

Supplementary Information

Engineering wetware and software for the predictive design of compressed genetic circuits for higher-state decision-making

Brian D. Huang^{1*}, Yongjoon Yu^{1*}, Junghwan Lee², Matthew W. Repasky², Yao Xie², Matthew J. Realff¹, & Corey J. Wilson^{1†}

¹School of Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA, USA

²H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

* authors contributed equally to this work

†To whom correspondence should be addressed: Corey J. Wilson, Georgia Institute of Technology, School of Chemical & Biomolecular Engineering, 311 Ferst Drive, Atlanta, GA 30332-0100. Telephone: 404-385-5397; E-Mail: corey.wilson@chbe.gatech.edu

Table of Contents

Supplementary Note 1: Defining the synthetic biology problem.

Supplementary Note 2: Defining inversion versus compression and the limit of predictive design.

Supplementary Note 3: Transcriptional Programming biocomputing wetware capacity

Supplementary Note 4: Combinatorial Space for 3-INPUT T-Pro circuits.

Supplementary Note 5: Assumptions of the expression units (EU) and ribosomal binding site (RBS) scaling.

Supplementary Note 6: Toxicity associated with lycopene biosynthesis.

Supplementary Note 7: Complexity analysis and other considerations.

Supplementary Table 1 | Computational resource requirements for design space configuration.

Supplementary Fig. 1 | Boolean logic and inversion versus T-Pro circuit compression.

Supplementary Fig. 2-17 | T-Pro Workflow and models: qualitative and quantitative design.

T-Pro Qualitative Design:

Supplementary Fig. 2 | T-Pro Qualitative Design Step 1.

Supplementary Fig. 3 | T-Pro Qualitative Design Step 2.

Supplementary Fig. 4 | T-Pro Qualitative Design Step 3.

T-Pro Quantitative Design:

Supplementary Fig. 5 | T-Pro Quantitative Design Step 1.

Supplementary Fig. 6 | T-Pro Quantitative Design Step 2.

Supplementary Fig. 7 | T-Pro Quantitative Design Step 3.

T-Pro Qualitative Design (flow diagram & algorithm):

Supplementary Fig. 8 | Algorithmic Enumeration Workflow.

Supplementary Fig. 9 | Recursive Search Algorithm.

Supplementary Fig. 10 | Code snippet to run enumeration.

Supplementary Fig. 11 | Circuits corresponding to unique input logic strings.

Supplementary Fig. 12 | CSR matrix representation converted to a human-readable circuit format.

Supplementary Fig. 13 | Identification of the optimized circuit.

T-Pro Quantitative Design Models & Software

Supplementary Fig. 14 | Single-Input Single-Output (SISO) performance optimization.

Supplementary Fig. 15 | Interactive Hill Modeling Software.

Supplementary Fig. 16 | SISO to MISO Logic.

Supplementary Fig. 17 | Cascading TFs with RBS Scaling and MISO Modeling.

*Source code is available at: <https://github.com/Jayaos/TPro>

** Software is publicly available at <https://doi.org/10.5281/zenodo.16988958>

Supplementary Fig. 18 | Non-modularity of RBS libraries applied to different synthetic promoters.

Supplementary Fig. 19 | Non-modularity of ribozymes.

Supplementary Fig. 20 | CelR and anti-CelR titration curves.

Supplementary Fig. 21 | RbsR and anti-RbsR titration curves.

Supplementary Fig. 22 | SISO predictions for RbsR-based circuits.

Supplementary Fig. 23 | LacI and anti-LacI titration curves.

Supplementary Fig. 24 | SISO predictions for LacI-based circuits.

Supplementary Fig. 25 | SISO and MISO statistical correlation analysis.

Supplementary Fig. 26 | Synthetic promoter orthogonality.

Supplementary Fig. 27 | Additional 2-input circuits.

Supplementary Fig. 28 | RBS context across variants and media conditions.

Supplementary Fig. 29 | Dynamic response comparison of T-Pro and inversion-based NOT and NOR gate.

Supplementary Fig. 30 | Validating a 25Nanoluc-GFP cassette.

Supplementary Fig. 31 | Probing recombinase functionality and degradation.

Supplementary Fig. 32 | Comparison of sequence-based models of expression.

Supplementary Fig. 33 | Genetic logic gate implementations across different design frameworks.

Supplementary Note 1: Defining the synthetic biology problem. *Descriptive legend* – below is a summary of the context variables for genetic circuit composability.

Despite our qualitative understanding of how to design fundamental genetic circuit architectures, our ability to accurately predict the quantitative performance of an engineered biological circuit remains to be a significant challenge. We define the inability to accurately predict the quantitative performance of an engineered biological circuit—*i.e.*, a circuit composed of known and well characterized parts—as ‘the synthetic biology problem’. We posit that genetic components characterized in one milieu (*i.e.*, genetic, cellular, and nutrient) may not behave the same in another setting due to a variety of context-dependent interactions, including:

1. Promoter and operator contexts
2. Ribosome binding site (RBS) contextual effects
3. Transcriptional read-through and leaky expression
4. Cryptic promoter activity
5. Insulator contexts
6. Gene of Interest (GOI) contextual effects
7. Chassis cell type
8. Growth media / environment

We posit that bridging the gap between our understanding of individual parts and composed performance will require a deeper understanding of context-dependent interactions (*i.e.*, genetic, chassis cell, and growth environment) and the development of new strategies for constructing genetic circuits with reliable, predictable performance. NOTE: To mitigate the issues arising from context-dependent interactions, synthetic biologists are increasingly focusing on designing modular and orthogonal genetic parts that minimize unwanted interactions¹⁻³. Additionally, empirical characterization of genetic components in multiple contexts is becoming more common to better predict and tune their behavior in different settings⁴⁻⁶. Despite our qualitative understanding of how to design fundamental genetic circuit architectures, the synthetic biology problem—*i.e.*, accurately predicting biological circuit quantitative performance—remains a significant challenge.

Supplementary Note 2: Defining inversion versus compression and the limit of predictive design. *Descriptive legend* – below is a summary of the distinction between classical genetic circuit design and genetic circuit design achieved *via* compression.

Automated genetic circuit design tools often produce qualitatively accurate architectures, but they fall short in providing precise quantitative predictions. This limitation arises primarily from the inability to fully model the complex biophysical characteristics governing genetic components. One practical approach to mitigate this issue is by minimizing the number of genetic components in the system, which reduces the overall complexity and the number of parameters that must be accounted for, thus enhancing predictive accuracy. In this context, synthetic anti-repressors offer an efficient solution by enabling NOT Boolean operations without the need for the traditional inversion mechanism, which involves two repressors connected in a feedforward cascade (see **Supplementary Fig. 1c-d**). By simplifying the design, synthetic anti-repressors reduce the number of promoters required from 2 to 1^{3,7}. Furthermore, the use of two coordinated synthetic anti-repressors can facilitate a NOR operation, achieving a 5 to 1 reduction in the number of promoters compared to a strict inversion-based NOR circuit⁸.

Supplementary Note 3: Transcriptional Programming biocomputing wetware capacity. *Descriptive legend* – below is a summary of the T-Pro biocomputing wetware expansion used in this study.

In this manuscript, we have expanded our Transcriptional Programming wetware to include five repressor/anti-repressor pairs that can be equipped with seven orthogonal synthetic alternate DNA recognition (ADR) functions (**Fig. 1**). We have previously engineered four sets of repressor/anti-repressor pairs using protein engineering workflows that we developed in^{3,9,10}. However, some engineered synthetic transcription factors exhibited either a

low dynamic range or a response to multiple inducers, which limits their utility in the context of expanding Boolean biocomputing *via* T-Pro. Moving forward, we chose to focus on a constrained design space based on orthogonality of ligand inputs and ADRs for this study (**Fig. 1d**).

Supplementary Note 4: Combinatorial space for 3-input T-Pro circuits. *Descriptive legend* – below is an outline of the design space and design rules for eight-state T-Pro logic.

Three signal orthogonal synthetic transcription factors—*i.e.*, repressors, anti-repressors, and super-repressors with 5 alternate DNA binding functions to synthetic promoters—comprise a total of 45 possible generalized transcription factors. Including the option of no transcription factor, we consider 46 possible combinations of components. Circuit topologies where 2 or more transcription factors are directed to a given promoter, and the ability of feedforward information processing up to 4 layers* is allowed. Assuming a single 3-input circuit can incorporate up to 9 transcription factors, the upper bound of the combinatorial space for 3-input T-Pro circuits is 46^9 , approximately 10^{14} without additional constraints***.

*Circuit topologies where two or more transcription factors regulate a given promoter are allowed, as well as hierarchical transcriptional regulation in a feedforward cascade**, enabling information processing across up to four sequential regulatory layers. Each layer represents a distinct level in the regulatory cascade where transcription factors influence downstream promoters, allowing multi-stage decision-making. Executing multiple independent Boolean operations within a single cell using orthogonal components is also feasible within our framework, provided that regulatory elements remain functionally isolated.

** In conventional network biology, feedforward loops involve three components (X, Y, and Z) with direct and indirect regulatory interactions, including an X-Z connection¹¹. However, our circuit architecture does not include a direct interaction between the first and third components. Instead, it follows a sequential feedforward cascade structure, where regulatory signals are transmitted unidirectionally without forming a closed feedback mechanism. Note, T-Pro inverters are implemented strategically and do not negatively impact circuit compression.

***To reduce the initial search space we implemented a set of rules to mitigate unnecessary conflicts in the circuit design space—*i.e.*, only one phenotype (repressor/anti-repressor) may be constitutively expressed for a given transcription factor, identical transcription factors cannot be regulated by multiple promoters, and a transcription factor may not regulate itself.

Supplementary Note 5: Assumptions of the expression units (EU) and ribosomal binding site (RBS) scaling. *Descriptive legend* – below is a summary detailing the EU metric, related context, and significance.

We assume that the dominant factors that determine a gene's expression level are:

1. The transcription rate of the associated promoter in the context of the specific 5' untranslated regions (UTR), RBS, and coding DNA sequence (CDS) of the gene.
2. The translation initiation rate, which is primarily determined by the processing and folding of the 5' UTR and initial ~75 bases of mRNA 3' of the RBS.

As such, a fusion of GFP to the first 16-25 amino acids of a gene of interest (GOI) is assumed to maintain the approximate expression level of the full-length GOI, which can be measured *via* the fluorescence of GFP.

We further assume that the relative strength of an RBS as measured using a GFP fusion reporter with a fixed promoter and 5' UTR will be the same if used with the full-length GOI. Extending this concept, we assume that the output expression level of a promoter being regulated by a TF can be linearly scaled when changing the RBS by multiplying the original output by the ratio of relative RBS strengths.

We define Expression Units (EU) as a relative measure of gene expression that captures both transcriptional and translational output. In contrast to Relative Promoter Units (RPU), which quantify promoter activity by

normalizing fluorescence output to a reference promoter under standardized conditions, EU incorporates the local sequence context downstream of the promoter—specifically the 5' untranslated region (5' UTR) and the coding sequence in proximity to the start codon. This region is critical for translation initiation efficiency making EU more representative of the overall protein expression potential of a gene.

Supplementary Note 6: Toxicity associated with lycopene biosynthesis. *Descriptive legend* – below is a summary of the challenges related to lycopene biosynthesis.

The process of cloning circuits containing the *crtE*, *crtB*, and *crtI* genes, essential for lycopene biosynthesis, presented substantial challenges. One of the primary difficulties encountered was the need to maintain low expression levels of these genes. Higher expression levels were observed to induce cellular mutations, often resulting in the alteration of promoters to a weaker variant, thereby reducing the expression of these genes. This phenomenon likely represents a cellular response to mitigate the metabolic burden or toxicity associated with high levels of lycopene biosynthesis pathway proteins. Additionally, we encountered issues related to the use of IPTG as an inducer. Incidentally, other studies have also indicated that IPTG causes toxicity in cells expressing *crtEBI* genes^{12, 13}. In line with these findings, our experiments revealed that IPTG concentrations above a certain threshold (10 μ M) led to noticeable growth defects. This observation emphasizes the delicate balance required in the inducer concentrations to achieve optimal gene expression without adversely affecting cell viability. Furthermore, we observed instances of homologous recombination and random mutations during the assaying of these genes, as confirmed by sequencing analysis. These genetic instabilities further complicate the cloning and expression process, requiring careful consideration of cloning strategies and expression conditions. Overall, the cloning and expression of *crtE*, *crtB*, and *crtI* genes for lycopene biosynthesis present a complex interplay of factors that need to be finely balanced. The issues of promoter weakening, IPTG-induced toxicity, and genetic instability highlight the need for precise control over gene expression and careful optimization of experimental conditions in synthetic biology applications involving these genes.

Supplementary Note 7: Complexity analysis and other considerations. *Descriptive legend* – below is a summary of the computational resource usage for algorithmic enumeration and limits of 3-input T-Pro design and scale-up.

The search space size grows as $O(C^{2^n})$, where C is a constant determined by the number of alternate DNA-binding functions (ADRs), regulatory control domains (RCDs), and phenotypes, and n is the number of circuit inputs. Although the total size of the search space is C^{2^n} , exhaustive enumeration is often unnecessary. By traversing the design space in order of increasing complexity, as in our method—starting with a design space that contains fewer promoters—the algorithm can identify optimized circuits early without exploring the entire space. For instance, while the complete search space size is $\sim 10^{14}$ in our study, making exhaustive enumeration infeasible with typical computational resources, our results show that over 95% of optimized circuits were discovered within a practical runtime. **Supplementary Table 1** summarizes the time and memory usage required to enumerate circuits under specific design space configurations, which collectively yielded 256 optimized circuits. All computations were performed on a machine equipped with two Intel Xeon Gold 6226 CPUs. No graphical processing units (GPUs) were used. Summary table of computational resource usage for algorithmic enumeration under each design space configuration given below.

NOTE: A qualitative comparison of T-Pro vs. Cello circuits is given in **Supplementary Data 3**. On average T-Pro circuits are ~ 3 times smaller than inverter-based (Cello) circuits that utilize tandem promoter genetic architectures, and ~ 4 times smaller than inverter-based circuits that use strict inversion (*i.e.*, generalized Cello).

Moving beyond 3-Input Boolean logic: Our current approach to algorithmic enumeration of 3-input T-Pro circuits and compression optimization is computationally expensive and it will be challenging to scale this approach to 16-states or greater—as this would require advancing from 256 operations to $>65,000$ operations. To address this challenge, we are actively exploring new strategies that balance biological feasibility and computational tractability.

