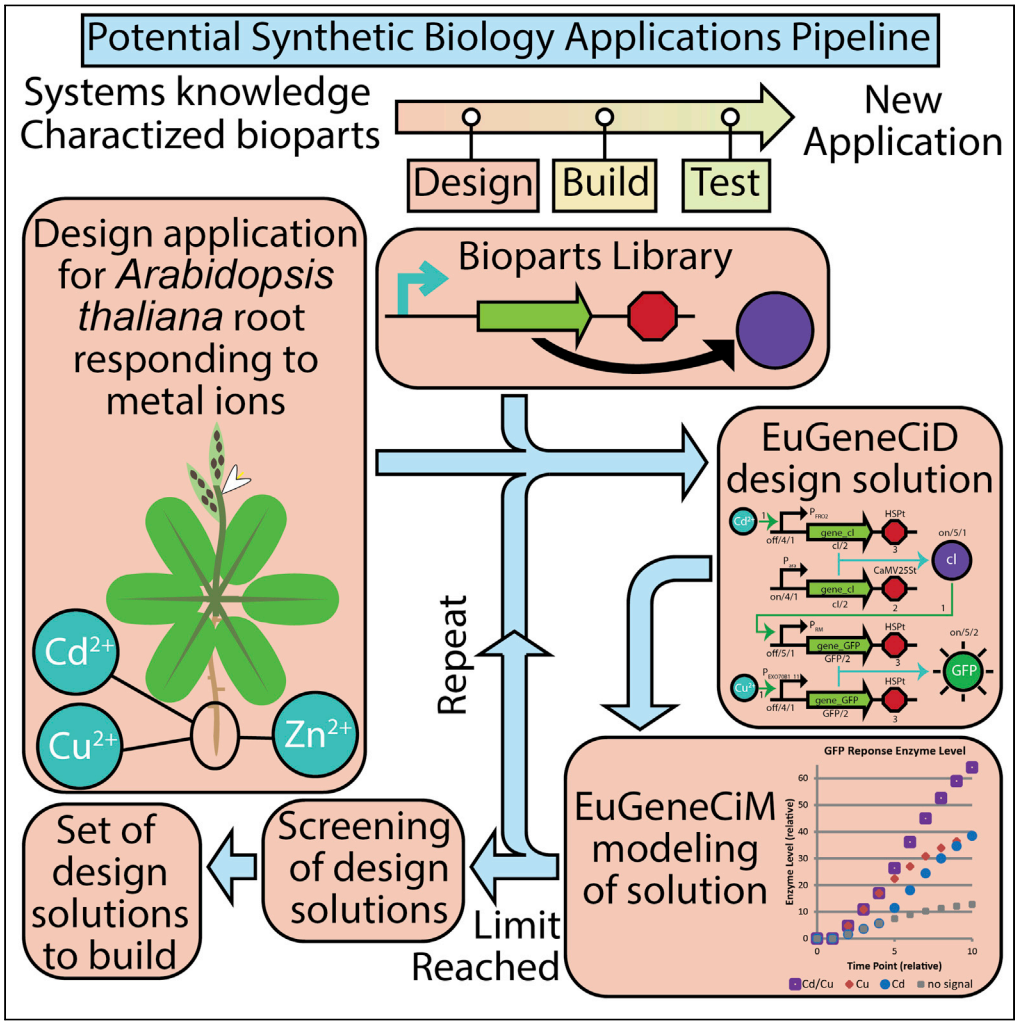
iScience

CellPress
OPEN ACCESS

Article

# Optimization-based Eukaryotic Genetic Circuit Design (EuGeneCiD) and modeling (EuGeneCiM) tools: Computational approach to synthetic biology




Wheaton L. Schroeder, Anna S. Baber, Rajib Saha

rsaha2@unl.edu

## Highlights

An *in silico* Eukaryotic Genetic Circuit Design (EuGeneCiD) tool is introduced

A complimentary Eukaryotic Genetic Circuit Modeling (EuGeneCiM) tool is developed

In a unified workflow, these tools generated thousands of designs and modeled them

The EuGeneCiM tool is also used to model a dynamic repressilator circuit

Schroeder et al., iScience *24*, 103000
September 24, 2021 
https://doi.org/10.1016/j.isci.2021.103000

iScience

CellPress
OPEN ACCESS

Article

# Optimization-based Eukaryotic Genetic Circuit Design (EuGeneCiD) and modeling (EuGeneCiM) tools: Computational approach to synthetic biology

Wheaton L. Schroeder,[1,2] Anna S. Baber,[2,3] and Rajib Saha[1,2,4,*]

## SUMMARY

**Synthetic biology has the potential to revolutionize the biotech industry and our everyday lives and is already making an impact. Developing synthetic biology applications requires several steps including design and modeling efforts which may be performed by *in silico* tools. In this work, we have developed two such tools, Eukaryotic Genetic Circuit Design (EuGeneCiD) and Modeling (EuGeneCiM), which use optimization concepts and bioparts including promotors, transcripts, and terminators in designing and modeling genetic circuits. EuGeneCiD and EuGeneCiM preclude problematic designs leading to future synthetic biology application development pipelines. EuGeneCiD and EuGeneCiM are applied to developing 30 basic logic gates as genetic circuit conceptualizations which respond to heavy metal ions pairs as input signals for *Arabidopsis thaliana*. For each conceptualization, hundreds of potential solutions were designed and modeled. Demonstrating its time-dependence and the importance of including enzyme and transcript degradation in modeling, EuGeneCiM is used to model a repressilator circuit.**

## INTRODUCTION

Synthetic biology is the design of living systems, utilizing engineering principles, to accomplish a desired task or purpose (Khalil and Collins, 2010). To date, applications include novel biochemical synthesis pathways and many biological analogs of electronic circuits such as logic gates, sensors, toggles, oscillators, and switches (Khalil and Collins, 2010; Kim and Winfree, 2011; Liu and Stewart, 2015; Scheller et al., 2020) with a long term goal of programmable biology (Xia et al., 2019). Commercial products which are the result of applications of synthetic biology are emerging in restaurants (the Impossible Burger), pharmacies (Januvia indicated for diabetes), electronics (Hyaline used in foldable smartphones), and hospitals (Kymriah, a cell-based therapy indicated for B-cell acute lymphoblastic leukemia) highlighting the emerging roles of synthetic biology throughout society (Voigt, 2020). Therefore, the tools which aid in the development of novel synthetic biology applications will be of both scientific and commercial value to accelerate the development of new applications. There are five major stages in the development of a new synthetic biology application: Conceptualization; design modeling; construction; probing, testing, and validation (Liu and Stewart, 2015). As a first step toward creating a synthetic biology application pipeline from a user-defined conceptualization, the design and modeling steps of this workflow will be explicitly linked in this work using two novel deterministic *in silico* optimization-based tools which can be largely automated.

Particularly in the design of new applications, synthetic biology often relies on the intuition of biologists and engineers; their knowledge of available promotors, genes, terminators, transcripts, enzymes, and proteins (collectively, bioparts) and the associated systems; and their design ability to create new applications. This approach is generally limited to system experts and to designs which are intuitive. Alternatively, a computational model-driven approach is advantageous in that it allows for non-intuitive designs and the quick *in silico* screening thereof, so that only designs with the greatest chance of success are constructed. Several design and modeling tools exist, such as Cello 2.0 (Chen et al., 2020), OptCircuit (Dasika and Maranas, 2008), the work of Zomorrodi and Maranas (2014) (the tool was unnamed), EQuIP (Davidsohn et al., 2015), SynBioSS (Hill et al., 2008), and several others which may be adapted to various systems and to screening of genetic circuits (Liu and Stewart, 2015). Figure 1 summarizes the unique approach to the problem of design along with advantages and disadvantages of each of these tools within the context of

[1]Department of Chemical and Biomolecular Engineering, University of Nebraska – Lincoln, Lincoln, NE 68588, USA

[2]Center for Root and Rhizobiome Innovation, University of Nebraska – Lincoln, Lincoln, NE 68588, USA

[3]Department of Biomedical Engineering, University of Rochester, Rochester, NY 14627, USA

