter because the input M_1 must sequester M_2 to induce the formation of the output dimer D_{33}. Additionally, the abundance of M_3 tunes the amplitude of the band-pass filter by directly promoting D_{33} formation. In another three-component network, adjusting protein expression levels can change a switch-off to a switch-on response (Figure 3B). Thus, even simple networks can show both quantitative and qualitative versatility.

Computational versatility extends to multi-input functions. For example, the midpoint of the two-input band-pass filter shown previously in Figure 2G can be tuned by independently

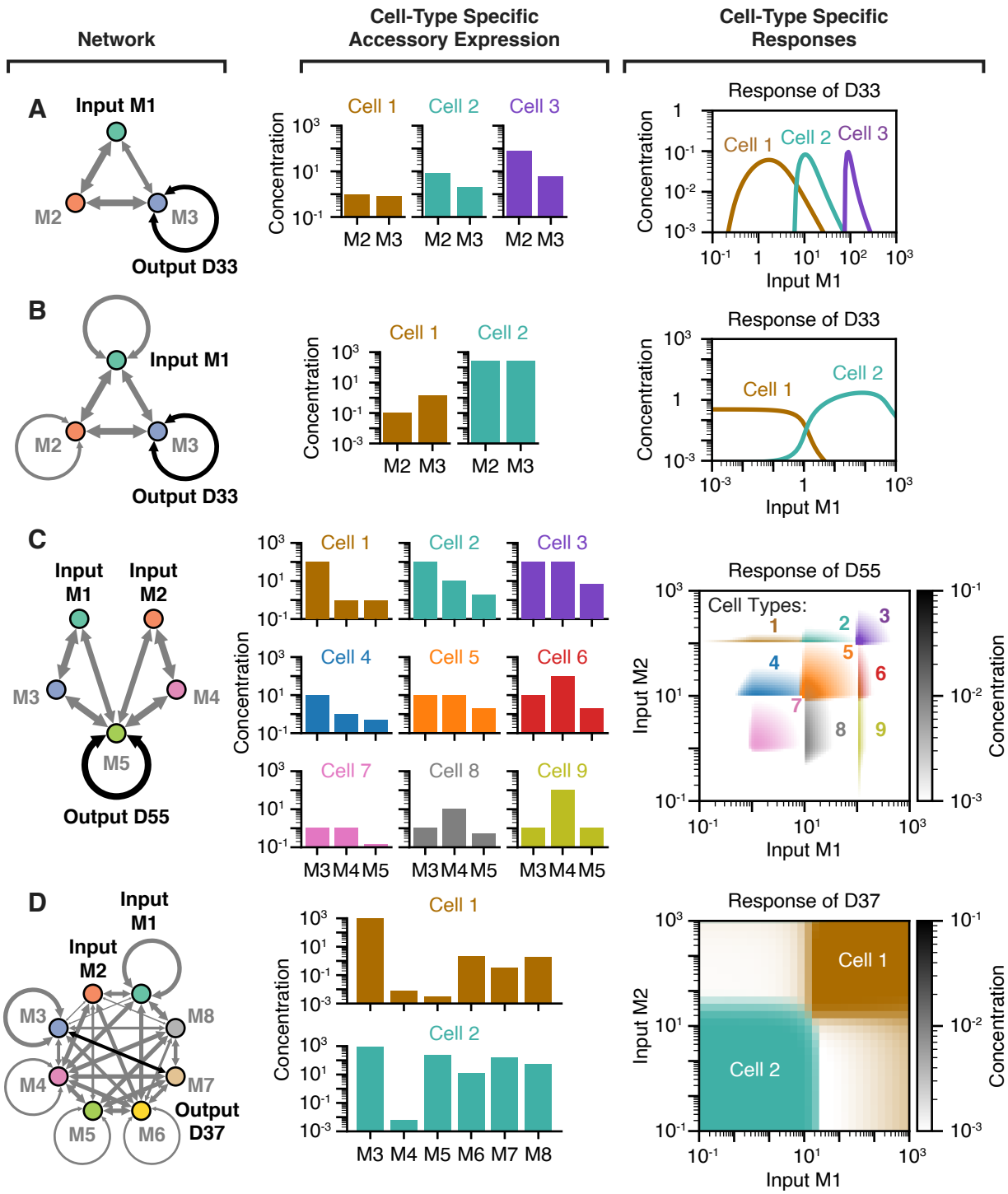

Network
Cell-Type Specific Accessory Expression
Cell-Type Specific Responses
A
Input M1
M2
M3
Output D33
Cell 1
Cell 2
Cell 3
Concentration
M2 M3
Response of D33
Input M1
B
Input M1
M2
M3
Output D33
C
Input M1
Input M2
M3
M4
M5
Output D55
Cell 4
Cell 5
Cell 6
Cell 7
Cell 8
Cell 9
M3 M4 M5
Response of D55
Cell Types:
Input M2
D
Input M1
Input M2
M3
M4
M5
M6
M7
M8
Output D37
M3 M4 M5 M6 M7 M8
Response of D37

Figure 3. A single set of dimerizing proteins can perform different input-output functions when expressed at different abundances, such as in different cell types. (A) Accessory monomer expression levels can be used to tune an input-output function, such as by tuning the midpoint of a one-input band-pass filter. Shown from left to right are the network diagram, accessory monomer expression levels, and simulated input-output responses for each hypothetical cell type. (B) More dramatically, accessory monomer expression levels can entirely transform the input-output function, converting a switch-on function to a switch-off function. (C) Two-input response functions may also be tuned, as the midpoint of a two-input band-pass filter may be tuned in both input dimensions. (D) Two-input functions may also be completely transformed, as an AND gate can be transformed into a NOR gate. The network shown in (D) was identified by screening random interaction affinities for networks of eight monomers (see "Large random networks are expressive and versatile").

---

adjusting the concentration thresholds for inputs M_1 and M_2 (Figure 3B and Supplementary Video 1). Biologically, these functions would allow different cell types to respond to different combinations of two inputs. Such cell-type- specific responses could then allow particular input combinations to selectively activate target cell types among others in a multicellular tissue, a process known as combinatorial addressing.[28]

Finally, a single network can be tuned to perform entirely different types of multi-input combinatorial logic. For example, the network shown in Figure 3D and Supplementary Video 2 computes an AND gate with dimer D_{37} with one set of monomer expression levels, but a NOR gate with another. A different network can compute both OR and NOR logic gates (Supplementary Video 3). These examples represent just a small set of the potential versatility of dimerization networks.

Expressivity grows with network size and connectivity

Evidently, dimerization networks can perform complex and versatile computations. How do the computational capabilities of dimerization networks scale with their size and connectivity? To address this question, we combined computational screening with optimization trials to systematically analyze the scope of possible network computations. First, we simulated large sets of networks of different sizes, with randomly chosen affinity (K_{ij}) and component expression levels. We sampled these parameters from broad parameter ranges consistent with experimentally measured affinities and protein expression levels (see Methods). Overall, this screen included 1 million networks of each network size, from m=2 to m=12 monomers. We also varied network connectivity, defined as the number of heterodimerization interactions present in a network.

For each random network, we determined the equilibrium concentrations of all species across a titration of the input monomer(s). By considering the response of each dimer as a distinct possible output, each network simulation produced multiple input-output responses. In order to classify these responses, we introduced a gridding scheme, binning the input and output into segments of equal size in logarithmic units (Figure 4A). With this scheme, two responses that pass

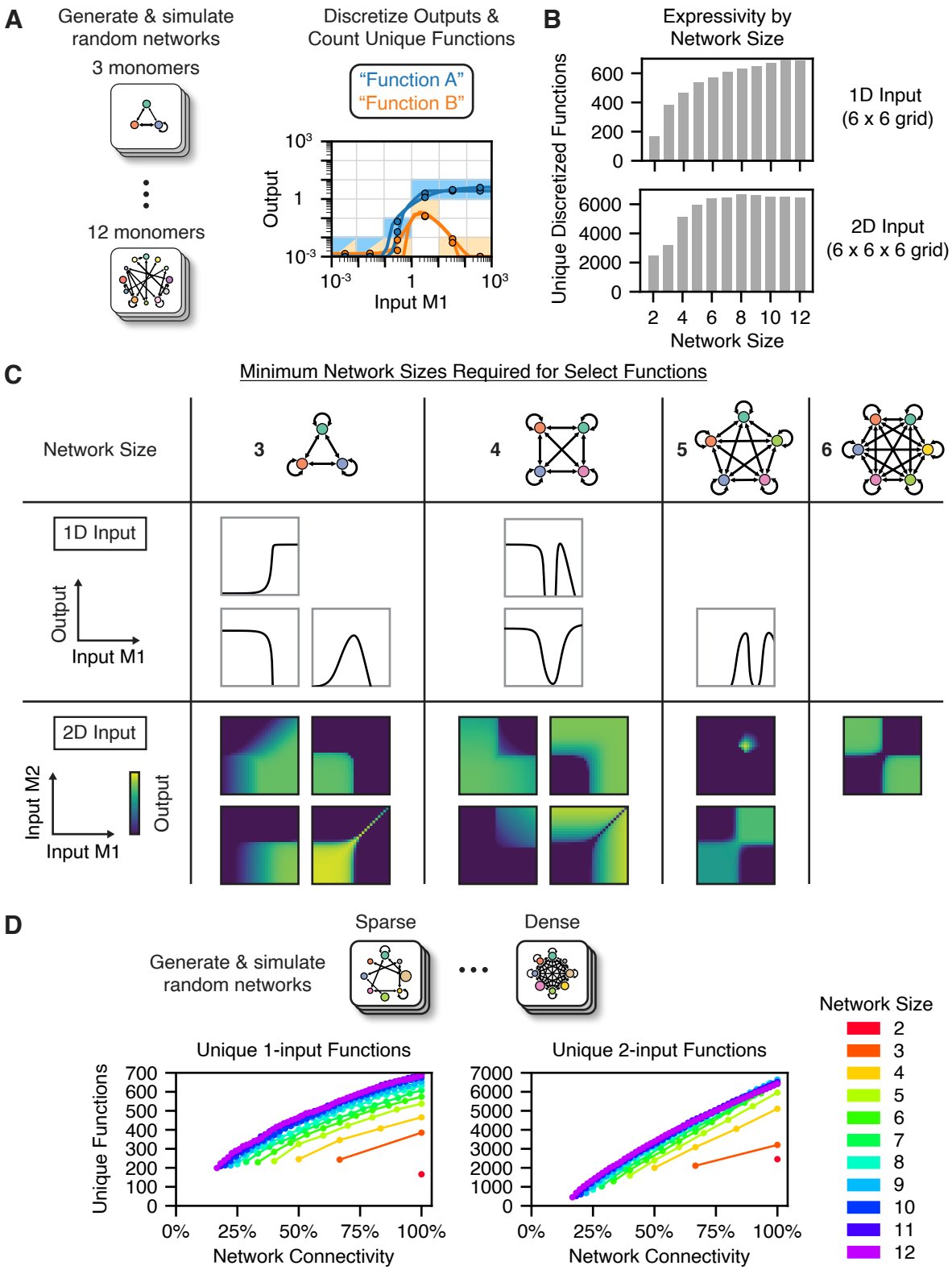

A
Generate & simulate random networks
3 monomers
12 monomers
Discretize Outputs & Count Unique Functions
"Function A"
"Function B"
Output
Input M1
B
Expressivity by Network Size
Unique Discretized Functions
1D Input (6 x 6 grid)
2D Input (6 x 6 x 6 grid)
Network Size
C
Minimum Network Sizes Required for Select Functions
Network Size
1D Input
Output
Input M1
2D Input
Input M2
Input M1
Output
D
Generate & simulate random networks
Sparse
Dense
Unique 1-input Functions
Unique 2-input Functions
Unique Functions
Network Connectivity
Network Size

Figure 4. Dimerization network expressivity grows with both network size and connectivity. (A) Many networks (n=10^6) of each network size with randomized parameters were simulated and their input-output responses were categorized by discrete binning. (B) Computational expressivity, measured as the number of unique, discretized response functions observed, increases with network size for both one-input and two-input functions. (C) Different classes of response functions require minimum network sizes, which were determined through optimization. Shown are the classes of response functions that become possible as network size is increased from three to six monomers. See Figure S1 and Figure S2 for schematics and quantitative plots of each response function shown. (D) Using the same parameter screen as in (A), network expressivity also grew with network connectivity, defined as the number of dimerization interactions present in a network.

---

through the same boxes (input-output bins) are classified as the same function type. This allows a more tractable, discrete analysis of the continuous input-output response space.

Unsurprisingly, larger networks performed more unique one-input and two-input functions (Figure 4B). For one-input functions, larger networks also exhibited a larger fraction of non-monotonic responses, including functions with up to 4 local extrema (Figure S3A). However, beyond network sizes of about m=6, we observed diminishing returns in the number of unique functions discovered. This trend was not impacted by the number of networks simulated, suggesting it was not an artifact of the random sampling (Figure S3C, Figure S3D). An additional observation was that, in larger networks, fewer dimers successfully form and respond to input perturbations (Figure S3B). This may be because, as network size increases, the number of dimers grows quadratically, increasing competition for the limited number of monomer constituents.

While screening many random networks allows for an unbiased exploration of possible response functions, this approach could fail to discover certain functions because of finite sampling depth. Thus, to gain more insight into the network size requirements for specific functions of interest, we turned to optimization approaches to identify parameter values that compute desired target functions. We found that both simulated annealing[39] and genetic algorithm[40] optimization could successfully optimize network parameters (see Methods). By defining an error threshold constituting a "successful" optimization, we optimized networks of decreasing size until no satisfactory parameter set could be identified. For example, using four network monomers, the optimization algorithms consistently identified parameter sets computing a one-input band-stop filter. Conversely, using three monomers, the optimizations consistently failed across many independent initializations, suggesting that the band-stop filter requires four network monomers. This approach revealed the network size requirements for the one- and two-input functions shown in Figure 4D, Figure S1, and Figure S2. These findings thus elucidate in more detail how expressivity grows as additional monomers are introduced. Together, the results of the parameter screen and optimizations suggest that most one- and two-input dimerization network computations can be performed by networks of just six monomers.

Networks of a given size can vary in connectivity, defined as the fraction of the possible heterodimerization interactions that are possible in a network. For networks of all sizes, more densely connected networks performed more unique functions in the parameter screen (Figure 4D). Additionally, in more densely connected networks, input monomers are typically connected to output dimers by shorter paths (Figure S3E). As a result, in more densely connected networks, more dimers successfully form and respond to perturbations of the input monomer(s) (Figure S3F). Thus, network expressivity appears to increase as networks become more densely connected.