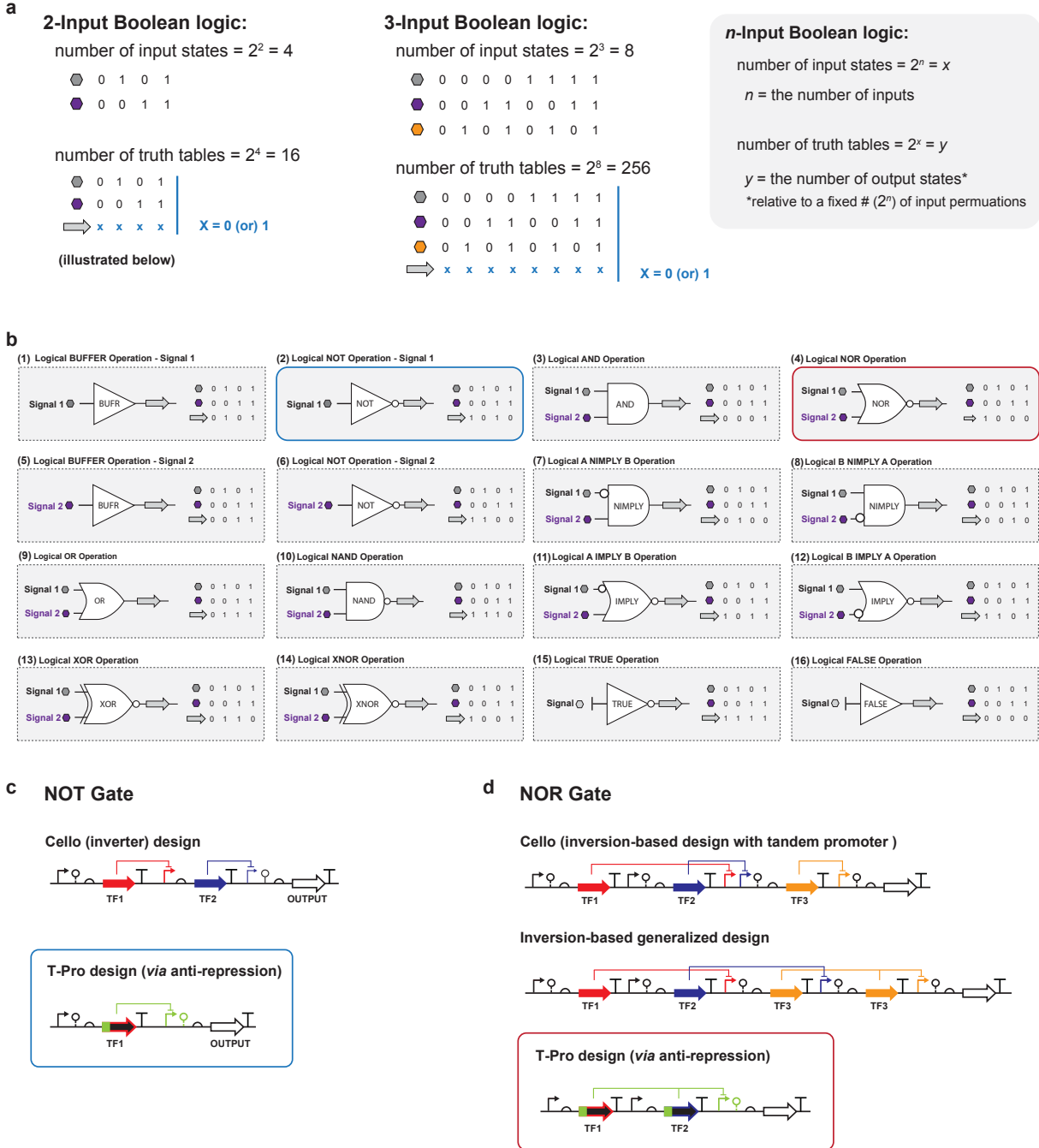
One approach under investigation involves using NOR-based logic structures, which can be simulated using digital logic generation software and remapping those architectures into biological circuit equivalents followed by circuit compression. The novel aspect of this approach is that it provides a foundation for identifying structural patterns and applying learned compression rules that reduce circuit complexity to its minimal functional form.

Evolutionary stability of T-Pro circuits: While our study did not conduct an extensive longitudinal analysis of mutational robustness, we can provide additional context regarding our observations and current suppositions. At the endpoint of our assay, the compressed circuits demonstrated consistent functionality under the specified growth conditions, suggesting a degree of stability in the short to medium term. We used *Escherichia coli* 3.32 cells, a daughter strain of K-12 MG1655, a well-characterized laboratory strain known for its relative genetic stability. While some mutation accumulation is expected during adaptive evolution, the controlled experimental conditions likely minimized significant genetic drift.

Additionally, it is well established—including in other circuit design studies (*e.g.*, Nielsen *et al.*¹⁴)—that circuit failure is often driven by homologous recombination between repeated genetic elements, rather than point mutations alone. In the present study, we did not specifically engineer circuits to minimize recombination risk, as the primary goal was to demonstrate technique development and predictive design capabilities. Nevertheless, future iterations of T-Pro circuits could incorporate sequence diversification strategies such as using orthogonal or non-repetitive genetic parts to mitigate recombination-driven failure and enhance evolutionary robustness for long-term applications.

# constitutive promoters	# regulating promoters	Max memory allocated	Execution time - wall clock
1	0	<4GB	<0.1hr
1	1	<4GB	<0.1hrs
2	0	<4GB	<0.1hrs
1	2	<4GB	<0.1hrs
2	1	<4GB	<0.1hrs
3	0	<4GB	<0.1hrs
2	2	<4GB	<2hrs
2	3	<32GB	<120hrs
3	2	<32GB	<120hrs
3	3	<32GB	<120hrs
3	4	<32GB	<120hrs
4	2	<32GB	<120hrs
4	3	<32GB	<120hrs
4	4	<32GB	<120hrs
5	4	<32GB	<120hrs

Supplementary Table 1 | Computational resource requirements for design space configuration. This table summarizes the memory allocation and execution time required to explore simple-to-complex genetic circuit topologies with the enumeration algorithm developed in this study. Each row represents a unique configuration defined by the number of constitutive and regulating promoters. As the total number of promoters increases, the required computational resources also increase.



Supplementary Fig. 1 | Boolean logic and inversion versus T-Pro circuit compression. **a** The number of input states and possible truth tables increases exponentially with the number of inputs in Boolean logic systems. For a 2-input system, there are 4 possible input states and 16 unique truth tables. For 3 inputs, this expands to 8 input states and 256 possible truth tables. This exponential scaling illustrates the need for scalable genetic logic architecture. **b** All 2-input Boolean logic functions are shown alongside with truth tables. **c** Comparison of NOT gate architectures between inversion-based (Cello) and T-Pro designs. Cello uses a repressive cascade requiring two transcription factors, while T-Pro implements a minimal anti-repression-based strategy. **d** Comparison of NOR gate designs shows that inversion-based topologies require tandem promoters (Cello) or multi-layered architectures, while the T-Pro design compresses the logic using fewer transcription factors and regulated elements.

T-Pro Workflow and models: qualitative and quantitative design

(Supplementary Figs. 2-17) *source code is available at: <https://github.com/Jayaos/TPro>

The T-Pro Workflow: Qualitative and Quantitative Designs. The T-Pro workflow is divided into two major stages: qualitative design and quantitative design. **Supplementary Figs. 2-4** outlines the qualitative design process, which involves three key steps to generate genetic circuits that implement user-defined logic functions. Additional sections follow describe the progression from qualitative design to quantitative design – including descriptions of the models and links to software **Supplementary Figs. 4-17**.

T-Pro Qualitative Design

Step 1: Truth Table Selection (user):

The process begins with the user defining a logic function in the form of a truth table. The system supports 2-input (**Supplementary Fig. 2a**) and 3-input (**Supplementary Fig. 2b**) logic functions, where each input corresponds to a biosensor-responsive transcription factor (TF). Inputs X, Y (and Z) can each represent one of several possible sensors (e.g., IPTG, Ribose, Cellobiose), and the desired output is specified for each input combination. Common logic functions such as AND, OR, and NOT combinations can be encoded.

</

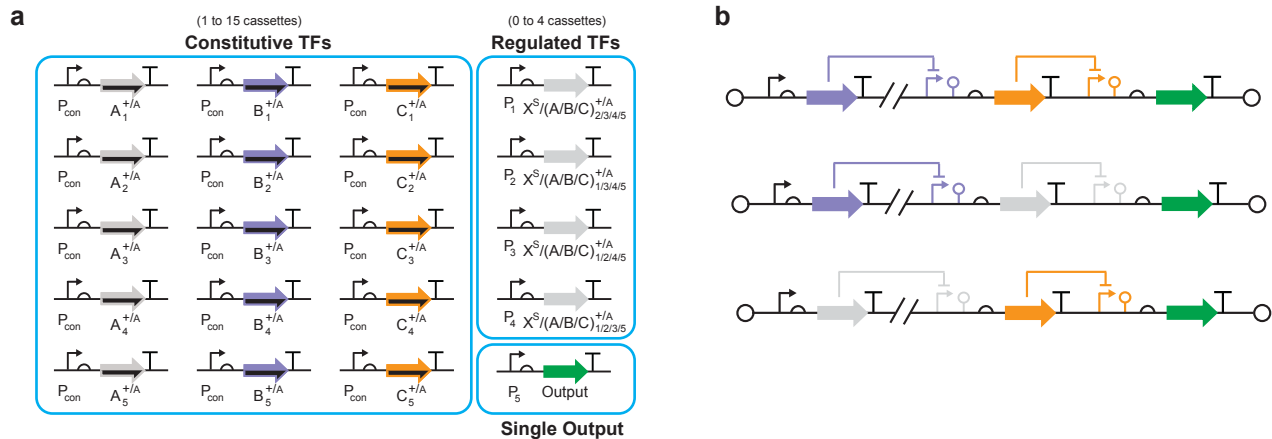
Supplementary Fig. 2 | T-Pro Qualitative Design Step 1. User defined logic functions in terms of a truth table (**a**) example of four-state logic and general 2-input truth-table description, and (**b**) eight-state logic and general 3-input truth-table description.

Step 2: Enumeration & Optimization (software)

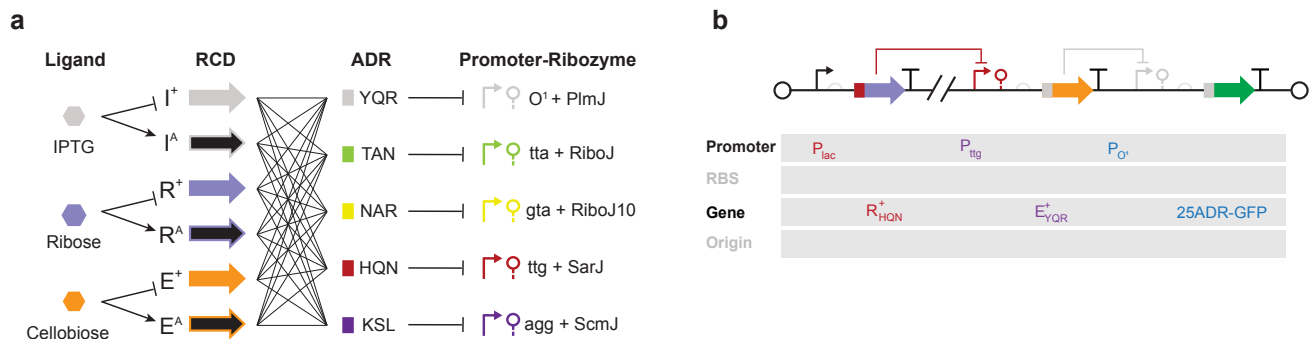
The specific logic gate is searched from a list of compressed circuits generated by an enumeration & optimization software (**Supplementary Fig. 3a**). The software systematically enumerates all possible genetic circuit topologies capable of implementing the desired 3-input logic function, progressing from simpler to more complex architectures. Each topology can contain up to 15 constitutive transcription factor (TF) cassettes and up to 4 regulated TF cassettes. At this stage, transcription factors are represented as placeholders (**Supplementary Fig. 3b**), enabling flexible matching in **Step 3**.

Step 3: Populate Qualitative Design with Desired Synthetic Transcription Factors and Output (user)

In the final step, the selected genetic architecture is populated with real, characterized synthetic transcription factors (wetware) from a defined library of 30 synthetic TFs (**Supplementary Fig. 4a**). Each TF is linked to a specific ligand, RCD (regulatory core domain), and ADR (alternate DNA recognition) unit, and controls a synthetic promoter-ribozyme combination. This step translates abstract designs into functional, testable genetic circuits by mapping logical inputs to biosensors and assigning a specific output gene, such as GFP (**Supplementary Fig. 4b**).



Supplementary Fig. 3 | T-Pro Qualitative Design Step 2. An illustration of the boundary conditions for qualitative T-Pro circuit design is given in (a). Circuit topology can contain up to 15 constitutive transcription factor (TF) cassettes and up to 4 regulated TF cassettes. An example of a circuit topology with placeholder TF cassettes is illustrated in (b) for iterations of an A IMPLY B operation.



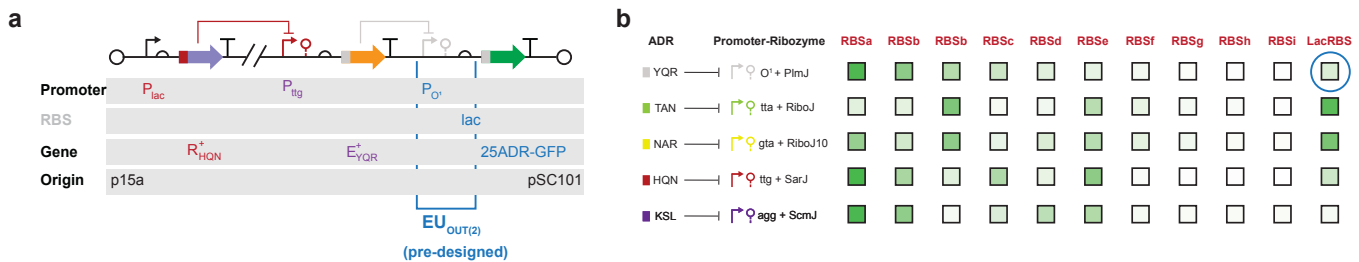
Supplementary Fig. 4 | T-Pro Qualitative Design Step 3. A defined library of 30 synthetic TFs and cognate synthetic promoters (a). An example of a populated A IMPLY B circuit (b) designed in Step 2 using parts from (a).

T-Pro Quantitative Design

The quantitative design phase fine-tunes the final genetic circuit architecture by selecting and optimizing regulatory parts for predictable and robust behavior in a biological context. This process involves three steps:

Step 1: Context-Specific Expression Cassette (CSEC) Selection for an Output

The output cassette is composed of modular promoter, RBS, and gene elements that can be tuned for desired quantitative response (**Supplementary Fig. 5a**). A pre-designed output unit (*e.g.*, P_{ol} , *lac*, 25ADR-GFP) is integrated with upstream transcriptional regulators (*e.g.*, R^+ , E^+) (**Supplementary Fig. 5b**).



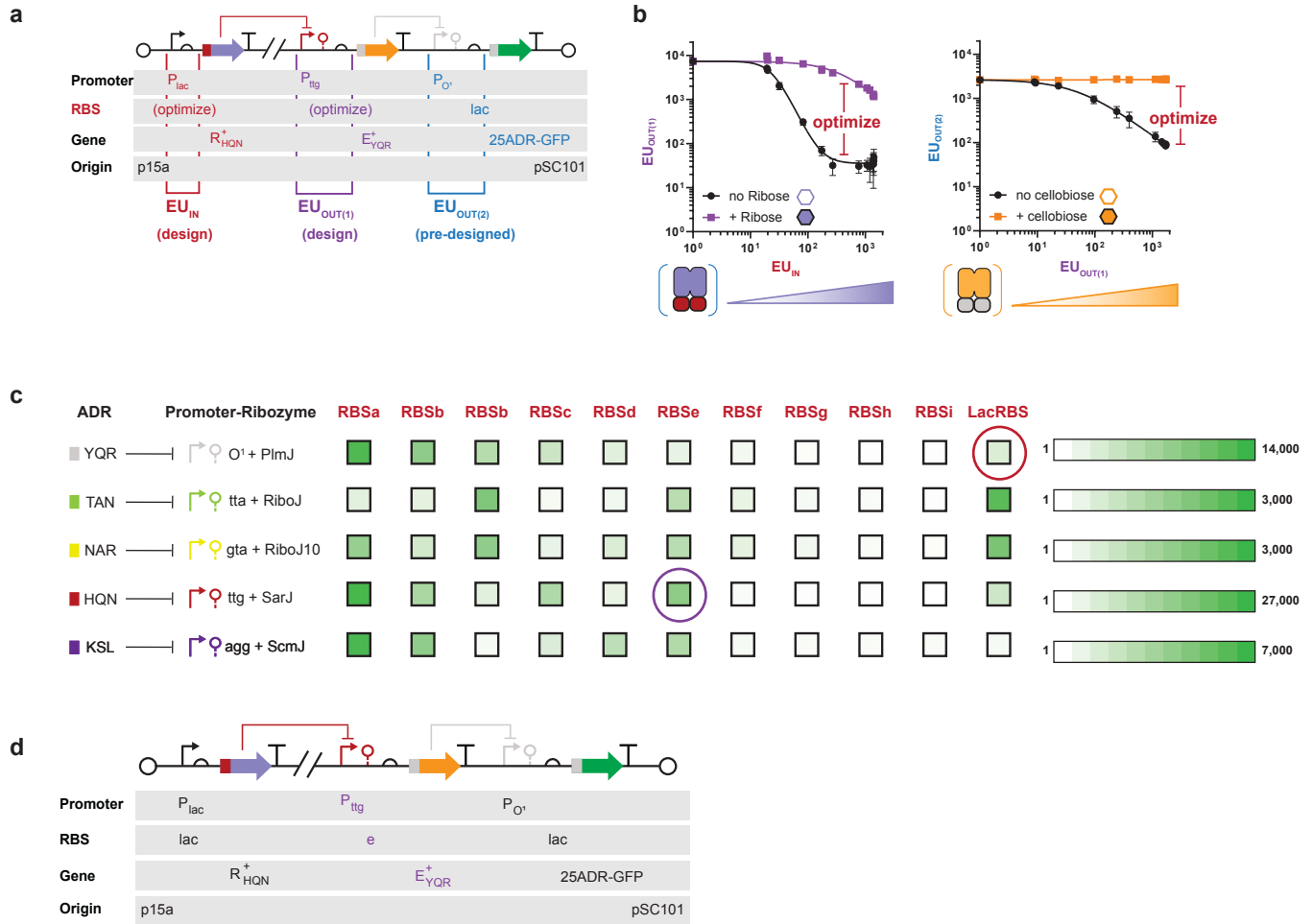
Supplementary Fig. 5 | T-Pro Quantitative Design Step 1. Granular A IMPLY B circuit composed of modular promoter, RBS, and gene elements that can be tuned for desired quantitative response (**a**). Example of an RBS look-up table cognate to synthetic promoters (**b**).

