

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

Corresponding Author: Professor Corey Wilson

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The manuscript provides a novel approach to compress the size of genetic circuits that can be used to implement different Boolean operations, which are the fundamental blocks of biological computing and large-scale genetic circuits. When the size of a genetic circuit is smaller, this can also contribute to fitter organisms due to less metabolic burden.

Significance and relation to the literature:

The work has significant results related to scaling up designs and future automation. The work builds on the authors' previous work, presenting incremental outcomes. The introduction section can introduce and acknowledge the work of others that was built upon more sufficiently, for example, related to the design and implementation of three input promoter architectures, existing and related genetic design automation and modular modelling approaches.

Support of the claims:

The manuscript is accompanied by several types of data. Characterisation-related data can especially be very valuable to others. The workflow is explained step by step, and the outcome is explained using two different use cases.

Data analysis and conclusions:

It is great that the manuscript provides different types of datasets. The manuscript also mentions that models are used in predictions. However, there is not much information about these models.

Methodology:

The methods related to the wet lab have been explained very well. However, the manuscript should provide more information about the computational methods used.

Reproducibility:

One of the main claims of the manuscript, which is also included in the title, is the software. However, there is not much information about this optimisation software in the manuscript. Moreover, providing sequences for synthetic biology is crucial for reusability and reproducibility. Although the manuscript contains different types of useful data, sequence information is not included.

Major recommendations:

- * If the authors do not have a concern to make the software available, then please provide a publicly available software repository link with a suitable licence. Perhaps, a short data availability section can be used to provide such information. Otherwise, please provide more information, such as the algorithm, the programming language, and the standards used.
- * Can the authors please provide sequence (e.g. DNA) information for all individual parts?
- * Ideally, sequences of all designs and constructs should also be available. Would it be difficult to provide all sequences? For example, The Cello dataset used as a benchmark for this manuscript uses the Synthetic Biology Open Language (SBOL) to make the designs computationally available.
- * Please provide more information about the computational modelling approach, the modelling language and the framework

used. If the models were built manually, an example can help explain the formation and application of the mathematical approaches.

* It is great that the authors used SBOL Visual. Can this please be explained in methods? This would be important to support the standardisation efforts and adoption of data standards for reproducibility.

* SBOL would particularly be useful for representing the directed acyclic graphs. Can the authors please explain how the optimisation software captured the required data and the role of any data standard for reusability and reproducibility?

* Please extend the discussion slightly to discuss the potential issues and drawbacks as well. The approach and the data presented are valuable but depend on the specific parts. Can the approach presented be applied to other designs?

Minor recommendations:

* Line 96: "... Transcriptional Programming (TPro) leverages synthetic TFs": Do authors refer to their previous work by TPro. If that is the case, please clarify and cite the work. Otherwise, it can be argued that transcriptional programming does not always use synthetic TFs.

* Line 99: Please clarify the particular work. Otherwise, this sentence suggests a general rule that would not always be correct for an introduction section.

* Line 103: Please briefly give two or three examples of the 16 boolean operations. It may be better to consistently refer to them as Boolean operations rather than truth tables.

* Line 100: Clarify the meaning of inversion-based circuits. It would still be possible to apply inversion with a single promoter. If you mean cascades of inversions, please explain accordingly.

* Line 114: Please explain where 256 boolean operations come from and provide two or three examples briefly.

* Line 177: It is not clear what authors refer to when they mention the feedforward loop (FFL). Can this please be clarified? FFLs typically involve the X-Y, Y-Z, and X-Z relationships for X, Y, and Z components. Perhaps a suitable paper can be cited to clarify.

* Lines 194-198: Reference previous work in this area. "We cannot quantitatively predict ...": This is an active research problem, and several papers have addressed this issue previously. It would be great if the authors could reword this section and explain how the approach in the manuscript differs or builds on others' work, if applicable.

* Lines 243-246: As above, please acknowledge the current literature and explain how you used the relevant data as parameters.

* Line 566: Reference RBS Calculator V2.1

* Fig 2C: Consider clarifying the use of letter A since it was previously used for Cellobiose in Fig 1c. This makes it challenging sometimes to follow the explanations.

* Fig 1f: Please double check this figure. Is there a terminator between R2+ and P2? If there is not, then Pcon also activates the downstream E1+ component, which can change the circuit's dynamics.

* Consider providing an overview workflow diagram at the beginning of the results. The manuscript provides a step-by-step workflow, and an overview diagram may help the reader more easily navigate through the content.

* Spell out DBD

* Supp. Note 4: Please clarify what is meant by 4 layers.

* Fig 1.d: Please double check the colouring for the "no ligand" option for YQR in the subfigures.

(Remarks on code availability)

The code is not available, and we included comments related to this.

Reviewer #2

(Remarks to the Author)

This manuscript presents a significant contribution to the field of synthetic genetic circuit design by integrating a novel set of synthetic transcription factors (wetware) with a computational modeling framework (software) to achieve high-compression genetic circuits. The proposed system enables multi-input Boolean logic implementation while minimizing genetic footprint, which is a crucial step toward scalable and efficient synthetic biocomputing. While the study introduces a valuable methodology and provides a solid experimental foundation, several critical aspects require further clarification and expansion. The key issues include circuit scalability beyond three inputs, the computational complexity of the enumeration algorithm, experimental validation under diverse growth conditions, and improved software accessibility for external researchers. Addressing these concerns will significantly enhance the manuscript's impact and utility.

This manuscript represents an important step toward rational and compressed genetic circuit design, and the integration of synthetic transcription factors with computational modeling is commendable. However, the study would benefit from additional experimental validation under diverse conditions, improved discussion on scalability, and greater transparency in software accessibility. With these revisions, the manuscript will be a strong candidate for publication in Nature Communications and a valuable resource for synthetic biology researchers.

Major Concerns

1. The manuscript convincingly demonstrates the transition from 2-input to 3-input logic circuits, yet there is no discussion or empirical validation on how well the design framework scales beyond three inputs. The enumeration algorithm evaluates 100+ trillion circuit configurations for 3-input circuits (Supplementary Note 4), but it is unclear if the approach remains computationally feasible for 4-input or higher-order circuits. The manuscript should include a computational complexity analysis detailing how the time and memory requirements of the algorithm scale with increasing circuit inputs. A comparison with existing circuit design automation tools, such as Cello (Nielsen et al., 2016), would strengthen the argument. If 4-input

circuits are computationally infeasible using the current algorithm, the authors should discuss potential optimizations, such as heuristic search methods, constraint-based pruning, or machine learning-based predictions.

2. While the authors validate their circuits in a single *E. coli* strain under defined conditions, there is no assessment of how well these circuits function across different bacterial strains, metabolic states, or environmental conditions. Given that metabolic burden and resource competition significantly impact genetic circuit performance (as acknowledged in Supplementary Note 6 regarding lycopene toxicity), additional experiments should be conducted to measure circuit performance in at least one alternative strain (e.g., BL21, MG1655) and in varying nutrient conditions (e.g., LB vs. M9 minimal media). If experimental validation in multiple strains is not feasible, the authors should at least discuss potential limitations of their framework under different cellular contexts and propose strategies for improving robustness.

3. The manuscript primarily compares its compressed circuits against traditional inverter-based designs but does not provide direct comparisons with other advanced genetic circuit frameworks such as CRISPRi-based logic gates (Nielsen et al., 2016) or RNA-based regulators (Green et al., 2017). Including experimental benchmarking against at least one alternative circuit architecture would strengthen the claim that the T-Pro framework provides superior compression and predictive accuracy. If direct comparisons cannot be performed, the discussion should at least provide a qualitative comparison of the strengths and weaknesses of different approaches in terms of genetic footprint, predictability, and tunability.

4. The authors describe an enumeration-based circuit design algorithm but do not provide access to the code or workflow implementation details. To enhance reproducibility, the manuscript should include: A GitHub repository link to the software code (or clear instructions on how to request access if proprietary). A step-by-step workflow diagram in the supplementary materials illustrating how researchers can use the software for their own circuit designs. Example scripts for running the enumeration algorithm on a sample dataset. Given the increasing emphasis on open-source computational tools in synthetic biology, making the software accessible would significantly increase the manuscript's impact.

5. While the authors claim that their predictive model achieves an average error below 1.4-fold, the statistical rigor of this claim is unclear. Supplementary Figures 5, 7, and 9 provide experimental validation of the predictions, but the manuscript lacks a direct statistical correlation analysis (e.g., Pearson or Spearman correlation coefficients) comparing predicted vs. measured outputs. A Bland-Altman plot or residual analysis should be included to visualize prediction deviations and identify potential sources of error (e.g., outliers, systematic biases). The authors should discuss potential failure modes of the model, such as cases where metabolic burden affects circuit behavior unpredictably.