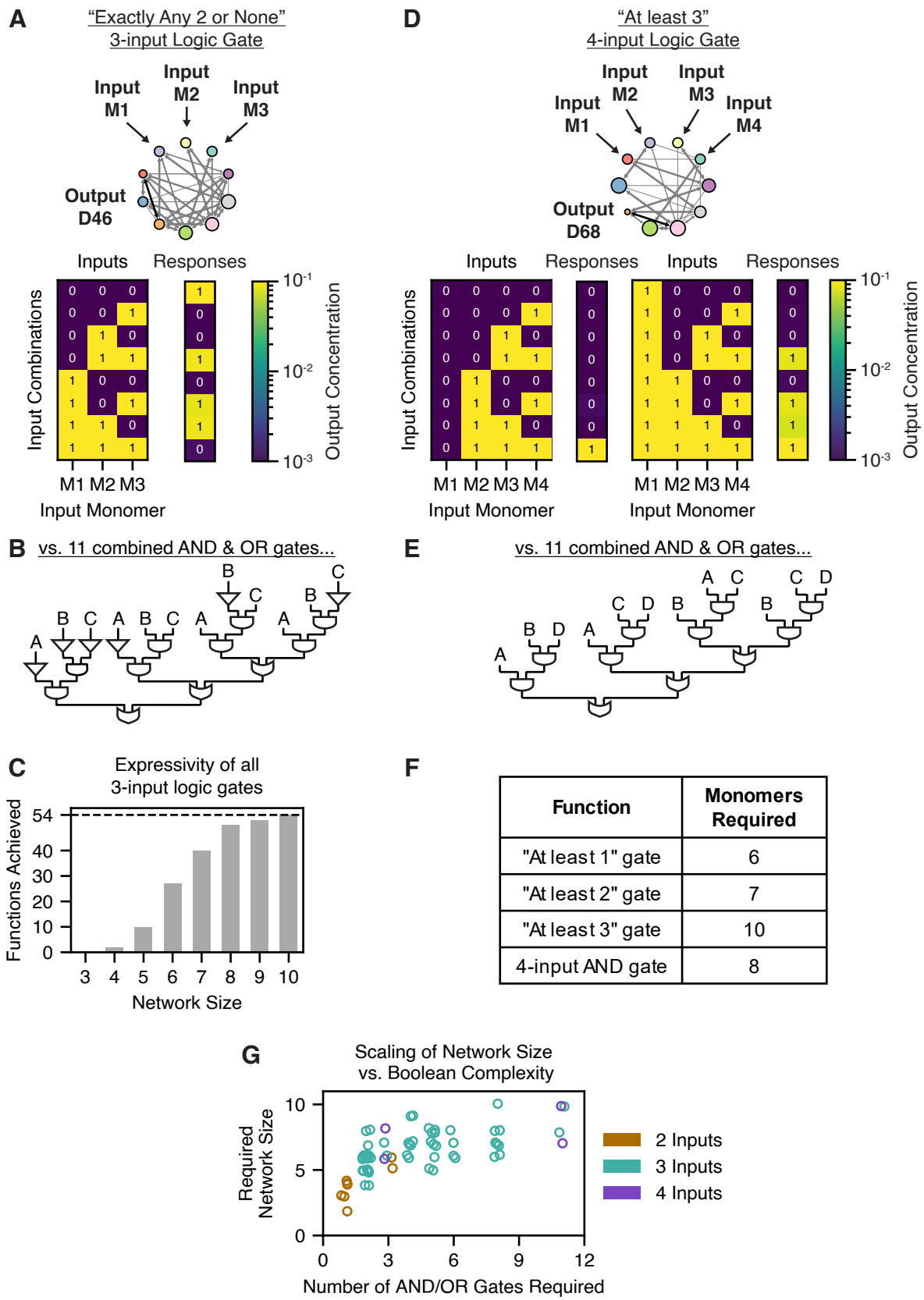
Competitive dimerization networks can compute multi-input functions

Cells commonly respond to many signals from other cells, their environment, and their own cell state.[18,34] For example, cell fate decisions can depend on inputs from multiple developmental signaling pathways, which may activate a common set of dimerizing bHLH transcription factors.[11,13] To understand how competitive dimerization networks could compute responses to multi-input signals, we used optimization to identify networks that perform various three- and four-input Boolean functions (logic gates).

Dimerization networks were capable of computing diverse multi-input functions. For example, a network of ten monomers can compute an "any 2 or none" function, in which the output dimer is formed if exactly two (but any two) inputs are present or no inputs are present (Figure 5A). This function would be difficult to implement by connecting elementary two-input functions, as it would require at least 11 AND and OR gates functioning orthogonally (Figure 5B). Thus, dimerization networks appear to offer the ability to compute multi-input functions in one computational layer, without the need for many orthogonal components. Using optimization trials, we demonstrated that dimerization networks of ten monomers can perform all 54 unique three-input logic gates, and networks of just six monomers could compute half of such functions (Figure 5C, Figure S4A).

This result encouraged us to ask whether dimerization networks could perform Boolean functions on even more inputs. We thus defined a set of four-input Boolean functions, which we call the "at least *n*" functions, whose output is high when at least *n* inputs, but any *n* inputs, are present. For instance, the network shown in Figure 5D computes an "at least 3" function using ten monomers. This function would also be difficult to achieve using elementary two-input functions, requiring 11 AND and OR gates at minimum (Figure 5E). We found that dimerization networks could readily perform all of the four-input "at least *n*" gates as well as a four-input AND gate (i.e., an "at least 4" gate) (Figure 5F, Figure S4B).

Thus, competitive dimerization networks can integrate several inputs in multi-dimensional computations. Importantly, while such computations could be performed using many orthogonal two-input logic operations, dimerization networks perform them all in the same computational layer. Indeed, while increasingly complex Boolean functions require larger dimerization networks – defining complexity by the number of two-input AND and OR gates they require – this relationship appears nonlinear, eventually reaching a point at which fewer monomers than elementary logic gates are required (Figure 5F).



Function	Monomers Required
"At least 1" gate	6
"At least 2" gate	7
"At least 3" gate	10
4-input AND gate	8

Figure 5. Competitive dimerization can integrate multiple inputs in multi-dimensional response functions. (A) A schematic is shown of a network computing a three-input logic gate, the "exactly any 2 or none" logic gate (explained in the text). Below is a heatmap of the simulated output concentrations for various input combinations (i.e., the truth table). (B) A schematic is shown decomposing the "exactly any 2 or none" logic gate into 11 elementary Boolean AND and OR gates. (C) A bar graph shows how the number of achievable three-input logic gates grows with network size. Dimerization networks with 10 monomers can perform all three-input logic gates. (D) A schematic is shown of a network computing a four-input logic gate, the "at least 3" logic gate (explained in the text). Below is a heatmap of the simulated output concentrations for various input combinations (i.e., the truth table). (E) A schematic is shown decomposing the "at least 3" logic gate into 11 elementary Boolean AND and OR gates. (F) A table displays the number of monomers required to compute the four-input Boolean "at least" functions. (G) Shown is a graph showing how the required network size grows with the Boolean complexity (measured as the number of elementary AND and OR gates required) of various logic gates. Colors denote the number of inputs used in each function. A jitter (random perturbation) was applied to each point to distinguish overlapping points.

Networks adapt over timescales of minutes to hours and function in the presence of noise

Competitive dimerization networks appear to compute diverse functions based on the deterministic modeling at equilibrium described above. But how quickly do these networks re-equilibrate, and can they continue to function despite various sources of biological noise? We first performed deterministic simulations of network re-equilibration dynamics in response to a perturbation of an input monomer. We found that networks re-equilibrate on a timescale of 100 s to 2 h (Figure S5A), without much dependence on network size. This timescale, which is comparable to the dissociation timescales of the highest-affinity dimers ($k_{\mathrm{off}} \approx 10^{-4}$ s^{-1} gives $t_{1/2} \approx$ 1.9 h) is faster than the hours to days timescales associated with transcriptional regulation.[41,42]

Biological circuits must be robust to various sources of both intrinsic and extrinsic noise.[43,44] In a cell, a competitive dimerization network would face both intrinsic noise from stochasticity in the dimerization equilibrium itself and extrinsic noise from fluctuations in the expression levels of network proteins. To characterize the impact of intrinsic noise, we performed stochastic Gillespie simulations of dimerization networks at equilibrium. Most species exhibited a noise coefficient of variation less than that of protein expression levels (0.2 to 0.5)[43,45] (Figure S5B, Figure S5C). The degree to which dimer concentrations are impacted by this intrinsic noise strongly depends on the average dimer abundance (Figure S5C), with low-abundance dimers being most sensitive to intrinsic noise, as is consistent with previous work.[46,47] Most species with abundances above 100 molecules exhibited noise coefficients of less than 0.1 (Figure S5B). To measure the effects of extrinsic fluctuations in monomer expression levels, we simulated networks performing each unique response function from the random parameter screen. Each network was simulated for 50 random perturbations of accessory expression levels using expression distributions consistent with experimental data (see Methods).[43,45] Most functions were robust to

such perturbations, with the median root-mean-square deviation (RMSD) in the log-scaled input-output function being 0.2 to 0.4, corresponding to a 1.5-fold to 2.5-fold change in the output (see Methods) (Figure S5D, Figure S5E).

Large random networks are expressive and versatile

It is unlikely that all the pairwise interaction affinities of a network can be precisely fine-tuned over evolution through protein mutations. However, protein expression levels could be tuned independently through several mechanisms, such as by adjusting promoter strength.[48,49] In the field of neural computation, it has been shown that sufficiently complex networks with random weights can still achieve arbitrary levels of expressivity, provided that only the weights of the final output layer are tuned.[50–53] Could dimerization networks with random interaction affinities similarly perform a wide variety of functions if only the expression levels of their accessory monomers can be tuned?

To address this question, we asked whether random networks could compute sets of various target functions (Figure 6A). We generated 50 networks with random interaction affinities at each network size from m=2 to m=12 monomers. As target functions, we selected representative subsets of the one-input functions previously identified in the parameter screen for networks of each size (Figure 4A). For each possible output dimer in each random network, we systematically optimized the accessory monomer expression levels to best fit each of the target functions. This approach allowed us to assess the versatility of many random networks of different sizes.

The versatility of individual dimers in random networks was broadly distributed, with some dimers exhibiting much greater versatility than others (Figure 6B). However, some dimers were capable of fitting over 70% of the target functions solely by adjusting the accessory monomer expression levels (Figure 6B, Figure S6C, Figure S6E). Versatility increased with network size, appearing to saturate at about m=6 monomers. In contrast, network connectivity did not appear to impact network versatility (Figure S6A). These results suggest that even random networks can potentially achieve versatile computation.

This analysis focused on the versatility of a single output dimer within a network. However, a feature of natural dimerization networks is that multiple dimers can be biochemically active outputs; for example, different transcription factor dimers often bind to distinct DNA binding sites, activating different sets of downstream genes.[2,54] We thus reasoned that the ability to use different output dimers in different contexts could further extend the versatility of random dimerization networks.

To test this hypothesis, we re-analyzed the optimization results above, allowing each random network to use different output dimers for each target function. With this additional flexibility, individual random networks were remarkably versatile. Networks with as few as 8 monomers were able to perform up to 90% of the functions observed across all 8-monomer networks (Figure 6C, Figure S6D, Figure S6F). Further, the entire distribution of networks achieved greater functional versatility, with the median network surpassing 80% of all functions. This expanded form of versatility increased rapidly with network size (Figure 6C) but did not

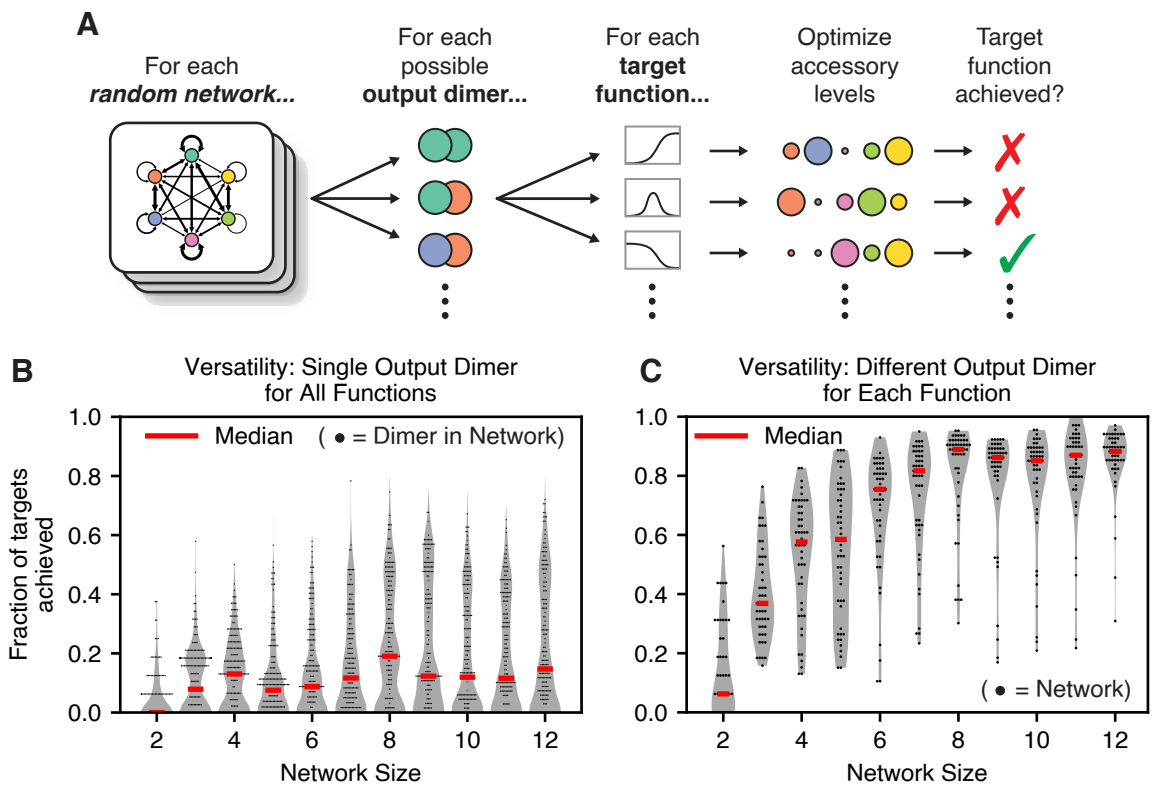

A
For each
random network...
For each
possible
output dimer...
For each
target
function...
Optimize
accessory
levels
Target
function
achieved?
B
Versatility: Single Output Dimer
for All Functions
Median
(● = Dimer in Network)
Fraction of targets
achieved
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
12
Network Size
C
Versatility: Different Output Dimer
for Each Function
Median
(● = Network)
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
12
Network Size

Figure 6. Large random networks are expressive and versatile. (A) For each possible output dimer in each network with random interactions (*n*=50), accessory monomer expression levels were systematically optimized to best fit each of the target functions. (B) A violin plot with scattered points shows the versatility of individual dimers, measured as the fraction of targets each individual dimer in each random network could achieve, for each network size tested. (C) A violin plot with scattered points shows the versatility of random networks, for each network size tested, when different output dimers may be used for each target function. For the violin plots, the gray violins show the kernel density estimate of the data distributions, red lines show the median values, and only a random subset of the data is displayed as scattered points.

appear to depend on network connectivity (Figure S6B). Evidently, individual networks do not require fine-tuned affinity parameters to function as powerful computational systems.