Step 2: Design Constraints - RBS Optimization & Selection (software)

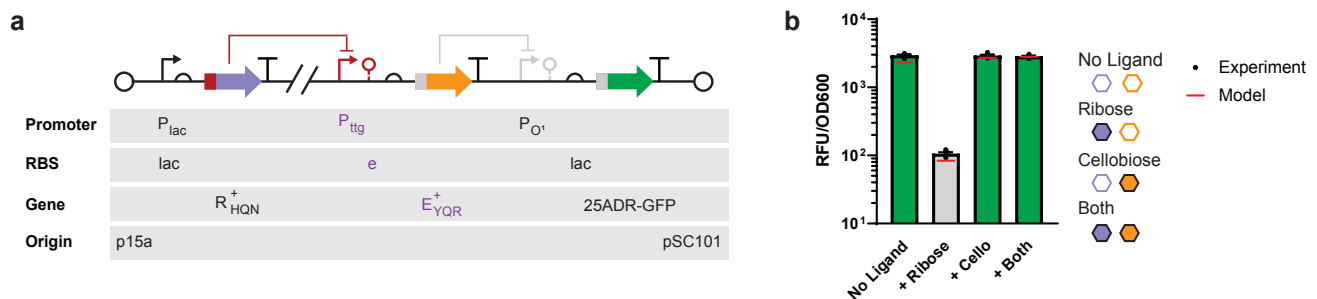
Based on the selected TFs and their ADR, ribosome binding sites (RBSs) upstream of constitutive and intermediate genes are optimized (**Supplementary Fig. 6a-b**). T-Pro CSEC software predicts the transfer function of each component (**Supplementary Fig. 6b**) and guide selection of RBSs (**Supplementary Fig. 6c**) to match the desired signal propagation. Plots of input-output behavior help identify relative transcription factor concentrations that optimize signal dynamic range and performance, ensuring each module performs predictably in its intended context (**Supplementary Fig. 6d**).

Step 3: Compressed Circuit Build & Test (user)

The final design (*i.e.*, including optimized RBSs and selected ADRs) is assembled and tested *in vivo* (**Supplementary Fig. 7a-b**). Experimental data from ligand-induction tests (*e.g.*, ribose, cellobiose, or both) are compared to model predictions (**Supplementary Fig. 7b**), validating the quantitative behavior of the constructed circuit. Successful builds exhibit logic-consistent fluorescence outputs across all input conditions.



Supplementary Fig. 6 | T-Pro Quantitative Design Step 2. Example of A IMPLY B circuit optimization. The variables open to optimization are shown for the target circuit (a), along with the transcription factor (TF) titration curves for the two chosen regulators (b). Look-up table for selection of RBSs (c) to match the desired signal propagation. The final circuit design is given in (d).

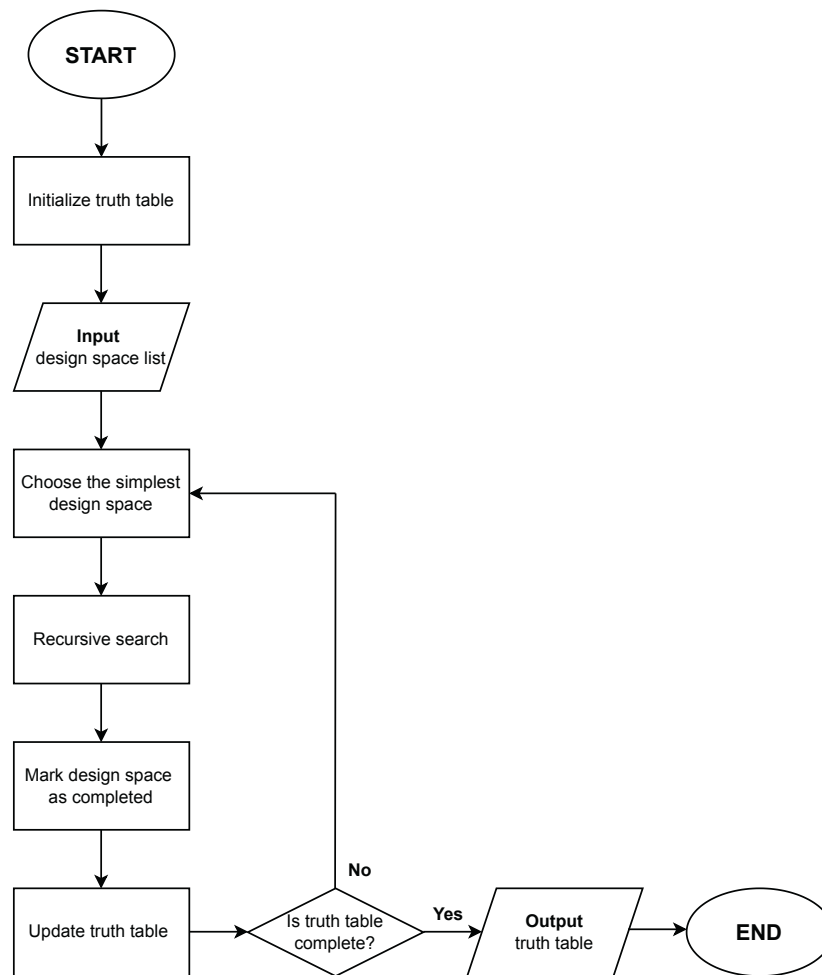


Supplementary Fig. 7 | T-Pro Quantitative Design Step 3. The final optimized genetic architecture is shown (a) along with the comparison of the predicted and experimental data (b).

T-Pro Qualitative Design (flow diagram & algorithm)

The software performs a recursive search over a user-defined design space to identify genetic circuit topologies that implement the desired logic function with minimal complexity (**Supplementary Fig. 8**). The complete workflow of the algorithmic enumeration is shown. The procedure begins by initializing the truth table, followed by inputting the design space list—each element consisting of a pair specifying the number of constitutive and regulated promoters (*e.g.*, (1,1) indicates one constitutive promoter and one regulating promoter). The algorithm selects the simplest design space from the list, and a recursive search is performed to explore all possible circuits, as described in **Algorithm 1** (**Supplementary Fig. 9**). Once the search for the selected space is completed, it is marked as completed, and the truth table is updated accordingly. This process is repeated until the truth table is fully populated.

*The algorithm was implemented in Python, and the source code is available at: <https://github.com/Jayaos/TPro>.



Supplementary Fig. 8 | Algorithmic Enumeration Workflow. The complete workflow of the algorithmic enumeration is outlined above.

Algorithm 1 Recursive Search

Input: Candidate circuit c , visited circuit set C , design space (α, β) , store dictionary D

Output: Updated store dictionary D

```
1: if  $c \in C$  then
2:   return
3: end if
4: Add  $c$  to  $C$ 
5:  $n_c \leftarrow$  number of constitutive promoters in  $c$ 
6:  $n_{ch} \leftarrow$  number of regulating promoters in  $c$ 
7: if  $n_c > \alpha$  or  $n_{ch} > \beta$  then
8:   return
9: end if
10: if  $n_c = \alpha$  and  $n_{ch} = \beta$  then
11:   if not valid circuit then
12:     return
13:   end if
14:   Evaluate  $c$  and store it in  $D$ 
15:   return
16: end if
17:  $I \leftarrow$  list of next valid candidate circuits from  $c$ 
18: for all  $c_i \in I$  do
19:   RecursiveSearch ( $c_i, C, (\alpha, \beta), D$ )
20: end for
```

Supplementary Fig. 9 | Recursive Search Algorithm. The algorithm selects the design space from the list, and a recursive search is performed in the context of all putative circuits.

Step-by-step algorithmic enumeration implementation example

1. Install dependency requirements and clone the GitHub repository.
2. Use the `search_circuit_recursive_run()` function from `utils/search_run.py` to enumerate circuits under a specified condition of constitutive and regulating promoters.

The function takes the following arguments in order:

- a. Number of constitutive promoters
- b. Number of regulating promoters
- c. Directory to save final results
- d. Directory to save intermediate (partial) results

If a directory is provided for `partial_save`, the function will automatically save cumulative results each time it discovers a new unique circuit for an input logic string.

For example, the code snippet given in **Supplementary Fig. 10** runs the enumeration under the condition of **1 constitutive promoter** and **0 regulating promoters**. The output shows that circuits were identified for **7 unique input logic strings**.

```
search_circuit_recursive_run(1, 0, "/content/Tpro/examples/results/", partial_save=None)

recursive search completed
100%|██████████| 7/7 [00:00<00:00, 23100.02it/s]
```

Supplementary Fig. 10 | Code snippet to run enumeration. The code excerpt runs the enumeration of 1 constitutive promoter and 0 regulating promoters.

- The search results are saved in the specified directory as a dictionary in the format: **{input logic string: compressed sparse row (CSR) matrix representation of the circuit}**. In **Supplementary Fig. 11**, the saved results show the circuits corresponding to the 7 unique input logic strings identified in the previous step.

```
results_1c0ch = load_data("/content/Tpro/examples/results/1c0ch")
results_1c0ch.items()

dict_items([('00000000', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>), ('11101000', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>), ('00010111', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>), ('11010100', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>), ('00101011', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>), ('10110010', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>), ('01001101', <Compressed Sparse Row sparse matrix of dtype 'int64'
  with 1 stored elements and shape (45, 5)>)])
```

Supplementary Fig. 11 | Circuits corresponding to unique input logic strings. Example of results for circuits corresponding to the 7 unique input logic strings identified in Supplementary Fig. 10.

- The CSR matrix representation can be converted to a human-readable circuit format using the `csr_to_circuit_string()` function from `utils/utils.py`. In **Supplementary Fig. 12**, the circuit corresponding to the input logic string '00000000' was converted to the representation '<C_5^S, <O>>' using this function.

```
csr_to_circuit_string(results_1c0ch["00000000"])

('00000000', '<C_5^S, <O>>')
```

Supplementary Fig. 12 | CSR matrix representation converted to a human-readable circuit format. Example of the circuit input logic string '00000000' converted to a human-readable circuit '<C_5^S, <O>>'.

- To find the optimized circuits—*i.e.*, those that use the fewest number of parts—for each input logic string, you can merge the enumeration results from multiple conditions using the `merge_circuit_search_results()` function in `utils/utils.py`. (also see **Supplementary Fig. 13**)

This function takes the input directory containing the saved search results and cognate outputs:

- The optimized circuit found for each input logic string
- The context (*i.e.*, promoter configuration) under which the smallest circuit was identified

```
merged_circuits, merged_circuit_desc = merge_circuit_search_results("/content/Tpro/examples/results/")

merging 1c0ch
100%|██████████| 7/7 [00:00<00:00, 8704.46it/s]
merging 1c1ch
100%|██████████| 20/20 [00:00<00:00, 4904.18it/s]
merging 2c0ch
100%|██████████| 32/32 [00:00<00:00, 24609.04it/s]
```

Supplementary Fig. 13 | Identification of the optimized circuit. This executable takes the input directory containing the saved search results and cognate outputs

6. Use the `merged_dict_to_table()` function from `utils/utils.py` to convert the merged search results into a summary table.

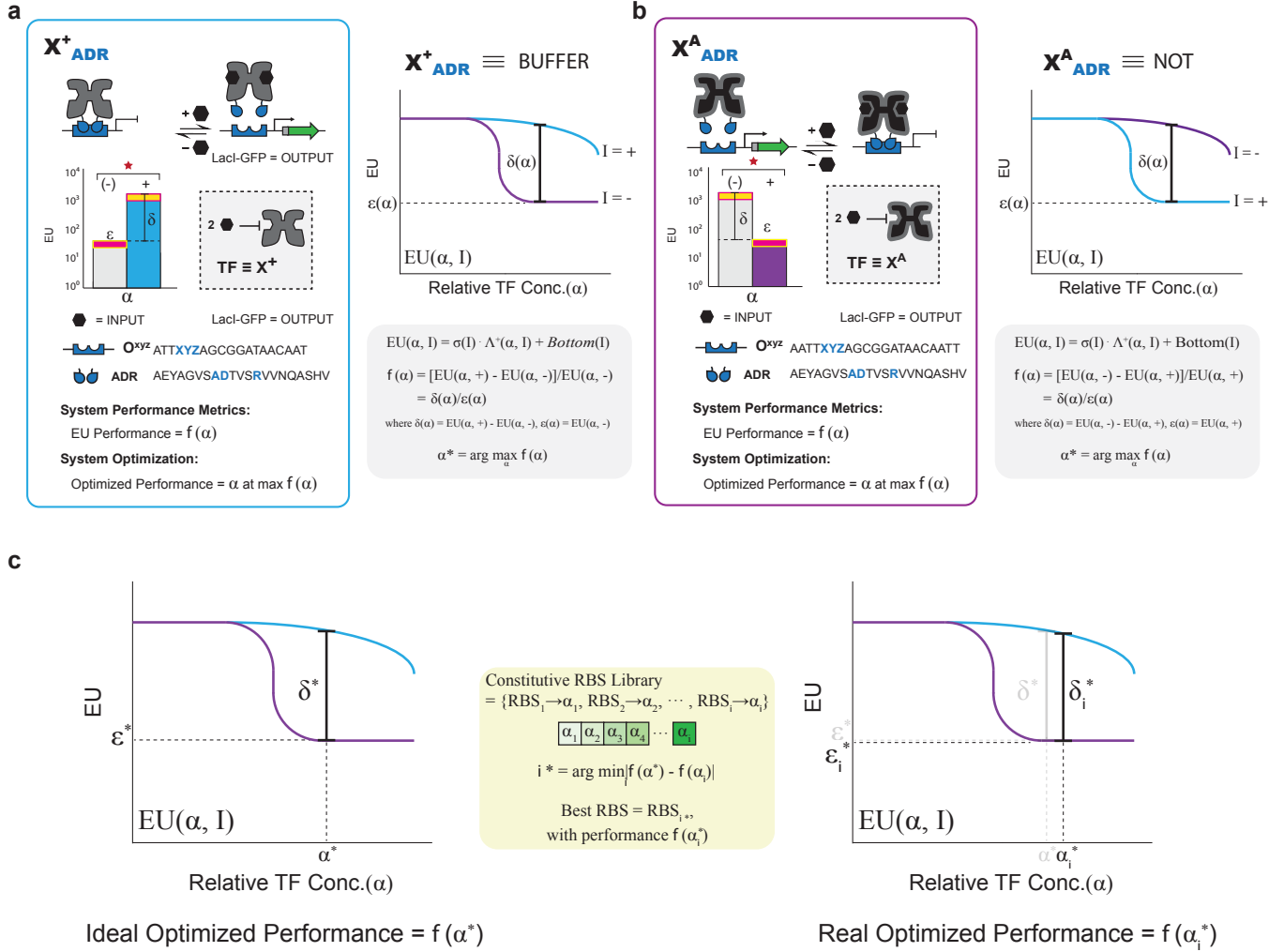
In the resulting table:

- a. The **"logic_string"** column indicates the input logic string
- b. The **"circuit_string"** column shows the smallest circuit identified
- c. The **"circuit_desc"** column specifies the condition under which the circuit was found

T-Pro Quantitative Design Models & Software

Part 1: Modeling & Calculating SISO Dynamic Range

Supplementary Fig. 14 illustrates how the dynamic performance of individual transcription factors is modeled and optimized at the single-input single-output (SISO) level. Each module is defined by an input-output transfer function, which describes the expression units (EU) of a reporter gene as a function of transcription factor (TF) concentration.