Minor Concerns

1. Figures depicting experimental results should explicitly indicate error bars and sample sizes where applicable. Terms like "compressed circuit" and "expression unit (EU)" should be formally defined in the main text rather than being introduced in the supplementary materials.

2. The manuscript focuses on single-layer Boolean logic circuits, but it does not explore how these circuits function when stacked into multi-layer architectures (e.g., feedforward loops or cascades). Some discussion on how the framework could be extended to hierarchical circuit designs would be beneficial.

3. Given that compressed circuits reduce redundancy, their evolutionary stability in long-term cultures remains uncertain. The authors should briefly discuss whether mutational robustness was considered in their design approach.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)

Review of Huang et al., Engineering wetware and software for the predictive design of compressed genetic circuits for higher-state decision-making

Huang et al. present a circuit compression strategy that integrates Transcriptional Programming (T-Pro) wetware with an algorithmic pipeline for designing multi-input genetic circuits with optimized Boolean logic. By developing synthetic transcription factors and computational workflows, the authors claim to achieve significantly smaller circuits (~5× smaller than canonical inverter-based circuits) while maintaining quantitative predictability. They validate their approach by designing and testing transcriptional programs, recombinase-based memory, and metabolic flux control.

This work addresses key limitations in synthetic circuit design, including limited modularity, high metabolic burden, and the lack of quantitative predictability. However, while the framework is rigorously tested, several critical gaps remain, particularly in demonstrating broader compression benefits over existing approaches, assessing circuit robustness, and evaluating practical limitations of compression strategies.

Comments:

1. The authors highlight circuit compression using two examples in Figure 5, but this is insufficient to establish their method as a superior compression strategy. Given that Cello is a widely used automated genetic circuit design tool, the authors should perform a direct, systematic comparison of compression rates across multiple large circuits previously designed using Cello. What is the average compression rate across different circuit sizes? Can their method reproduce and significantly reduce the size of the largest circuits ever designed with Cello? A table comparing circuit sizes, number of promoters, and number of TFs between their compressed circuits and Cello-designed circuits would be highly informative.

2. The manuscript demonstrates successful compression of 3-input circuits, but it would be valuable to discuss how well the

approach scales beyond this. Given that combinatorial complexity grows exponentially with additional inputs, what are the practical limits of this method? Are there theoretical constraints in their optimization workflow that would limit its applicability to 4-input or 5-input circuits? If the approach is computationally scalable, what additional considerations would be needed for experimental implementation? A deeper discussion on these points would enhance the clarity of the manuscript and help define the broader applicability of the framework.

3. The authors claim that compressed circuits retain equivalent function, but reducing circuit size may introduce performance compromises such as: weaker signal strengths (lower gene expression levels), slower response times due to fewer available transcriptional regulators, increased expression noise due to reduced redundancy. Have they quantified these effects? A direct comparison of circuit dynamics (e.g., response time, signal amplitude) between compressed and uncompressed versions of the same circuit would clarify these trade-offs.

4. The Context-Specific Expression Cassette (CSEC) workflow is an interesting approach to predicting expression levels, but it remains unclear whether this metric is robust across different genetic or environmental contexts. Have they tested CSEC under different metabolic conditions? Does the EU metric remain stable in various contexts, or does it require re-calibration?

5. The Figure 7 section, which focuses on tuning expression levels of metabolic pathway genes, appears misaligned with the core focus of the paper. The authors primarily screen RBSs to match target expression levels, which is not fundamentally about genetic circuit design, but rather expression level optimization. Since the main claim of the paper is about predictable genetic circuit design, not empirical tuning of metabolic pathways, removing this section would improve the focus of the manuscript.

6. The authors report <1.4-fold prediction error but do not specify how prediction accuracy varies across different types of circuits. Are some circuit topologies harder to predict than others?

7. The authors should consider how their circuit design pipeline could be integrated into existing synthetic biology tools. How easily can non-specialists use this algorithm for circuit design? Could they share their engineered transcription factors and genetic parts through Addgene? Could they provide a user-friendly modeling software interface? Their current GitHub repository is not user-friendly. Improving documentation and usability would increase adoption of their approach.

8. The study relies on synthetic transcription factors responsive to IPTG, ribose, and cellobiose. Have the authors considered expanding this approach to transcription factors that respond to biologically relevant signals?

(Remarks on code availability)

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

I recommend that the manuscript is accepted.

Thanks for addressing the comments and updating the manuscript accordingly. I have no further comments.

(Remarks on code availability)

The code is freely available on GitHub and examples are provided to demonstrate its use. The README file provides instructions to install the software.

Reviewer #2

(Remarks to the Author)

The authors have successfully and thoroughly addressed my original concerns. While the predictive framework remains context-dependent and requires recalibration when applied across different strains and media, the authors have addressed this limitation by demonstrating its adaptability through new experiments. They clearly explain the need for context-specific lookup tables and provide a systematic approach to generating them. I consider this an acceptable and well-acknowledged limitation in synthetic biology, and the presented method represents a significant advance in circuit design methodology. I consider this revised manuscript suitable for publication in Nature Communications.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)

The authors have thoroughly addressed all of my previous comments. The revised manuscript is substantially improved,

with clearer explanations, expanded methodological details, additional experimental validation, and improved data and code availability. I have no further comments. I recommend the manuscript for publication in its current form.

(Remarks on code availability)

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

REVIEWER 1

Comment 1: The manuscript provides a novel approach to compress the size of genetic circuits that can be used to implement different Boolean operations, which are the fundamental blocks of biological computing and large-scale genetic circuits. When the size of a genetic circuit is smaller, this can also contribute to fitter organisms due to less metabolic burden.

Response to Comment 1: We sincerely appreciate the reviewer's positive evaluation of our work and recognition of the significance of compressed genetic circuits in reducing metabolic burden and improving fitness.

Comment 2: Significance and relation to the literature: The work has significant results related to scaling up designs and future automation. The work builds on the authors' previous work, presenting incremental outcomes. The introduction section can introduce and acknowledge the work of others that was built upon more sufficiently, for example, related to the design and implementation of three input promoter architectures, existing and related genetic design automation and modular modelling approaches.

Response to Comment 2: Done. We have revised the introduction section to acknowledge and discuss key prior studies relevant to our work more explicitly. Excerpt below – note given the word length restrictions and other considerations we had to write this as concisely as possible.

“Synthetic genetic circuits facilitate the reprogramming of cells to perform novel functions, which has enabled myriad developments and applications that span biotechnology, chemical biology, and the like. Biological computing and large-scale genetic circuit engineering that enable said developments leverages the implementation of multi-input promoter architectures^{1,2}, modular modeling approaches³⁻⁶, and in a limited qualitative capacity via genetic design automation^{7,8}. The successful design of genetic...”

Comment 3: Data analysis and conclusions: It is great that the manuscript provides different types of datasets. The manuscript also mentions that models are used in predictions. However, there is not much information about these models.

Response to Comment 3: We have included a complete description of the models and workflow in the SI – see **Supplementary Fig. 2** for details.

Comment 4: Methodology: The methods related to the wet lab have been explained very well. However, the manuscript should provide more information about the computational methods used.

Response to Comment 4: We have provided a better description in the **Methods** section and included a complete description of the workflow including the computational methods and related materials in **Supplementary Fig. 2**. In addition, the source code with implementation instructions is available at <https://github.com/Jayaos/TPro>. NOTE: This link is now cited in the main text **Methods** section and in **Supplementary Fig. 2**.

Comment 5: Reproducibility: One of the main claims of the manuscript, which is also included in the title, is the software. However, there is not much information about this optimisation software in the manuscript. Moreover, providing sequences for synthetic biology is crucial for reusability and reproducibility. Although the manuscript contains different types of useful data, sequence information is not included.

Response to Comment 5: We have now included additional information for the enumeration software that systematically identifies compressed genetic circuits based on specified logic functions. As noted in our response to Comment 4, we have revised the **Methods** section to provide a more detailed description of this process. A flowchart of the workflow and pseudocode implementation details have also been included in the **Supplementary Material** – see **Methods** and **Supplementary Fig. 2** for details.

To support circuit design, we have also included a user-friendly software tool that allows users to visualize each transcription factor's optimal performance point and identify the corresponding RBS sequence needed to achieve performance closest to the ideal. This tool facilitates the design of SISO circuits with optimized performance. For MISO designs involving feedforward cascades, users can manually input the effective units (EU) of the upstream transcription factor along with chosen scaling values. This process is also illustrated in the newly added workflow diagram for clarity – see **Supplementary Fig. 2**.