[4]Lead contact

*Correspondence: rsaha2@unl.edu

https://doi.org/10.1016/j.isci.2021.103000

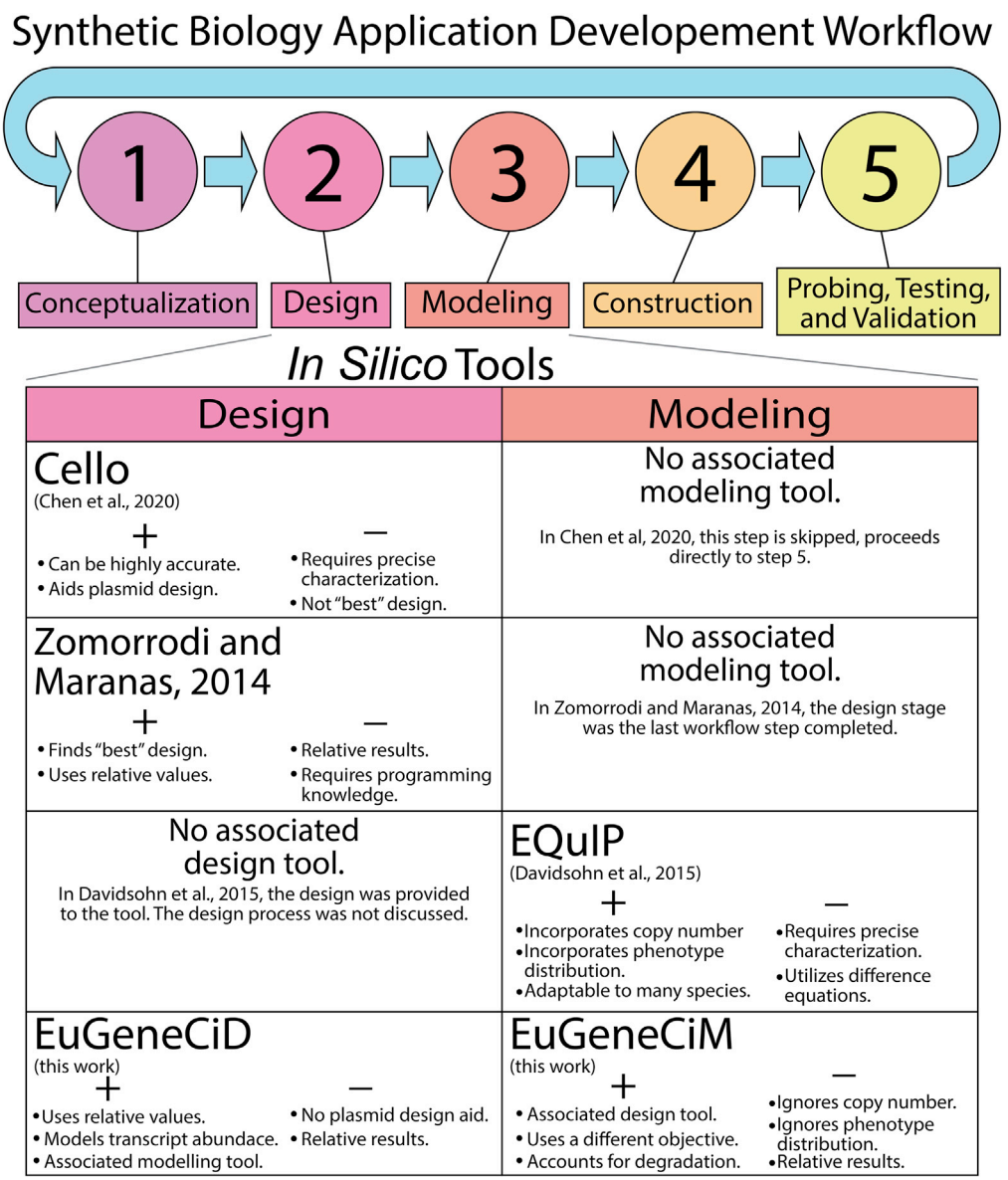



**Figure 1. Steps of synthetic biology application development and some *in silico* tools**

Synthetic biology applications generally have five steps: Conceptualization, design, modeling, construction, and probing, testing, and validation. Of these steps, three can be performed *in silico*. Several independent design and modeling tools exist for the second and third stages of this workflow, including Cello, the work of Zomorrodi and Maranas (2014) (in addition to their previous OptCircuit), and EQuIP. Introduced here are the EuGeneCiD and EuGeneCiM tools which integrate the design and modeling steps as design solutions are passed from EuGeneCiD to be modeled by EuGeneCiM. For the listed tools, a short list of strengths and weaknesses is included to help better position this work in the context of the current state of the field.

developing synthetic biology applications. Although these tools have successfully designed or simulated behaviors replicated *in vivo*, the most overarching challenge associated with these tools is their specialization for design or modeling tasks with no clear workflow or method by which to link the two activities. This is highlighted in that some design tools, such as Cello 2.0, published synthetic biology workflows which skip the modeling step altogether and used more expensive and time-consuming *in vivo* screening processes (Borujeni et al., 2020). A particularly difficult problem in current optimization-based design tools such as Zomorrodi and Maranas (2014), and OptCircuit (Dasika and Maranas, 2008) are Bistable Orthogonal Designs (BODs). These produced design solutions that would not function as desired. For instance, consider

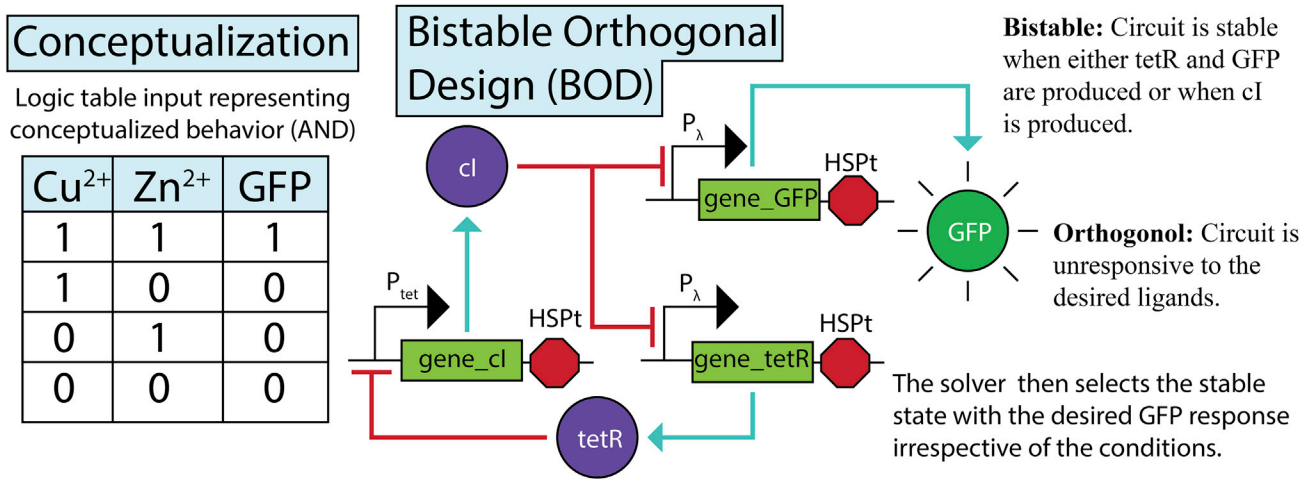



**Figure 2. Example bistable orthogonal design (BOD)**

This figure illustrates a major category of problematic potential designs which may be produced by optimization-based genetic circuit design tools. From a conceptualized $Cu^{2+}/Zn^{2+}$ responsive AND circuit, it is possible, without attribution equations, to create Bistable Orthoganol Design (BOD) which can produce the desired response, yet not be responding to the desired signals. Text in the image describes why this occurs. One of the major innovations in EuGeneCiD is the development of attribution equations to avoid BODs.

the example shown in Figure 2, where it is desired to produce a circuit with an AND response to copper and zinc ions using a GFP reporter. Using only a handful of parts, it is possible to produce a circuit with two stable states (where both tetR and GFP are produced or only cI is produced). Further, these two stable states are independent of (or orthogonal to) the signals which the circuit should respond (e.g., the copper and zinc ions). For such a BOD, a solver might then pick whichever state is necessary to match the desired conceptualized circuit behavior irrespective of the conditions, rendering the circuit effectively useless for the proposed application. These BODs are technically correct solutions to the conventional optimization-based tools but require further manual scrutiny to identify and remove these problematic solutions. When producing large numbers of solutions, BODs generally outnumber true designs and can overwhelm a researcher's ability to screen.

One promising area for synthetic biology applications is in plants, particularly commercially important crops such as maize (*Zea mays*), rice (*Oryza sativa*), and barley (*Hordeum vulgare*). Applications in plants include increasing nutrient content (Beyer et al., 2002; Gonzali et al., 2009), synthesizing novel chemicals (Liu and Stewart, 2015; Mortimer, 2019), improving crop resilience (Pixley et al., 2019), and synthetic sensors (Liu and Stewart, 2015). Here, we have chosen to demonstrate our novel tools using the model plant species *Arabidopsis thaliana* (hereafter, *Arabidopsis*) because it is well studied and has been used for many synthetic biology applications (Holland and Jez, 2018). We have further chosen to design and model plant-based synthetic sensors of heavy metal in the root of *Arabidopsis*. Heavy metal pollution occurs as a result of human activities (such as mining or manufacturing), and is toxic to living organisms at sufficient concentrations, even essential elements such as Zinc. These metal ions can enter the soil via several possible routes including from water and the air (Vardhan et al., 2019; Vareda et al., 2019). Three of the most common heavy metal pollutants are Copper, Cadmium, and Zinc (Vardhan et al., 2019), to which *Arabidopsis* has some natural response mechanisms. By creating reporter systems which respond to these heavy metal ions, it may be possible in the future to develop synthetic biology applications in crop species for metal ion removal or mitigation from contaminated soils through phytoremediation (Jacob et al., 2018). Different logical combinations of present ions might require different phytoremediation strategies; therefore, the construction of logic gates responding metal ion signals would be a logical first step in the long-term development of these strategies and applications.

For developing a combined design and modeling workflow, in this work, we developed two deterministic optimization-based tools, namely the Eukaryotic Genetic Circuit Design (EuGeneCiD) and Modeling (EuGeneCiM) tools, which utilize an input of the conceptualized circuit behavior and perform an automated simulation of the optimal and suboptimal circuit designs for manual screening. EuGeneCiD provides one key improvement upon previous optimization-based tools (Zomorrodi and Maranas, 2014; Dasika

and Maranas, 2008) by developing constraints (called the attribution constraints) which precludes BODs. In addition, several other distinct differences and improvements distinguish the EuGeneCiD tool from either of these previous works. First, EuGeneCiD is designed for eukaryotic systems where Ribosome Binding Sites (RBSs) are not a critical design element, but replaces such elements with terminators which are important in eukaryotic gene expression, particularly for plants (de Felippes et al., 2020; Nagaya et al., 2010). Second, the rate of mRNA and protein degradation on circuit behavior is incorporated, which leads to new design possibilities. Third, the tool was made more granular so that concentration values are not always integer values. Fourth, the layers of the central dogma (transcription and translation) are mathematically separated so that, aside from relative concentration levels, relative levels of mRNA for genes might also be designed and simulated. EuGeneCiM takes these unique elements and, utilizing a design passed from EuGeneCiD, simulates circuit behavior over a given number of hypothetical time points, which will allow for screening of circuit behavior before constructing these proposed synthetic biology applications.