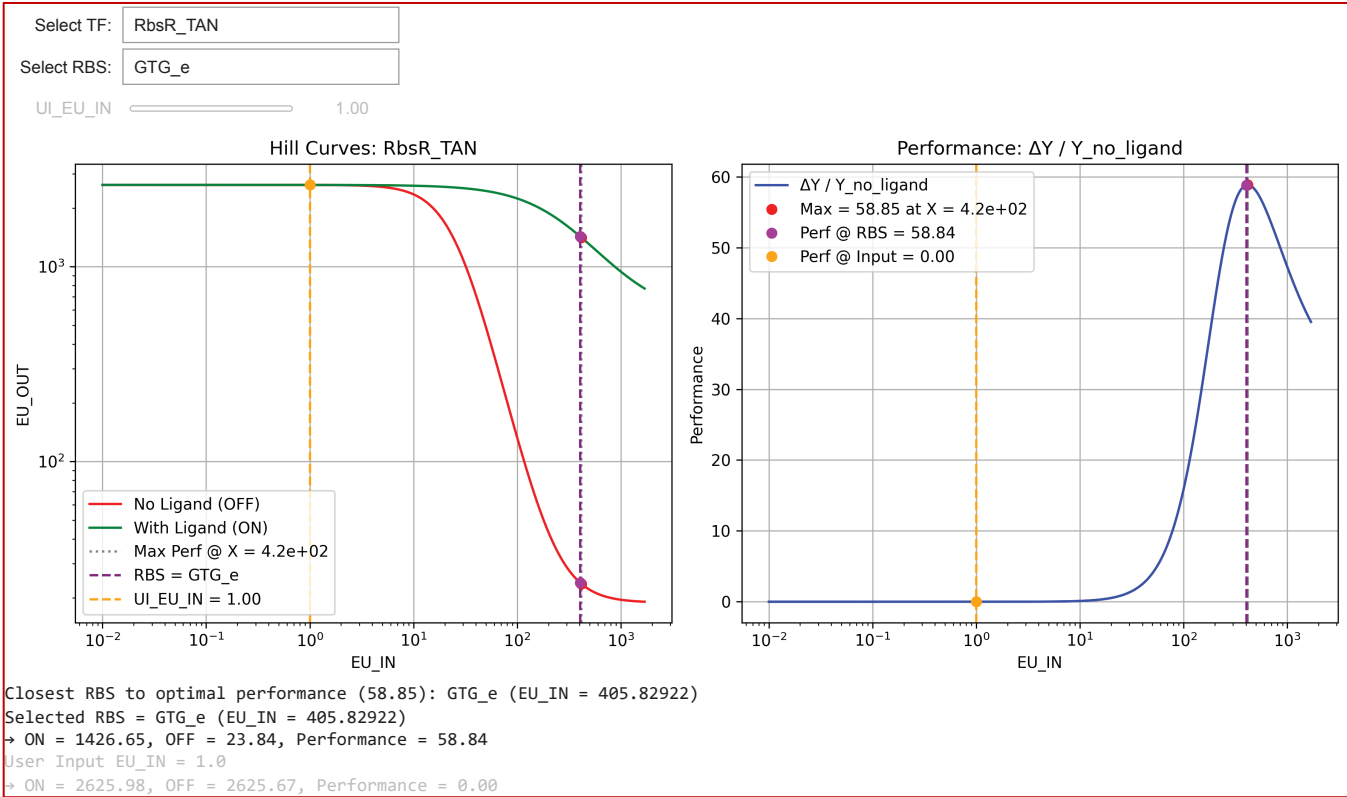


Supplementary Fig. 14 | Single-Input Single-Output (SISO) performance optimization. (a) BUFFER Gate Behavior: A repressor-type transcription factor is used to model a BUFFER logic operation. For each TF, two Hill curves—corresponding to the liganded and unliganded states—are generated. The system calculates a performance metric $f(\alpha)$, defined as the dynamic range normalized to the OFF state. This prioritizes achieving low leakage in the OFF state while maximizing signal separation. The optimal TF concentration (α^*) is identified as the point that yields this maximum normalized dynamic range. **(b) NOT Gate Behavior.** An anti-repressor-type transcription factor is used to model a NOT logic function. As in (a), the performance is evaluated using the normalized dynamic range to determine α^* . **(c) Ideal vs. Real Performance.** The system compares ideal RBS scaling (left) with the closest match from an experimental RBS library (right), selecting the RBS variant (RBS_{i^*}) that best approximates the ideal behavior using performance deviation minimization.

Interactive Hill Modeling Software

For each transcription factor, the software generates Hill curves for both the ON and OFF states, calculates the associated performance metric, and recommends the optimal RBS variant. Users may alternatively select from a predefined list of RBS options or manually input a desired EU_IN value, which is used in downstream multi-input-single-output (MISO) optimizations. The software then computes and displays the resulting output expression level, screenshot given in **Supplementary Fig. 15**.

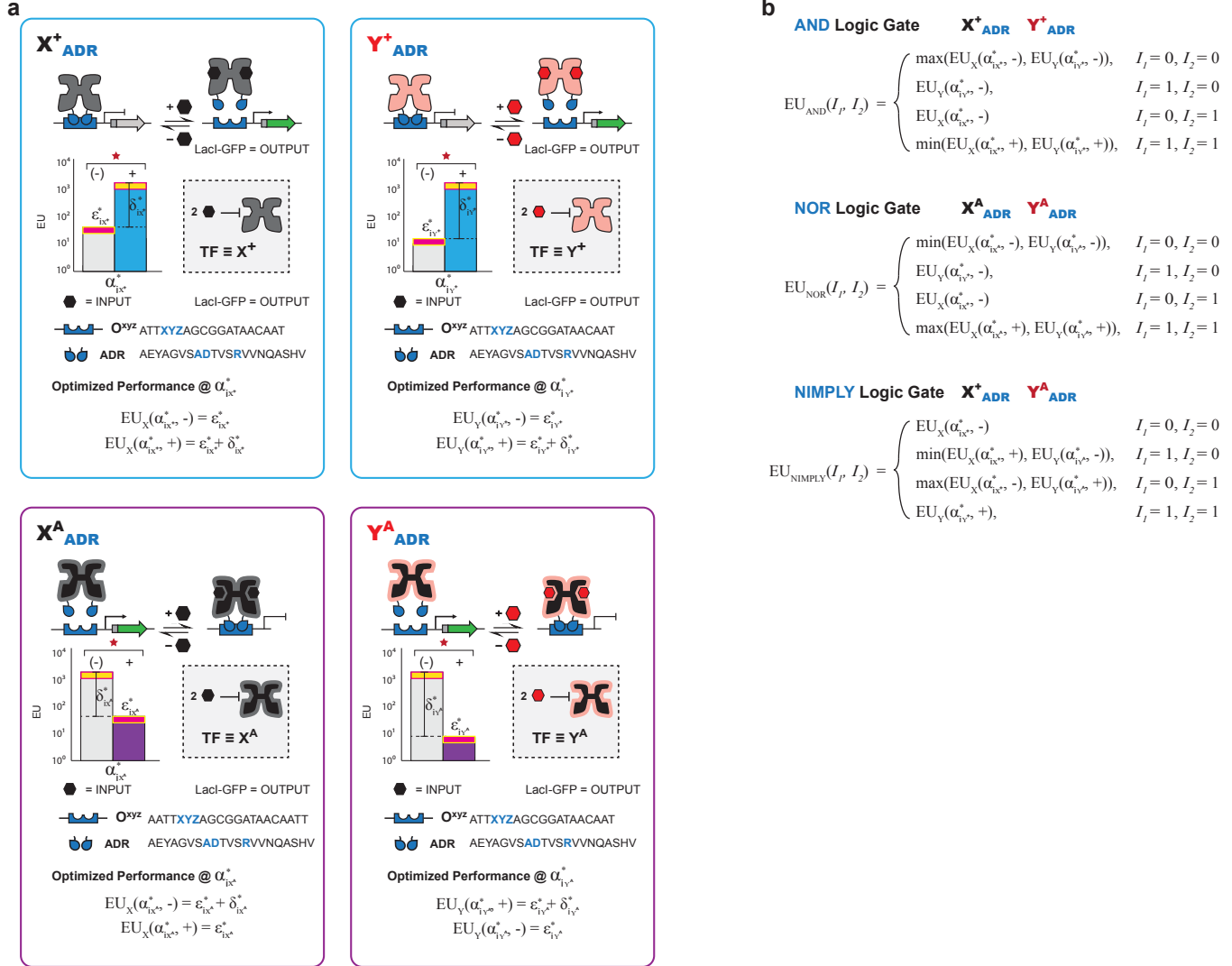
*source code is available at: <https://github.com/Jayaos/TPro>



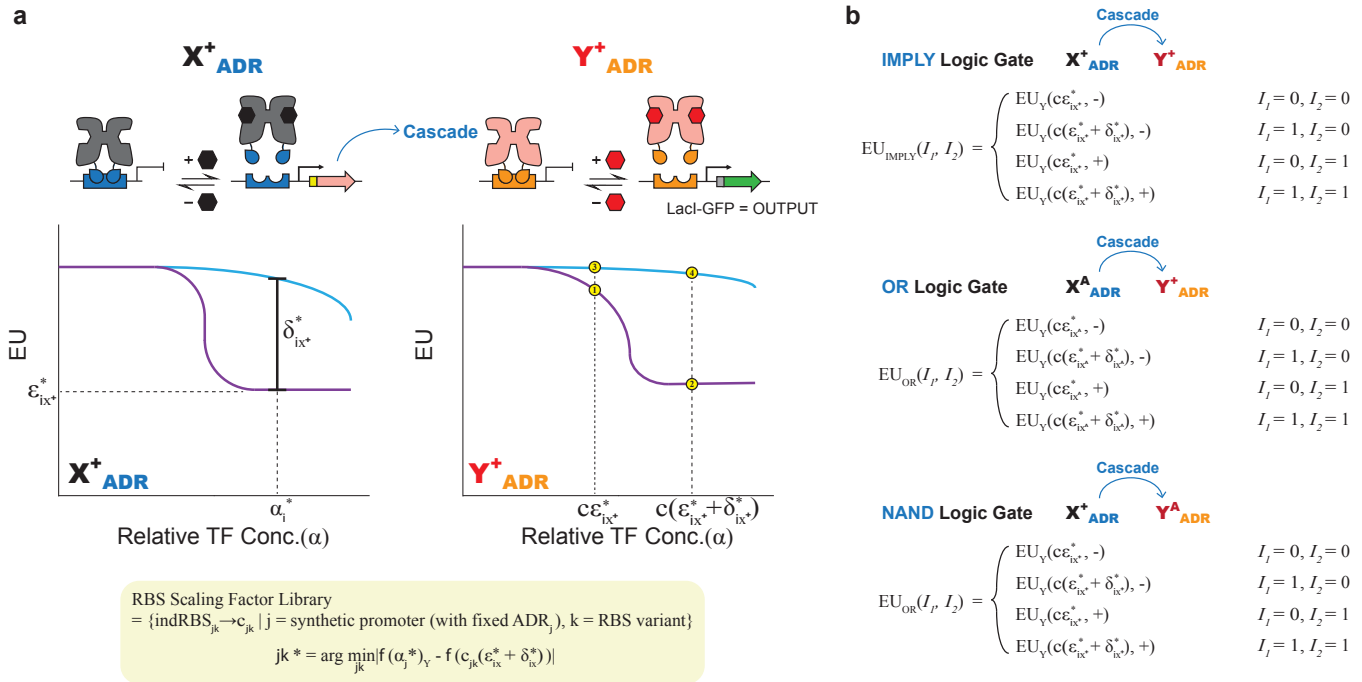
Supplementary Fig. 15 | Interactive Hill Modeling Software. Screenshot of the software generating Hill curves for both the ON and OFF states, calculating the associated performance metric, and recommendation for the optimal RBS variant.

Part 2: Extension from SISO to MISO Logic

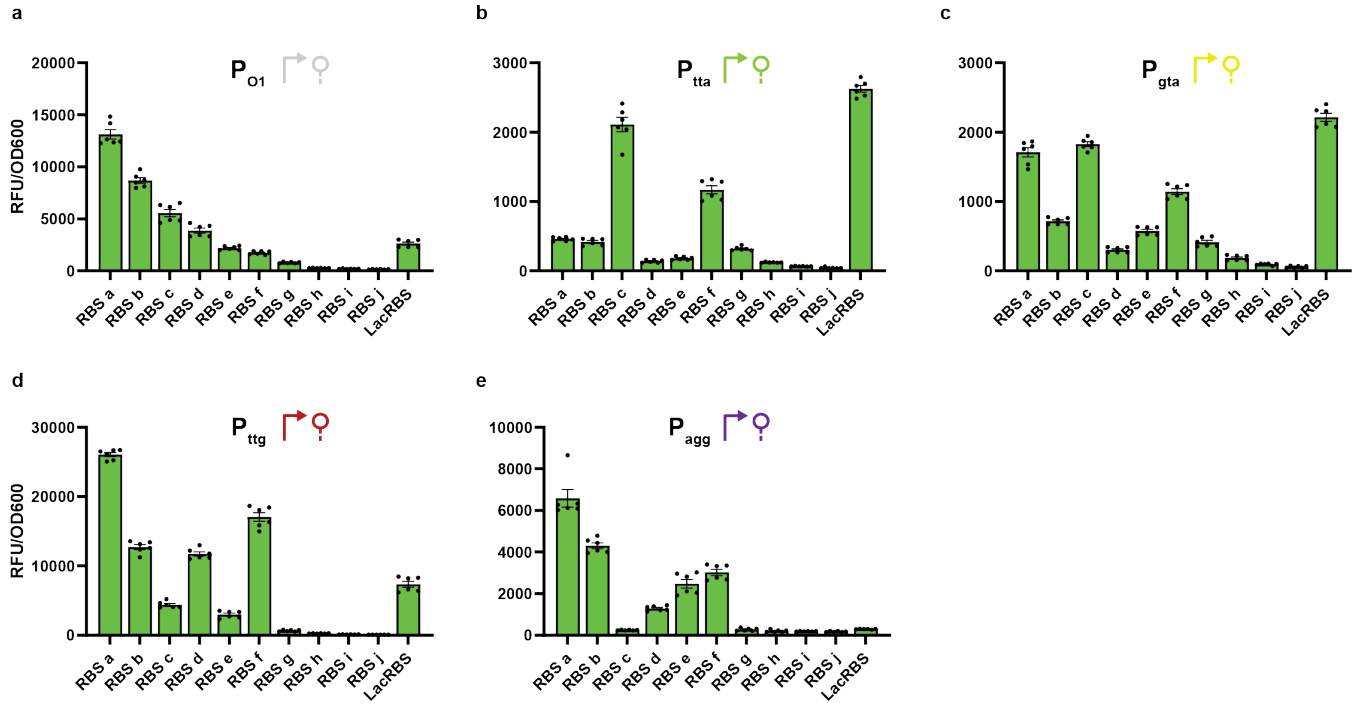
Supplementary Figs. 16-17 illustrates how to build functional 2-input logic gates, independently optimized SISO models are composed into two-input circuits (X and Y).