Regarding reproducibility and reusability, we fully agree that providing sequence information is essential. We have included all relevant DNA sequences and plasmids used as a supplementary data file. Additionally, all plasmids used in this study have been deposited in GenBank, with the accession numbers provided in the main text. **Data Availability** excerpt given below.

“Data Availability

All data collected in this study are provided in the Supplementary Information and Source Data Files. Source Data are provided with this paper. Plasmid sequences are available from GenBank with accession numbers PV806910 - PV807102. Materials generated in this study are available from the corresponding author. Source code available at <https://github.com/Jayaos/TPro>.”

Comment 6: If the authors do not have a concern to make the software available, then please provide a publicly available software repository link with a suitable licence. Perhaps, a short data availability section can be used to provide such information. Otherwise, please provide more information, such as the algorithm, the programming language, and the standards used.

Response to Comment 6: Done. The source code is publicly available at <https://github.com/Jayaos/TPro> under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0). A detailed description of the algorithm and implementation, including the programming language used, is provided in the Methods section and the Supplementary Material.

Comment 7: Can the authors please provide sequence (e.g. DNA) information for all individual parts?

Response to Comment 7: Done. We have included all relevant DNA sequences and plasmids used as a supplementary data file. Additionally, all plasmids used in this study have been deposited in GenBank, with the accession numbers provided in the main text – see response to **Comment 5**.

Comment 8: Ideally, sequences of all designs and constructs should also be available. Would it be difficult to provide all sequences? For example, The Cello dataset used as a benchmark for this manuscript uses the Synthetic Biology Open Language (SBOL) to make the designs computationally available.

Response to Comment 8: Done: All plasmids developed in the study have been deposited in GenBank and are available in the supplementary data file – also see response to **Comment 5**.

Comment 9: Please provide more information about the computational modelling approach, the modelling language and the framework used. If the models were built manually, an example can help explain the formation and application of the mathematical approaches.

Response to Comment 9: Done. As described above in **Responses to Comments 3-5** we have expanded the Methods section to include additional details on algorithmic enumeration, covering both the modeling approach and the programming language used. Likewise, we have fully disclosed the models used in the quantitative design in the SI. Additionally, implementation details of the algorithm are provided in the supplementary material – see **Supplementary Fig. 2**.

Comment 10: It is great that the authors used SBOL Visual. Can this please be explained in methods? This would be important to support the standardization efforts and adoption of data standards for reproducibility.

Response to Comment 10: We did not use SBOL Visual all of our circuit figures were made in-house – *i.e.*, we manually created all figures using common symbols for genetic circuit representation. While integrating SBOL Visual into our enumeration algorithm to automate circuit generation would be highly beneficial, this remains an area for future work. We acknowledge the potential of such an approach to further support standardization and automation, and we aim to potentially explore this direction in subsequent studies.

Comment 11: SBOL would particularly be useful for representing the directed acyclic graphs. Can the authors please explain how the optimization software captured the required data and the role of any data standard for reusability and reproducibility?

Response to Comment 11: The algorithm stores all valid circuit configurations that meet the specified logic conditions and merges them across increasing levels of complexity. The output from our enumeration algorithm is currently represented in a compact one-line notation, as illustrated in **Figure 2**. A full description of the software and models is given in **Supplementary Fig. 2**. As noted in **Response to Comment 8** we potentially plan to incorporate SBOL in future generations of the software.

Comment 12: Please extend the discussion slightly to discuss the potential issues and drawbacks as well. The approach and the data presented are valuable but depend on the specific parts. Can the approach presented be applied to other designs?

Response to Comment 12: Done.

Minor recommendations:

Comment 13: Line 96: “... Transcriptional Programming (TPro) leverages synthetic TFs”: Do authors refer to their previous work by TPro. If that is the case, please clarify and cite the work. Otherwise, it can be argued that transcriptional programming does not always use synthetic TFs.

Response to Comment 13: Yes, Transcriptional Programming (T-Pro) is refer to in previous work and is now better cited in the current manuscript.

Comment 14: Line 99: Please clarify the particular work. Otherwise, this sentence suggests a general rule that would not always be correct for an introduction section.

Response to Comment 14: Done. We have included references to support this statement. We have also provided an illustration of comparing inversion vs. compression in **Supplementary Fig. 1** with an expanded discussion in **Supplementary Note 2**. Excerpt from the main text given below for clarity.

*“Notably, T-Pro utilizes engineered repressor⁹ and anti-repressor^{10,11} TFs that support coordinated binding to cognate synthetic promoters—mitigating the need for inversion (**Supplementary Fig. 1, Supplementary Note 2**). Specifically, synthetic anti-repressors facilitate objective NOT/NOR Boolean operations that utilize fewer promoters relative to inversion-based circuits^{11,12}.”*

Comment 15: Line 103: Please briefly give two or three examples of the 16 boolean operations. It may be better to consistently refer to them as Boolean operations rather than truth tables.

Response to Comment 15: Done. We have provided a summary of all 16 2-input operations in **Supplementary Fig. 1**. Note, we illustrated this in the context of scaling to 3-inputs or greater. In addition, the use of the term “truth tables” is deliberate and important as this is the input for the design software. Moreover, once you move beyond 2-inputs all operations will not have a convenient name.

Comment 16: Line 100: Clarify the meaning of inversion-based circuits. It would still be possible to apply inversion with a single promoter. If you mean cascades of inversions, please explain accordingly.

Response to Comment 16: We have now included an illustration of an inverter relative to a compressed circuit for a NOT and NOR operation – see **Supplementary Fig. 1** – where the class of inverter is from the Cello (Voigt *et al.*) designs.

Comment 17: Line 114: Please explain where 256 boolean operations come from and provide two or three examples briefly.

Response to Comment 17: Done. We have included a description in the supplement on how you scale Boolean logic as you increase the number of inputs – see **Supplementary Fig. 1** – also see **Response to Comment 15**.

Comment 18: Line 177: It is not clear what authors refer to when they mention the feedforward loop (FFL). Can this please be clarified? FFLs typically involve the X-Y, Y-Z, and X-Z relationships for X, Y, and Z components. Perhaps a suitable paper can be cited to clarify.

Response to Comment 18: We have revised the text to explicitly state that our design represents a feedforward cascade, rather than a classical feedforward loop (FFL). We have also expanded **Supplementary Note 4** for a broader discussion of the differences and intent in this paper.

Comment 19: Lines 194-198: Reference previous work in this area. “We cannot quantitatively predict ...”: This is an active research problem, and several papers have addressed this issue previously. It would be great if the authors could reword this section and explain how the approach in the manuscript differs or builds on others’ work, if applicable.

Response to Comment 19: Done – see excerpt below. Note, given the limits on the word count and other considerations for the main text we have included an additional comment in **Supplementary Note 1**.

“Qualitatively, we have a clear understanding of the design rules for a basic genetic circuit. However, given a promoter of strength “x” and an RBS of strength “y” we cannot quantitatively predict the expression level of a gene of interest (GOI) with a given reading frame “z” a priori. We define the discrepancy between qualitative design and the prediction of a genetic circuit with a prescribed quantitative performance as the synthetic biology problem (Supplementary Note 1, Fig. 3a). The aforesaid can be regarded as a grand challenge in synthetic biology and is currently an active research problem^{6, 13, 14}.

Comment 20: Lines 243-246: As above, please acknowledge the current literature and explain how you used the relevant data as parameters.

Response to Comment 20: We did not use previously published data to set the parameters *per se*, rather the parameters were constrained by consensus elements from which designed components were generated in this study. Moreover, all putative references overlap with those given in **Response to Comment 19**, and the additional context is given in **Supplementary Note 1**.

Comment 21: Line 566: Reference RBS Calculator V2.1

Response to Comment 21: Done. We have added the following reference for RBS calculator V2.1 (<https://pubs.acs.org/doi/10.1021/acssynbio.0c00394>)

Comment 22: Fig 2C: Consider clarifying the use of letter A since it was previously used for Cellobiose in Fig 1c. This makes it challenging sometimes to follow the explanations.

Response to Comment 22: Noted – we have clarified the use of “A” relative to the superscript “X^A”
Note, super script “A” indicates the anti-repressor phenotype, and the non-scripted A represents a generalized TF or the ligand cognate to the biosensor. This is now noted in the figure caption.

Comment 23: Fig 1f: Please double check this figure. Is there a terminator between R2+ and P2? If there is not, then Pcon also activates the downstream E1+ component, which can change the circuit's dynamics.