Using bioparts, which are either a part of natural *Arabidopsis* heavy metal response mechanisms, or shown to function in *Arabidopsis* from other species, and fluorescent proteins as state reporters, EuGeneCiD is applied to developing these synthetic heavy metal sensors in *Arabidopsis*. EuGeneCiD was used to create design solutions for 30 different genetic circuits formed from combining nine unique two-input logic gates with three different input signal pairs. These input signals are the presence of Cadmium, Copper, and Zinc ions at high or toxic concentrations. For each genetic circuit conceptualization which was able to be designed from the given biopart library, EuGeneCiD generated hundreds of feasible solution designs, each with a corresponding dynamic simulation from EuGeneCiM. Aside from basic logic circuits, repressilators have also proven to be a useful control schema in synthetic biology, allowing for oscillating gene expression (English et al., 2021). Therefore, EuGeneCiM is used to model the dynamic behavior of a repressilator circuit to demonstrate its utility as a stand-alone dynamic modeling tool and the value of incorporating mRNA and protein degradation in modeling efforts. Together, the EuGeneCiD and EuGeneCiM tools can hypothesize genetic circuit designs and simulate their behavior to increase the chances that a plant might have the desired behavior when transformed, potentially saving time and resources. This work could be the basis for the development of a synthetic biology application pipeline. Therefore, for the ease of use and the facilitation of this pipeline, various programs have been developed to make EuGeneCiD and EuGeneCiM user-friendly, and a related protocol paper on the use of these tools will be published to accompany this work. Further, the design solutions produced here could form the basis of future heavy metal phytoremediation applications of synthetic biology particularly in important crops like *Zea mays* (maize). Maize has been identified as both Cadmium tolerant (Rizwan et al., 2017) and as a Cadmium hyperaccumulator (Wuana and Okieimen, 2010), and is already used for heavy metal phytoremediation (Rizwan et al., 2017). Additionally, maize has been identified as a bioaccumulator of both Zinc and Copper (Sekara et al., 2005; Wuana and Okieimen, 2010). From this, maize is already particularly well suited for phytoremediation applications, and could be engineered through synthetic biology to be superb, solving multiple problems at a stroke by providing food from otherwise toxic farmland while cleansing it of heavy metal ions toxic to both humans and other plants.

## RESULTS

### Selection of test system and synthetic biology conceptualizations

*Arabidopsis* was chosen as the test system for the development and subsequent application of the EuGeneCiD tool since it is a model plant system to which systems biology has often been applied (Holland and Jez, 2018). It was decided to develop heavy metal ion biosensors in the *Arabidopsis* root, which would report sensor state using fluorescent proteins. A plant system, in particular, was chosen for this work because in the future EuGeneCiD and EuGeneCiM will be applied to plants of biotechnological and agronomic importance (e.g., *Zea mays*) for various applications related to plant health and fitness, potentially including phytoremediation of heavy metal pollution. Since phytoremediation strategies may change depending on the metal ion(s) present, basic logic gates are conceptualized here which report on the presence or absence of the metal ions.

### Development of the Eukaryotic Genetic Circuit Design (EuGeneCiD) tool

EuGeneCiD was conceived and developed to address the limitation of the current state-of-the-art optimization-based design tools for synthetic biology applications (Zomorrodi and Maranas, 2014; Dasika and Maranas, 2008). Particularly, by changing the focus to eukaryotic systems, allowing granularity, modeling transcript abundance, adding terminators as a design element (which are particularly important in plant

synthetic biology), and creating the attribution constraints. The initial EuGeneCiD formulation was inspired by other optimization-based circuit design works (Ali R Zomorrodi and Maranas, 2014; Dasika and Maranas, 2008) and was formulated specifically to apply to eukaryotic systems and incorporate biopart degradation. This involved using terminators, as opposed to RBSs, as part of the design; incorporating mRNA and protein degradation; having a more granular values of concentration; and reporting relative mRNA abundance for particular genes. Attempts were made to incorporate time to make EuGeneCiD a dynamic design tool. This would influence various design variables, such as concentration, yet this proved computationally intractable and was abandoned. At this stage in development, it was decided to separate the formulations of design and modeling tools. When applying this first version of the EuGeneCiD tool to a modest sized biopart database, the issue of BODs became apparent and pressing. The final stages of the development of EuGeneCiD involved the creation of the attribution constraints to prevent BODs. These attribution constraints account for a high fraction of all constraints (about 42%) and variables (about 42%) in the formulation of EuGeneCiD and thus account for a fair amount of the tools' computational expense. This tradeoff is considered worthwhile in that it allows for the preclusion of BOD solutions which can account for greater than 90% of solutions in some instances when the attribution equations are not included. The final formulation of EuGeneCiD is a Mixed Integer Linear Programming (MILP) problem, with a single-level objective function maximizing the concentration of desired enzymes and minimizing that of undesired enzymes. Initial testing of both EuGeneCiD and EuGeneCiM was conducted using hypothetical bioparts, details of which are provided in the GitHub repository associated with this work (github.com/ssbio/EuGeneCiDM or https://doi.org/10.5281/zenodo.4762590). The final formulation has over three dozen constraints and variables which are detailed in the STAR Methods section.

### Development of the Eukaryotic Genetic Circuit Modeling (EuGeneCiM) tool

EuGeneCiM was conceived and developed to address the lack to optimization-based tools for the modeling of proposed synthetic biology application designs, particularly one which might readily be passed designs for screening. As previously stated, EuGeneCiM initial development began when it was noticed that including time-based simulations inside the EuGeneCiD tool was computationally intractable. EuGeneCiM is similar to EuGeneCiD in formulation with three major exceptions. First, the design variable is made a parameter in EuGeneCiM as these values are passed from an optimal or suboptimal solution of EuGeneCiM. Second, as EuGeneCiM does not design, the attribution constraints are unnecessary and therefore unused, thus considerably boosting solution speed. Third, as the design is not variable, this allows certain simplifications in the formulation. The final formulation has approximately two dozen constraints and variables which are detailed in the STAR Methods section.

### Initial testing of the EuGeneCiD and EuGeneCiM tools

Initial testing of both EuGeneCiD and EuGeneCiM was conducted using a test bioparts database and test codes, provided in the GitHub repository associated with this work (github.com/ssbio/EuGeneCiDM). This test bioparts library consists of 33 promotors, 13 transcripts, 10 terminators, and 13 proteins and enzymes. Versions of the EuGeneCiD and EuGeneCiM workflow were created which allow for one-, two-, and three-input circuit designs. These circuits include the following logics: ADDER (three inputs), AND (two inputs), BUFFER (one input), HALF ADDER (two inputs), NAND (two inputs), NOR (two inputs), NOT (one input), OR (two inputs), XNOR (two inputs), and XOR (two inputs). Through these tests, numerical issues such as BODs were discovered. The final EuGeneCiD and EuGeneCiM workflow was not applied to these test circuits and database, though these test applications show that EuGeneCiD and EuGeneCiM can be adapted to circuits with other than two input signals.

### Definition of the bioparts database

Following the creation and initial testing of the EuGeneCiD and EuGeneCiM tools, a database of real bioparts was created for the design of genetic circuits which respond to Cadmium ($Cd^{2+}$), Copper ($Cu^{2+}$), or Zinc ($Zn^{2+}$) ions, or combinations thereof to design and simulate various logic gates. Note that bioparts which are responsive to the metal ions do not directly respond to those ions, but rather make use of the native metal sensing or signaling pathways of *Arabidopsis* and are bioparts whose activity is affected by these signaling pathways. This approach is used because it was decided that it would be too complex to introduce the various signal pathways in a target organism with each design. Promotors included in the biopart database are shown in Figure 3. Details on the sources for these bioparts, their parameterization, and their reason for inclusion in the database can be found in Table S1.