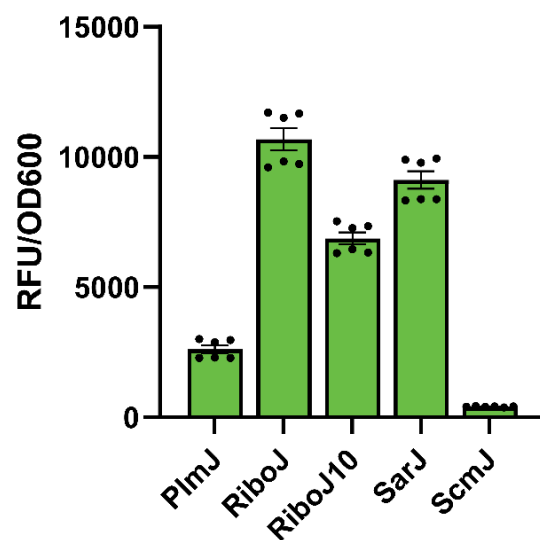
Supplementary Fig. 16 | SISO to MISO Logic. (a) Performance Metrics for Dual Inputs. Each TF–ADR module (X or Y) is individually tuned, and its optimized ON/OFF responses are used to calculate combined output. (b) Logical Operations (AND, NOR, NIMPLY), said logic operations are constructed by pairing two transcription factors that share the same alternate DNA recognition (ADR) unit and bind to a common operator site upstream of the output gene. The overall expression output (EU) is computed by combining ON and OFF states at their respective optimized TF concentrations. Additional details for the fundamental models are reported in our earlier work—Milner *et al.*⁶



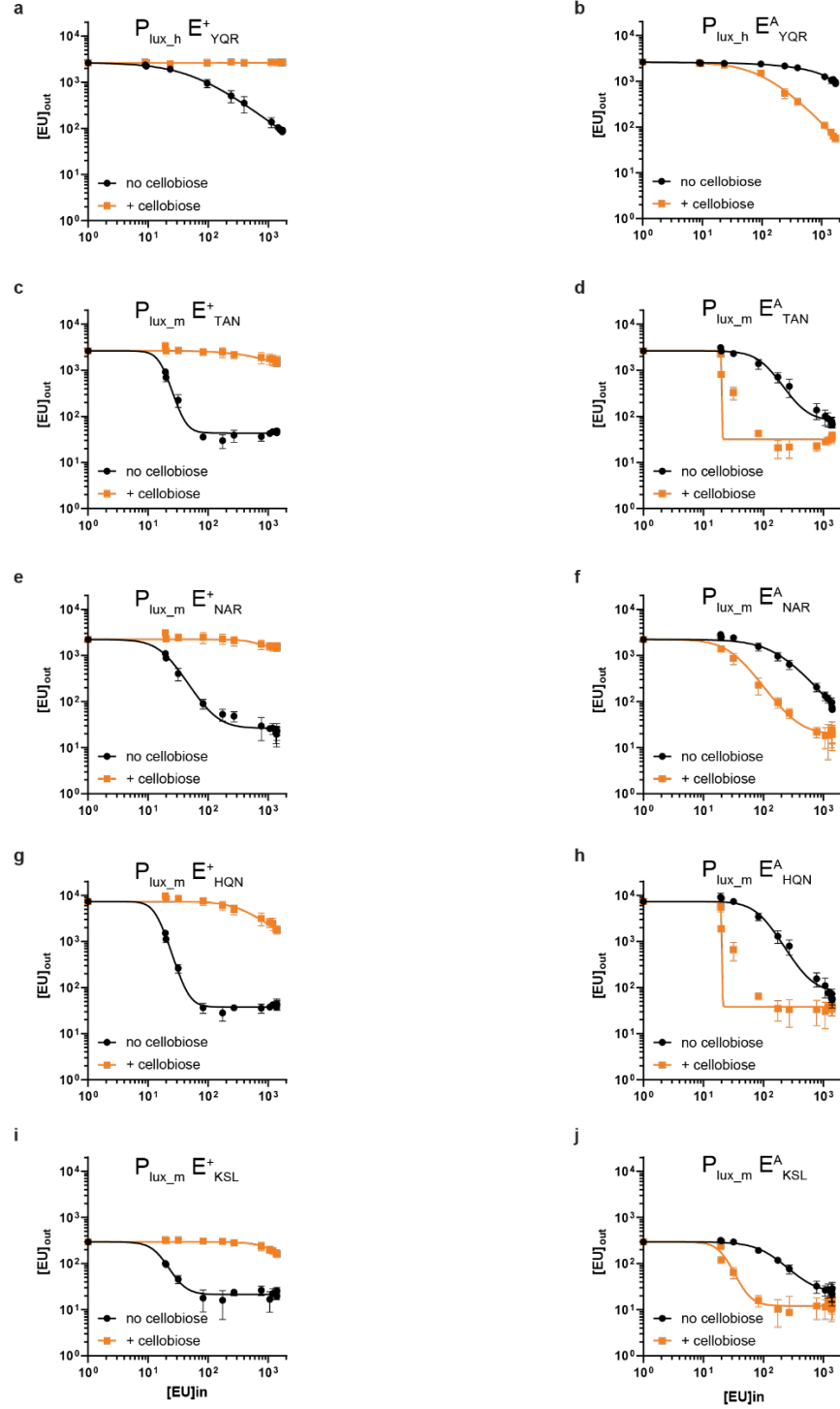
Supplementary Fig. 17 | Cascading TFs with RBS Scaling and MISO Modeling. Feedforward cascade designs involve chaining transcription factors such that the output of an upstream TF module serves as the input for a downstream module **(a)**. This is achieved by applying RBS scaling factors to the regulated CSEC of the downstream TF. The optimization focuses on maximizing performance in the high-expression (high- α) of the downstream TF, to ensure highly distinct ON and OFF states while minimizing leakiness. Yellow numbered circles correspond to the four different ligand input conditions and their associated output expression levels. **(b)** Modeling cascaded MISO logic gates: IMPLY, OR, and NAND. Each gate is built by chaining two transcription factors, where the output expression of the upstream TF (X) regulates the expression level of the downstream TF (Y) through a feedforward cascade. Unlike co-localized logic gates, the two TFs in these cascaded designs have distinct ADRs and bind to separate operators. For each gate, the overall expression output (EU) is computed for the four possible input states, taking the scaled output level of TF(X) as the input for TF(Y). Users can manually input the desired EU_IN for the downstream TF and visualize the output using the interface described in Supplementary Fig. 15, allowing flexible modeling of signal propagation through the cascade.



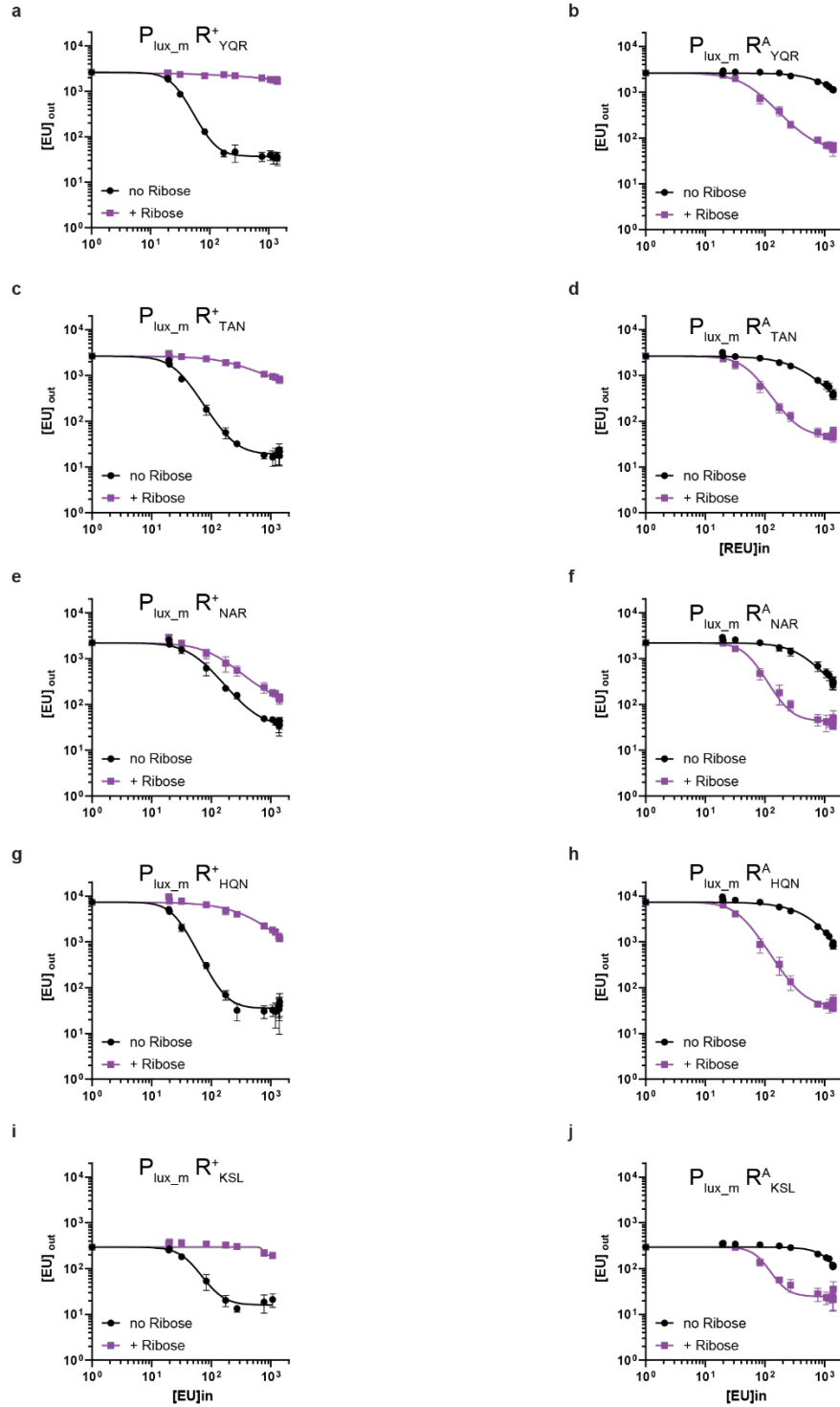
Supplementary Fig. 18 | Non-modularity of RBS libraries applied to different synthetic promoters. **a** The RBS library originally characterized with the O^1 promoter and the 25ADR-GFP fusion reporter was applied to the other 4 promoters used in this study: **(b)** *tta*, **(c)** *gta*, **(d)** *ttg*, **(e)** *agg*. Synthetic promoter names are at the top of each graph. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



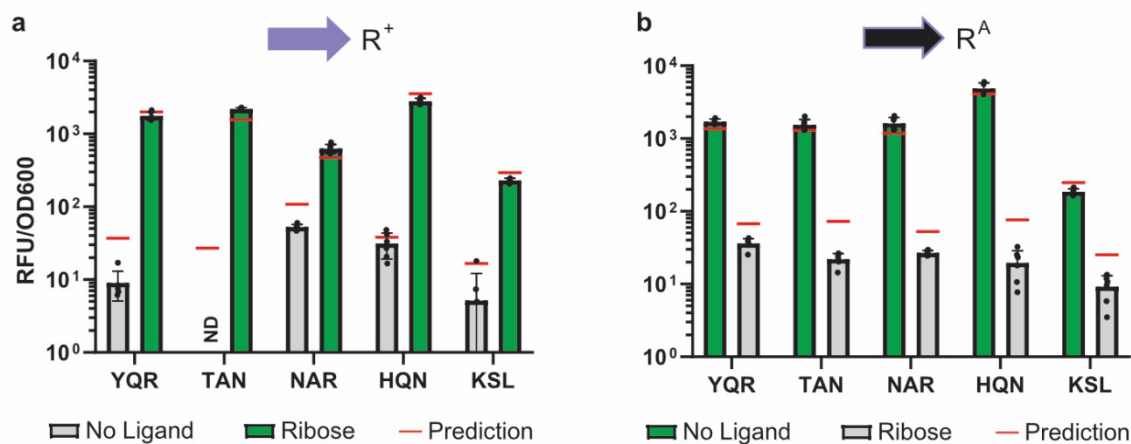
Supplementary Fig. 19 | Non-modularity of ribozymes. All 5 ribozymes used in this study were applied in the context of the O^1 promoter, lacRBS, and the 25ADR-GFP fusion reporter. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



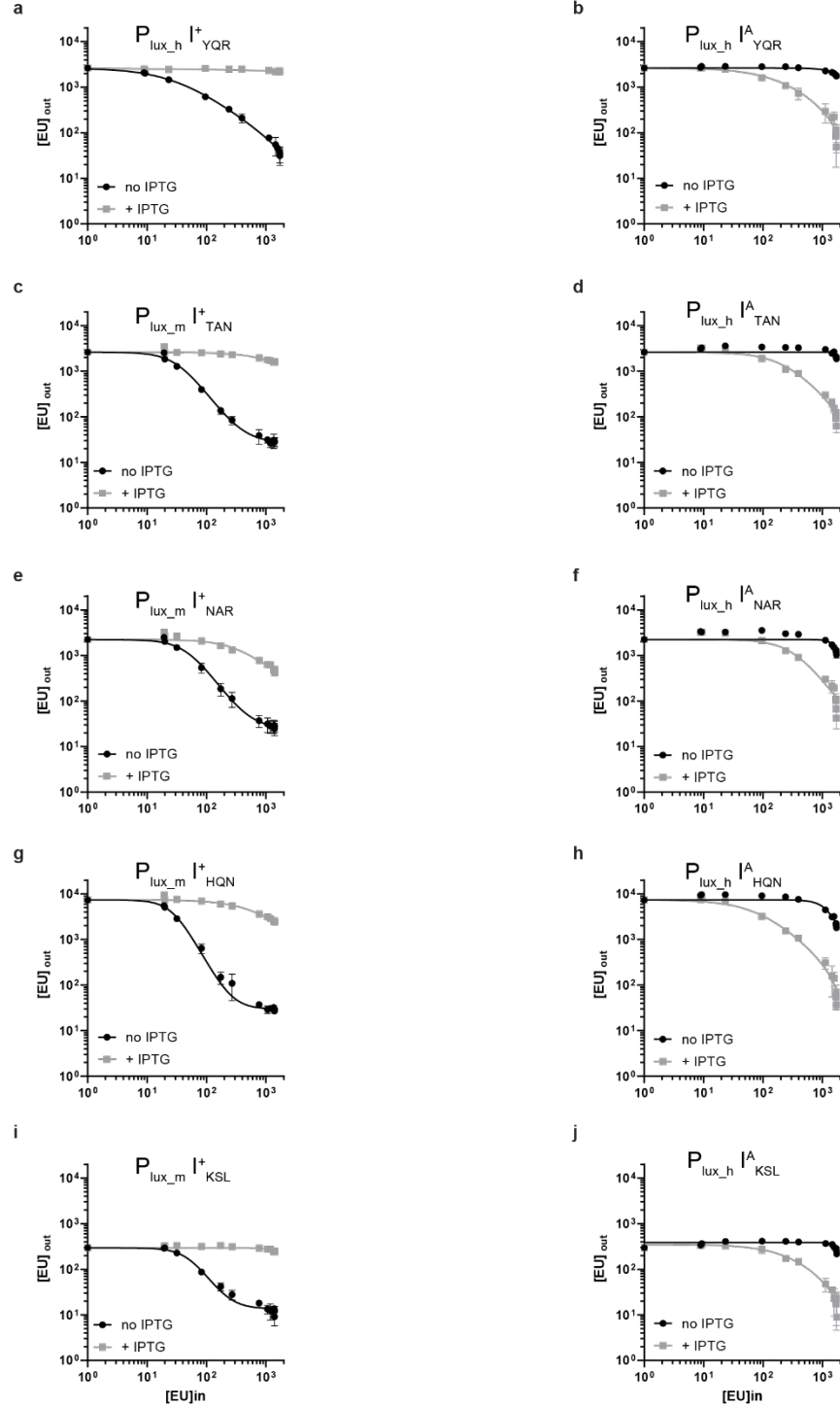
Supplementary Fig. 20 | CelR and anti-CelR titration curves. The circuit described in **Fig. 4a-b** was constructed and tested for all 5 ADR variants of CelR (left) and anti-CelR (right) regulating cognate promoters. From top to bottom row of varying ADRs: **(a, b)** YQR, **(c, d)** TAN, **(e, f)** NAR, **(g, h)** HQN, **(i, j)** KSL. Note that some TFs are expressed from a medium-strength promoter and some are expressed from a higher-strength promoter. The x-axis input Expression Units ($[EU]_{in}$; see Methods for definition of EU) is based on the corresponding standard curve for the appropriate P_{lux} promoters. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