Discussion

Across many biological contexts, transcription factor dimerization networks interpret signals and regulate genes to control differentiation, proliferation, and stress responses. Here, we demonstrate that competitive dimerization networks are computationally *expressive*, computing a wide variety of input-output functions, and *versatile*, performing different functions solely by tuning the expression levels of network monomers. Dimerization networks can use multiple monomers as inputs, such as in the three- and four-input Boolean functions shown here (Figure 5). Further, such multi-input computations are performed in one computational step, without the need for many orthogonal modules. Dimerization networks are therefore well-suited to perform the multi-input computations necessary in biological decision-making, such as in the control of cell fate decisions by diverse signaling ligands, cell contacts, and environmental cues.[18,34] Additionally, dimerization networks appear to equilibrate on biologically reasonable timescales and function despite both the intrinsic noise of the dimerization equilibria and the extrinsic noise of protein expression levels (Figure S5). Overall, these results strongly suggest that computational expressivity and versatility could explain the ubiquity of transcription factor dimerization networks in natural signaling pathways.

Our results show that modestly sized dimerization networks can be quite powerful. For one- and two-input functions, network expressivity and versatility saturated at a network size of only 6 monomers. However, larger networks were required for multi-input computations or computations with random interaction affinities. For example, 10 monomers were required to compute all three-input logic gates (Figure 5C) and about 8 monomers were required to compute most one-input functions with random interactions (Figure 6C). These network sizes compare well to those of natural dimerization networks. For instance, there are 57 human bZIP transcription factors, at least 21 of which have been observed to interact in a network with 20% connectivity.[4] A median of 42 human bZIP proteins appear to be co-expressed in any given cell type (Figure S7), and previous work has already demonstrated that different cell types express components of these protein families at different abundances.[2,24] Thus, the bZIP dimerization network exhibits the

potential for complex and cell-type-specific computation. Further, similar statistics characterize the mouse and *Arabidopsis thaliana* bZIP families, the mouse and human nuclear receptor families, and the *Arabidopsis thaliana* MADS-box family (Figure S7B, Figure S7C). In the context of the results presented here, it thus appears that many natural networks have the potential to exhibit high computational expressivity and versatility.

Why are competitive dimerization networks such effective computational systems? Even a single dimerization reaction exhibits nonlinear input-output behavior, which can be accentuated by competition in the process of molecular titration.[55,56] In dimerization networks, multiple dimerization reactions are chained together, producing strong ultrasensitivity; for instance, a 10-fold change in an input monomer might produce a 1000-fold change in an output dimer (Figure 2A, Figure 2B). Perturbations to an input monomer radiate throughout a network in "paths," directly and indirectly affecting the concentrations of dimers. For example, increasing the monomer M_1 could directly increase the concentration of its heterodimer, D_{12}, indirectly decreasing the concentration of a D_{23} dimer, thereby indirectly increasing the D_{34} dimer, and so on. Maslov and Ispolatov previously identified conditions – relating to the total and free concentrations of each monomer along such a dimerization chain – that allow a perturbation in a dimerization network to propagate successfully along paths as long as four monomers.[25] In more complex dimerization networks, many such paths intersect, allowing for complex, non-monotonic responses. For example, in the simple network computing a band-pass filter (Figure 2C), one path favors output dimer formation at medium input concentrations, but another path then disfavors the same dimer's formation at high input concentrations. In this view, computational versatility is possible because one may tune the accessory monomer expression levels to leverage different paths in a network, thereby achieving different input-output functions.

Are there functions that cannot be computed through competitive dimerization alone? One key limitation is that input signals necessarily decay throughout dimerization networks; changing the abundance of an input monomer by N molecules can, at most, increase or decrease the abundance of another dimer by N molecules. As a result, perturbations decay with each dimerization step, and have only been observed to propagate through chains up to four monomers in length (Figure S3G).[25] Downstream biochemical mechanisms, such as enzymatic catalysis or gene transcription, could potentially be used to amplify the outputs of dimerization network computations. However, signal decay along dimerization chains still poses limits on the complexity of dimerization network computations. For example, while the band-stop filter (Figure 2D) requires paths of only three monomers, more complex functions, such as the "down-up-down" and "up-down-up" functions shown in Figure S1, require paths of up to four monomers and thus achieved smaller output dynamic ranges. Some functions are likely so complex that they cannot be performed with a dynamic range larger than the intrinsic noise of the system. Nevertheless, as evidenced by the functions shown here, a wide variety of complex computations can be computed without reaching this limit.

Dimerization networks are versatile: tuning their accessory monomer expression levels modulates their input-input computations (Figure 3, Figure 6). However, the same property could

also make these computations overly sensitive to protein expression noise. Our results suggest that at least some networks can balance these opposing properties through a separation of concentration scales. In such networks, the input-output computation is robust to modest (<3-fold) perturbations of accessory expression levels, but qualitatively sensitive to larger (>10-fold) perturbations. For example, when the network shown in Figure 3B computes a switch-off function in the presence of typical protein expression noise (<3-fold variation), the computation exhibits a proportional (approximately 3-fold) variation without a qualitative change (Figure S5D); the network still computes a switch-off function. By contrast, the large perturbation shown in Figure 3B is sufficient to convert this switch-off function into a switch-on function. Nevertheless, beyond examples like this, a more systematic investigation of the robustness-versatility tradeoff could provide insight into how protein dimerization networks manage the influence of extrinsic noise.

The theory of versatile computation by dimerization networks can be experimentally tested in several ways. In specific natural networks, one could systematically perturb the expression levels or pairwise interactions of dimerizing transcription factors and evaluate how the levels of various network dimers change in response. From a synthetic biology perspective, however, *synthetic* dimerization networks could be created using synthetic transcription factors and sets of non-orthogonal binding domains. A wide variety of synthetic binding domains have previously been developed.[57–59] While such efforts largely focused on creating orthogonal pairs of binding partners, they also generated "failed" (non-orthogonal) designs that exhibit many-to-many binding interactions.[57,60,61] The ability of these synthetic networks to perform complex and versatile computation could thus be evaluated in living cells.

It can be tempting to regard the complexity of protein interaction networks as an accidental by-product of duplication and divergence during evolution. However, the field of neural network computation has shown that simple but nonlinear elements, when connected in a complex network, can act as powerful computational systems.[62–64] Dimerization networks are prevalent across biological signaling pathways, and, as seen here, offer powerful computational capabilities. These observations strongly suggest that many-to-many dimerization networks could be used in biology as adaptable, multi-input computers whose specific functions can be readily tailored to diverse cellular needs. This system-level viewpoint, complemented with predictive mathematical models, should facilitate the control of natural cellular functions as well as the engineering of synthetic ones.

Limitations of this work

In the parameter screens, a wide range of affinity constants was chosen so as to capture many diverse network behaviors. Protein abundances were allowed to vary over six orders of magnitude, as observed in biological systems[65–67]; the weakest allowed interaction affinity would only dimerize 1% of proteins present at the highest concentration, whereas the strongest would dimerize 99% of proteins present at the lowest concentration (see Methods). However, it is unlikely that real protein networks could exhibit such a wide range of interaction affinities. Thus, some extraneous functions identified in our parameter screen may require affinities too extreme to

be biologically feasible. To assess whether this is the case, we performed a subsequent parameter screen using a restricted range of affinity values (four orders of magnitude) and still observed many complex input-output functions (containing up to 2 local extrema; Figure S3H). Finally, dimerization networks likely possess additional capabilities beyond those examined here. Transcription factor dimers may regulate the expression of network monomers, generating feedback, which can produce dynamic behaviors such as oscillations and multistability.[68,69] This property could also allow the output of one network to activate the inputs of another. Additionally, multiple dimers could be used as outputs, such as in combination (to compute sums of dimer concentrations) or separately (to compute multi-output functions).

Acknowledgments

We thank Matthew Langley, Pietro Perona, Jordi Garcia-Ojalvo, Arvind Murugan, Evan Mun, and James Linton for their conversations and scientific input. We thank Jan Gregrowicz, Martin Tran, Rachael Kuintzle, and Rongrong Du for feedback on the manuscript. This research was supported by the National Institutes of Health R01MH116508, the Allen Discovery Center, a Paul G. Allen Frontiers Group advised program of the Paul G. Allen Family Foundation (award number UWSC10142), the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health R01EB030015, and Caltech AI4Science Amazon Web Services cloud computing credits. J.P.G. was supported by the National Science Foundation Graduate Research Fellowship and the Ford Foundation Pre-Doctoral Fellowship. A.S. is grateful for support from a Department of Defense Vannevar Bush Faculty Fellowship. M.B.E. is a Howard Hughes Medical Institute investigator.

Author Contributions

J.P.G., B.E., and M.B.E. conceived of dimerization network computation. The computational approaches used were designed by all authors and implemented by J.P.G. and M.L. J.P.G. and M.B.E. wrote the manuscript, with input from all authors.

Declaration of Interests

M.B.E. is a scientific advisory board member or consultant at TeraCyte, Primordium, and Spatial Genomics.

Methods

Resource Availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Michael B. Elowitz (melowitz@caltech.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

All simulation data and all original code have been deposited at CaltechDATA and are publicly available as of the date of publication at https://doi.org/10.22002/kas2z-0fe41. Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

Method Details

All analysis was performed in Python version 3.8.13. Parameter screens and optimization trials were performed using Amazon Web Services (AWS) c5.4xlarge and c6a.48xlarge EC2 instances, respectively.

Network simulations

The input-output functions of dimerization networks were simulated using the Equilibrium Toolkit (EQTK, version 0.1.3), a Python-based, computationally efficient numerical solver for systems of reversible biochemical reactions.[70] To more precisely specify the problem of simulating a network's input-output function, we consider the vector $\boldsymbol{c}$, the concentrations of all species in the network, as described more thoroughly in the EQTK documentation (https://eqtk.github.io/user_guide/core_concepts.html). Thus,

$$\boldsymbol{c} = \left[[\mathrm{M}_1], [\mathrm{M}_2], \ldots [\mathrm{M}_m], \ldots [\mathrm{D}_{11}], [\mathrm{D}_{12}], \ldots [\mathrm{D}_{mm}]\right]$$

Each dimerization network specifies a set of n_{dimer} chemical reactions in which two monomers dimerize, e.g., $\mathrm{M}_1 + \mathrm{M}_1 \rightleftharpoons \mathrm{D}_{11}$. All such dimerization reactions can be written as a stoichiometric matrix $\mathbf{N}$, whose rows correspond to dimerization reactions and whose columns correspond to chemical species. Each row specifies how the counts of each chemical species increase or decrease with each chemical reaction.

We seek to identify the unique set of species concentrations $\boldsymbol{c}_{\mathrm{eq}}$ at equilibrium. To do this, we impose two constraints. For each reaction involving monomers i and j, we first define the equilibrium (or affinity) constant that must be satisfied:

$$\mathrm{K}_{ij} = \frac{\left[\mathrm{D}_{ij}\right]_{\mathrm{eq}}}{[\mathrm{M}_i]_{\mathrm{eq}}\left[\mathrm{M}_j\right]_{\mathrm{eq}}}$$

Secondly, we impose the conservation of mass according to the stoichiometry matrix $\mathbf{N}$; the total abundance of all monomers must remain constant. To do this, we define the conservation matrix $\mathbf{A}$ (using notation consistent with the EQTK documentation; not to be confused with the accessory protein expression levels $\boldsymbol{a}$), such that the quantity $\mathbf{A} \cdot \boldsymbol{c}$ is conserved. For this to be true, $\mathbf{A}$ must satisfy $\mathbf{A} \cdot \mathbf{N}^{\top} = 0$.

Given the total monomer concentrations as the initial condition, EQTK then uses trust region optimization to identify the equilibrium concentrations of all species consistent with both the dimerization affinities K_{ij} and the conservation of $\mathbf{A} \cdot \boldsymbol{c}$. Thus, to simulate the input-output computation performed by a particular network, the equilibrium species concentrations were solved over a titration of the input monomer(s), holding the total abundance of the accessory (non-input) monomers constant.

Parameter screen

To perform the large parameter screen, networks of two to twelve monomers were generated, with equal numbers of networks of each possible connectivity. In this parameter screen, approximately 10^6 networks were generated for one-input simulations, 250,000 of which were used for two-input simulations. For example, a network of eight proteins can have between 7 and 28 heterodimer "edges," (22 options); thus, for each number of edges, 45,455 networks were generated for a total of 1,000,010 networks. We later sub-sampled these networks to assess whether the number of networks sampled impacted the analysis. When more networks were sampled, the log of the number of unique functions discovered increased linearly with the log of the number of networks sampled (Figure S3C, Figure S3D), but the expressivity trends presented in Figure 4B remained consistent. The networkx Python package (version 2.7.1) was used to randomly generate graphs with a desired number of edges from an Erdős–Rényi model; each graph was checked for connectedness (i.e., that there are no fully separate networks) and re-generated if necessary to achieve connectedness. Homodimer edges were chosen with a probability of 75% (i.e., approximately 75% of monomers across the parameter screen were allowed to homodimerize). This fraction is within the range observed in natural dimerization networks, such as bZIP proteins (70-80%)[4] and nuclear receptor proteins (68%).[1] Upon choosing which edges would be present in a network, the values of the edge affinities were randomly chosen on a log-uniform range of dimensionless values 10^{-5} to 10^7 using Latin hypercube sampling (LHS). Finally, the expression levels of network proteins were also randomly chosen on a log-uniform range of dimensionless values 10^{-3} to 10^3 using LHS.