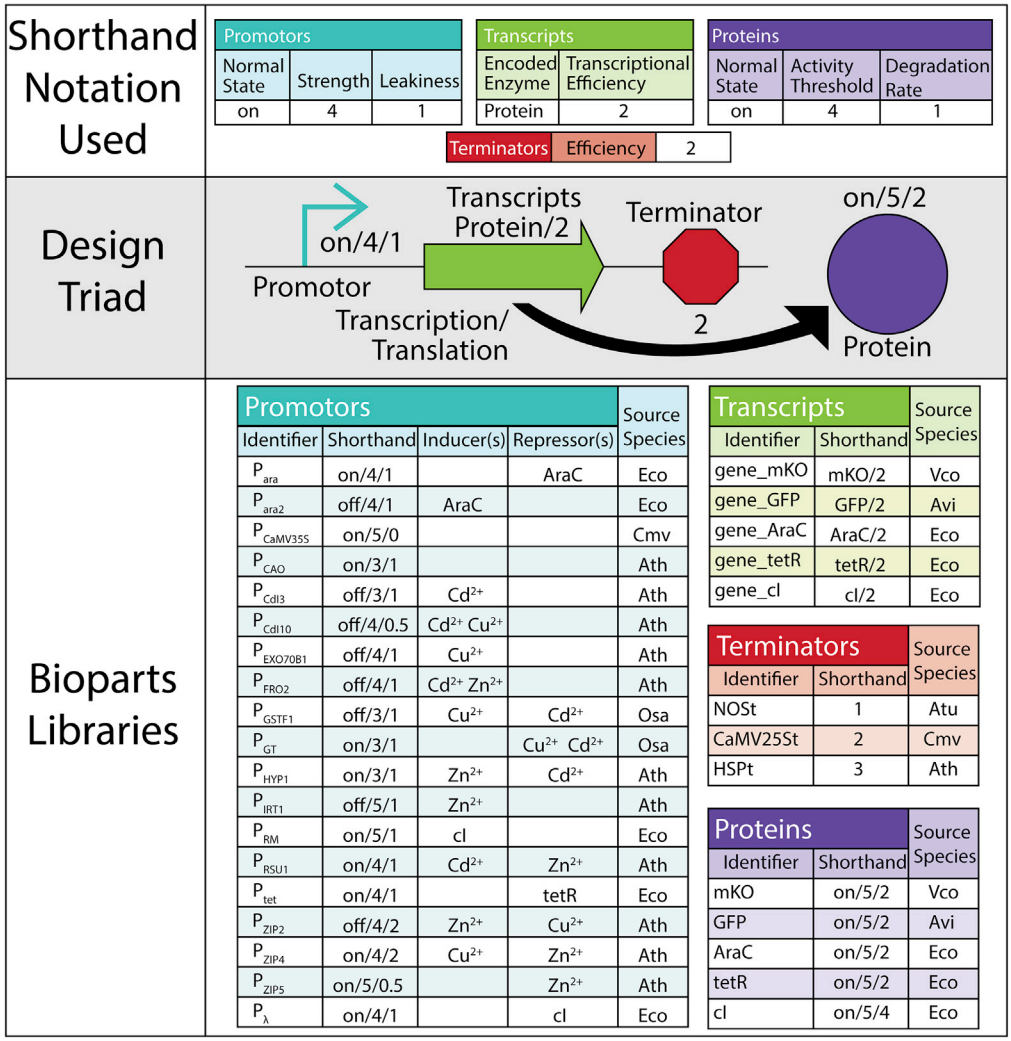



**Figure 3. Bioparts database for the current work**

The EuGeneCiD and EuGeneCiM Tools designed require the definition of bioparts databases from which to pick design elements and to define the properties of those elements for both design and modeling. For compactness in other images, introduced here is a shorthand for promotor, transcript, terminator, and protein characteristics. The shorthand here is then used to define each biopart included in the bioparts library used for this work, which includes promotors, transcripts, terminators, and proteins. Source species acronyms for listed bioparts are as follows: Ath –*Arabidopsis thaliana*, Osa –*Oryzae sativa*, Eco –*Escherichia coli*, Vco –*Verrillofungia coninna*, Avi –*Aequorea victoria*, Atu–*Agrovacterium tumefaciens*, Cmv–Califlower Mosaic Virus.

## Application of EuGeneCiD and EuGeneCiM

The EuGeneCiD and EuGeneCiM tools are embedded in the workflow shown in Figure 4. In summary, this workflow uses the bioparts library and the synthetic biology application conceptualization as inputs from which the EuGeneCiD problem is attempted. Should a solution be found, EuGeneCiM is solved across several time points to model the designed circuit. If a solution is not found, there are two possibilities: all possible designs with the specified parameters (primarily circuit size) have been identified, or that all possible designs have been identified which are smaller than some maximum allowed circuit size. In the former case, the size of the sought design is incremented, and EuGeneCiD is attempted again. Otherwise, the selection of designs is returned, and the user may select a design from the design and modeling information. For greater details, see STAR Methods.

To demonstrate the utility of EuGeneCiD and EuGeneCiM tools, it was decided to use these tools to design and model 30 unique genetic circuit conceptualizations using the defined real bioparts database. Each

. Continued

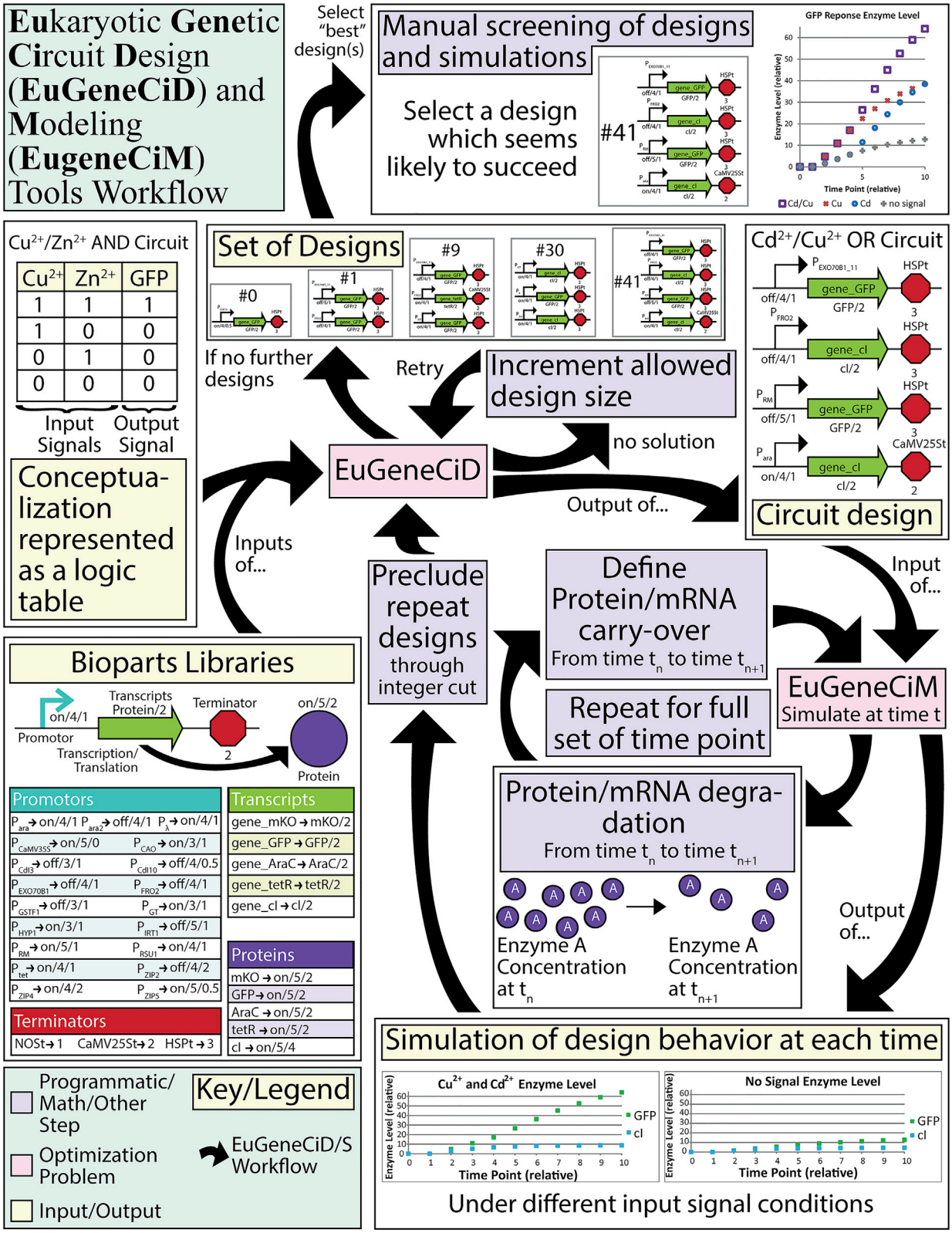

Eukaryotic Genetic Circuit Design (EuGeneCiD) and Modeling (EugeneCiM) Tools Workflow
Select "best" design(s)
Manual screening of designs and simulations
Select a design which seems likely to succeed
#41
GFP Reponse Enzyme Level
Enzyme Level (relative)
Time Point (relative)
Cd/Cu
Cu
Cd
no signal
Cu2+/Zn2+ AND Circuit
Cu2+ Zn2+ GFP
1 1 1
1 0 0
0 1 0
0 0 0
Input Signals
Output Signal
Conceptua-lization represented as a logic table
Set of Designs
#0
#1
#9
#30
#41
If no further designs
Retry
Increment allowed design size
EuGeneCiD
no solution
Output of...
Inputs of...
Cd2+/Cu2+ OR Circuit
Circuit design
Input of...
Preclude repeat designs through integer cut
Define Protein/mRNA carry-over
From time tn to time tn+1
EuGeneCiM
Simulate at time t
Repeat for full set of time point
Bioparts Libraries
Promotor
Transcription/ Translation
Transcripts Protein/2
Terminator
Protein
Promotors
Transcripts
Terminators
NOSt→1 CaMV25St→2 HSPt→3
Proteins
Protein/mRNA degra-dation
From time tn to time tn+1
Enzyme A Concentration at tn
Enzyme A Concentration at tn+1
Output of...
Simulation of design behavior at each time
Cu2+ and Cd2+ Enzyme Level
No Signal Enzyme Level
GFP
cl
Under different input signal conditions
Key/Legend
Programmatic/ Math/Other Step
Optimization Problem
Input/Output
EuGeneCiD/S Workflow

**Figure 4. Workflow of the EuGeneCiD and EuGeneCiMtools**

The EuGeneCiD and EuGeneCiM tools were designed to be used in concert to complete the design and modeling steps of synthetic biology applications development together. This workflow begins with a defined conceptualization of the application (in the form of a logic table) and a bioparts library which defines and describes potential design elements (see Figure 2). Then an attempt to solve EuGeneCiD is made, with three possible outcomes. First, no solution is found at the current design size limit (limiting the number of allowed triads), in which case this limit is incremented, and EuGeneCiD is attempted again. Should design or run limits be reached, or if no further designs exist within specified restrictions, the set of designs is returned which can be manually screened for candidates likely to succeed. Should the attempt to solve EuGeneCiD be successful, a circuit design is the result, which is passed to EuGeneCiM for modeling. This modeling solves EuGeneCiM at each time point and applies protein and transcript degradation between time points for the full set of desired model time points. This results in a simulation of design behavior at each time point which will be reported. The current solution is then precluded by defining a new integer cut and the cycle is repeated.