Response to Comment 23: We appreciate the reviewer for pointing this out. Upon review, we confirmed that a terminator was missing. To address this, we have revised Figure 2f to include the missing terminator, ensuring the circuit representation accurately reflects its intended function.

Comment 24: Consider providing an overview workflow diagram at the beginning of the results. The manuscript provides a step-by-step workflow, and an overview diagram may help the reader more easily navigate through the content.

Response to Comment 24: As discussed in responses to earlier comments 4,5,9 we have an extensive overview of the overall workflow in **Supplementary Fig. 2**. In addition, all relevant code of the enumeration algorithm is provided along with a link to a github repository of an implementation of the enumeration scheme.

Comment 25: Spell out DBD

Response to Comment 25: For consistency we changed DBD to synthetic alternate DNA recognition (ADR), which is used exclusively throughout the manuscript.

Comment 26: Supp. Note 4: Please clarify what is meant by 4 layers.

Response to Comment 26: We have clarified “what is meant by 4 layers” in the revised Supplementary Note 4 – excerpt given below:

“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.”

Comment 27: Fig 1.d: Please double check the colouring for the “no ligand” option for YQR in the subfigures.

Response to Comment 27: The coloring is correct. The dynamic range for each variant is shown on a white-blue scale for repressors, and white-purple scale for anti-repressors. The dynamic range is defined as the high output state divided by the low output state. To better clarify this, we have revised the figure and figure caption.

REVIEWER 2

Comment 1: This manuscript presents a significant contribution to the field of synthetic genetic circuit design by integrating a novel set of synthetic transcription factors (wetware) with a computational modeling framework (software) to achieve high-compression genetic circuits. The proposed system enables multi-input Boolean logic implementation while minimizing genetic footprint, which is a crucial step toward scalable and efficient synthetic biocomputing. While the study introduces a valuable methodology and provides a solid experimental foundation, several critical aspects require further clarification and expansion. The key issues include circuit scalability beyond three inputs, the computational complexity of the enumeration algorithm, experimental validation under diverse growth conditions, and improved software accessibility for external researchers. Addressing these concerns will significantly enhance the manuscript's impact and utility. This manuscript represents an important step toward rational and compressed genetic circuit design, and the integration of synthetic transcription factors with computational modeling is commendable. However, the study would benefit from additional experimental validation under diverse conditions, improved discussion on scalability, and greater transparency in software accessibility. With these revisions, the manuscript will be a strong candidate for publication in Nature Communications and a valuable resource for synthetic biology researchers.

Response to Comment 1: We appreciate the reviewer's thorough evaluation of our work and their recognition of the significance of integrating synthetic transcription factors with computational modeling to enable compressed genetic circuit design. We acknowledge the key concerns regarding scalability beyond three-input circuits, computational complexity, validation under diverse growth conditions, and software accessibility. We have addressed all of the concerns raised by the reviewer detailed below.

Comment 2: The manuscript convincingly demonstrates the transition from 2-input to 3-input logic circuits, yet there is no discussion or empirical validation on how well the design framework scales beyond three inputs. The enumeration algorithm evaluates 100+ trillion circuit configurations for 3-input circuits (Supplementary Note 4), but it is unclear if the approach remains computationally feasible for 4-input or higher-order circuits. The manuscript should include a computational complexity analysis detailing how the time and memory requirements of the algorithm scale with increasing circuit inputs. A comparison with existing circuit design automation tools, such as Cello (Nielsen et al., 2016), would strengthen the argument. If 4-input circuits are computationally infeasible using the current algorithm, the authors should discuss potential optimizations, such as heuristic search methods, constraint-based pruning, or machine learning-based predictions.

Response to Comment 2: Done. We agree with the reviewer's concern regarding the scalability of our current enumeration and optimization framework beyond 3-input logic. While full-scale implementation of higher-order (*i.e.*, > 3-inputs) circuits is beyond the scope of this study, we revised the **Discussion** and have included a note in the SI (**Supplementary Note 7**) to address the aforesaid – excerpts below.

In addition, we have included a qualitative comparison of T-Pro vs. Cello circuits in the **Source Data File** – see the Tab labeled “Enumeration Lookup Table”

“Discussion

While this study successfully illustrated the advance of T-Pro from four-state programming (2-input Boolean logic) to eight-state programming (3-input Boolean logic), a key question is can the wetware, software, and workflows be extended to even higher-states of synthetic decision-making. Undoubtedly, we can expand our wetware to support >3-input Boolean logic – as we have projected our design space to be on the order of 10^6 biosensors and 10^3 synthetic promoters¹⁵. Moreover, we posit that the workflow and models will continue to be effective given that circuit composition is predicated on SISO to MISO construction – barring metabolic limits of the chassis cell. However, 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 (**Supplementary Note 7**)."

“Supplementary Note 7: Complexity analysis: The search space size grows as $O(C^n)$, 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^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 $\sim 40^n$ 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. **Table below** summarizes the time and memory usage required to enumerate circuits under specific design space configurations, which collectively yielded 246 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.”

Design space configuration

# 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

NOTE: A qualitative comparison of T-Pro vs. Cello circuits is given in the **Source Data File** – see the Tab labeled “Enumeration Lookup Table”. 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.”

Comment 3: While the authors validate their circuits in a single *E. coli* strain under defined conditions, there is no assessment of how well these circuits function across different bacterial strains, metabolic states, or environmental conditions. Given that metabolic burden and resource competition significantly impact genetic circuit performance (as acknowledged in Supplementary Note 6 regarding lycopene toxicity), additional experiments should be conducted to measure circuit performance in at least one alternative strain (e.g., BL21, MG1655) and in varying nutrient conditions (e.g., LB vs. M9 minimal media). If experimental validation in multiple strains is not feasible, the authors should at least discuss potential limitations of their framework under different cellular contexts and propose strategies for improving robustness.

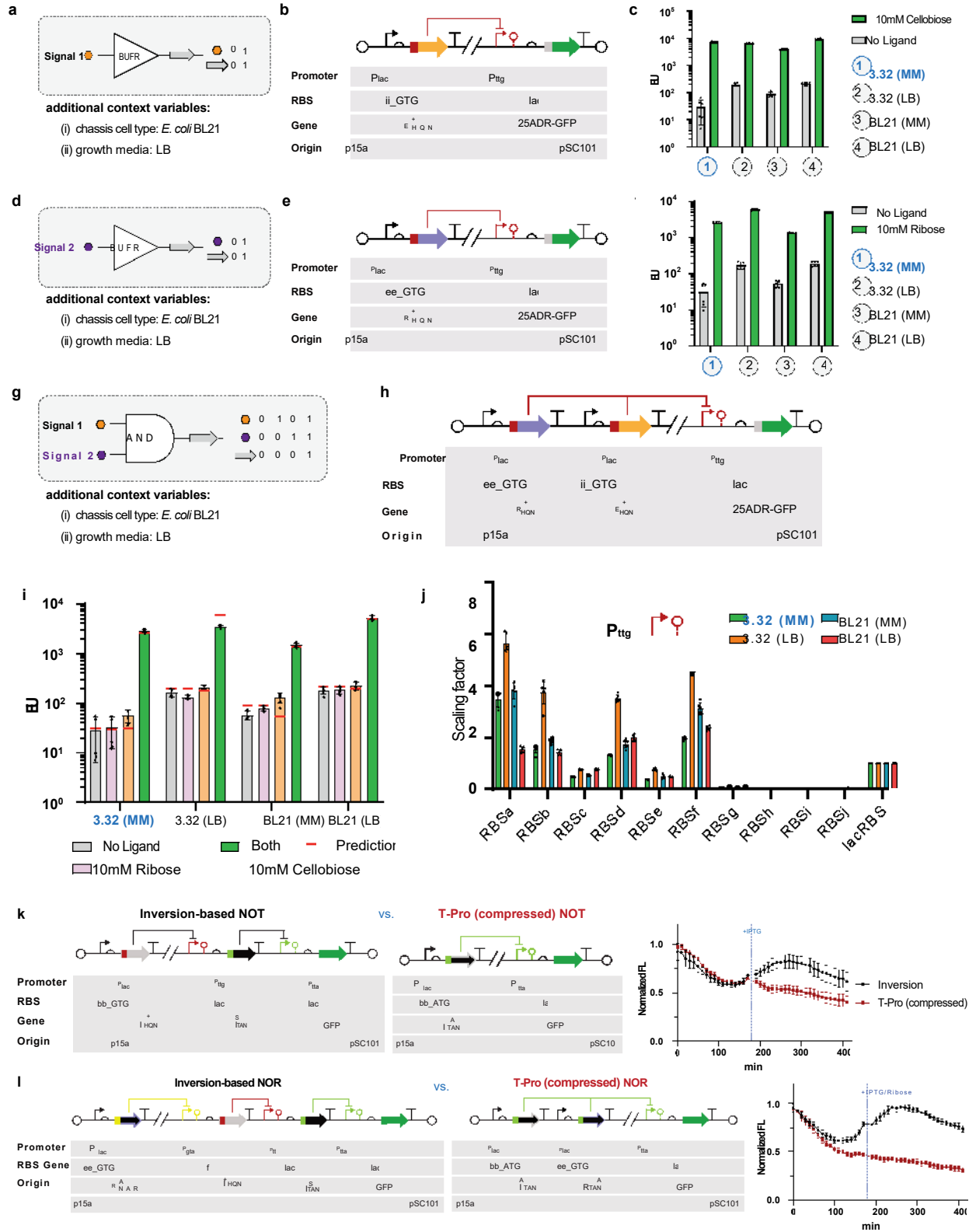
Response to Comment 3: We would like to thank the reviewer for this comment as it allowed us to expand (and better understand) context with respect to circuit design. We acknowledge the reviewer’s concern regarding the robustness of our circuits across different bacterial strains, metabolic states, and environmental conditions. As suggested by the reviewer we reevaluated select SISO and MISO using different cell lines and media. To address this, we have conducted additional experiments evaluating the performance of a representative MISO circuit in LB media and in an alternative *E. coli* strain (BL21) under both LB and minimal media conditions. These experiments confirm that T-Pro circuits are highly context-dependent, not only in terms of genetic architecture but also with respect to environmental factors such as nutrient availability and metabolic state. The observed differences in circuit performance highlight the need for recalibration when transitioning circuits across different cellular contexts.