The aforementioned parameter ranges were defined generously so as to encompass as many biologically feasible behaviors as possible, even those that may require extreme parameters. However, the parameter ranges are biologically inspired. Protein expression levels in whole mammalian cells, as well as of transcription factors in the nuclei, have been observed to vary over six orders of magnitude,[65–67] ranging from approximately 10^{-10} M (10 pM; single-digit copy numbers) to 10^{-4} M (100 μM), assuming cell volumes of approximately 2-4 pL.[66,71] The wide range for dimerization affinities was chosen such that the strongest affinity sampled (10^7) would be strong enough to dimerize 99% of monomers at the lowest concentration sampled (10^{-3}), and the weakest affinity sampled (10^{-5}) would be weak enough such that only 1% of monomers at the highest concentration sampled (10^3) would be dimerized. Biological affinity values span an

enormous range, with dissociation constants (K_D) from approximately 10^{-3} M (mM) to 10^{-12} M (pM),[72] and certain RNase inhibitor proteins have been reported with even stronger affinities (K_D < 0.2 fM).[73] While we acknowledge that the affinities characterizing competitive dimerization networks are unlikely to take on such a wide range due to biochemical constraints on affinity and multi-specificity,[74] many-to-many interactions of SYNZIP coiled-coil proteins have been observed to vary in affinity (K_D) over four orders of magnitude from approximately 10^{-10} M (100 pM) to 10^{-6} M (1 μM).[57]

All generated parameter sets were simulated over a titration of the input monomer(s), with 30 titration points for one-input functions and 12 titration points for two-input functions; titration points were spaced evenly in log space from 10^{-3} to 10^{3} (the same range as for the accessory monomers). The ray Python package (version 1.11.1) was used for parallelization. For each network, the concentrations of each dimer over this input titration constituted the "responses." Any concentrations below 10^{-3} were rounded to 10^{-3}, as we consider such concentrations outside of the biochemically feasible window (i.e., less than one molecule of dimer per cell). The dataset was filtered to include only responses with a dynamic range greater than 10-fold; all other dimers either did not form at all or did not change significantly in response to the input monomer(s). The remaining responses were categorized into "unique" functions by discretizing the space of possible outputs into ten-fold bins (i.e., bin edges at 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, and 10^{3}, for both input and output). For each input bin (e.g., input between 10^{-3} and 10^{-2}), the response points were averaged in log space, and this value was categorized into one of the output bins (e.g., 0.5 categorized into the 10^{-1} to 1 bin). Thus, each response was transformed into a sequence, such as [0, 0, 0, 1, 1, 1], constituting its unique response function. To analyze expressivity, all unique response functions for a given dataset were counted. Lastly, we used these unique functions to create a library of target functions for optimization (see "Optimizations"). For each unique function observed, all responses categorized as performing that function were averaged in log space to create the corresponding target function in the library.

<u>Optimizations</u>

Two classes of optimization algorithms were used in this work: (1) a dual annealing algorithm was used to identify optimal parameter sets for particular functions, testing the minimum number of monomers required to achieve each; (2) a genetic algorithm was used in the much larger effort characterizing the versatility of networks with random interaction affinities. In both cases, the ray Python package (version 1.11.1) was used for parallelization.

Dual annealing optimization was used to determine whether networks of a particular size could achieve pre-defined target functions (in Figure 4D) or logic gates (in Figure 5). A dual annealing algorithm was used to optimize the affinities ($\boldsymbol{K}$) and accessory monomer expression levels ($\boldsymbol{a}$) of networks with a defined size to best fit a target function, with the loss defined as the sum of squares of residuals in log space:

$$\min_{\boldsymbol{K},\boldsymbol{a}} \sum_{i}^{N} (\log_{10}(\text{target}_i) - \log_{10}(x_i))^2$$

The optimize.dual_annealing function in the scipy Python package (version 1.10.1) was used to perform such optimizations. The dual annealing algorithm combines simulated annealing with a local search algorithm.[39] In simulated annealing, a perturbation of the parameter set (a "step") is proposed based on a visiting distribution. If the proposed step improves the loss function, it is accepted; otherwise, it may still be accepted with some probability based on a "temperature" factor (which decreases over the course of the optimization) so as to promote the identification of globally optimal solutions. In dual annealing, a local search is subsequently performed on the solutions identified by simulated annealing.

To optimize networks for three- and four-input logic gates, four titration points in each input dimension were used, at concentrations of 10^{-3}, 10^{-1}, 10^{1}, and 10^{3}. The lower two concentrations were considered "off," and the higher two concentrations were considered "on." The optimization was considered successful if the highest response of any input combination in which the output should be "off" is less than the lowest response of any input combination in which the output should be "on."

Separately, a genetic algorithm was used to measure the versatility of networks with random binding affinities $\boldsymbol{K}$ (Figure 6). In this process, we define a library of target functions and evaluate the ability of each sampled network to reproduce each target by optimizing the accessory protein expression levels $\boldsymbol{A}$. In a genetic algorithm, an initial population of parameter sets is generated and the best of these sets are allowed to "reproduce," producing a new population of parameter sets. This new population is then "mutated" (perturbed) and the process is repeated. All optimizations in this section were performed using the genetic algorithm (GA) function of the pymoo Python package (version 0.5.0),[75] with 20 iterations and a population size of 100.

For each network size *m*, we defined a library of n_{target}^{m} target functions from the unique functions that were identified in the parameter screen (see "Parameter screen"). Note that each network size thus has a distinct library size; this allows us to study versatility as a fraction of functions we know to be possible for a given network size.

We consider two loss metrics for evaluating the ability of a network, with affinities $\boldsymbol{K}$ and accessory monomer expression levels $\boldsymbol{a}$, to perform a target f using dimer index j:

$$\mathcal{L}_2(\boldsymbol{a}; j, \boldsymbol{K}, f) = \frac{1}{n_{\text{titration}}} \sum_{i}^{n_{\text{titration}}} \left|\log\big(f(x_i)\big) - \log\Big(g_j(x_i; \boldsymbol{K}, \boldsymbol{a})\Big)\right|^2 \quad (1)$$

$$\mathcal{L}_\infty(\boldsymbol{a}; j, \boldsymbol{K}, f) = \max_i \left|\log\big(f(x_i)\big) - \log\Big(g_j(x_i; \boldsymbol{K}, \boldsymbol{a})\Big)\right| \quad (2)$$

where the function g_j evaluates the equilibrium concentration of the jth dimer in the network at a particular input concentration x_i. Networks are simulated with $n_{\text{titration}}$ titration points.

In our experiments, we tune $\boldsymbol{a}$ in order to optimize Equation (1) but evaluate the quality of the resulting fit using Equation (2). This was done because the $\mathcal{L}_\infty$ norm measures the loss at the "worst point," which would be most relevant in practice. That is, we have the following:

$$\boldsymbol{a}^* = \underset{\mathbf{a}}{\operatorname{argmin}}\, \mathcal{L}_2\,(\boldsymbol{a}; j, \boldsymbol{K}, f)$$

We then define a *versatility* metric $\mathcal{V}$ measuring the fraction of targets each network could perform with a loss below the tolerance $\gamma = 1$. For each $\boldsymbol{K}$, dimer j, and network size m, versatility is calculated as the following:

$$\mathcal{V}^m(\boldsymbol{K}, j) \coloneqq \frac{1}{n^m_{\text{target}}} \sum_{n=1}^{n^m_{\text{target}}} [\mathcal{L}_\infty(\boldsymbol{a}^*; j, \boldsymbol{K}, f^m_n) \leq \gamma]$$

where f^m_n is the nth target for network size m. We term $\mathcal{V}$ as versatility because it reports the fraction of target functions that dimer j in a network with affinities $\boldsymbol{K}$ of size m can perform simply by tuning $\boldsymbol{a}$. This definition requires that the same dimer must always be used, such that no "rewiring" would be required at the molecular level.

We randomly sampled 50 networks with affinities $\boldsymbol{K}$ chosen on a log-uniform range of dimensionless values from 10^{-5} to 10^7. We subsequently performed the necessary inner optimizations of $\boldsymbol{a}$ to measure the versatility of each dimer j.

In Figure 6C, we broaden our definition of versatility to allow a network to perform different target functions using different output dimers. In this case, we first redefine versatility with the following:

$$\boldsymbol{a}^*,\, j^* = \operatorname{argmin}_{\boldsymbol{a}, j=1\ldots n_{\text{dimer}}} \mathcal{L}_2(\boldsymbol{a}; j, \boldsymbol{K}, f)$$

$$\mathcal{V}^m_{\text{any dimer}}(\boldsymbol{K}) \coloneqq \frac{1}{N_m} \sum_{n=1}^{N_m} [\mathcal{L}_\infty(\boldsymbol{a}^*; j^*, \boldsymbol{K}, f^m_n) \leq \gamma]$$

where each network has n_{dimer} dimers. This metric allows us to quantify the ability of a single network $\boldsymbol{K}$ to perform different functions by tuning $\boldsymbol{a}$ when given the freedom to use different dimers for each function.

<u>Simulating the Kinetics of Network Re-equilibration</u>

Dimerization network re-equilibration kinetics were simulated by numerical integration of ordinary differential equations (ODEs) using the integrate.odeint function of the scipy Python package (version 1.10.1). The ODEs were of the following form:

$$\frac{\mathrm{d}[\mathrm{M}_i]}{\mathrm{dt}} = \left(2\,\mathrm{k}^{ii}_{\text{off}}[\mathrm{D}_{ii}] + \sum_j^m \mathrm{k}^{ij}_{\text{off}}[\mathrm{D}_{ij}]\right) - \left(2\mathrm{k}_{\text{on}}[\mathrm{M}_i]^2 + \sum_j^m \mathrm{k}_{\text{on}}[\mathrm{M}_i][\mathrm{M}_j]\right)$$

$$\frac{d[\mathrm{D}_{ij}]}{dt} = k_{\mathrm{on}}[\mathrm{M}_i][\mathrm{M}_j] - k_{\mathrm{off}}^{ij}[\mathrm{D}_{ij}]$$

The rate of change in the concentration of monomer $\mathrm{M_i}$ is the rate of all dimer dissociation reactions involving monomer M_i minus the rate of all dimer association reactions involving monomer M_i, adjusted for stoichiometry. The rate of change in the concentration of dimer D_{ij} is the rate of the D_{ij} association reaction minus the rate of the D_{ij} dissociation reaction. We assume that all association reactions have similar rate constants, following a minimal kinetic model in which there is a single high-energy transition state for dimerization. In this model, various dimers only differ in the free energy of their dimerized states and thus only differ in their dissociation rates. We chose an associate rate constant of 5×10^5 M^{-1} s^{-1}, following experimental measurements of coiled-coil dimerization kinetics.[76] The dissociation rate constants were chosen randomly on a log scale between 10^{-4} and 1 s^{-1}, giving dimerization affinity constants (K_D) of about 200 pM to 2 μM, matching experimental measurements of coiled-coil interaction affinities.[57] All parameters were made unitless by converting concentration units into "molecules per cell" counts, assuming a cellular volume of about 3 pL.[66,71]

For each network size from *m*=2 to *m*=12 monomers, 20 random networks were simulated. The initial state of each network was the equilibrium state in which the input monomer was at its lowest concentration (1 molecule per cell), but the concentration of the input monomer was then set to its maximum concentration (10^6 molecules per cell). Networks were simulated with time increments of 10 s until equilibrium, with a maximum simulation time of 10^7 s. The exact equilibrium concentrations were calculated using EQTK (see "Network Simulations"), and a network species was defined as having reached equilibrium when its concentration was within 1 molecule per cell of the exact equilibrium concentration.

<u>Assessing the Intrinsic Noise of Dimerization Equilibria</u>

The intrinsic noise of the dimerization equilibria was simulated using stochastic Gillespie simulations, implemented using a modified version of the code from the biocircuits python package (version 0.1.14), using the numba Python package (version 0.55.1) to speed up simulation functions and the ray Python package (version 1.11.1) for parallelization. In the Gillespie algorithm, a population of molecules as well as a set of possible reactions among those molecules (with rates calculated from the current population) is first defined. To perform a step of the simulation, the time until the next reaction is sampled from an exponential distribution based on the rates of all reactions. The identity of the reaction that occurs is chosen based on the relative rates of all the possible reactions, and the population is updated to reflect that reaction.

Using the same parameter ranges as in the deterministic simulations of network equilibration, 20 random networks were simulated for 1000 s for each network size from *m*=2 to *m*=12 monomers. The noise η in the concentration of each species was defined as the coefficient of variation in the number of molecules per cell n over time[43,45]:

$$\eta = \frac{\langle n^2 \rangle - \langle n \rangle^2}{\langle n \rangle^2}$$

Assessing Robustness to Extrinsic Monomer Expression Noise

Protein expression levels are subject to both intrinsic expression noise, which independently affects different genes, as well as extrinsic expression noise, in which overall fluctuations in machinery for transcription and translation could affect the expression of all genes in a concerted manner. The steady-state distribution of mRNA counts produced by bursty transcription is negative binomial,[77] and steady-state protein concentrations appear similarly distributed.[45] Thus, expression noise was modeled here using a gamma distribution, the continuous analog of the negative binomial distribution. This distribution was parameterized to produce expression noise coefficients similar to those observed experimentally (0.4 for intrinsic and 0.6 for extrinsic).[45] To accomplish this, the shape parameter was set to $1/\eta^2$ and the scale parameter was set to η^2.

We simulated both cases in which there was either purely intrinsic (independent) or purely extrinsic (concerted) expression noise. For each unique function identified in the aforementioned parameter screen, a network performing that function was randomly selected and simulated for 50 random perturbations of the accessory monomer expression levels. Each of the 50 resulting input-output functions (simulated at 30 input points) was compared to the original input-output response using the root-mean-square difference (RMSD) in log space:

$$\mathrm{RMSD} = \frac{1}{\mathrm{n_{points}}} \sqrt{\sum_{i}^{\mathrm{n_{points}}} (\log_{10}(\mathrm{original}_i) - \log_{10}(\mathrm{perturbed}_i))^2}$$

The RMSD is one metric for the typical difference, in log space, between the original and perturbed responses at each input point.