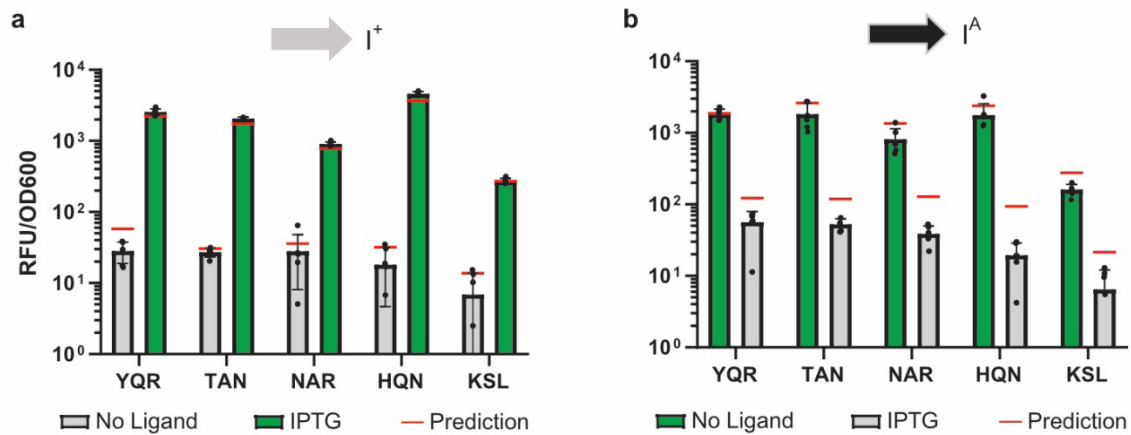
Supplementary Fig. 21 | RbsR and anti-RbsR titration curves. The circuit described in **Fig. 4a-b** was constructed and tested for all 5 ADR variants of RbsR (left) and anti-RbsR (right) regulating cognate promoters. From top to bottom row of varying ADRs: **(a, b)** YQR, **(c, d)** TAN, **(e, f)** NAR, **(g, h)** HQN, **(i, j)** KSL. Here, all TFs are expressed from a medium-strength promoter. The x-axis input Expression Units ($[EU]_{in}$; see Methods for definition of EU) is based on the corresponding standard curve for the appropriate P_{lux} promoter. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



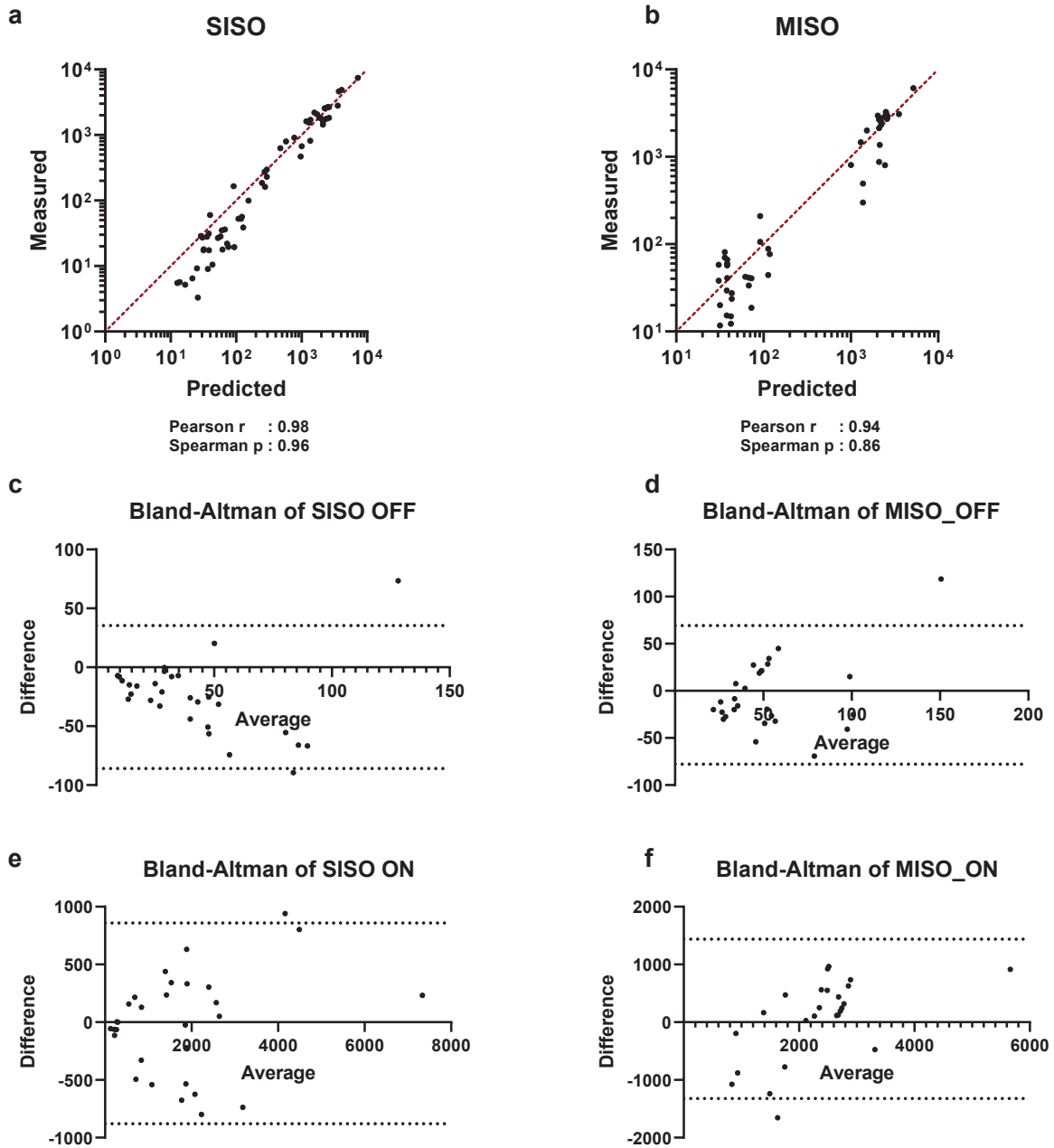
Supplementary Fig. 22 | SISO predictions for RbsR-based circuits. **a** The predicted and experimental comparisons are shown for five R^+_{ADR} BUFFER circuits. **b** The predicted and experimental comparisons are shown for five R^A_{ADR} NOT circuits. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



Supplementary Fig. 23 | LacI and anti-LacI titration curves. The circuit described in Fig. 4a-b was constructed and tested for all 5 ADR variants of LacI (left) and anti-LacI (right) regulating cognate promoters. From top to bottom row of varying ADRs: (a, b) YQR, (c, d) TAN, (e, f) NAR, (g, h) HQN, (i, j) KSL. Note that some TFs are expressed from a medium-strength promoter and some are expressed from a higher-strength promoter. The x-axis input Expression Units ([EU]_{in}; see Methods for definition of EU) is based on the corresponding standard curve for the appropriate P_{lux} promoters. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



Supplementary Fig. 24 | SISO predictions for LacI-based circuits. **a** The predicted and experimental comparisons are shown for five I^+_{ADR} BUFFER circuits. **b** The predicted and experimental comparisons are shown for five I^A_{ADR} NOT circuits. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



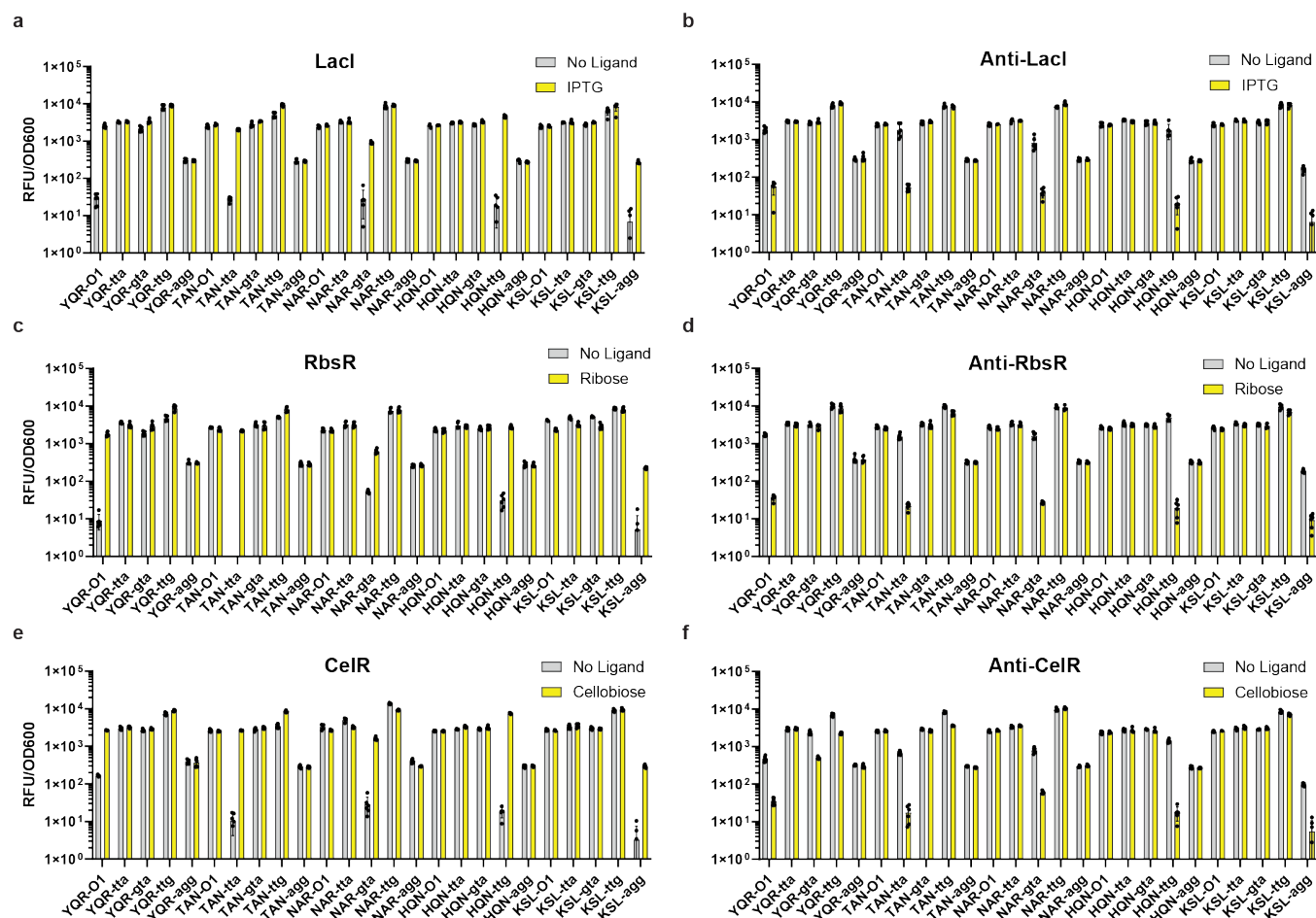
Supplementary Fig. 25 | SISO and MISO statistical correlation analysis. **a** Predicted versus measured Expression Unit (EU) values for SISO (Single-Input Single-Output) circuits show strong agreement with a Pearson correlation coefficient of 0.98 and Spearman correlation of 0.96. The red dashed line represents the line of identity ($x = y$). **b** Predicted versus measured EU values for MISO (Multiple-Input Single-Output) circuits also show strong correlation, with a Pearson coefficient of 0.94 and Spearman coefficient of 0.86. **c** Bland–Altman analysis of SISO circuits in the OFF state shows that most predicted EU values fall within acceptable agreement with measured data, with a slight bias toward overprediction. Variability increases at higher OFF-state expression levels, likely due to greater biological noise as expression approaches the ON-state threshold. **d** MISO OFF-state predictions show tight agreement, with a single outlier above the upper limit of agreement. The remaining data are symmetrically distributed, indicating no systematic bias. **e** Bland–Altman analysis of SISO circuits in the ON state shows increased

variability compared to the OFF state, though most values remain within the 95% limits of agreement. **f** Bland–Altman analysis of MISO circuits in the ON state shows broader variation, particularly at higher expression levels, but the majority of predictions remain within limits of agreement. Dotted lines in all Bland–Altman plots represent the mean difference and 95% limits of agreement. Error analysis and Source data are provided as a Source Data file.