This experimental assessment is now part of the main text results with a corresponding figure, in addition to an expanded **Supplementary Note 1**. Main text excerpts and figure given below:

*“Additional context variables and relative circuit performance. In addition to genetic contexts, we posited that chassis cell type and nutrient conditions will also have an impact on circuit performance (Supplementary Note 1). To test this assertion, we evaluated two BUFFER gate circuits in nutrient rich media and in BL21 *E. coli* cells, relative to the reference system – i.e., 3.32 *E. coli* chassis cells in minimal media – see Fig. 6a-f. In other words, the genetic context (promoter, RBS, GOI, plasmid) remained constant while the environment was changed. Qualitatively, the BUFFER operations performed as expected in all environments; however, quantitatively there were moderate differences in circuit performance upon change in chassis cell and nutrient conditions. Given the performance of the two SISO we were able to accurately predict the quantitative performance of the cognate MISO (i.e., BUFFERs to AND gate) substantiating our engineering workflow and central claim regarding improved circuit composability upon reconciliation of context variables (Fig. 6g-i and Supplementary Fig. 13).*

To further investigate the origin of how variation in growth media and chassis cell type impact circuit performance we reevaluated RBS strength under different environmental contexts (Fig. 6j). In brief, in all cases variation in chassis cell and growth media impacted RBS performance. This observation implies that RBS lookup table recalibration in the context of the selected chassis cell and growth media will be necessary to facilitate quantitative predictions; otherwise, our workflows and cognate models remain intact.”

Figure 6



Comment 4: The manuscript primarily compares its compressed circuits against traditional inverter-based designs but does not provide direct comparisons with other advanced genetic circuit frameworks such as CRISPRi-based logic gates (Nielsen et al., 2016) or RNA-based regulators (Green et al., 2017). Including experimental benchmarking against at least one alternative circuit architecture would strengthen the claim that the T-Pro framework provides superior compression and predictive accuracy. If direct comparisons cannot be performed, the discussion should at least provide a qualitative comparison of the strengths and weaknesses of different approaches in terms of genetic footprint, predictability, and tunability.

Response to Comment 4: Done. While a direct experimental comparison would be an enormous undertaking and more appropriate for a separate study, we did directly compare fundamental NOT/NOR inverter-based and compressed circuits. In addition, we revised the **Discussion** and have included an SI figure (**Supplementary Fig.18**) to more broadly address the aforesaid. Excerpts below:

“Additional context variables and relative circuit performance – paragraph 3. Finally, we conducted a direct comparison between inverter-based circuits and T-Pro analogs (Fig. 6k-l). Here, we engineered NOT and NOR gates as inverter-based circuits, and as compressed (T-Pro) iterations. The justification being that NOT/NOR Boolean operations can be used to construct any program in vivo or in silico. Accordingly, an assessment of said operations can serve as a reasonable proxy for more complex operations. Moreover, we constructed both iterations (inverter and non-inverter circuits) using T-Pro parts. In general, both compressed (T-Pro) NOT and NOR operations displayed significantly faster response times upon induction (Fig. 6k-l and Supplementary Fig. 14).”

“Discussion – paragraph 3: While this study primarily focused on comparing T-Pro to traditional inverter-based circuits, other circuit frameworks can be used to generate Boolean operations such as CRISPRi-based logic gates¹⁶ or RNA-based regulators¹⁷. To glean how different iterations of circuit frameworks compare to T-Pro we conducted a qualitative comparison of fundamental NOR logical circuits (Supplementary Fig. 18). As indicated before NOR logic can be used to construct any higher-state logical operation and can serve as a reasonable first approximation of how effective T-Pro circuit compression compares to other genetic circuit technologies in a broader context. Consistent with our supposition the T-Pro circuit required the fewest number of parts; this is also the case when comparing AND gate qualitative designs. While this is not a comprehensive assessment it is unlikely that the genetic footprint of more complex gates will fare any better relative to T-Pro given the fundamental constraints of the design rules for RNA-based and CRISPRi-based gates. Overall, we believe our technology represents a significant advance in genetic circuit design, as well circuit compression will have an impact on future applications of synthetic biology¹⁸⁻²⁴ that are anticipated to have major impacts in a broad range of areas.”

“Supplementary Fig.18 [figure on next page] | 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: Cello⁸, inversion-based logic, CRISPR/dCas9-mediated circuits²⁵, RNA-computing architectures²⁶, 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 logic.”