Transcriptomics data for analysis of transcription factor co-expression

To characterize how many dimerizing transcription factors are potentially co-expressed in individual cell types, two pre-existing datasets were used: the integrated mouse transcriptomics dataset from Granados et al.,[24] which integrated multiple pre-existing mouse datasets, and the Human Protein Atlas (version 23.0) "RNA consensus tissue gene data" dataset[78] (https://www.proteinatlas.org), which reports normalized expression levels from a consensus of multiple scRNA-seq datasets. The Human Protein Atlas dataset was used to demonstrate the cell-type-specific expression of nuclear receptor proteins in Figure 1C. The names of genes belonging to the bZIP and nuclear receptor families in mice and humans were obtained from Uniprot,[79] Reinke et al.,[54] and Amoutzias et al.[1]

Quantification and Statistical Analysis

The numbers of samples (*n*) used in each analysis are described in the Method Details section as well as the figure captions. In analyses involving the comparison of data distributions, such as the

comparisons of versatility scores in Figure 6, violin plots were used to show the distribution (more specifically, the kernel density estimate) of data points. Points were also directly plotted with the violins; points were sub-sampled if it was not practical to display all data points. On such violin plots, the medians of the data are displayed as red lines.

Additional Resources

An interactive notebook hosted on Google Colaboratory, in which users can simulate the input-output functions of dimerization networks, can be found at the following link: https://bxky.short.gy/interactive_dimerization_networks

Key Resources Table

Deposited data		
Human Protein Atlas	[78]Uhlén et al., 2015	https://doi.org/10.1126/science.1260419
Integrated Mouse Atlas	[24]Granados et al., 2022	https://doi.org/10.1101/2022.08.21.504714
Software and algorithms		
Simulation, optimization, and analysis code	This paper	https://doi.org/10.22002/kas2z-0fe41

Supplemental Figures

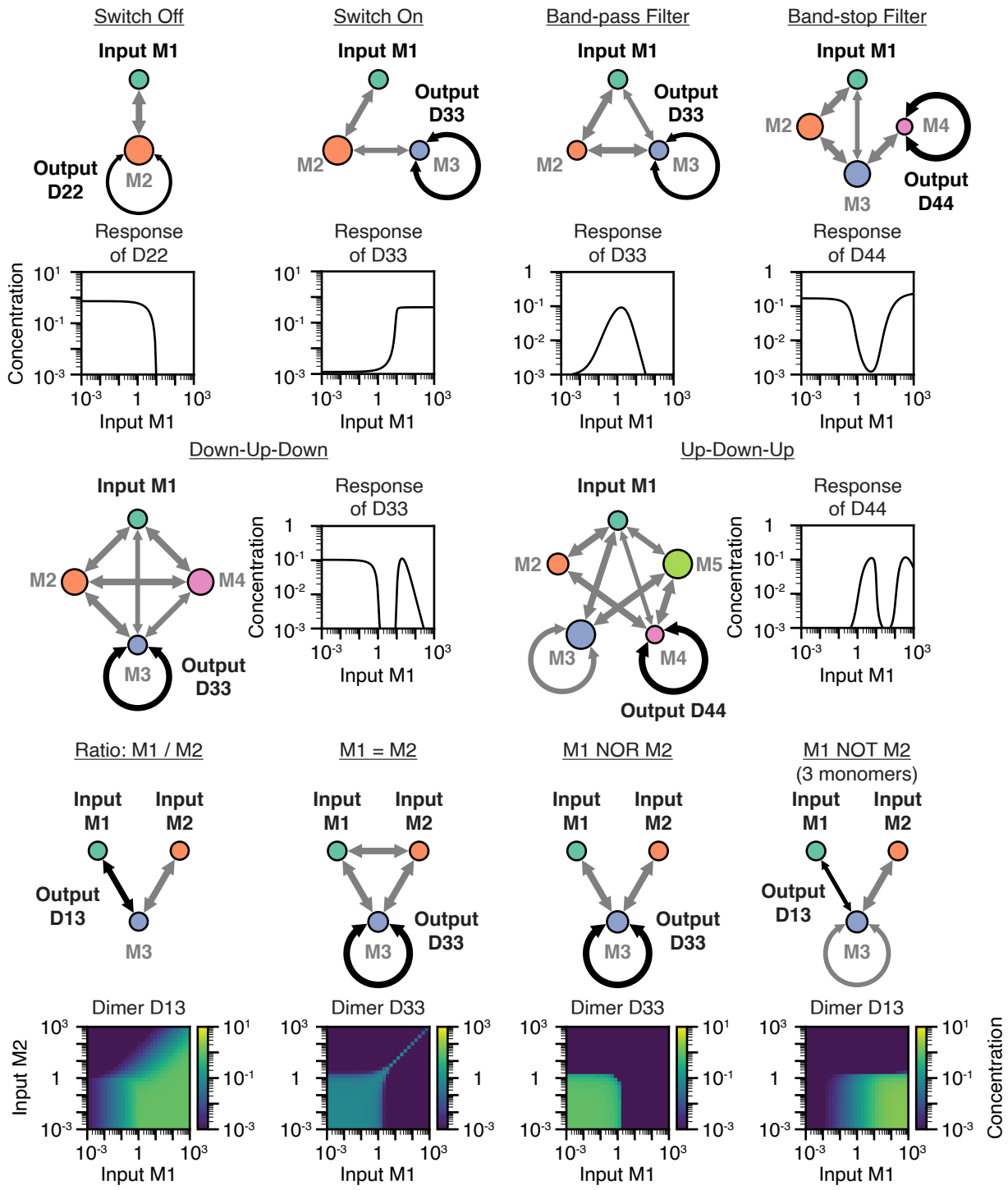

Switch Off
Input M1
Output D22
M2
Response of D22
Concentration
Input M1
Switch On
Input M1
Output D33
M2
M3
Response of D33
Input M1
Band-pass Filter
Input M1
Output D33
M2
M3
Response of D33
Input M1
Band-stop Filter
Input M1
M2
M4
M3
Output D44
Response of D44
Input M1
Down-Up-Down
Input M1
M2
M4
M3
Output D33
Response of D33
Concentration
Input M1
Up-Down-Up
Input M1
M2
M5
M3
M4
Output D44
Response of D44
Concentration
Input M1
Ratio: M1 / M2
Input M1
Input M2
Output D13
M3
Dimer D13
Input M2
Input M1
M1 = M2
Input M1
Input M2
M3
Output D33
Dimer D33
Input M1
M1 NOR M2
Input M1
Input M2
M3
Output D33
Dimer D33
Input M1
M1 NOT M2
(3 monomers)
Input M1
Input M2
Output D13
M3
Dimer D13
Input M1
Concentration

Figure S1. An atlas of elementary dimerization network computations, part 1. Shown for each function is a schematic of the network parameters and a plot of the corresponding input-output function, for both one-input functions (top) and a partial set of two-input functions (bottom). The remaining two-input functions can be found in Figure S2.

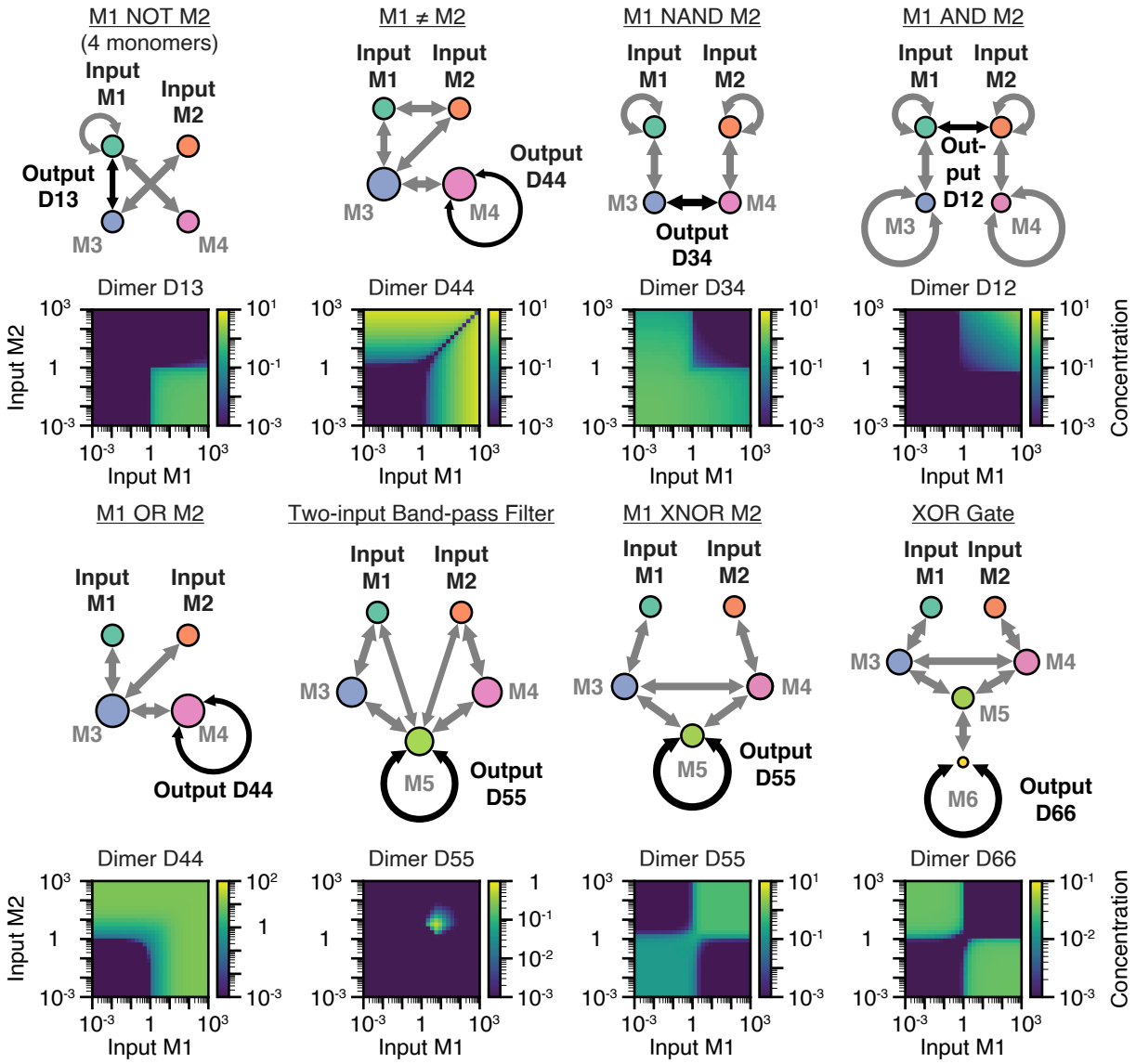

M1 NOT M2
(4 monomers)
Input M1
Input M2
Output D13
M3
M4
M1 ≠ M2
Input M1
Input M2
Output D44
M3
M4
M1 NAND M2
Input M1
Input M2
M3
M4
Output D34
M1 AND M2
Input M1
Input M2
Out-put D12
M3
M4
Dimer D13
Dimer D44
Dimer D34
Dimer D12
Input M2
Input M1
Concentration
M1 OR M2
Input M1
Input M2
M3
M4
Output D44
Two-input Band-pass Filter
Input M1
Input M2
M3
M4
M5
Output D55
M1 XNOR M2
Input M1
Input M2
M3
M4
M5
Output D55
XOR Gate
Input M1
Input M2
M3
M4
M5
M6
Output D66
Dimer D44
Dimer D55
Dimer D55
Dimer D66
Input M2
Input M1
Concentration

Figure S2. An atlas of elementary network computations, part 2. Shown for each function is a schematic of the network parameters and a plot of the corresponding input-output function. Displayed are the two-input computations not included in Figure S1.