conceptualization will have its own input file; an example is provided in Table S2, containing all information from Table S1 in addition to a logic table, and a parameter specifying the number of time points to model. These unique conceptualizations were defined both by the logic circuit and the ligand pair to which that circuit is to respond. The logic circuits to which EuGeneCiD and EuGeneCiM were used to design and model include BUFFER (also known as a toggle circuit), AND, NIMPLY, converse non-implication (abbreviated CNI), HALF ADDER, NAND, NOR, OR, XNOR, and XOR. Note that CNI is included because it is logically equivalent to NIMPLY with a reversed ligand order. Further, this study does not purport to study all possible or useful logic gates, but rather these 30 conceptualizations will show the usefulness of the EuGeneCiD and EuGeneCiM workflow to apply to a variety of genetic circuit conceptualizations. Divalent heavy metal ion pairs, representing common heavy metal pollutants (Vardhan et al., 2019), were selected to serve as the signals for the logic gates by their presence or absence. The metal ion signal pairs used are Cadmium and Copper; Cadmium and Zinc; and Copper and Zinc. The number line shown in Figure 5 shows each combination of metal ion signal and logic gate.

It should be noted that the applications of EuGeneCiD and EuGeneCiM when applied to the real bioparts database do not make full use of the in-built capabilities of these algorithms. First, these algorithms have the potential to consider alternative splicing, through definitions of the variable which maps transcripts to its encoded enzyme ($\rho_{je}$) and transcriptional efficiency ($\eta_j$). The former can be used to define more than one transcript-enzyme encoding relationships and the latter can be lowered to reflect fractions of transcript being used to encode each alternative splice. In addition, the capability exists for enzymes to be regulated by environmental cues and other enzymes. These capabilities are not exploited in this application because it was desired to apply these tools to a plant system, and *Arabidopsis* appears to not have such sophisticated bioparts natively (at least for heavy metal signaling and resonse pathways), nor have such parts been engineered for *Arabidopsis*. However, these capabilities will function in the event that they are needed and defined in the input bioparts library, as these functions have been tested using the test database described earlier.

### *General EuGeneCiDsolution trends*

Several general trends emerge from the sets of solutions produced by EuGeneCiD and can be identified in Figure 5. First, as highlighted in Figure 5, using the given database, it appears that certain simpler logic gates such as BUFFER, AND, NIMPLY, NOR, and OR are easier to find design solutions for. This is indicated by high numbers of solutions after the seven day run time, short solution times (minimum, average, and maximum), and a large percentage of reported solutions being proven optimal solutions (as opposed to integer solutions which do not guarantee optimality). On the other hand, circuits such as XNOR, XOR, and HALF ADDER are generally more difficult to find design solutions as indicated by fewer solutions, longer solve time, and low percentage of reported solutions being proven optimal. For these circuits, the majority of solutions are integer solutions without proven local or global optimality. In addition, these more difficult circuits generally also have higher minimum and mode circuit sizes, as well as longer solution times. These circuits are also more likely to have been terminated by reaching the seven day time limit, as opposed to the easier circuits which were more likely to be terminated by reaching the maximum number of allowed solutions. As shown in Figure 5, more complex solutions generally require more triads (solution size is reported as the number of triads in the design) to achieve the desired logic.

A particularly interesting trend in EuGeneCiD solutions, shown in Figure 5, is that the maximum objective function value rarely occurs in the first solution, with the exception of the $Cu^{2+}$/$Zn^{2+}$ XOR and $Zn^{2+}$ BUFFER responsive circuits, though the minimum objective value sometimes occurs at this point. This can be for multiple reasons. The objective function is defined as the difference of response strength under desired

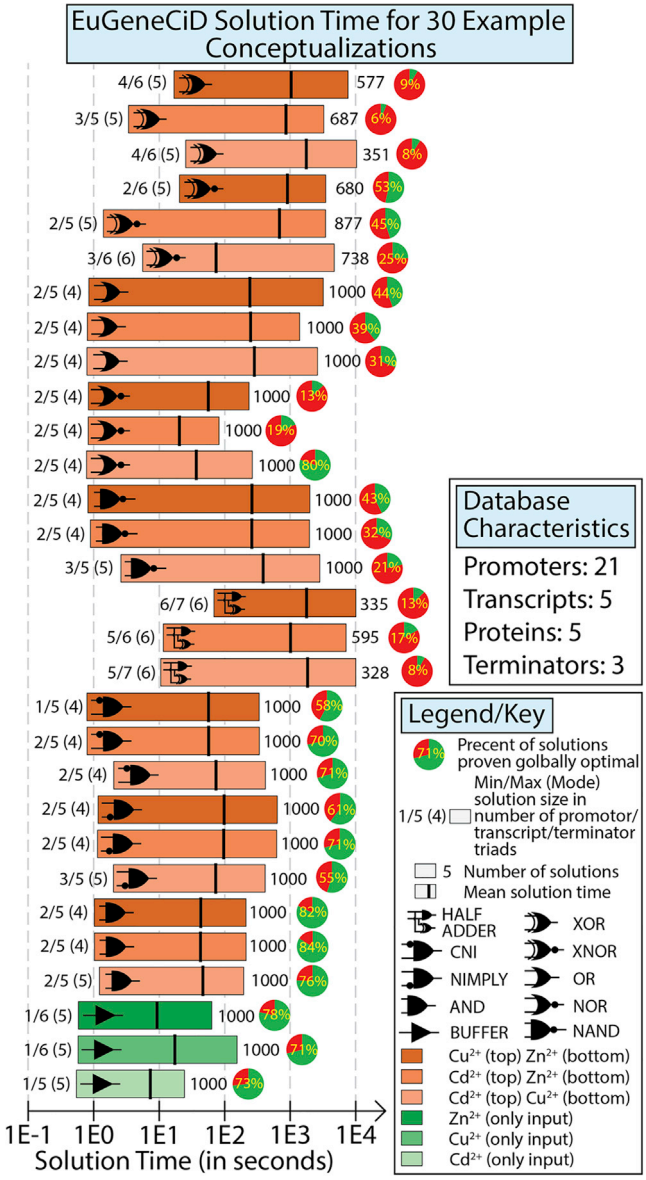



**Figure 5. Visualized holistic EuGeneCiDresults**

Thisfigure highlights several key result metrics of the application of EuGeneCiD to a large number of synthetic biology application conceptualizations seeking a large number of solutions. Each bar indicates the minimum, mean, and maximum solution time for specified inputs (indicated by color) and logic gate (indicated by the logic gate drawing on the bar). The number to the right of the bar indicates the number of solutions achieved, and the pie chart to the right of the bar indicates what percent (rounded to the nearest whole percent) of those solutions were proven to be globally optimal. To the left of the bar are given the solution size (in number of triads in each design) statistics, formatted as "min/max (mode)". Above the legend, which defines the symbols and colors used, is given the characteristics of the database used to derive these solution sets.

response conditions and response strength under undesired condition. This formulation ideally will favor solutions with strong responses and low expression leakiness. See STAR Methods for the mathematical formulation. The first possibility is that a biopart with this inherent function might be leaky or not particularly strong, yet would be the simplest possible solution. A second possibility is, due to the nature of the EuGeneCiD objective function, different circuit conceptualizations will have slightly different priorities in their optimal designs. In summary, depending on the sparsity of the response vector(s) in the input logic table, a slight favoritism for low leakiness of the response protein(s) or for a strong response pulse may be favored. A full discussion of this can be found in the STAR Methods.

### *Dissecting selected circuit designs*

This study produced a very large number of design and modeling results, more than 23,000 to be precise. This volume allows for analysis of the broad solution trends discussed while precluding the analysis of each individual solution. All solutions may be found in the associated GitHub repository (github.com/ssbio/EuGeneCiDM). Additionally, code is provided in the repository which will plot any given solution (see the provided documentation in GitHub). This code was used in part to generate the graphs in Figures 6 and S4 (showing example BUFFER solutions). By investigating several solutions using this code, we have selected three representative circuit results (two of which could be defined as likely to be successful and one is unlikely to be successful) as example results, shown in Figure 6. One general feature of interest in the EuGeneCiM tool can be seen in each of the modeling results graphs: the start-up time. EuGeneCiM essentially assumes that the genetic circuit is newly introduced into the target organism at time point 0; therefore, there is some delay (2 time points) between introduction of the circuit and the response of the circuit to environmental conditions. This delay is caused by the enforced delays in the EuGeneCiM algorithm. The first enforced delay is between transcription and translation (allowing for phenomena such as time for RNA processing and transport). The second is between translation and protein activity (allowing for phenomena like protein folder and localization). A second point of interest is that, while both tools use Mixed Integer Linear Programming, the curves produced are non-linear. This is because, in EuGeneCiM, the half-life based degradation of transcripts and proteins is calculated between time steps as a "carry over" value from one time point to the next (as shown in the workflow image Figure 4 and described in the STAR Methods).