Supplementary Fig.18

a

NOR Gate

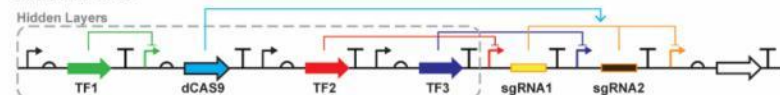
Cello



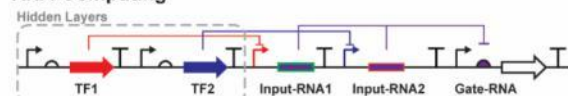
Inversion-based



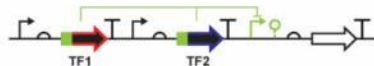
CRISPR/CAS



RNA-Computing



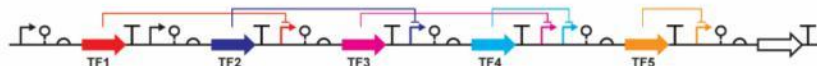
T-pro



b

AND Gate

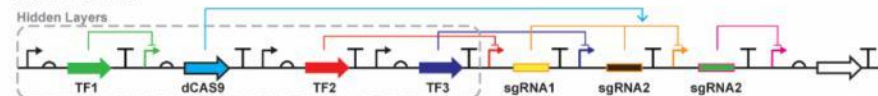
Cello



Inversion-based



CRISPR/CAS



RNA-Computing



T-pro



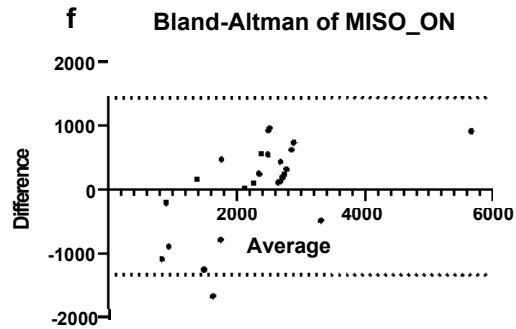
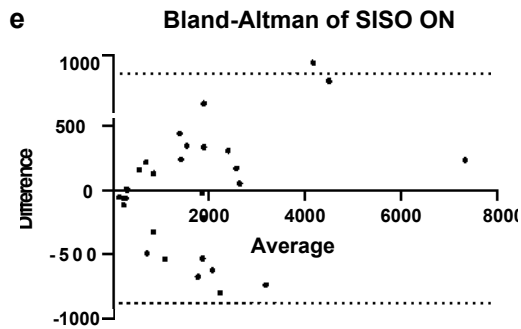
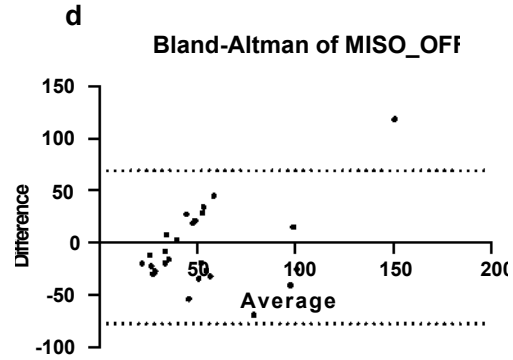
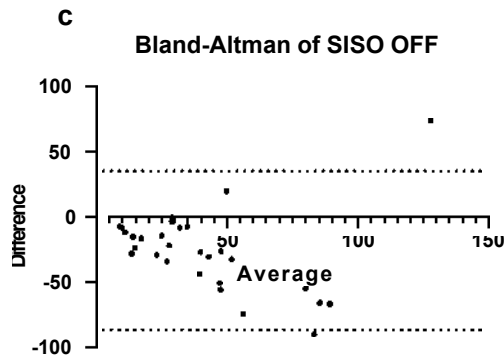
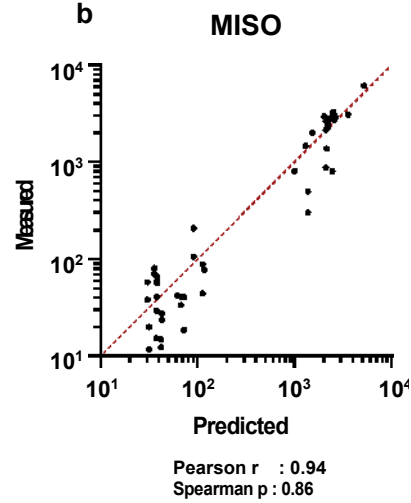
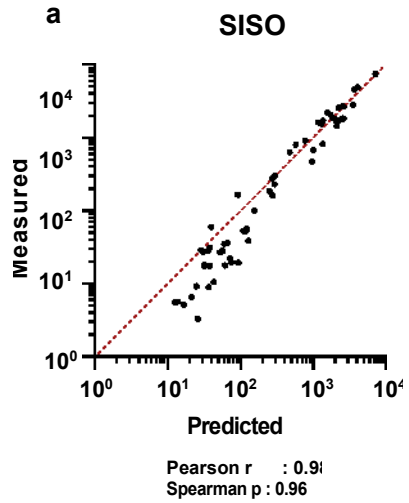
Comment 5: The authors describe an enumeration-based circuit design algorithm but do not provide access to the code or workflow implementation details. To enhance reproducibility, the manuscript should include: A GitHub repository link to the software code (or clear instructions on how to request access if proprietary). A step-by-step workflow diagram in the supplementary materials illustrating how researchers can use the software for their own circuit designs. Example scripts for running the enumeration algorithm on a sample dataset. Given the increasing emphasis on open-source computational tools in synthetic biology, making the software accessible would significantly increase the manuscript's impact.

Response to Comment 5: Done. The source code is available at <https://github.com/Jayaos/TPro>. The repository includes a Jupyter Notebook to facilitate implementation. Additionally, we have provided a flowchart of the algorithmic enumeration workflow in the Supplementary Material to help readers navigate the step-by-step process described in the manuscript – see **Supplementary Fig. 2**.

Comment 6: While the authors claim that their predictive model achieves an average error below 1.4-fold, the statistical rigor of this claim is unclear. Supplementary Figures 5, 7, and 9 provide experimental validation of the predictions, but the manuscript lacks a direct statistical correlation analysis (e.g., Pearson or Spearman correlation coefficients) comparing predicted vs. measured outputs. A Bland-Altman plot or residual analysis should be included to visualize prediction deviations and identify potential sources of error (e.g., outliers, systematic biases). The authors should discuss potential failure modes of the model, such as cases where metabolic burden affects circuit behavior unpredictably.

Response to Comment 6: Done. We have prepared an SI Figure (**Supplementary Fig. 10**) with a summary of the direct statistical correlation analysis for SISO and MISO – see excerpt below. In addition, we have prepared a detailed analysis spreadsheet to substantiate the “average error below 1.4-fold” claim summarized in the **Source Data File** – see the Tab labeled “**Supplementary Fig. 10**”

*“**Supplementary Fig. 10 [figure on next page] | SISO and MISO statistical correlation analysis.** **a.** Predicted versus measured expression 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 expression 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 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 data are provided as a Source Data file.”*



Minor Concerns:

Comment 7: Figures depicting experimental results should explicitly indicate error bars and sample sizes where applicable. Terms like "compressed circuit" and "expression unit (EU)" should be formally defined in the main text rather than being introduced in the supplementary materials.

Response to Comment 7: Done. We have explicitly indicated error bars and sample sizes in the figure legends where applicable. The term "compressed circuit" has been formally defined in the Introduction, and "expression unit (EU)" has been introduced in the **Methods** and **Supplementary Note 5** section to ensure clarity and consistency in terminology. Excerpt below:

*“**Methods - Expression Unit (EU) Metric.** 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) ²⁶, 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 5** elaborates on the assumptions and rationale underlying the use of the EU metric in our study.”*

Comment 8: The manuscript focuses on single-layer Boolean logic circuits, but it does not explore how these circuits function when stacked into multi-layer architectures (e.g., feedforward loops or cascades). Some discussion on how the framework could be extended to hierarchical circuit designs would be beneficial.

Response to Comment 8: Our approach inherently supports sequential cascades of layers, as most Boolean operations in our system are implemented through hierarchical transcriptional regulation. Additionally, if the reviewer is referring to executing multiple independent Boolean operations within a single cell using orthogonal components, this is indeed feasible within our framework, provided that regulatory elements remain functionally isolated. We have included an SI note (**Supplementary Note 4**) to clarify these aspects and highlight the potential for scaling T-Pro circuits to more complex layered architectures.

Comment 9: Given that compressed circuits reduce redundancy, their evolutionary stability in long-term cultures remains uncertain. The authors should briefly discuss whether mutational robustness was considered in their design approach.

Response to Comment 9: Done. Given the limits on the word count and other considerations for the main text we have included an additional comment in **Supplementary Note 7** – excerpt below:

*“**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.”

REVIEWER 3

Comment 1: Huang et al. present a circuit compression strategy that integrates Transcriptional Programming (T-Pro) wetware with an algorithmic pipeline for designing multi-input genetic circuits with optimized Boolean logic. By developing synthetic transcription factors and computational workflows, the authors claim to achieve significantly smaller circuits ($\sim 5\times$ smaller than canonical inverter-based circuits) while maintaining quantitative predictability. They validate their approach by designing and testing transcriptional programs, recombinase-based memory, and metabolic flux control.

This work addresses key limitations in synthetic circuit design, including limited modularity, high metabolic burden, and the lack of quantitative predictability. However, while the framework is rigorously tested, several critical gaps remain, particularly in demonstrating broader compression benefits over existing approaches, assessing circuit robustness, and evaluating practical limitations of compression strategies.

Response to Comment 1: We appreciate the reviewer's thorough evaluation of our work and their recognition of what our work brings to key limitations in synthetic circuit design, including limited modularity, high metabolic burden, and the lack of quantitative predictability. We acknowledge the key concerns, and we have addressed all of the concerns raised by the reviewer detailed below.

Comment 2: The authors highlight circuit compression using two examples in Figure 5, but this is insufficient to establish their method as a superior compression strategy. Given that Cello is a widely used automated genetic circuit design tool, the authors should perform a direct, systematic comparison of compression rates across multiple large circuits previously designed using Cello. What is the average compression rate across different circuit sizes? Can their method reproduce and significantly reduce the size of the largest circuits ever designed with Cello? A table comparing circuit sizes, number of promoters, and number of TFs between their compressed circuits and Cello-designed circuits would be highly informative.

Response to Comment 2: Done. We have included a qualitative comparison of T-Pro vs. Cello circuits in the **Source Data File** – see the Tab labeled “Enumeration Lookup Table” (partial screenshot below). *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).*

</

Addressing: “Can their method reproduce and significantly reduce the size of the largest circuits ever designed with Cello?” **YES** - the 3-input XNOR circuit in Figure 5 corresponds to the largest circuit designed by Cello in the Science paper, further reinforcing the scalability and efficiency of our approach.