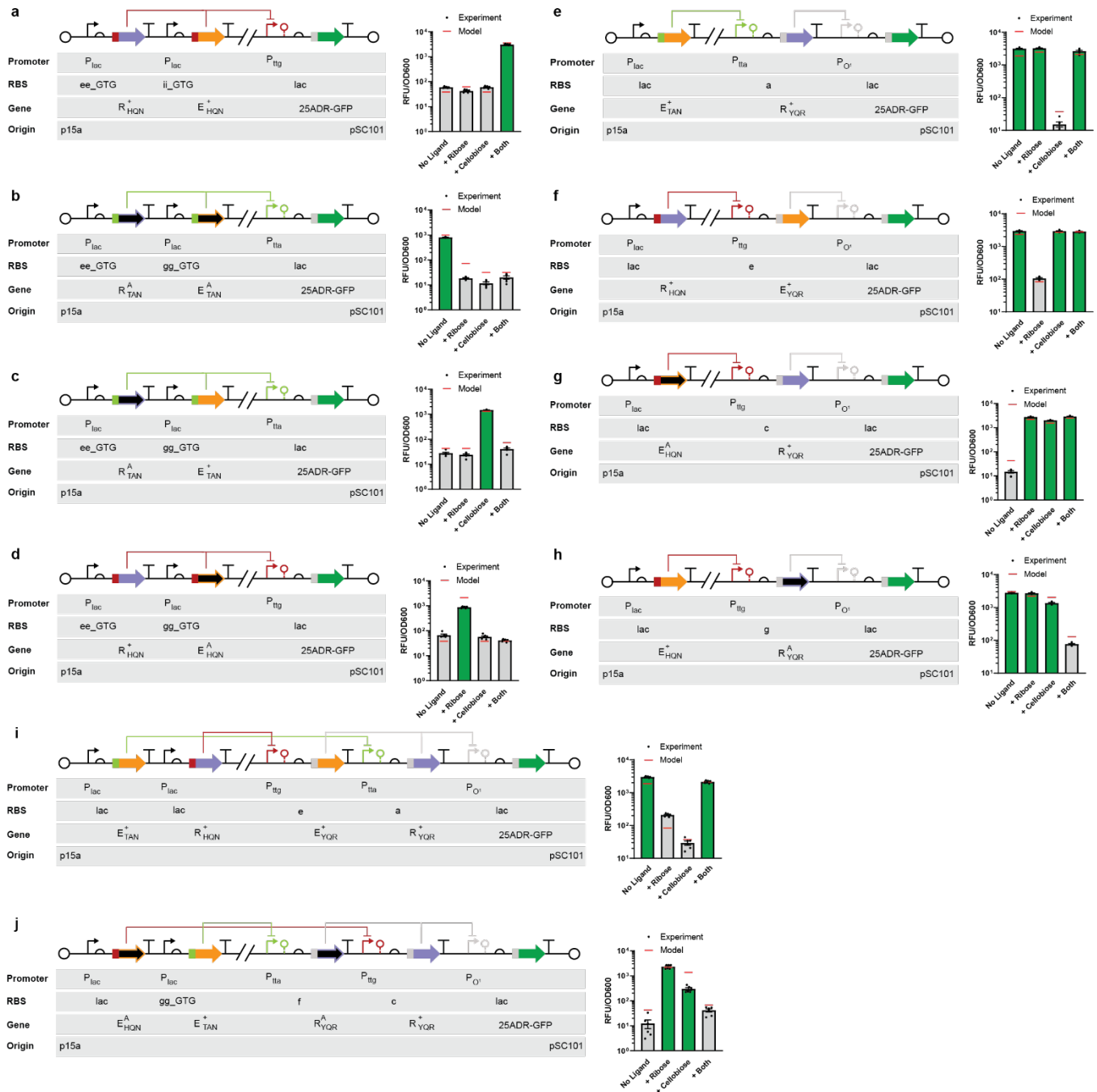
NOTES: To assess the accuracy of T-Pro circuit predictions, we compared experimental outputs to model predictions for all 30 single-input single-output (SISO) gates constructed using the 30 transcription factors in our study, as well as 12 multi-input single-output (MISO) gates, including both 2-input and 3-input logic functions. For each circuit, we calculated the percentage difference between the predicted and observed expression levels across all input states (see **Source Data** file).

In all cases tested, the deviation between predicted and measured output levels did not exceed 2.3-fold, with an average deviation of approximately 1.4-fold across all circuits, demonstrating consistent predictive accuracy across circuit architectures and logic functions.

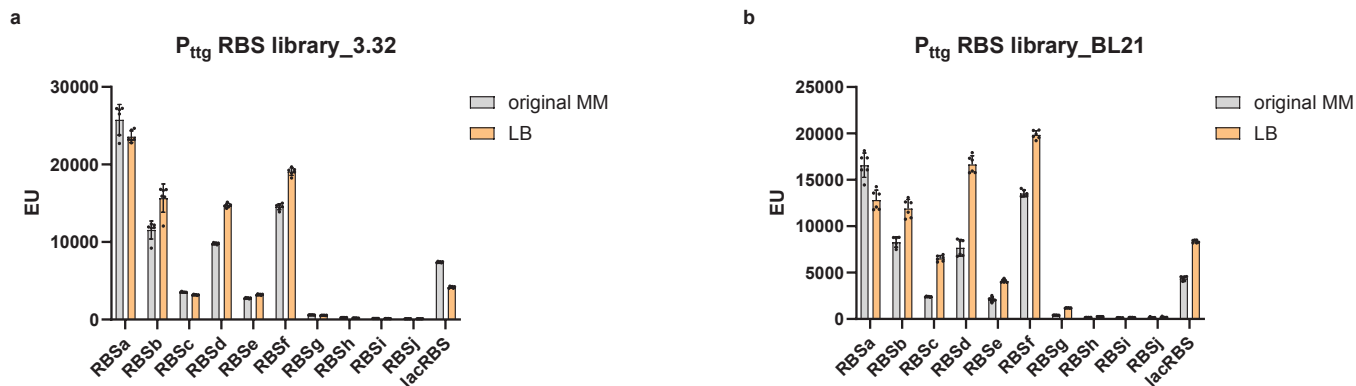
After constructing and testing the 3-input program given in **Fig. 5e**, no state was greater than ~1.5-fold off from the prediction, which is substantially more accurate than the Cello design which often exhibited >10-fold deviations from the prediction. For this comparison, we performed a rough visual estimation of the predictive performance reported in Cello, as raw numerical data were not available. Based on inspection of published figures (*e.g.*, fluorescence output distributions), many output states in Cello circuits appeared to deviate by more than 10-fold from predicted values. Additionally, they report circuits with one to three failed output states, indicating lower predictive accuracy.



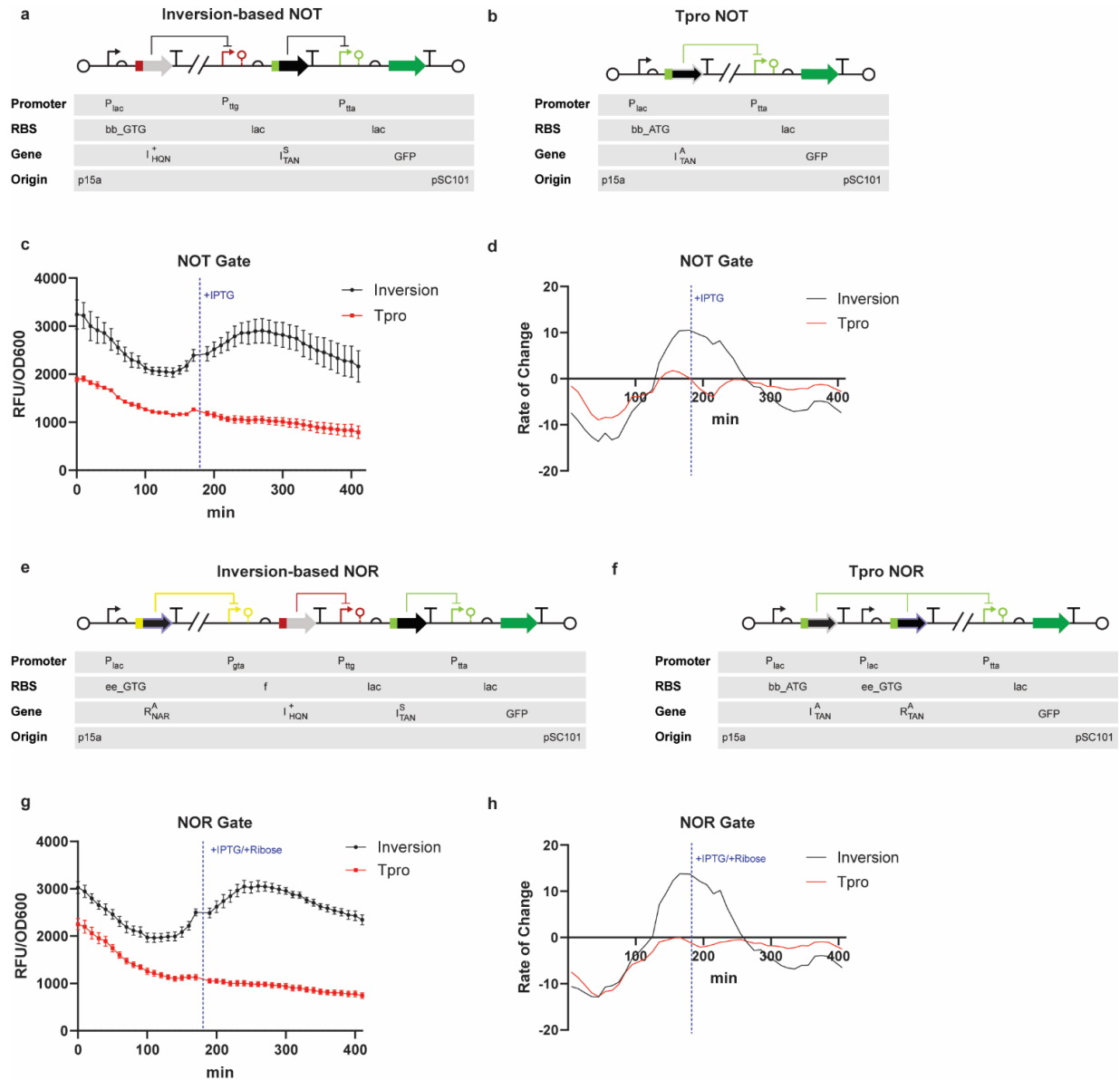
Supplementary Fig. 26 | Synthetic promoter orthogonality. The TFs used in this study were assayed for potential regulation of each of the 5 promoters. Each pair of bars corresponds to a single TF with the labeled ADR regulating a promoter with the 25ADR-GFP fusion reporter. From top to bottom rows: **(a)** LacI, **(b)** Anti-LacI, **(c)** RbsR, **(d)** Anti-RbsR, **(e)** CelR, **(f)** Anti-CelR. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



Supplementary Fig. 27 | Additional 2-input circuits. The workflow described in **Supplementary Figs. 2-17** was executed for all 2-input circuits with D-ribose and cellobiose as the inputs. Circuits are as follows: AND (**a**), NOR (**b**), E NIMPLY R (**c**), R NIMPLY E (**d**), E IMPLY R (**e**), R IMPLY E (**f**), OR (**g**), NAND (**h**), XNOR (**i**), XOR (**j**). The circuit diagram is shown on the left with measured performance on the right. Red bars indicate the model predictions for each state. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.



Supplementary Fig. 28 | RBS context across variants and media conditions. Reporter expression levels from the P_{ttg} promoter driving the 25ADR-GFP across an RBS library in two chassis strains: *E. coli* 3.32 (**a**) and BL21 (**b**). Expression Unit (EU) measurements were taken in minimal medium (MM; gray) and LB (orange) to characterize context-specific differences in expression strength. Expression levels vary significantly between RBSs and between media, underscoring the need for recharacterization when applying models to new contexts. Source data are provided as a Source Data file. Data represents the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.

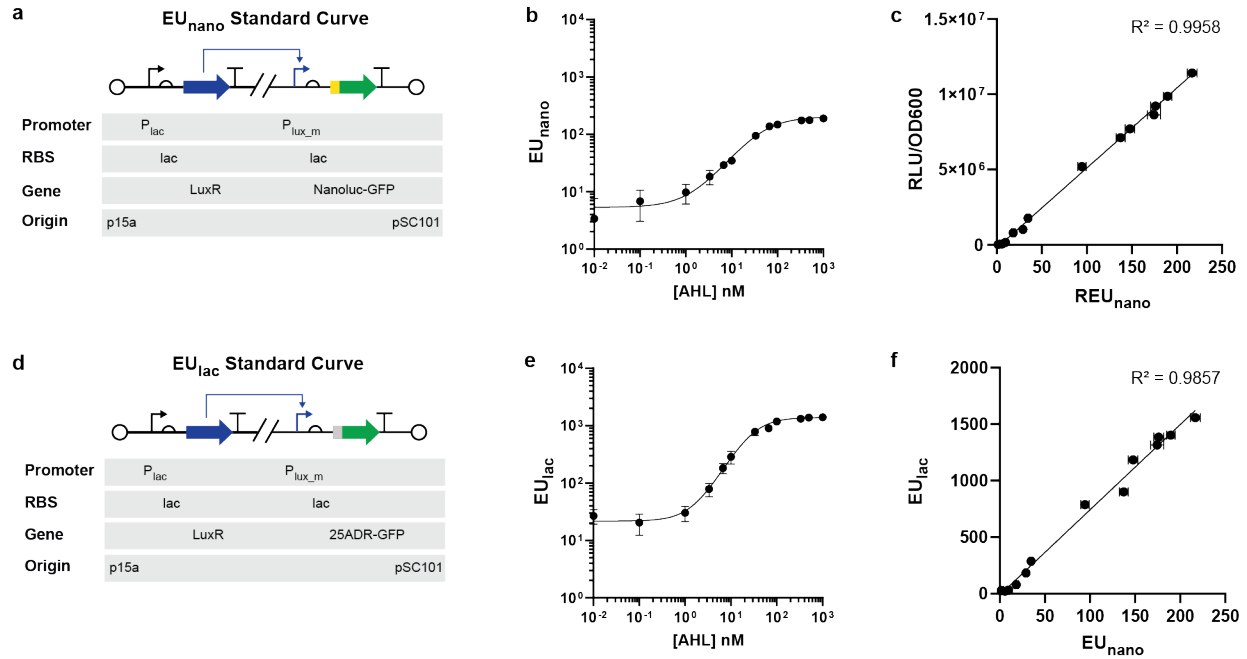


Supplementary Fig. 29 | Dynamic response comparison of T-Pro and inversion-based NOT and NOR gate. **a** The genetic schematic of a NOT gate built with inversion-based design is shown. **b** The genetic schematic of a NOT gate built with T-Pro style design is shown. **c** Plate reader time-course assay results for the circuits in **a** and **b** are shown. 10 mM IPTG was added to the cultures at the 180 minute mark depicted with a blue dotted line. **d** First derivative analysis of the graphs in **c** are shown. Four neighboring points were averaged with a 2nd order polynomial to smooth the curve. **e** The genetic schematic of a NOR gate built with a simplified inversion-based design is shown. **f** The genetic schematic of a NOR gate built with T-Pro style design is shown. **g** Plate reader time-course assay results for the circuits in **e** and **f** are shown. 10 mM IPTG was added to the cultures at the 180 minute mark depicted with a blue dotted line. **h** First derivative analysis of the graphs in **g** are shown. Four neighboring points were averaged with a 2nd order polynomial to smooth the curve. Source data are provided as a Source Data file. Data for **c** and **g** represent the average of $n = 3$ biological replicates. Error bars correspond to the SEM of these measurements. **Note, for the inverter-based NOR gate in order to avoid the use of tandem promoters or two output channels (see **Supplementary Fig. 1c-d**) we designed a partial NOR inverter circuit (**Fig. 6I**). In addition, we posited that said design will minimize differences in metabolic burden imposed on the chassis cell by a given NOR circuit facilitating a better relative comparison.