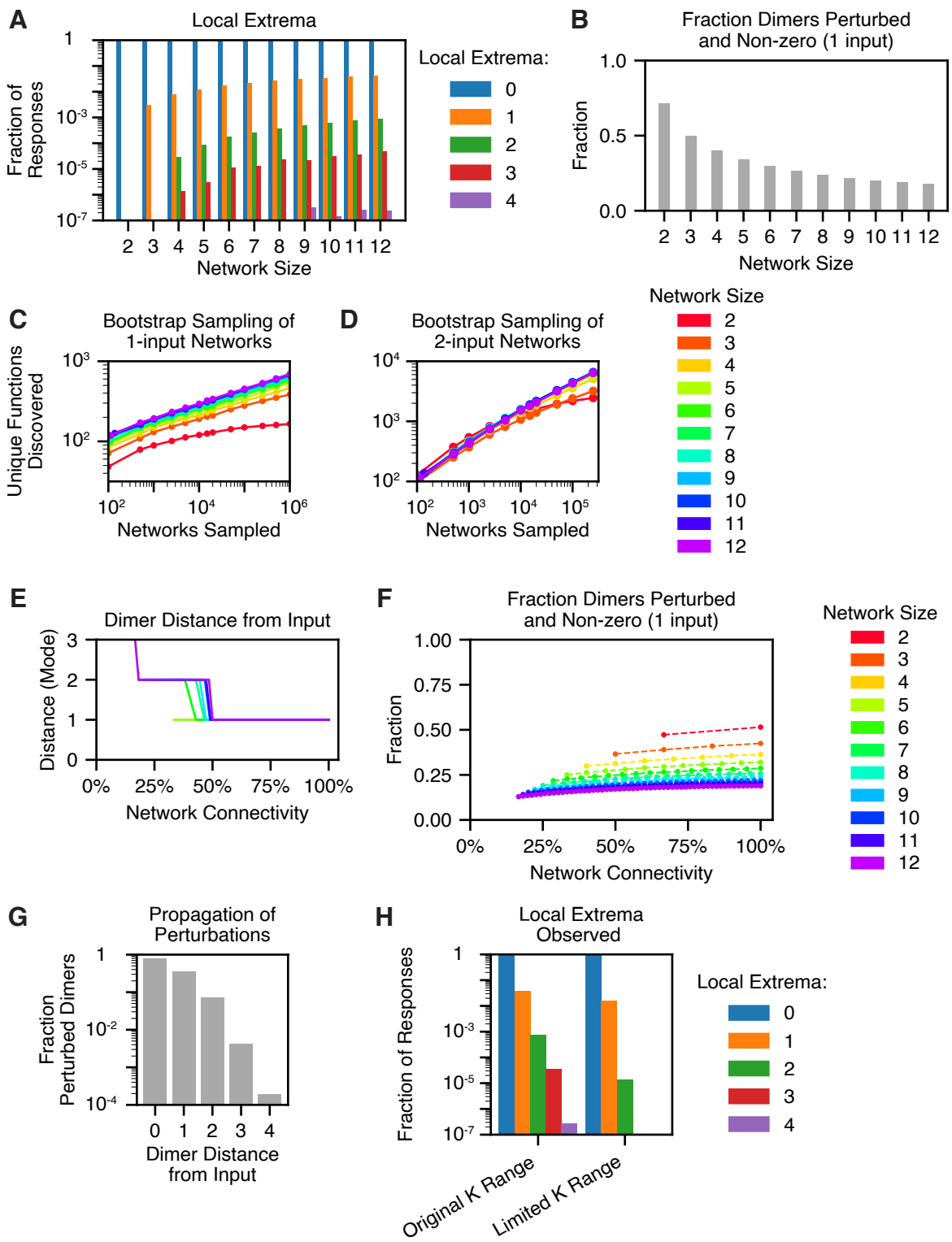
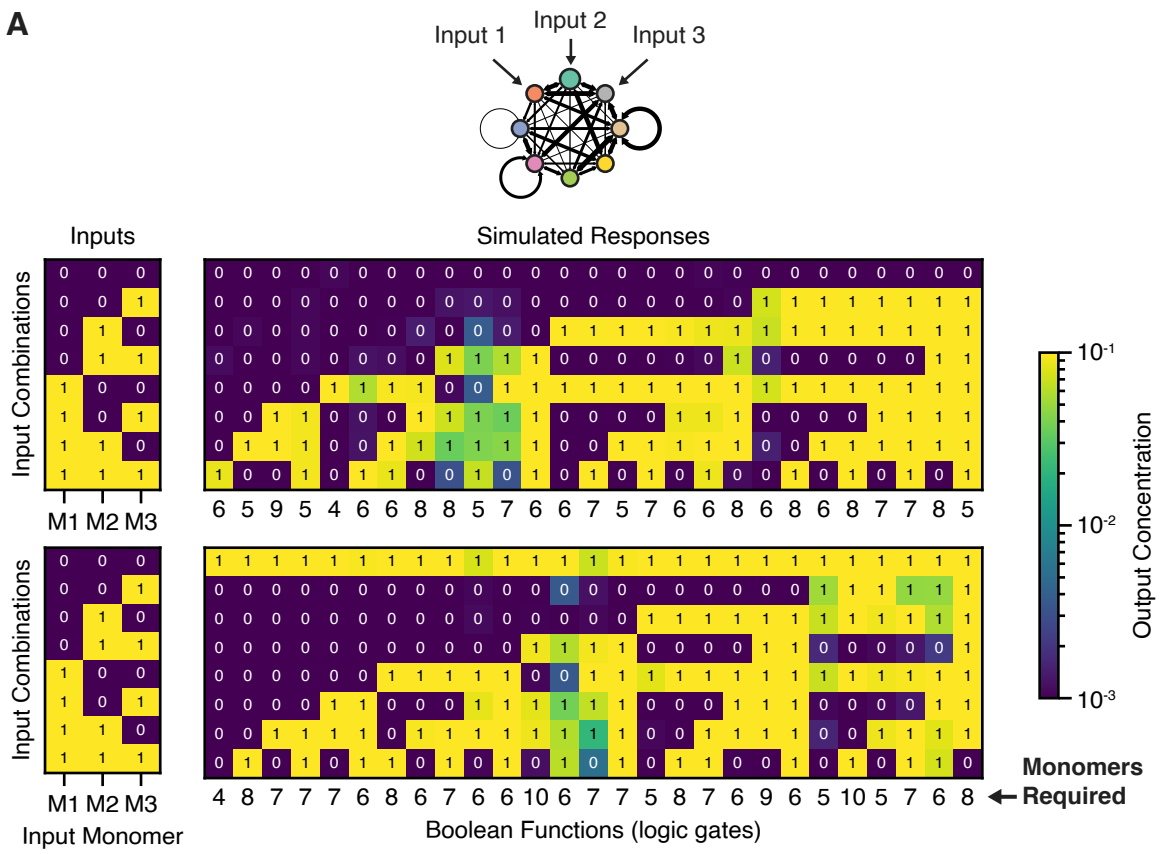

A
Local Extrema
Fraction of Responses
1
10-3
10-5
10-7
2 3 4 5 6 7 8 9 10 11 12
Network Size
Local Extrema:
0
1
2
3
4
B
Fraction Dimers Perturbed and Non-zero (1 input)
Fraction
1.0
0.5
0.0
2 3 4 5 6 7 8 9 10 11 12
Network Size
C
Bootstrap Sampling of 1-input Networks
Unique Functions Discovered
10³
10²
10² 10⁴ 10⁶
Networks Sampled
D
Bootstrap Sampling of 2-input Networks
10⁴
10³
10²
10² 10³ 10⁴ 10⁵
Networks Sampled
Network Size
2
3
4
5
6
7
8
9
10
11
12
E
Dimer Distance from Input
Distance (Mode)
3
2
1
0
0% 25% 50% 75% 100%
Network Connectivity
F
Fraction Dimers Perturbed and Non-zero (1 input)
Fraction
1.00
0.75
0.50
0.25
0.00
0% 25% 50% 75% 100%
Network Connectivity
Network Size
2
3
4
5
6
7
8
9
10
11
12
G
Propagation of Perturbations
Fraction Perturbed Dimers
1
10-2
10-4
0 1 2 3 4
Dimer Distance from Input
H
Local Extrema Observed
Fraction of Responses
1
10-3
10-5
10-7
Original K Range
Limited K Range
Local Extrema:
0
1
2
3
4

Figure S3. A global parameter screen reveals the diversity and nature of dimerization network computations. (A) A bar graph shows, for each network size, the fraction of responses with zero to four local extrema (i.e., local minima and maxima). (B) A bar graph shows, for each network size, the fraction of dimers that both form at significant concentrations and are perturbed more than 10-fold by a titration of the input monomer. (C-D) The number of unique one-input (C) and two-input (D) functions observed is plotted versus the number of networks sampled in the random parameter screen. (E) Shown is the most common distance (the mode) between the input monomer and network dimers for various levels of network connectivity. (F) The fraction of dimers that both form at significant concentrations and are perturbed more than 10-fold by a titration of the input monomer is plotted versus network connectivity for various network sizes. (G) A bar graph shows, for increasing distances between the input monomer and output dimer, the fraction of dimers (out of all dimers that form at appreciable concentrations) that are perturbed more than 10-fold in response to a titration of the input monomer. (H) A bar graph shows, for a parameter screen of 12-monomer networks using a more limited range of affinities (K_{ij}), the fraction of responses with zero to four local extrema (i.e., local minima and maxima).


A
Input 1
Input 2
Input 3
Inputs
Simulated Responses
Input Combinations
M1 M2 M3
Input Monomer
Boolean Functions (logic gates)
Monomers Required
Output Concentration
10^{-1}
10^{-2}
10^{-3}

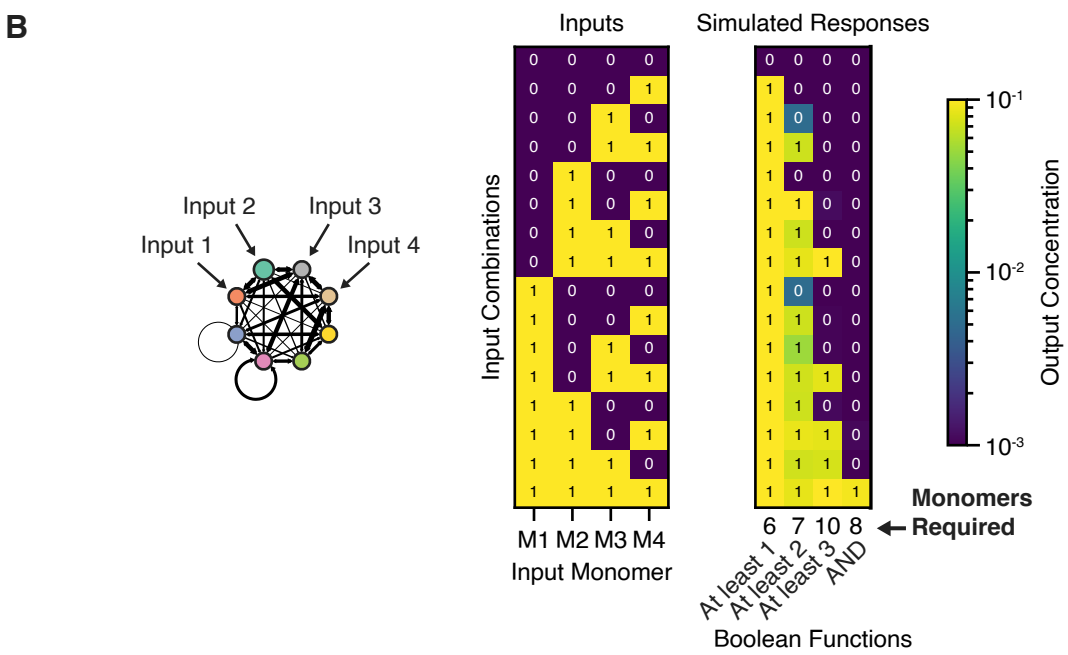

B
Input 1
Input 2
Input 3
Input 4
Inputs
Simulated Responses
Input Combinations
M1 M2 M3 M4
Input Monomer
At least 1
At least 2
At least 3
AND
Boolean Functions
Monomers Required
Output Concentration
10^{-1}
10^{-2}
10^{-3}

Figure S4. Competitive dimerization networks can compute multi-input functions. (A) Dimerization networks can perform all three-input logic gates. (left) Rows represent different combinations of inputs that are presented to each network. (right) A heatmap of responses is shown, where each column represents a unique logic gate and the color of the response heatmap represents the output dimer concentration. The number of network monomers required to perform each gate is noted below each column. (B) Dimerization networks can further perform four-input logic gates. Shown are four examples, the AND and "at least *n*" gates, which output 1 if at least *n* inputs are present.

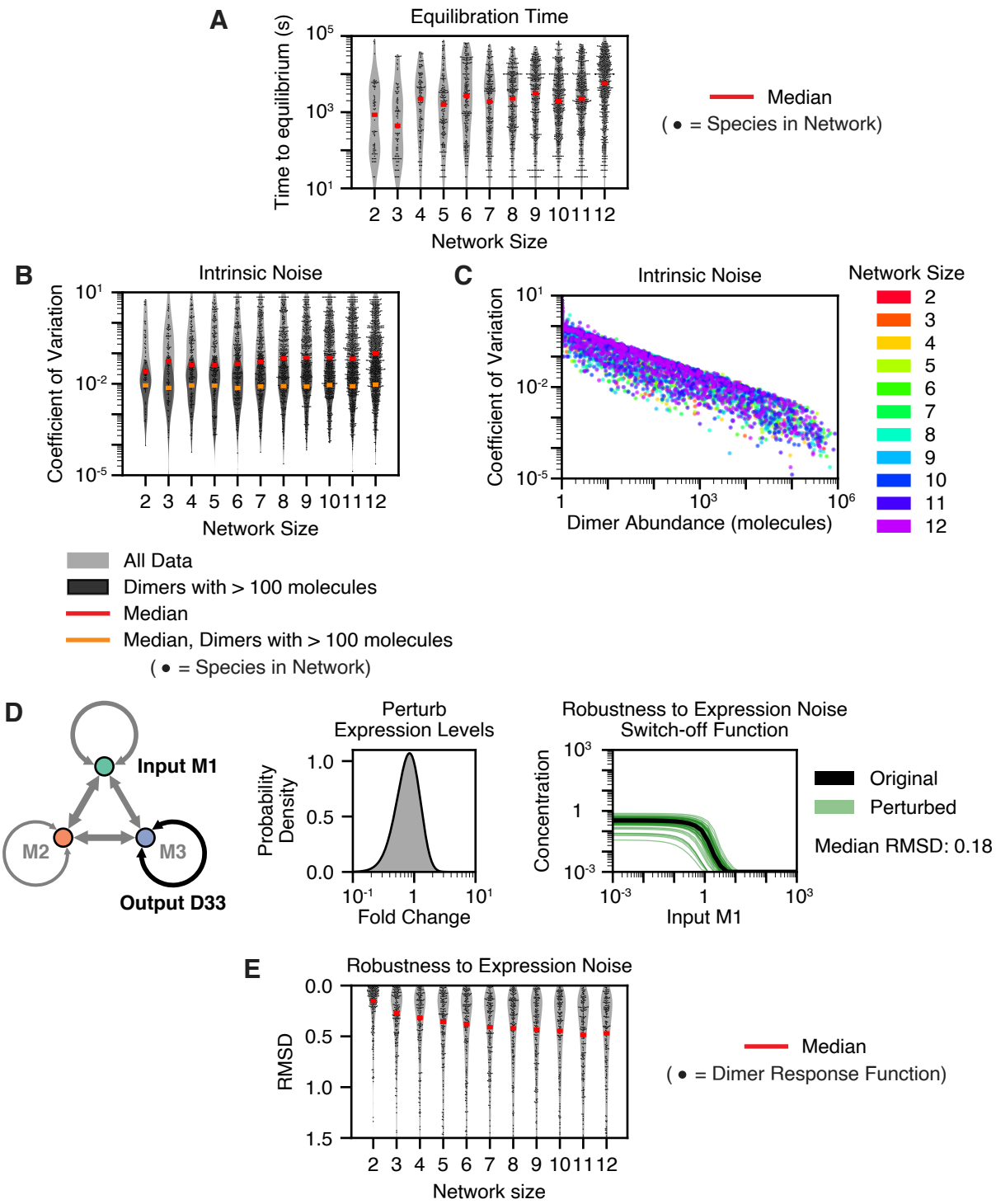

A
Equilibration Time
Time to equilibrium (s)
Network Size
Median
(• = Species in Network)
B
Intrinsic Noise
Coefficient of Variation
Network Size
All Data
Dimers with > 100 molecules
Median
Median, Dimers with > 100 molecules
(• = Species in Network)
C
Intrinsic Noise
Coefficient of Variation
Dimer Abundance (molecules)
Network Size
D
Input M1
M2
M3
Output D33
Perturb
Expression Levels
Probability Density
Fold Change
Robustness to Expression Noise
Switch-off Function
Concentration
Input M1
Original
Perturbed
Median RMSD: 0.18
E
Robustness to Expression Noise
RMSD
Network size
Median
(• = Dimer Response Function)

Figure S5. Competitive dimerization networks exhibit biologically reasonable equilibration kinetics and robustness to noise. (A) Network equilibration kinetics were simulated by numerically integrating ordinary differential equations (ODEs) describing dimer association and dissociation kinetics. The time for each species in n=20 networks to re-equilibrate after a perturbation of an input monomer is displayed as a violin plot with scattered points. (B-C) The intrinsic noise of the binding equilibrium was simulated using the Gillespie algorithm with 100 steps of 10 s each. (B) A violin plot (light gray) with scattered points shows the coefficient of variation, a measure of noise, for each species. A dark gray violin shows the data specifically for species present at high abundances (>100 molecules/cell, median shown by the orange line). (C) The versatile switch-off function from Figure 3B (right, black) was perturbed with typical protein expression noise (probability density function shown in the middle), and each perturbed computation shown (right, green, n=50). The median root mean square deviation (RMSD), in log space, between the original and the perturbed curves was 0.18, corresponding to a 1.5-fold change in output. (E) Networks performing each unique function from the parameter screen were perturbed with extrinsic expression noise (n=50). A violin plot with scattered points shows the RMSD in log space between the original and perturbed input-output functions. For all violin plots, the violins show the kernel density estimate of the data distributions, red lines show the median values, and only a random subset of the data is displayed as scattered points.