Comment 3: The manuscript demonstrates successful compression of 3-input circuits, but it would be valuable to discuss how well the approach scales beyond this. Given that combinatorial complexity grows exponentially with additional inputs, what are the practical limits of this method? Are there theoretical constraints in their optimization workflow that would limit its applicability to 4-input or 5-input circuits? If the approach is computationally scalable, what additional considerations would be needed for experimental implementation? A deeper discussion on these points would enhance the clarity of the manuscript and help define the broader applicability of the framework.

Response to Comment 3: Done. We agree with the reviewer’s concern regarding the scalability of our current enumeration and optimization framework beyond 3-input logic. While full-scale implementation of higher-order (*i.e.*, > 3-inputs) circuits is beyond the scope of this study, we revised the **Discussion** and have included a note in the SI (**Supplementary Note 7**) to address the aforesaid – excerpts below.

“Discussion: While this study successfully illustrated the advance of T-Pro from four-state programming (2-input Boolean logic) to eight-state programming (3-input Boolean logic), a key question is can the wetware, software, and workflows be extended to even higher-states of synthetic decision-making. Undoubtedly, we can expand our wetware to support >3-input Boolean logic – as we have projected our design space to be on the order of 10^6 biosensors and 10^3 synthetic promoters¹⁵. Moreover, we posit that the workflow and models will continue to be effective given that circuit composition is predicated on SISO to MISO construction – barring metabolic limits of the chassis cell. However, 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 (**Supplementary Note 7**).”

“Supplementary Note 7: Complexity analysis: The search space size grows as $O(C^n)$, 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^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 $\sim 40^n$ 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. **Table below** summarizes the time and memory usage required to enumerate circuits under specific design space configurations, which collectively yielded 246 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.”

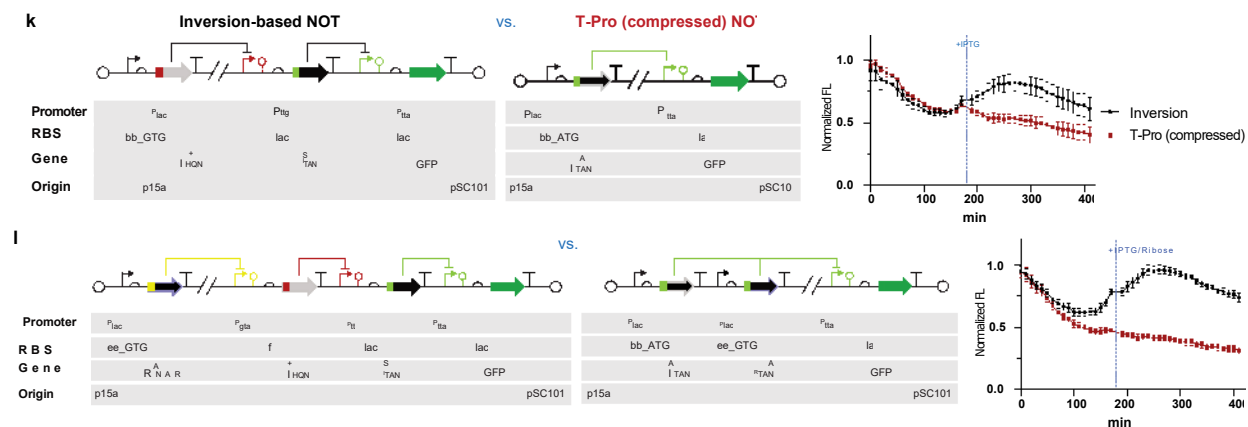
“Moving Beyond 3-Input Boolean Logic: Our current approach to algorithmic enumeration of 3-input T-Pro circuits and compression optimization is computationally expensive and it will be challenging to scale this approach to 16-states or greater – as this would require advancing from 256 operations to >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.”

Comment 4: The authors claim that compressed circuits retain equivalent function, but reducing circuit size may introduce performance compromises such as: weaker signal strengths (lower gene expression levels), slower response times due to fewer available transcriptional regulators, increased expression noise due to reduced redundancy. Have they quantified these effects? A direct comparison of circuit dynamics (e.g., response time, signal amplitude) between compressed and uncompressed versions of the same circuit would clarify these trade-offs.

Response to Comment 4: Done. We conducted a direct comparison of fundamental NOT/NOR inverter-based and compressed circuits. This comparison is now included in the main text. Note, given the limits on the word count and other considerations for the main text we have included additional comments in **Supplementary Fig. 14**. Excerpts below:

“Additional context variables and relative circuit performance – paragraph 3. Finally, we conducted a direct comparison between inverter-based circuits and T-Pro analogs (Fig. 6k-l). Here, we engineered NOT and NOR gates as inverter-based circuits, and as compressed (T-Pro) iterations. The justification being that NOT/NOR Boolean operations can be used to construct any program in vivo or in silico. Accordingly, an assessment of said operations can serve as a reasonable proxy for more complex operations. Moreover, we constructed both iterations (inverter and non-inverter circuits) using T-Pro parts. In general, both compressed (T-Pro) NOT and NOR operations displayed significantly faster response times upon induction (Fig. 6k-l and Supplementary Fig. 14).”

Figure 6 (excerpt)



“Supplementary Fig. 14 | 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. 10mM 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. 10mM 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. 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. 6l). 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.”

Comment 5: The Context-Specific Expression Cassette (CSEC) workflow is an interesting approach to predicting expression levels, but it remains unclear whether this metric is robust across different genetic or environmental contexts. Have they tested CSEC under different metabolic conditions? Does the EU metric remain stable in various contexts, or does it require re-calibration?

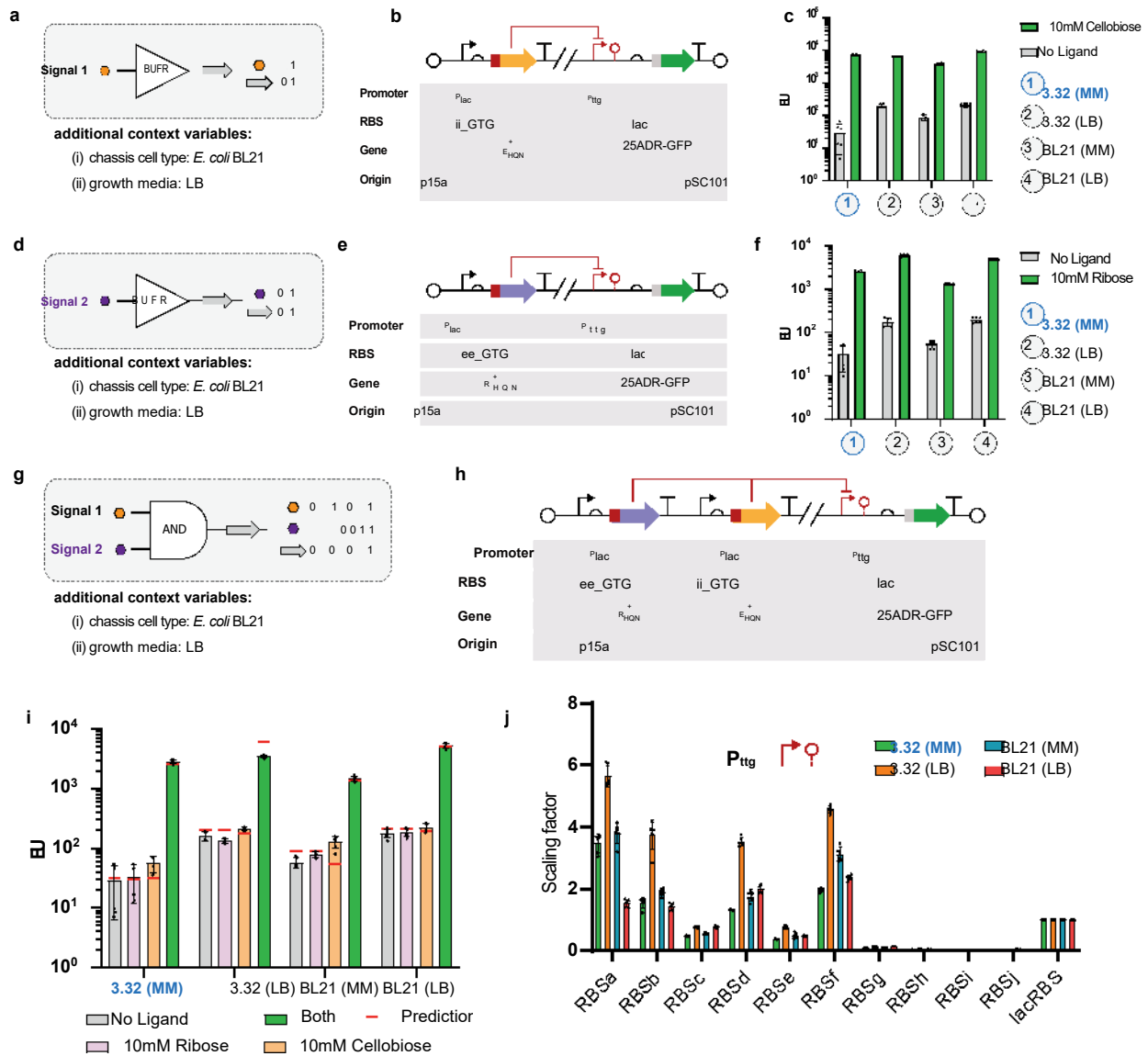
Response to Comment 5: We would like to thank the reviewer for this comment as it allowed us to expand (and better understand) context with respect to circuit design. We acknowledge the reviewer’s concern regarding the robustness of our circuits across different bacterial strains, metabolic states, and environmental conditions. As suggested by the reviewer we reevaluated select SISO and MISO using different cell lines and media. To address this, we have conducted additional experiments evaluating the performance of a representative MISO circuit in LB media and in an alternative *E. coli* strain (BL21) under both LB and minimal media conditions. These experiments confirm that T-Pro circuits are highly context-dependent, not only in terms of genetic architecture but also with respect to environmental factors such as nutrient availability and metabolic state. The observed differences in circuit performance highlight the need for recalibration when transitioning circuits across different cellular contexts.