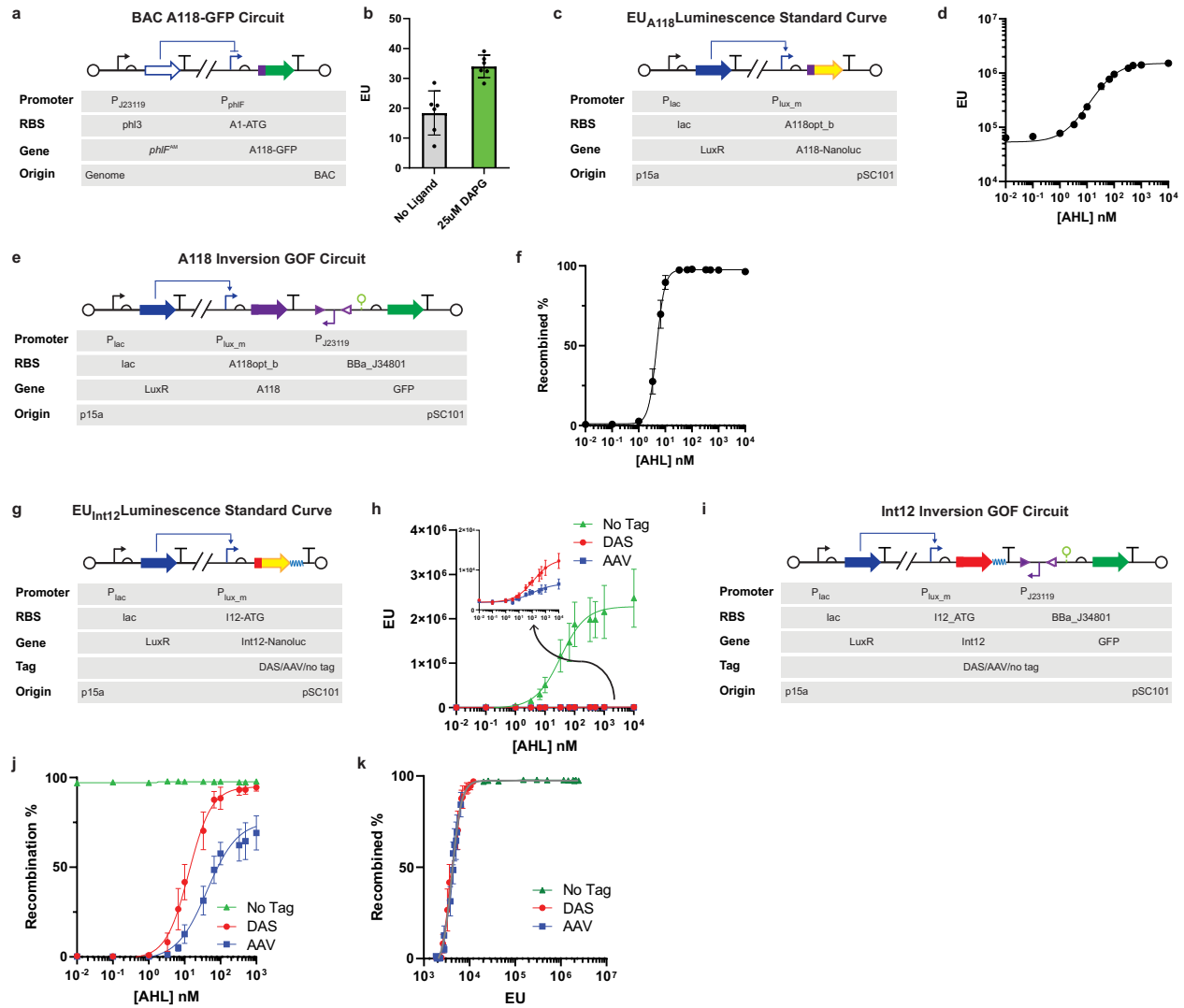
NOTE: T-Pro circuits are designed to minimize weak signal strength by leveraging transcription factors with strong dynamic ranges – *i.e.*, ON/OFF states. Since T-Pro directly regulates the output without additional inverters, gene expression levels are primarily determined by the choice of transcription factor(s) regulating the output promoter and the RBS of the output gene, rather than circuit size alone. Thus, despite having a slightly lower ON-state in the given compressed circuits this is paired with a significantly lower OFF-state – which is critical for dynamic range performance, and most applications. Moreover, we have the ability to engineer circuits with low OFF-states paired with higher ON-states (data not shown).

Compressed circuits exhibit faster response times. Cello-type circuits that rely on multi-layered inverters require signal propagation through multiple transcriptional and translational steps, which introduces additional delays. In contrast, T-Pro circuits bypass these extra layers, directly inducing or repressing outputs based on ligand-responsive transcription factors (*i.e.*, *via* maximizing post-translation events). By reducing the number of regulatory steps, T-Pro circuits achieve faster transitions between states.

In terms of expression noise, redundancy in circuit design can potentially buffer stochastic fluctuations; however, larger circuits also introduce more opportunities for cumulative variability. Each additional transcription factor in a circuit can contribute to variations in expression due to differences in protein stability, binding efficiency, or metabolic burden. By minimizing the number of regulatory elements required for logic execution, T-Pro circuits typically reduce overall noise accumulation – as shown. However, the impact of compression on noise levels can vary depending on specific circuit configurations.



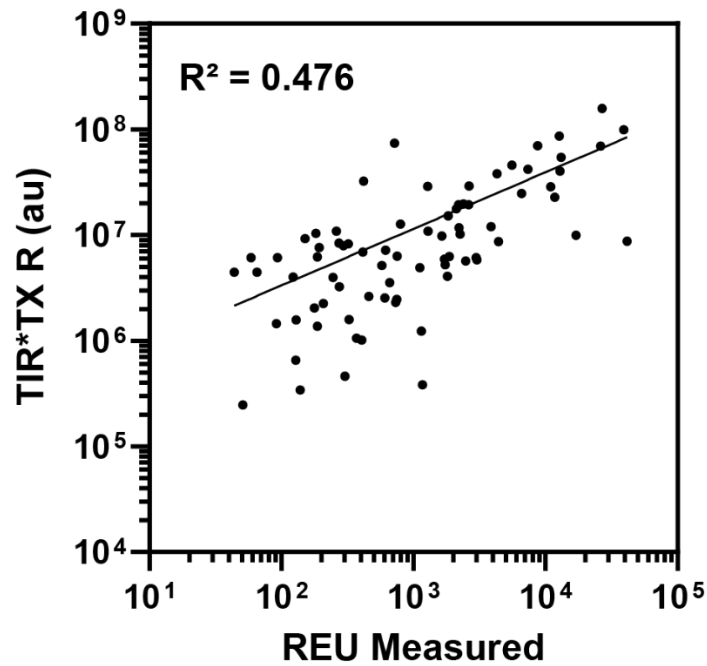
Supplementary Fig. 30 | Validating a 25Nanoluc-GFP cassette. **a** The genetic schematic of an EU standard curve for Nanoluc is shown. **b** The measured Expression Unit (EU) as a function of AHL concentration is shown for **a**. **c** Activity of full-length Nanoluc as a function of EU is shown to correlate linearly. **d** The genetic schematic of an EU standard curve for LacI is shown. **e** The measured EU as a function of AHL concentration is shown for **d**. **f** EU(LacI) correlates linearly with EU(Nanoluc) over varying transcription levels. Source data are provided as a Source Data file. Data represent the average of $n = 6$ biological replicates, with groups of three taken on two separate days except for **c** ($n = 3$, one-day measurement). Error bars correspond to the SEM of these measurements.



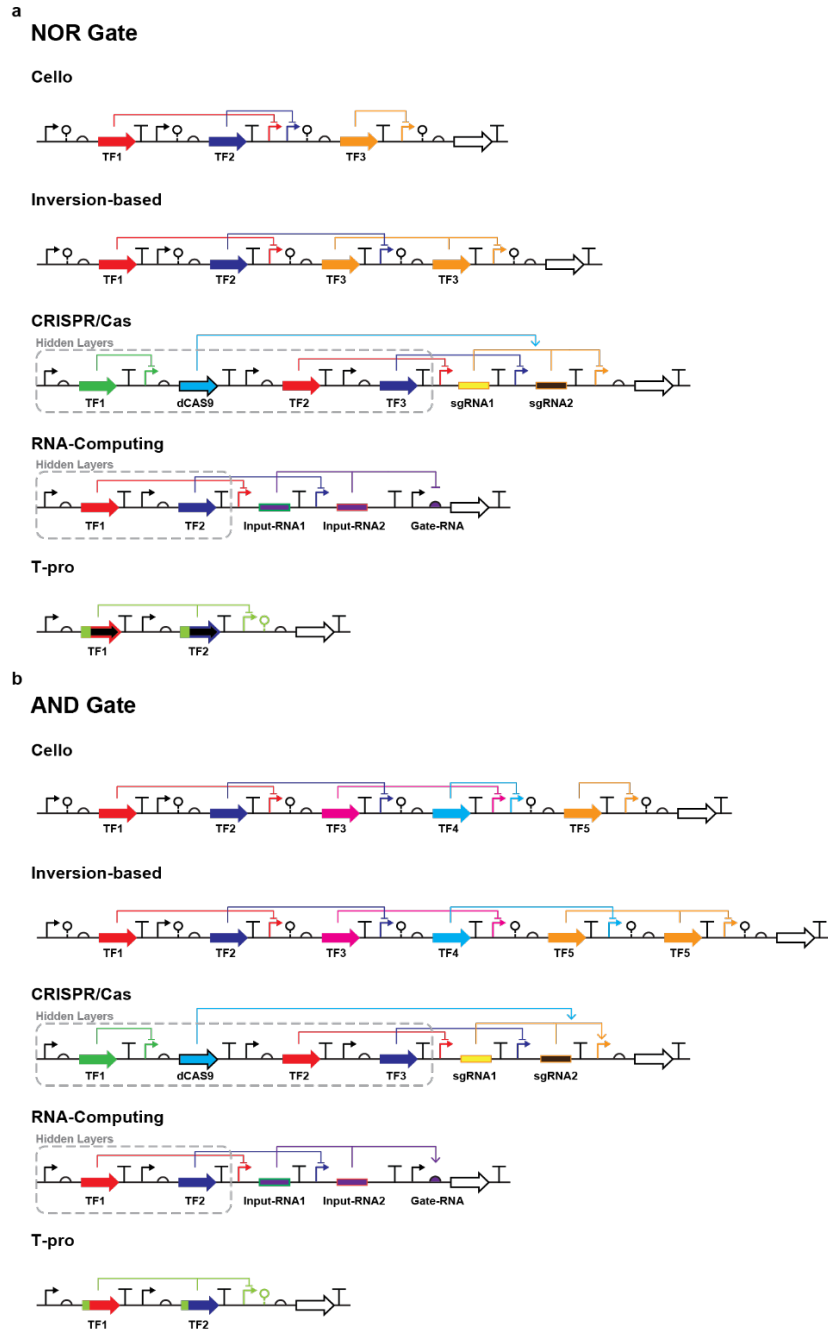
Supplementary Fig. 31 | Probing recombinase functionality and degradation. **a** A circuit expressing an A118-GFP fusion reporter with BAC origin is shown. **b** The GFP fusion reporter to measure fluorescence levels in Expression Units (EU) is not ideal for measuring low expression levels. **c** The genetic schematic of an EU standard curve for A118 is shown. **d** The measured EU as a function of AHL concentration is shown for **c**. **e** The genetic schematic of A118 activating a promoter-flip for GFP expression is shown. **f** The measured recombination efficiency as a function of AHL concentration is shown for **e**. **g** The genetic schematic of an EU standard curve for Int12 is shown. Three degradation tag variants are used (DAS, AAV, and no degradation tag). **h** The measured EU as a function of AHL concentration is shown for **g**. **i** The genetic schematic of Int12 with different degradation tags activating a promoter-flip for GFP expression is shown. **j** The measured recombination efficiency as a function of AHL concentration is shown for **i**. **k** Data correlating EU with functionality for Int12 with three different degradation profiles all fit on under a single dose response curve highlighting the robustness of the metric. Source data are provided as a Source Data file. Data represents the average of $n = 6$ biological replicates, with groups of three taken on two separate days. Error bars correspond to the SEM of these measurements.

NOTE: We sought to extend the application of our CSEC architecture and EU metric by examining its utility in modeling the impact of different SsrA protein degradation tags on targeted protein functionality. For this purpose, we selected another serine integrase (Int12) and explored three Nanoluc fusion variants: one with a DAS tag, another with an AAV tag, and a third without any tag. Similar to the A118 experiment, we generated standard curves for

each variant and measured their functionality through a gain-of-function GFP expression assay. Our results showed distinct ranges of expression levels and functional performance for each variant, as anticipated. Notably, despite these variations, all the data points from the DAS, AAV, and no-tag variants fit within a single nonlinear curve, reinforcing the robustness and versatility of our workflow. This unified correlation validated our EU metric as a reliable tool for evaluating protein functionality across varying expression and degradation profiles.



Supplementary Fig. 32 | Comparison of sequence-based models of expression. Measured EU values for 77 constitutively expressed constructs were compared to predicted expression levels based on the RBS and Promoter Calculators. See **Methods** for details. Source data are provided as a Source Data file.



Supplementary Fig. 33 | Genetic logic gate implementations across different design frameworks. **a** NOR gate and **b** AND gate designs implemented using various synthetic biology frameworks. Each row represents a different design strategy, shown from top to bottom: Cello¹⁴, generalized inversion-based logic, CRISPR/dCas9-mediated circuits¹⁵, RNA-computing architectures¹⁶, and Transcriptional Programming (T-Pro). The Cello framework compresses NOR logic from inversion-based circuitry by using tandem promoters and layered transcription factor arrangements. In CRISPR/dCas9 systems, logic is encoded through cascades of sgRNAs, with hidden layers consisting of additional transcription factors and dCas9 expression cassettes that are typically omitted from schematic representations. RNA-computing architectures function at the post-transcriptional level, utilizing structured RNAs as inputs and gate elements to perform logic operations. The T-Pro architecture, by contrast, enables a minimalistic circuit design highlighting how T-Pro can facilitate streamlined and resource-efficient genetic circuit construction.

NOTE: From the vantage point of predictability and tunability we posit that consideration of the context variables identified in this study will be equally important for all of the aforesaid technologies. Moreover, we believe that our workflow can be adapted to reconcile the context factors that influence the qualitative production and performance of non-coding RNA and we regard this as the next challenge moving forward, in addition to scaling to higher-state (*i.e.*, greater than 3-inputs) logic.

SUPPLEMENTARY REFERENCES

1. Wang, B.J., Kitney, R.I., Joly, N. & Buck, M. Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology. *Nat Commun* **2** (2011).
2. Zhang, Y., Ding, W.T., Wang, Z.H., Zhao, H.M. & Shi, S.B. Development of Host-Orthogonal Genetic Systems for Synthetic Biology. *Adv Biol* **5** (2021).
3. Groseclose, T.M., Rondon, R.E., Herde, Z.D., Aldrete, C.A. & Wilson, C.J. Engineered systems of inducible anti-repressors for the next generation of biological programming. *Nat Commun* **11** (2020).
4. Cardinale, S. & Arkin, A.P. Contextualizing context for synthetic biology - identifying causes of failure of synthetic biological systems. *Biotechnology Journal* **7**, 856-866 (2012).
5. Yeung, E. et al. Biophysical Constraints Arising from Compositional Context in Synthetic Gene Networks. *Cell Systems* **5**, 11-+ (2017).
6. Milner, P.T. et al. Performance Prediction of Fundamental Transcriptional Programs. *Acs Synth Biol* **12**, 1094-1108 (2023).
7. Rondon, R.E., Groseclose, T.M., Short, A.E. & Wilson, C.J. Transcriptional programming using engineered systems of transcription factors and genetic architectures. *Nat Commun* **10**, 4784 (2019).
8. Huang, B.D., Groseclose, T.M. & Wilson, C.J. Transcriptional programming in a Bacteroides consortium. *Nat Commun* **13**, 3901 (2022).
9. Richards, D.H., Meyer, S. & Wilson, C.J. Fourteen Ways to Reroute Cooperative Communication in the Lactose Repressor: Engineering Regulatory Proteins with Alternate Repressive Functions. *Acs Synth Biol* **6**, 6-12 (2017).
10. Groseclose, T.M., Hersey, A.N., Huang, B.D., Realff, M.J. & Wilson, C.J. Biological signal processing filters via engineering allosteric transcription factors. *Proc Natl Acad Sci U S A* **118**, e2111450118 (2021).
11. Mangan, S. & Alon, U. Structure and function of the feed-forward loop network motif. *Proc Natl Acad Sci U S A* **100**, 11980-11985 (2003).
12. Gallego-Jara, J. et al. Lycopene overproduction and in situ extraction in organic-aqueous culture systems using a metabolically engineered Escherichia coli. *AMB Express* **5**, 65 (2015).
13. Yoon, K.W., Doo, E.H., Kim, S.W. & Park, J.B. In situ recovery of lycopene during biosynthesis with recombinant Escherichia coli. *J Biotechnol* **135**, 291-294 (2008).
14. Nielsen, A.A. et al. Genetic circuit design automation. *Science* **352**, aac7341 (2016).
15. Nielsen, A.A. & Voigt, C.A. Multi-input CRISPR/Cas genetic circuits that interface host regulatory networks. *Molecular Systems Biology* **10** (2014).
16. Green, A.A. et al. Complex cellular logic computation using ribocomputing devices. *Nature* **548**, 117-+ (2017).