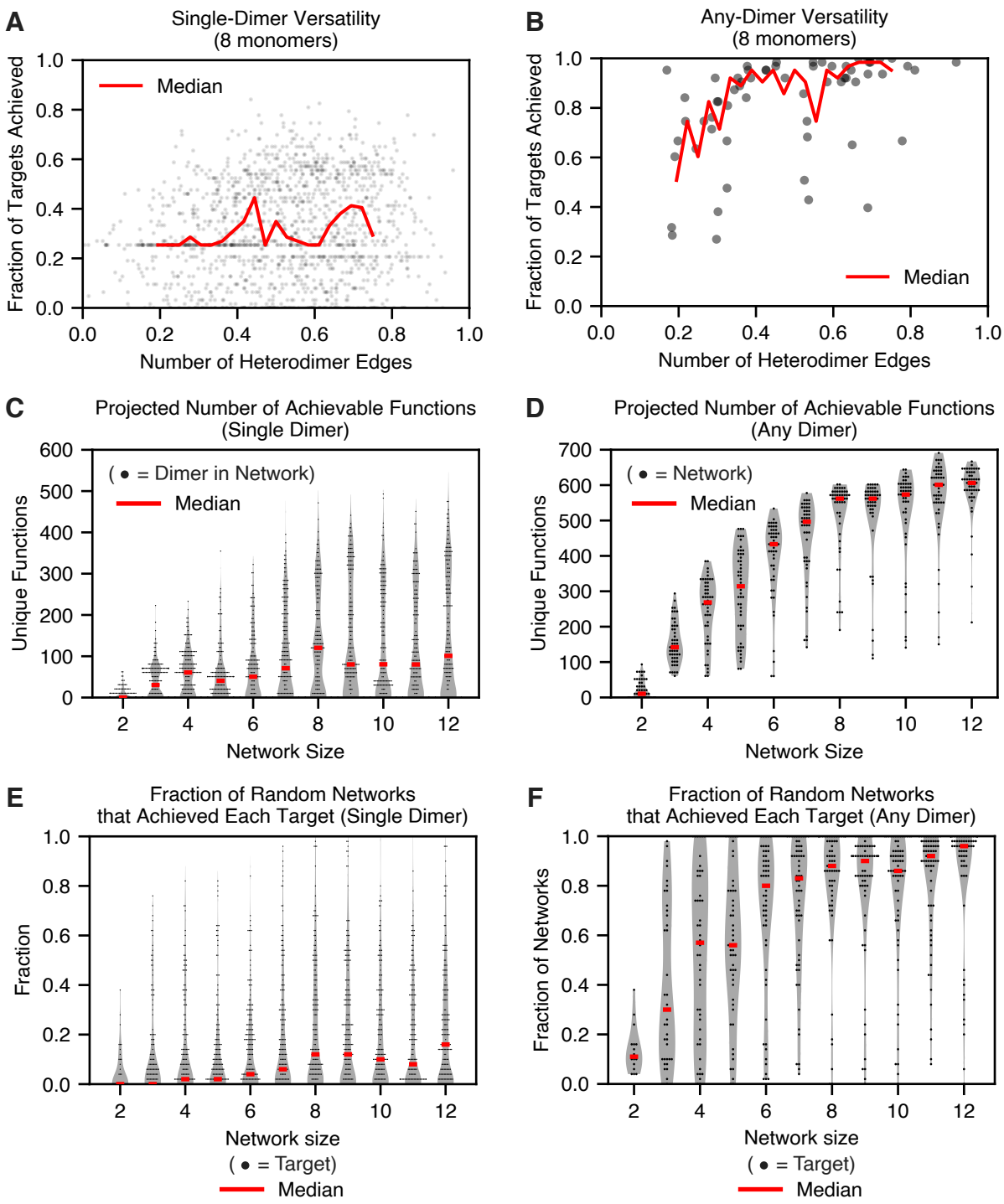
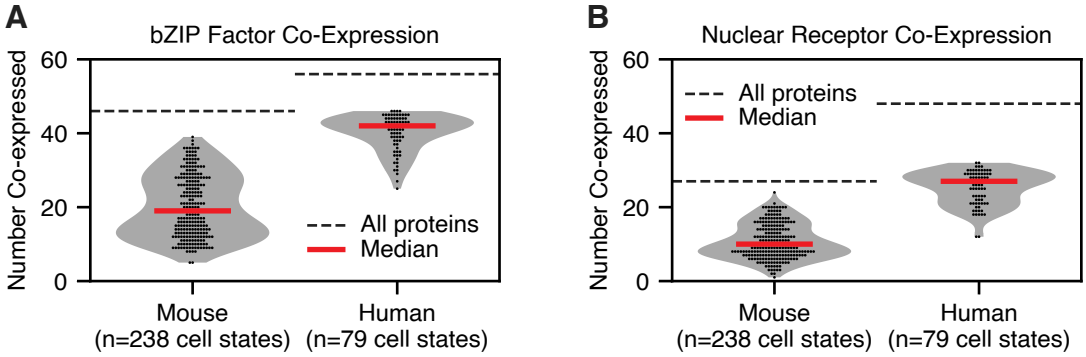

A
Single-Dimer Versatility
(8 monomers)
Fraction of Targets Achieved
Median
Number of Heterodimer Edges
B
Any-Dimer Versatility
(8 monomers)
Fraction of Targets Achieved
Median
Number of Heterodimer Edges
C
Projected Number of Achievable Functions
(Single Dimer)
Unique Functions
(• = Dimer in Network)
Median
Network Size
D
Projected Number of Achievable Functions
(Any Dimer)
Unique Functions
(• = Network)
Median
Network Size
E
Fraction of Random Networks
that Achieved Each Target (Single Dimer)
Fraction
Network size
(• = Target)
Median
F
Fraction of Random Networks
that Achieved Each Target (Any Dimer)
Fraction of Networks
Network size
(• = Target)
Median

Figure S6. The versatility of random networks increases with network size. (A-B) A scatter plot shows how network versatility using only a single output dimer (A) or any output dimer (B) varies with network connectivity. (C-D) While Figure 6B shows only the fraction of targets that could be achieved by each network, plotted here is the projected total number of targets each network is expected to achieve – accounting for differences in the overall expressivities of networks differing in size – for both cases in which only a single dimer (C) or any dimer (D) may be used as the output. (E-F) We also addressed the inverse question: how "easy" is each target function to achieve? A violin plot with scattered points shows the fraction of random networks that could achieve each target function, separated by network size and whether only a single dimer (E) or any dimer (F) may be used as the output. For the violin plots, the gray violins show the kernel density estimate of the data distributions, red lines show the median values, and only a random subset of the data is displayed as scattered points.



C

Organism, Protein Family	Family Members	Interacting Members	Co-expressed (median)	Connectivity	Refs
Human (*homo sapiens*) bZIP	57	> 21	42	20%	Reinke et al. 2013
Mouse (*mus musculus*) bZIP	46	N. D.	15	N. D.	
Arabidopsis thaliana bZIP	78	> 14	N. D.	24%	Li et al. 2022, Ehlert et al. 2006
Human (*homo sapiens*) Nuclear Receptor	48	> 30	28	28%	Amoutzias et al. 2007
Mouse (*mus musculus*) Nuclear Receptor	27	N. D.	7	N. D.	
Arabidopsis thaliana MADS-box	107	> 11	N. D.	40%	de Folter et al. 2005

Figure S7. Natural dimerization networks are of sufficient size to exhibit high expressivity. (A, B) Co-expression of bZIP (A) and nuclear receptor (NR) (B) transcription factors was assessed for many cell types across both mouse and human datasets. A violin plot with scattered points shows the number of network proteins co-expressed in each cell type. Gray violins show the kernel density estimate of the data distributions, red lines show the median values, and black dotted lines indicate the total number of genes assessed. (C) Table summarizing the size, number of known interacting members, number of co-expressed members, and connectivity of several natural dimerization networks. N.D. indicates that an entry was not determined.

Supplementary Video 1: Tunable Two-input Band-pass Filter

Supplementary Video 2: Versatile Network for Both AND and NOR Logic Gates

Supplementary Video 3: Versatile Network for Both OR and NOR Logic Gates

References

1. Amoutzias, G.D., Pichler, E.E., Mian, N., De Graaf, D., Imsiridou, A., Robinson-Rechavi, M., Bornberg-Bauer, E., Robertson, D.L., and Oliver, S.G. (2007). A protein interaction atlas for the nuclear receptors: properties and quality of a hub-based dimerisation network. BMC Syst. Biol. *1*, 34. 10.1186/1752-0509-1-34.

2. Li, M., Lin, W., Hinckley, W., Yao, T., Muchero, W., Chen, J.-G., and Carol Huang, S. (2022). Uncovering the "ZIP code" for bZIP dimers reveals novel motifs, regulatory rules and one billion years of cis-element evolution. bioRxiv, 2022.04.17.488518. 10.1101/2022.04.17.488518.

3. Amoutzias, G.D., Veron, A.S., Weiner, J., 3rd, Robinson-Rechavi, M., Bornberg-Bauer, E., Oliver, S.G., and Robertson, D.L. (2007). One billion years of bZIP transcription factor evolution: conservation and change in dimerization and DNA-binding site specificity. Mol. Biol. Evol. *24*, 827–835. 10.1093/molbev/msl211.

4. Reinke, A.W., Baek, J., Ashenberg, O., and Keating, A.E. (2013). Networks of bZIP protein-protein interactions diversified over a billion years of evolution. Science *340*, 730–734. 10.1126/science.1233465.

5. Murre, C. (2019). Helix-loop-helix proteins and the advent of cellular diversity: 30 years of discovery. Genes Dev. *33*, 6–25. 10.1101/gad.320663.118.

6. de Folter, S., Immink, R.G.H., Kieffer, M., Parenicová, L., Henz, S.R., Weigel, D., Busscher, M., Kooiker, M., Colombo, L., Kater, M.M., et al. (2005). Comprehensive interaction map of the Arabidopsis MADS Box transcription factors. Plant Cell *17*, 1424–1433. 10.1105/tpc.105.031831.

7. Amoutzias, G.D., Robertson, D.L., Van de Peer, Y., and Oliver, S.G. (2008). Choose your partners: dimerization in eukaryotic transcription factors. Trends Biochem. Sci. *33*, 220–229. 10.1016/j.tibs.2008.02.002.

8. Mechta-Grigoriou, F., Gerald, D., and Yaniv, M. (2001). The mammalian Jun proteins: redundancy and specificity. Oncogene *20*, 2378–2389. 10.1038/sj.onc.1204381.

9. Lourenco, C., Resetca, D., Redel, C., Lin, P., MacDonald, A.S., Ciaccio, R., Kenney, T.M.G., Wei, Y., Andrews, D.W., Sunnerhagen, M., et al. (2021). MYC protein interactors in gene transcription and cancer. Nat. Rev. Cancer *21*, 579–591. 10.1038/s41568-021-00367-9.

10. Dai, C., Heemers, H., and Sharifi, N. (2017). Androgen Signaling in Prostate Cancer. Cold Spring Harb. Perspect. Med. *7*. 10.1101/cshperspect.a030452.

11. Hidaka, R., Miyazaki, K., and Miyazaki, M. (2022). The E-Id Axis Instructs Adaptive Versus Innate Lineage Cell Fate Choice and Instructs Regulatory T Cell Differentiation. Front. Immunol. *13*, 890056. 10.3389/fimmu.2022.890056.

12. Sharma, S., Shen, T., Chitranshi, N., Gupta, V., Basavarajappa, D., Sarkar, S., Mirzaei, M., You, Y., Krezel, W., Graham, S.L., et al. (2022). Retinoid X Receptor: Cellular and Biochemical Roles of Nuclear Receptor with a Focus on Neuropathological Involvement. Mol. Neurobiol. *59*, 2027–2050. 10.1007/s12035-021-02709-y.

13. Puig-Barbé, A., and de Navascués, J. (2019). Homo- and heterodimerization of bHLH transcription factors balance stemness and bipotential differentiation. bioRxiv. 10.1101/685347.

14. De Bosscher, K., Desmet, S.J., Clarisse, D., Estébanez-Perpiña, E., and Brunsveld, L. (2020). Nuclear receptor crosstalk—defining the mechanisms for therapeutic innovation. Nat. Rev. Endocrinol. *16*, 363–377.

15. Jia, M., Dahlman-Wright, K., and Gustafsson, J.-Å. (2015). Estrogen receptor alpha and beta in health and disease. Best Pract. Res. Clin. Endocrinol. Metab. *29*, 557–568. 10.1016/j.beem.2015.04.008.

16. Korbecki, J., Simińska, D., Gąssowska-Dobrowolska, M., Listos, J., Gutowska, I., Chlubek, D., and Baranowska-Bosiacka, I. (2021). Chronic and Cycling Hypoxia: Drivers of Cancer Chronic Inflammation through HIF-1 and NF-κB Activation: A Review of the Molecular Mechanisms. Int. J. Mol. Sci. *22*. 10.3390/ijms221910701.

17. Rijo-Ferreira, F., and Takahashi, J.S. (2019). Genomics of circadian rhythms in health and disease. Genome Med. *11*, 82. 10.1186/s13073-019-0704-0.

18. Antebi, Y.E., Linton, J.M., Klumpe, H., Bintu, B., Gong, M., Su, C., McCardell, R., and Elowitz, M.B. (2017). Combinatorial Signal Perception in the BMP Pathway. Cell *170*, 1184–1196.e24. 10.1016/j.cell.2017.08.015.

19. Klumpe, H.E., Langley, M.A., Linton, J.M., Su, C.J., Antebi, Y.E., and Elowitz, M.B. (2022). The context-dependent, combinatorial logic of BMP signaling. Cell Syst *13*, 388–407.e10. 10.1016/j.cels.2022.03.002.