This experimental assessment is now part of the main text results with a corresponding figure, in addition to an expanded **Supplementary Note 1**. Main text excerpts and figure given below:

*“Additional context variables and relative circuit performance. In addition to genetic contexts, we posited that chassis cell type and nutrient conditions will also have an impact on circuit performance (Supplementary Note 1). To test this assertion, we evaluated two BUFFER gate circuits in nutrient rich media and in BL21 *E. coli* cells, relative to the reference system – i.e., 3.32 *E. coli* chassis cells in minimal media – see Fig. 6a-f. In other words, the genetic context (promoter, RBS, GOI, plasmid) remained constant while the environment was changed. Qualitatively, the BUFFER operations performed as expected in all environments; however, quantitatively there were moderate differences in circuit performance upon change in chassis cell and nutrient conditions. Given the performance of the two SISO we were able to accurately predict the quantitative performance of the cognate MISO (i.e., BUFFERs to AND gate) substantiating our engineering workflow and central claim regarding improved circuit composability upon reconciliation of context variables (Fig. 6g-i and Supplementary Fig. 13).*

To further investigate the origin of how variation in growth media and chassis cell type impact circuit performance we reevaluated RBS strength under different environmental contexts (**Fig. 6j**). In brief, in all cases variation in chassis cell and growth media impacted RBS performance. This observation implies that RBS lookup table recalibration in the context of the selected chassis cell and growth media will be necessary to facilitate quantitative predictions; otherwise, our workflows and cognate models remain intact.”

Figure 6 (excerpt)



Comment 6: The Figure 7 section, which focuses on tuning expression levels of metabolic pathway genes, appears misaligned with the core focus of the paper. The authors primarily screen RBSs to match target expression levels, which is not fundamentally about genetic circuit design, but rather expression level optimization. Since the main claim of the paper is about predictable genetic circuit design, not empirical tuning of metabolic pathways, removing this section would improve the focus of the manuscript.

Response to Comment 6: We appreciate the reviewer's perspective on the relevance of Figure 7 to the core focus of the manuscript. The goal of this section is to demonstrate the versatility of the CSEC framework beyond logic circuits by showing its applicability to predict expression levels for diverse gene types, including metabolic pathway components. Importantly, this also highlights the ability of CSEC to accurately model gene expression for both individually expressed genes and genes organized within operon structures. Operons introduce additional complexity due to transcriptional coupling effects, and demonstrating predictive accuracy in these contexts provides a more rigorous test of the framework's generalizability. This section also directly addresses Comment 5 made by the reviewer, by further validating CSEC's predictive capabilities across different genetic and environmental contexts.

Additionally, this section highlights how T-Pro transcription factors can be utilized not only for genetic logic circuits but also for metabolic pathway control, expanding the potential applications of our framework. Given the increasing interest in using synthetic circuits for dynamic metabolic regulation, we believe this example strengthens the impact of our work by illustrating how our design principles can be extended beyond strict Boolean logic implementation.

Comment 7: The authors report <1.4-fold prediction error but do not specify how prediction accuracy varies across different types of circuits. Are some circuit topologies harder to predict than others?

Response to Comment 7: Done. We have prepared an SI Figure (**Supplementary Fig. 10**) with a summary of the direct statistical correlation analysis for SISO and MISO and we have prepared a detailed analysis spreadsheet to substantiate the "average error below 1.4-fold" claim summarized in the **Source Data File** – see the Tab labeled "**Supplementary Fig. 10**". In addition, we now report a complexity analysis in the SI – see **Supplementary Note 7** – which directly addresses the comment "*Are some circuit topologies harder to predict than others?*"

Excerpts below:

"Supplementary Fig. 10| SISO and MISO statistical correlation analysis. a. Predicted versus measured expression 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 expression 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 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 data are provided as a Source Data file."

"Supplementary Note 7: Complexity analysis: The search space size grows as $! (\#!)$, where $\#$ is a constant determined by the number of alternate DNA-binding functions (ADRs), regulatory control

domains (RCDs), and phenotypes, and % is the number of circuit inputs. Although the total size of the search space is $2^{\#}$, 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 $\sim 10^{\#}$ 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. **Table below** summarizes the time and memory usage required to enumerate circuits under specific design space configurations, which collectively yielded 246 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.”

Design space configuration

# 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

Comment 7: The authors should consider how their circuit design pipeline could be integrated into existing synthetic biology tools. How easily can non-specialists use this algorithm for circuit design? Could they share their engineered transcription factors and genetic parts through Addgene? Could they provide a user-friendly modeling software interface? Their current GitHub repository is not user-friendly. Improving documentation and usability would increase adoption of their approach.

Response to Comment 7: We agree that accessibility is key to increasing the adoption of our circuit design pipeline. To this end, we have prepared a detail workflow now given in **Supplementary Fig. 2**. In addition, we have prepared a complementary Jupyter Notebook that includes step-by-step instructions and example input data to demonstrate how to use the enumeration algorithm etc. for circuit design. This notebook is available in the updated GitHub repository and is intended to make implementation straightforward, even for users with limited computational background. <https://github.com/Jayaos/TPro>.

We also recognize the importance of integrating our approach with existing synthetic biology tools, and we are currently discussing this as part of our future work. Finally, we have deposited all engineered transcription factors and plasmids in GenBank (accession numbers provided in the main text) and included all sequence information and plasmid names as a supplementary data file – see “Parts and Plasmids” tab.

Comment 8: The study relies on synthetic transcription factors responsive to IPTG, ribose, and cellobiose. Have the authors considered expanding this approach to transcription factors that respond to biologically relevant signals?

Response to Comment 8: YES – we have an entire suite of synthetic transcription factors either developed or in the process of being developed. We have already engineered sets of T-Pro biosensors that are responsive to biologically relevant signals. For example, previous work from the lab has demonstrated the feasibility of engineering T-Pro transcription factors to recognize signal such as caffeine, and small molecule drugs like theophylline (DOI: 10.1021/acssynbio.0c00582). In addition, we are currently preparing two new manuscripts that demonstrate T-Pro biosensors that respond to other “biologically relevant signals” that we plan to publish later this year (or early next year).

Below is a point-by-point response to each of the reviewer's comments:

Comment Reviewer #1: The code is freely available on GitHub and examples are provided to demonstrate its use. The README file provides instructions to install the software.

Response to Reviewer 1: We would like to thank the reviewer for confirming the software – no revisions required.

Comment Reviewer #2: The authors have successfully and thoroughly addressed my original concerns. While the predictive framework remains context-dependent and requires recalibration when applied across different strains and media, the authors have addressed this limitation by demonstrating its adaptability through new experiments. They clearly explain the need for context-specific lookup tables and provide a systematic approach to generating them. I consider this an acceptable and well-acknowledged limitation in synthetic biology, and the presented method represents a significant advance in circuit design methodology. I consider this revised manuscript suitable for publication in Nature Communications.

Response to Reviewer 2: We would like to thank the reviewer for the recommendation for publication – no revisions required.

Comment Reviewer #3: The authors have thoroughly addressed all of my previous comments. The revised manuscript is substantially improved, with clearer explanations, expanded methodological details, additional experimental validation, and improved data and code availability. I have no further comments. I recommend the manuscript for publication in its current form.

Response to Reviewer 3: We would like to thank the reviewer for the recommendation for publication – no revisions required.