The first successful example, solution #41 for a $Cd^{2+}/Cu^{2+}$ responsive AND circuit, is shown in the top third of Figure 6. Solution #41 was chosen as it is the $Cd^{2+}/Cu^{2+}$ responsive AND circuit with the maximum objective function value, likely due to the multiple gated encoding of GFP. This solution contains four triads (promotor/gene/terminator groupings which specify the circuit design): $P_{FRO2}$/gene_cI/HSPt, $P_{ara}$/gene_cI/CaMV25St, $P_{RM}$/gene_GFP/HSPt, and $P_{EXO70B1\_11}$/gene_GFP/HSPt. There are two responsive elements to the signal ions, promotors $P_{FRO2}$ (responding to $Cd^{2+}$) and $P_{EXO70B1\_11}$ (responding to $Cu^{2+}$). These then regulate the expression of GFP indirectly and directly, respectively. Note that while $P_{ara}$ is regulated by araC, because araC is not encoded, it will act like a constitutive promotor. Due to the short half-life of cI, this circuit maintains a constitutive pool of cI which is below the concentration threshold necessary for a cI-expressing phenotype unless $Cd^{2+}$ is present. This gates the expression of GFP from the $P_{RM}$/gene_GFP/HSPt triad, preventing GFP expression from this triad unless $Cd^{2+}$ is present. GFP expression induced by $Cu^{2+}$ is regulated directly. This causes the circuit to be quicker to respond to the presence of $Cu^{2+}$ than to $Cd^{2+}$ in the modeling results. The double-encoding of the GFP results in the significantly stronger response of the circuit to both conditions, than to a single condition. This is one potential drawback of the binary encoding of the conceptualization in that there is no mechanism to ensure equal expression in all cases where expression is desired, since phenotype is what is desired, rather than strength of that phenotype.

The second successful example, solution #11 of a $Cu^{2+}$ NIMPLY $Zn^{2+}$ circuit, is shown in the middle third of Figure 6 also uses cI as the desired control enzyme which gates expression of GFP. This circuit uses three triads in the design: $P_{GSTF1}$/gene_cI/HSPt, $P_{FDR3}$/gene_cI/NOSt, and $P_{RM}$/gene_GFP/HSPt. For controlling the expression of cI, a moderately strong promotor, $P_{FDR3}$ (which is repressed by $Zn^{2+}$), is paired with a relatively inefficient terminator NOSt, which results in a pool of cI transcripts which can quickly build or degrade in the absence or presence of $Zn^{2+}$ but which is not sufficient for cI-expression phenotype. The $P_{GSTF1}$/gene_cI/HSPt triad then is also a deciding factor in cI phenotype, encoding stable RNA (from an efficient terminator, HSPt) from a moderate promotor ($P_{GSTF1}$). This second promotor results in a slowly building yet stable pool of cI transcripts. When both triads produce cI, the concentration is high enough for cI expression. When cI is expressed, the very strong promotor $P_{RM}$ is activated, resulting in strong GFP expression. When modeled, this mixed approach to cI production (using from quick- and slow-accumulating pools of cI transcript) in combination with the sort half-life of cI results in a slow-responding circuit (only beginning to diverge from other conditions at time point 7), as expression from both triads is required. Yet, when cI is at sufficient concentration, the circuit responds very strongly. It is highly possible that the response strength would be greater than what is shown if the circuit were modeled for more time points. Theoretically, this circuit could be quickly "shut off" by lack of a $Cu^{2+}$ signal or especially the presence of a $Zn^{2+}$ signal. Due to the single-encoded gene_GFP, GFP expression is uniform and low in non-expressive conditions.

The provided unsuccessful solution is solution #26 for a $Cd^{2+}/Zn^{2+}$ responsive NAND circuit, shown in the bottom third of Figure 6. As with the previous example, three triads are used, two of which gate the

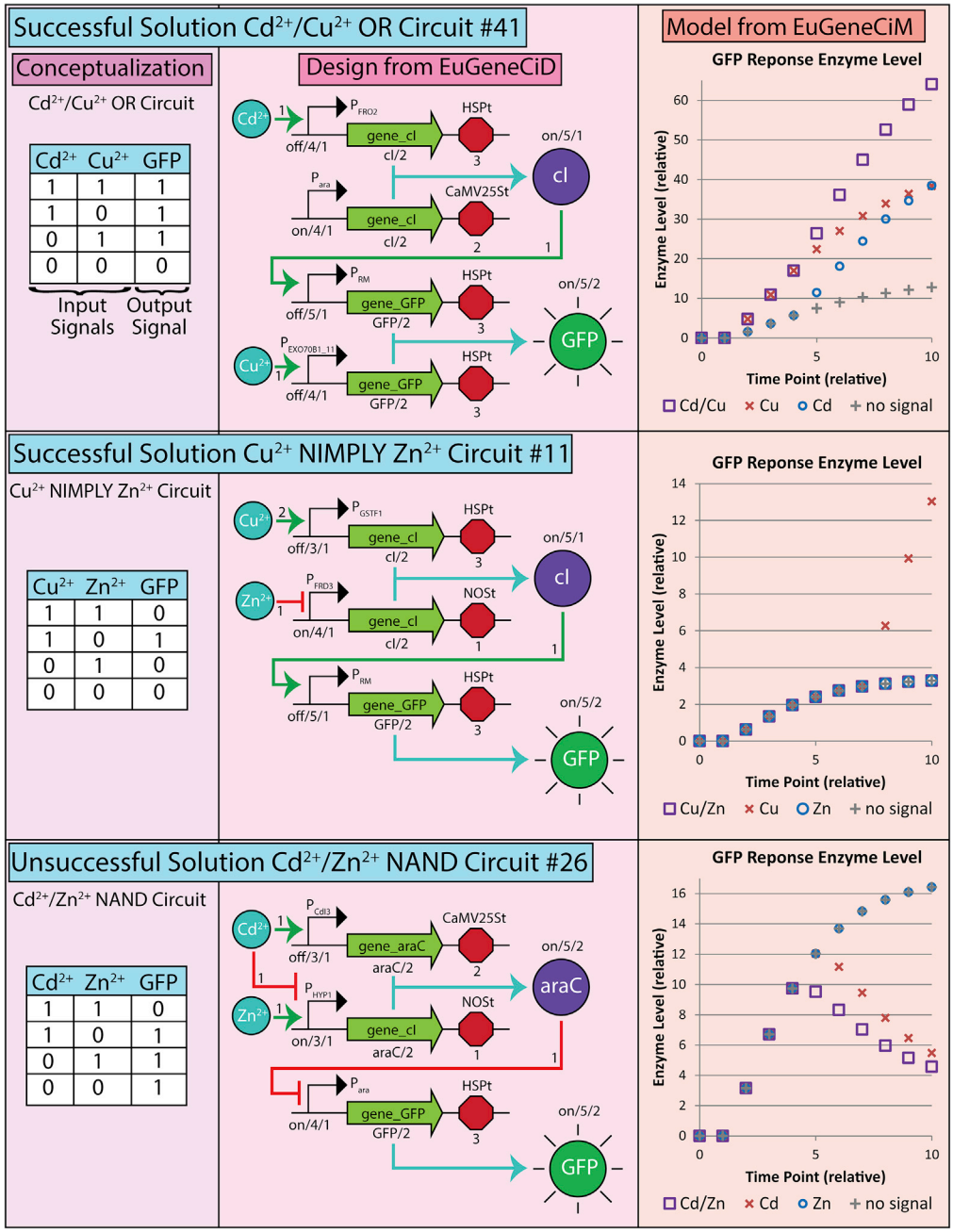



**Figure 6. Example EuGeneCiD and EuGeneCiMsolutions**

Shown here are three circuit conceptualizations, EuGeneCiD design solutions, and their associated EuGeneCiM models. The conceptualization is shown as the input logic table. The solution is shown with the design triads and produced enzymes with regulatory relations shown (green for activation, red for inhibition), including their relative strengths (shown as numbers on top of the regulation line). The modeled design responses are shown in the rightmost panel; where purple squares indicate the presence of both signals; blue circles and red crosses denote only one signal (see individual legends); and gray plus signs indicate no signal. Of the provided solutions, two of were shown to be potentially successful ($Cd^{2+}/Cu^{2+}$ OR circuit solution #41 and $Cu^{2+}$ NIMPLY $Zn^{2+}$ circuit solution #11) and one shown to be potentially unsuccessful ($Cd^{2+}/Zn^{2+}$ NAND Circuit solution #26) by EuGeneCiM.

expression of GFP through a control enzyme, in this case, araC. The triads of this design are $P_{Cdl3}$/gene_araC/CaMV25St, $P_{HYP1}$/gene_araC/NOSt, and $P_{ara}$/gene_GFP/HSPt. One interesting point to note is that the used promotors are weaker and terminators are less efficient than those generally used with cI

because the control enzyme, araC, has a longer half-life. In this unsuccessful example, the circuit responds correctly to the presence of both $Cd^{2+}$ and $Zn^{2+}$; of $Zn^{2+}$; and to no signal. This circuit fails in the condition at which only $Cd^{2+}$ is present. This is because, while EuGeneCiD partially accounts for enzyme degradation, it does not account for accumulation as it predicts that under this condition araC will not accumulate sufficiently to be active. However, when accounting for accumulation, EuGeneCiM predicts that araC will accumulate enough for an araC-expressed phenotype around time point 5, resulting in a sharp decline in GFP response from this point. This circuit could be potentially corrected by replacing the terminator in the $P_{CdI3}$/gene_araC/CaMV25St triad with a less efficient terminator. Unlike the other examples, this also illustrates that the trend of EuGeneCiM models might change direction and even sign during the simulation. This change during the simulation may result in a correct circuit response, whereas previous time points might suggest an incorrect response (consider the condition where both $Cd^{2+}$ and $Zn^{2+}$ are present). This suggests that for some circuits it may be useful to look at longer-term behavior in some cases where a designed circuit may be modeled to show an incorrect response.