20. Sprinzak, D., Lakhanpal, A., Lebon, L., Santat, L.A., Fontes, M.E., Anderson, G.A., Garcia-Ojalvo, J., and Elowitz, M.B. (2010). Cis-interactions between Notch and Delta generate mutually exclusive signalling states. Nature *465*, 86–90. 10.1038/nature08959.

21. Schreiner, D., and Weiner, J.A. (2010). Combinatorial homophilic interaction between gamma-protocadherin multimers greatly expands the molecular diversity of cell adhesion. Proc. Natl. Acad. Sci. U. S. A. *107*, 14893–14898. 10.1073/pnas.1004526107.

22. Thu, C.A., Chen, W.V., Rubinstein, R., Chevee, M., Wolcott, H.N., Felsovalyi, K.O., Tapia, J.C., Shapiro, L., Honig, B., and Maniatis, T. (2014). Single-cell identity generated by combinatorial homophilic interactions between α, β, and γ protocadherins. Cell *158*, 1045–1059. 10.1016/j.cell.2014.07.012.

23. Klumpe, H.E., Garcia-Ojalvo, J., Elowitz, M.B., and Antebi, Y.E. (2023). The computational capabilities of many-to-many protein interaction networks. Cell Syst *14*, 430–446. 10.1016/j.cels.2023.05.001.

24. Granados, A.A., Kanrar, N., and Elowitz, M.B. (2022). Combinatorial expression motifs in signaling pathways. bioRxiv, 2022.08.21.504714. 10.1101/2022.08.21.504714.

25. Maslov, S., and Ispolatov, I. (2007). Propagation of large concentration changes in reversible protein-binding networks. Proc. Natl. Acad. Sci. U. S. A. *104*, 13655–13660. 10.1073/pnas.0702905104.

26. Celaj, A., Schlecht, U., Smith, J.D., Xu, W., Suresh, S., Miranda, M., Aparicio, A.M., Proctor, M., Davis, R.W., Roth, F.P., et al. (2017). Quantitative analysis of protein interaction network dynamics in yeast. Mol. Syst. Biol. *13*, 934. 10.15252/msb.20177532.

27. Neuhold, L.A., and Wold, B. (1993). HLH forced dimers: tethering MyoD to E47 generates a dominant positive myogenic factor insulated from negative regulation by Id. Cell *74*, 1033–1042. 10.1016/0092-8674(93)90725-6.

28. Su, C.J., Murugan, A., Linton, J.M., Yeluri, A., Bois, J., Klumpe, H., Langley, M.A., Antebi, Y.E., and Elowitz, M.B. (2022). Ligand-receptor promiscuity enables cellular addressing. Cell Syst *13*, 408–425.e12. 10.1016/j.cels.2022.03.001.

29. Hjelmfelt, A., Weinberger, E.D., and Ross, J. (1991). Chemical implementation of neural networks and Turing machines. Proc. Natl. Acad. Sci. U. S. A. *88*, 10983–10987. 10.1073/pnas.88.24.10983.

30. Chen, H.-L., Doty, D., and Soloveichik, D. (2012). Deterministic Function Computation with Chemical Reaction Networks. arXiv [cs.CC]. 10.48550/ARXIV.1204.4176.

31. Chan, C.-H., Shih, C.-Y., and Chen, H.-L. (2022). On the Computational Power of Phosphate Transfer Reaction Networks. New Generation Computing *40*, 603–621. 10.1007/s00354-022-00154-6.

32. Bengio, Y., and Delalleau, O. (2011). On the expressive power of deep architectures. In Discovery Science Lecture notes in computer science. (Springer Berlin Heidelberg), pp. 1–1. 10.1007/978-3-642-24477-3_1.

33. Gühring, I., Raslan, M., and Kutyniok, G. (2022). Expressivity of Deep Neural Networks. In Mathematical Aspects of Deep Learning (Cambridge University Press), pp. 149–199. 10.1017/9781009025096.004.

34. Kramer, B.A., Sarabia Del Castillo, J., and Pelkmans, L. (2022). Multimodal perception links cellular state to decision-making in single cells. Science *377*, 642–648. 10.1126/science.abf4062.

35. Yao, J., Pilko, A., and Wollman, R. (2016). Distinct cellular states determine calcium signaling response. Mol. Syst. Biol. *12*, 894. 10.15252/msb.20167137.

36. Cheong, R., Rhee, A., Wang, C.J., Nemenman, I., and Levchenko, A. (2011). Information transduction capacity of noisy biochemical signaling networks. Science *334*, 354–358. 10.1126/science.1204553.

37. Shakiba, N., Jones, R.D., Weiss, R., and Del Vecchio, D. (2021). Context-aware synthetic biology by controller design: Engineering the mammalian cell. Cell Syst *12*, 561–592. 10.1016/j.cels.2021.05.011.

38. Maslov, S. (2008). Topological and Dynamical Properties of Protein Interaction Networks. In Protein-protein Interactions and Networks: Identification, Computer Analysis, and Prediction, A. Panchenko and T. Przytycka, eds. (Springer London), pp. 115–137. 10.1007/978-1-84800-125-1_7.

39. Xiang, Y., Sun, D.Y., Fan, W., and Gong, X.G. (1997). Generalized simulated annealing algorithm and its application to the Thomson model. Phys. Lett. A *233*, 216–220. 10.1016/S0375-9601(97)00474-X.

40. Lambora, A., Gupta, K., and Chopra, K. (2019). Genetic Algorithm- A Literature Review. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 380–384. 10.1109/COMITCon.2019.8862255.

41. Hargrove, J.L., Hulsey, M.G., and Beale, E.G. (1991). The kinetics of mammalian gene expression.

Bioessays *13*, 667–674. 10.1002/bies.950131209.

42. Rosenfeld, N., Elowitz, M.B., and Alon, U. (2002). Negative autoregulation speeds the response times of transcription networks. J. Mol. Biol. *323*, 785–793. 10.1016/s0022-2836(02)00994-4.

43. Elowitz, M.B., Levine, A.J., Siggia, E.D., and Swain, P.S. (2002). Stochastic gene expression in a single cell. Science *297*, 1183–1186. 10.1126/science.1070919.

44. Barkai, N., and Shilo, B.-Z. (2007). Variability and robustness in biomolecular systems. Mol. Cell *28*, 755–760. 10.1016/j.molcel.2007.11.013.

45. Quarton, T., Kang, T., Papakis, V., Nguyen, K., Nowak, C., Li, Y., and Bleris, L. (2020). Uncoupling gene expression noise along the central dogma using genome engineered human cell lines. Nucleic Acids Res. *48*, 9406–9413. 10.1093/nar/gkaa668.

46. Bundschuh, R., Hayot, F., and Jayaprakash, C. (2003). The role of dimerization in noise reduction of simple genetic networks. J. Theor. Biol. *220*, 261–269. 10.1006/jtbi.2003.3164.

47. Yan, K.-K., Walker, D., and Maslov, S. (2008). Fluctuations in mass-action equilibrium of protein binding networks. Phys. Rev. Lett. *101*, 268102. 10.1103/PhysRevLett.101.268102.

48. Halligan, D.L., Kousathanas, A., Ness, R.W., Harr, B., Eöry, L., Keane, T.M., Adams, D.J., and Keightley, P.D. (2013). Contributions of protein-coding and regulatory change to adaptive molecular evolution in murid rodents. PLoS Genet. *9*, e1003995. 10.1371/journal.pgen.1003995.

49. Gao, R., and Stock, A.M. (2013). Evolutionary tuning of protein expression levels of a positively autoregulated two-component system. PLoS Genet. *9*, e1003927. 10.1371/journal.pgen.1003927.

50. Rahimi, A., and Recht, B. (2007). Random Features for Large-Scale Kernel Machines. Adv. Neural Inf. Process. Syst. *20*.

51. Rudi, A., and Rosasco, L. (2017). Generalization properties of learning with random features. In Advances in neural information processing systems *30*.

52. Grigoryeva, L., and Ortega, J.-P. (2018). Echo state networks are universal. Neural Netw. *108*, 495–508. 10.1016/j.neunet.2018.08.025.

53. Stern, M., and Murugan, A. (2023). Learning Without Neurons in Physical Systems. Annu. Rev. Condens. Matter Phys. *14*, 417–441. 10.1146/annurev-conmatphys-040821-113439.

54. Rodríguez-Martínez, J.A., Reinke, A.W., Bhimsaria, D., Keating, A.E., and Ansari, A.Z. (2017). Combinatorial bZIP dimers display complex DNA-binding specificity landscapes. Elife *6*. 10.7554/eLife.19272.

55. Buchler, N.E., and Louis, M. (2008). Molecular titration and ultrasensitivity in regulatory networks. J. Mol. Biol. *384*, 1106–1119. 10.1016/j.jmb.2008.09.079.

56. Buchler, N.E., and Cross, F.R. (2009). Protein sequestration generates a flexible ultrasensitive response in a genetic network. Mol. Syst. Biol. *5*, 272. 10.1038/msb.2009.30.

57. Thompson, K.E., Bashor, C.J., Lim, W.A., and Keating, A.E. (2012). SYNZIP protein interaction toolbox: in vitro and in vivo specifications of heterospecific coiled-coil interaction domains. ACS Synth. Biol. *1*, 118–129. 10.1021/sb200015u.

58. Chen, Z., Boyken, S.E., Jia, M., Busch, F., Flores-Solis, D., Bick, M.J., Lu, P., VanAernum, Z.L., Sahasrabuddhe, A., Langan, R.A., et al. (2019). Programmable design of orthogonal protein heterodimers. Nature *565*, 106–111. 10.1038/s41586-018-0802-y.

59. Gradišar, H., and Jerala, R. (2011). De novo design of orthogonal peptide pairs forming parallel coiled-coil heterodimers. J. Pept. Sci. *17*, 100–106. 10.1002/psc.1331.

60. Baryshev, A., Fleur, A.L., Groves, B., Michel, C., Baker, D., Ljubeti*f*ç, A., and Seelig, G. (2023). Massively parallel protein-protein interaction measurement by sequencing (MP3-seq) enables rapid screening of protein heterodimers. bioRxiv. 10.1101/2023.02.08.527770.

61. Boldridge, W.C., Ljubetič, A., Kim, H., Lubock, N., Szilágyi, D., Lee, J., Brodnik, A., Jerala, R., and Kosuri, S. (2023). A multiplexed bacterial two-hybrid for rapid characterization of protein-protein interactions and iterative protein design. Nat. Commun. *14*, 4636. 10.1038/s41467-023-38697-x.

62. Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Netw. *2*, 359–366. 10.1016/0893-6080(89)90020-8.

63. Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems *2*, 303–314. 10.1007/BF02551274.

64. Hertz, J., Krogh, A., and Palmer, R.G. (2018). Introduction. In Introduction to the Theory of Neural Computation (CRC Press), pp. 1–10. 10.1201/9780429499661-1.

65. Schwanhäusser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J., Wolf, J., Chen, W., and Selbach, M. (2011). Global quantification of mammalian gene expression control. Nature *473*, 337–342. 10.1038/nature10098.

66. Beck, M., Schmidt, A., Malmstroem, J., Claassen, M., Ori, A., Szymborska, A., Herzog, F., Rinner, O., Ellenberg, J., and Aebersold, R. (2011). The quantitative proteome of a human cell line. Mol. Syst. Biol. *7*, 549. 10.1038/msb.2011.82.

67. Gillespie, M.A., Palii, C.G., Sanchez-Taltavull, D., Shannon, P., Longabaugh, W.J.R., Downes, D.J., Sivaraman, K., Espinoza, H.M., Hughes, J.R., Price, N.D., et al. (2020). Absolute Quantification of Transcription Factors Reveals Principles of Gene Regulation in Erythropoiesis. Mol. Cell *78*, 960–974.e11. 10.1016/j.molcel.2020.03.031.

68. Kageyama, R., Shimojo, H., and Ohtsuka, T. (2019). Dynamic control of neural stem cells by bHLH factors. Neurosci. Res. *138*, 12–18. 10.1016/j.neures.2018.09.005.

69. Zhu, R., Del Rio-Salgado, J.M., Garcia-Ojalvo, J., and Elowitz, M.B. (2022). Synthetic multistability in mammalian cells. Science *375*, eabg9765. 10.1126/science.abg9765.

70. Bois, J.S. (2020). justinbois/eqtk: Version 0.1.1 (CaltechDATA) 10.22002/D1.1430.

71. Lang, F., Ritter, M., Wöll, E., Weiss, H., Häussinger, D., Hoflacher, J., Maly, K., and Grunicke, H. (1992). Altered cell volume regulation in ras oncogene expressing NIH fibroblasts. Pflugers Arch. *420*, 424–427. 10.1007/BF00374615.

72. Wang, R., Fang, X., Lu, Y., and Wang, S. (2004). The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. J. Med. Chem. *47*, 2977–2980. 10.1021/jm030580l.

73. Lomax, J.E., Bianchetti, C.M., Chang, A., Phillips, G.N., Jr, Fox, B.G., and Raines, R.T. (2014). Functional evolution of ribonuclease inhibitor: insights from birds and reptiles. J. Mol. Biol. *426*, 3041–3056. 10.1016/j.jmb.2014.06.007.

74. Schreiber, G., and Keating, A.E. (2011). Protein binding specificity versus promiscuity. Curr. Opin. Struct. Biol. *21*, 50–61. 10.1016/j.sbi.2010.10.002.

75. Blank, J., and Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. IEEE Access *8*, 89497–89509. 10.1109/access.2020.2990567.

76. Chao, H., Houston, M.E., Jr, Grothe, S., Kay, C.M., O'Connor-McCourt, M., Irvin, R.T., and Hodges, R.S. (1996). Kinetic study on the formation of a de novo designed heterodimeric coiled-coil: use of surface plasmon resonance to monitor the association and dissociation of polypeptide chains. Biochemistry *35*, 12175–12185. 10.1021/bi9530604.

77. Amrhein, L., Harsha, K., and Fuchs, C. (2019). A mechanistic model for the negative binomial distribution of single-cell mRNA counts. bioRxiv, 657619. 10.1101/657619.

78. Uhlén, M., Fagerberg, L., Hallström, B.M., Lindskog, C., Oksvold, P., Mardinoglu, A., Sivertsson, Å., Kampf, C., Sjöstedt, E., Asplund, A., et al. (2015). Tissue-based map of the human proteome. Science *347*, 1260419. 10.1126/science.1260419.

79. UniProt Consortium (2023). UniProt: the Universal Protein Knowledgebase in 2023. Nucleic Acids Res. *51*, D523–D531. 10.1093/nar/gkac1052.