### EuGeneCiM-modeled repressilator

To demonstrate the utility of EuGeneCiM as an independent modeling tool, it was decided to model a repressilator circuit. Repressilator circuits rely on the degradation of proteins whose expression is repressed to allow a downstream protein to be expressed, and therefore could not be modeled by non-dynamic genetic circuit modeling tools, or tools which do not consider transcript or protein degradation. A five-triad repressilator circuit was manually designed (because a repressilator cannot be designed by the non-dynamic EuGeneCiD) and is shown in Figure 7. This circuit utilizes araC, cI, and tetR control enzymes from *E. coli*, which have been reported to be used in synthetic biology applications in *Arabidopsis* (Messing, 1998), are well characterized, and which control promotor expression. All these enzyme inhibit one promotor in the biopart library, and importantly two of these enzymes have corresponding promotors which they activate, araC and cI. No promotor could be found which was activated by tetR. These activated promotors encode reporting fluorescent enzymes mKO (activated by araC) and GFP (activated by cI) identified through the fluorescent protein database (fpbase.org). Using EuGeneCiM, it was decided to model the first 100 relative time points of the simulation of the repressilator.

This simulation highlights several important features of the EuGeneCiM for which there was no opportunity for discussion when modeling EuGeneCiD-created designs. First, transcript production, transcript level (shown in Figure 7C), enzyme production, and enzyme level (shown in Figure 7B) are all modeled and tracked by EuGeneCiM (complete results can be found in the GitHub associated with this work at github.com/ssbio/EuGeneCiDM). Second, the shape of the response curves is of interest. As shown best by the tetR response curve (purple), EuGeneCiM models can achieve steady state (or near steady state) and be perturbed from that state. This curve also shows that EuGeneCiM is capable of modeling oscillatory circuit designs. This indicates that EuGeneCiM is not wholly dependent on EuGeneCiD and can be used as an independent modeling tool. Further, upon introducing three enzymes, there is some unsteady-state start-up period where the enzymes in question are all produced prior to some control enzyme taking dominance. Using GFP as an example, this period is approximately the times from time points 0 to 12. This is the start-up period, and varies to some extent between enzymes, though it appears that GFP has the longest such period. It can also be seen in these graphs that the amplitude of enzyme responses are uneven between enzymes. This is due to differences in promotor strength (stronger promotor, higher peak), terminator efficiency (more efficient terminator, higher peak), and enzyme half-life (longer half-life, higher peak). These factors also influence the breadth of the peaks, with shallower peaks also being broader, and taller peaks being narrower, with cI and GFP as the two more extreme cases in each direction, respectively. However, it should be noted that regardless of the breadth or height of the peaks, all enzyme expressions have a period of 22 time points, a period which is indefinitely stable (this repressilator has been modeled out to 500 time points).

One potential discrepancy with *in vivo* behavior is that repressilator responses *in vivo* are generally sinusoidal in behavior, in EGeneCiM models, the behavior is not perfectly sinusoidal in shape with sharp discontinuities at peak and trough. This is because transcription of a triad is modeled as a binary (either transcribed or not), rather than as a more continuous process as might occur *in vivo*. However, this wave has several similarities to a sine wave including a well-defined period (22 time points), amplitude (approximately 8 units), y-intercept (varies depending on the enzyme of interest, for GFP it is 10.37 units, defined from the average post-start-up), and x-intercept (varies depending on the enzyme of interest, for GFP this is

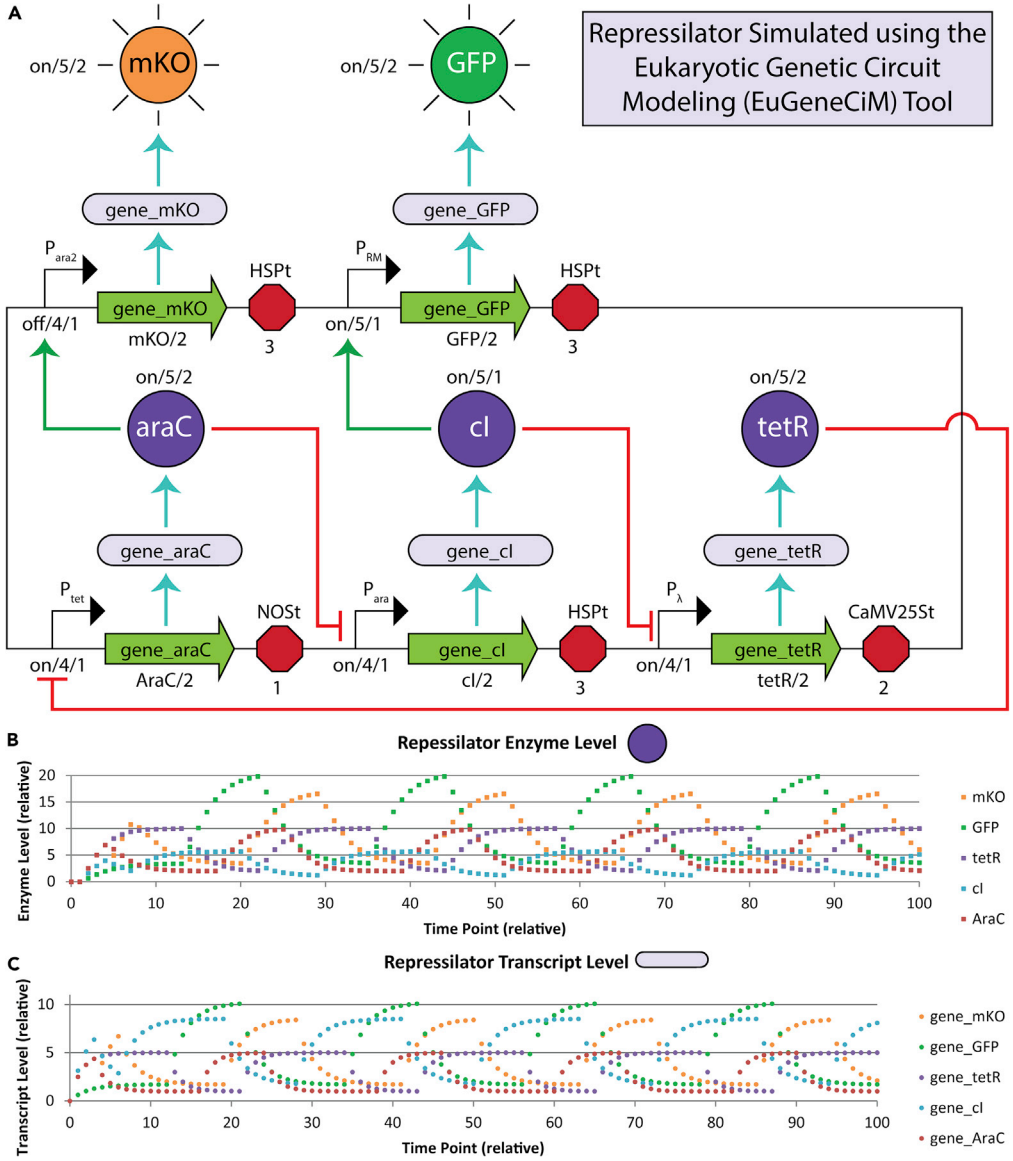



**Figure 7. Repressilator simulated using the EuGeneCiMtool**

While the EuGeneCiD and EuGeneCiM tools were designed to use in concert, they can be used independently, as evidenced here where EuGeneCiM is used to model a manually-designed repressilator.

(A) Shows the repressilator design with promotors (black), transcripts (green), and terminators (red) (collectively the design triads) in addition to the transcripts (light purple) and proteins (purple) produced thereby. The shorthand used throughout this work is used to show the characterization of these parts. Further, regulatory relations are shown (green for activation, red for inhibition).

(B) Scatterplot showing the dynamic behavior of the enzyme level for each of the enzymes included in the repressilator.

(C) Scatterplot showing the dynamic behavior of the transcript level for each of the enzymes included in the repressilator.

2 units). Despite their slightly different shape, they still are quite similar to sine waves nonetheless. As a demonstration of the modeled GFP enzyme level's similarity to a sine wave, a sine wave with the aforementioned characteristics of the GFP expression curve, graphs are provided in Table S3 which highlight the similarity of the GFP enzyme level curve shape and that of a sine wave. This has also been done for cI. The Pearson correlations between these curves are r = 0.91 and r = 0.97, respectively, showing a strong linear relationship between the curves produced by EuGeneCiM and the sine waves produced by using the characteristics of those curves, suggesting that the shape of the curves are very similar. Further, these

curves have the same mean value (about 10.4 units), and similar standard deviations (5.7 units for the sine wave and 6.0 units for the GFP curve) suggesting very similar magnitude, in addition to similar shape.

## DISCUSSION

Synthetic biology holds great potential for technological advancements and applications in a wide variety of fields. The designing of a new application involves five distinct steps, of which the first three (conceptualization, design, and modeling) can be performed *in silico*. Designing and modeling synthetic biology applications *in silico* holds several advantages including speed, tractability, advantages associated with certain types of mathematics such as optimization, and the potential to develop a pipeline for synthetic biology applications. This has been recognized by other researchers, who have developed *in silico* tools for either design or modeling of genetic circuits, which are generally not paired with a complimentary tool in the other step (see Figure 1). This work seeks to address this lack and work toward pipeline development by explicitly and easily linking the modeling and design steps, as well as expanding and improving upon optimization-based circuit design algorithms. In this work, it was decided to design and model plant-based heavy metal ion biosensors in *Arabidopsis*. These biosensors were designed to detect Cadmium, Copper, and Zinc, which are common metal ion pollutants, as a potential basis for future synthetic biology applications for phytoremediation. *Arabidopsis* was chosen as a model plant system with many previous synthetic biology applications, and it is eventually intended to apply EuGeneCiD and EuGeneCiM tools for applications in other plants (e.g., maize).

In the current work, two deterministic optimization-based tools for the design and modeling steps of the development of synthetic biology applications are introduced, the Eukaryotic Genetic Circuit Design (EuGeneCiD) and Modeling (EuGeneCiM) tools. These tools together can hypothesize and screen for potential genetic circuit designs which will be most likely to success when built *in vivo*. The first tool uses inputs of a bioparts database and a conceptualization of the desired application (in the form of a logic table) to design hypothetical genetic circuits. This tool is unique compared to previous tools in that it models transcript production; focuses on eukaryotic systems; accounts for transcript and enzyme degradation; and is more granular in its predictions than previous optimization-based tools. EuGeneCiD is paired with the dynamic circuit modeling tool EuGeneCiM, which uses the EuGeneCiD design and the bioparts databases as inputs. See Figure 4 for a visualization of the workflow.

Once these tools were developed, they were applied to 30 different systems biology conceptualizations which were created by pairing a logic gate (BUFFER, AND, NIMPLY CNI, HALF ADDER, NAND, NOR, OR, XNOR, and XOR) with ligands for that gate to respond to (single ligands for the BUFFER, namely Cd, Cu, and Zn, paired ligands for the other circuits, namely Cd/Cu, Cd/Zn, and Cu/Zn). These conceptualizations were chosen so as to make *Arabidopsis* roots as biosensors for heavy metal pollution, which can eventually be used as a basis for synthetic biology phytoremediation applications. The combined EuGeneCiD and EuGeneCiM workflow was run for seven days for each of the 30 conceptualization. The results of all 30 circuits are shown broadly in Figure 5, with some specific solutions to both EuGeneCiD and EuGeneCiM shown in Figure 6, while those of much simpler BUFFER circuit are shown in Figure S4. Briefly, EuGeneCiM solves more quickly and with higher fractions of optimal solutions for simpler circuit logic, for example BUFFER, AND, NIMPLY, and NOR and more slowly for more difficult logics like XOR, HALF ADDER, and XNOR. As shown in Figure 6, when modeled dynamically, while many EuGeneCiD-created designs functioned correctly, designs did not always function correctly under dynamic modeling. This showed that EuGeneCiM adds value by screening potentially unsuccessful solutions. This is in part because EuGeneCiD does not design circuits with respect to time, so accumulation of enzymes and transcripts are not accounted for at the design stage. We also wished to emphasize that the EuGeneCiM tool could be used as a stand-alone dynamic genetic circuit modeling tool, and to this end, EuGeneCiM is successfully applied to a manually designed repressilator (see Figure 7). This highlights how the EuGeneCiM tool crucially accounts for enzyme and transcript degradation allowing modeling of important dynamic circuits such as repressilators.

As shown in Figure 5, no set of EuGeneCiD solutions for any of the 30 synthetic biology application conceptualizations produced only optimal solutions. For all, some fraction of solutions were integer solutions with no guarantee of optimality (local or global). The two-input conceptualization with the highest fraction of optimal solutions is the $Cd^{2+}/Zn^{2+}$ responsive AND circuit with 84% and that with the lowest fraction is the $Cd^{2+}/Cu^{2+}$ responsive HALF ADDER and XOR circuits with slightly less more than 9% of solutions being optimal. The lack of any conceptualization identifying only optimal solutions has a few possible explanations. The first is that there is some "best" set of solver settings which would achieve only optimal solutions

which we have not been able to identify. Due to the long run time of some circuit designs (seven days), it was not deemed worth the time and effort to identify this set. A second possibility is the sheer number of solutions sought in that the runs were set only to terminate when 1000 solutions had been identified, the sought circuit size exceeded ten triads, or seven days had passed. A third possibility is that stretches of non-optimal solutions occur when the optimal solution lies along an edge, and the solutions along that edge are not globally optimal because equivalent designs exist. As shown in output files such as for the $Cd^{2+}$ NIMPLY $Zn^{2+}$ circuit conceptualization, stretches of sequential non-optimal solutions occur which have the same objective value (such as solutions #13 through #15), followed by an optimal solution with the same objective value. In the output of EuGeneCiD, it was found that for the $Cd^{2+}$/$Zn^{2+}$ responsive AND circuit, of the 160 non-optimal solutions returned, 81 of these occur in the last 150 solutions identified. Other non-optimal solutions occur when only a single solution remains at a given circuit size. In some instances, a non-optimal solution code might also be returned for a solution with the same objective value as an immediately preceding optimal solution (to two decimal points), suggesting that in some cases the non-optimality is inconsequential. Similar patterns occur for many of the easy to solve conceptualizations such as AND, NIMPLY, and CNI. By this point, a large number of integer cuts have been defined in the model to prevent repeat solutions, increasing the difficulty of finding a solution. When more difficult, this result in longer run times and an increased likelihood of heuristic termination from the solver. These heuristic terminators include lack of improvement on solution bounds in a certain time frame and reaching the maximum allowed time for a single solution (set at $1 \times 10^4$ seconds). These heuristic terminations also might explain the differences between optimality ratios, such as between the $Cd^{2+}$/$Zn^{2+}$ responsive AND and $Cd^{2+}$/$Cu^{2+}$ responsive HALF ADDER circuits, in that solving the latter is significantly more difficult than the former. Given the relative positions of optimal to non-optimal solutions, the positions of solutions with the maximum objective value, and the lengthening solution times at higher solution numbers, for users of the EuGeneCiD tool it is recommended that only the first 100 solutions need be identified and investigated.

As noted earlier, EuGeneCiD is not a dynamic design tool, although it does attempt to model one half-live of degradation to attempt to overcome this issue and to include degradation in design criteria. This results in some design solutions being non-functional under dynamic modeling in EuGeneCiM. EuGeneCiD was made non-dynamic for one primary reason: computational expense. Given the number of binary variables inherent in the EuGeneCiD problem, the already long solution times for certain conceptualizations, and the frequent non-optimality of solutions, it was decided not to create a dynamic EuGeneCiD out of concern for creating a non-viable tool (or one viable only in niche instances). In future, it is desired to improve the EuGeneCiD tool, and one of the primary improvements we will aim to implement is to make the tool dynamic, potentially creating a hybrid design and modeling tool. Another issue arising from pairing a static and dynamic tool such as this, is the cumulative effects of concentration buildup in the dynamic model. This resulted in the need to halve terminator and enzyme half-lives to attempt to reach similar enzyme production levels in EuGeneCiD as in EuGeneCiM. Without this adjustment, EuGeneCiM predicted levels often were one to two order of magnitude larger than in EuGeneCiD, resulting in all enzymes in the design being "active" regardless of regulation. This approach to reduce the half-live seemed best to both minimize the changes the parameters (such as enzyme concentration level thresholds, half-life, transcriptional efficiency, etc.) and to still produce results on a similar order of magnitude.

Overall, EuGeneCiD and EuGeneCiM have the potential to design with respect to and model biopart interactions which do not exist in the current bioparts database. Some of these functionalities include alternative splicing, changeable transcriptional efficiency (such as might be tuned through codon optimization), and protein-protein regulatory interactions. In creating a more capable tool, we hope to encompass new bioparts with sophisticated functionality and regulation which are even now being created by synthetic biologists for fine-tuned control of designed systems. One example is the Two-Component Systems (TCSs) for phosphoregulated, chemically induced signal transduction in mammalian cells, a work which shows great potential for the future designs of sophisticated synthetic biology bioparts (Scheller et al., 2020). In addition to making EuGeneCiD and EuGeneCiM potentially compatible with future synthetic bioparts, the choice of system and knowledge of that system has limited the biopart interactions which might be present in the library. *Arabidopsis* was chosen as a test system because it is a model plant to which synthetic biology applications have previously been applied. A plant system was chosen for the application because, in future, we hope to use the EuGeneCiD and EuGeneCiM tools to create synthetic biology applications for *Zea mays*, particularly those which activate in response to stress conditions to increase plant health and fitness under these conditions. One potential application is for heavy metal phytoremediation, hence

the use of heavy metal ligands as signals for designed genetic circuits. Given these desired goals and future applications, the breadth and types of interactions in the bioparts database was further limited.

### Limitations of the study

The EuGeneCiD and EuGeneCiM system essentially applies to a single small cell, as there is no explicit inclusion of transport mechanisms, diffusion, or cell differentiation. Cell differentiation, and the resulting differential expression of genes, in particular must be considered when defining the bioparts database. This may be problematic when attempting to model behavior in a multi-cellular organism) and has limited ability to account for individual variations between cells. In contrast to other techniques, EuGeneCiD and EuGeneCiM produce relative concentration predictions, rather than exact levels. Additionally, as already discussed, while *in vivo*repressilators have sinusoidal behavior, EuGeneCiM-modeled repressilators due not due to their underlying binary mathematics, though their shape is similar as already discussed. Further, some current tools (with a more biophysical focus) include considerations of copy number and phenotypic ranges, which are not accounted for in the EuGeneCiD and EuGeneCiM tools. As shown in Figure 5, a large number of returned solutions are non-optimal, particularly for circuit logic which are more difficult to construct such as HALF ADDER, XOR, and others.

Another limitation of these tools which may be addressed in future is their deterministic nature, as opposed to a stochastic approach. These tools were developed as deterministic tools for ease of characterization (e.g., a deterministic model requires no distribution for the generation of "noise"), and lower computational cost. While gene expression is often noisy and stochastic in nature, these tools will suffice to design and model circuit behavior to allow for hypothesizing and screening of potential design solutions. Future improvements (such as a dynamic EuGeneCiD) may be accompanied by the changing of these tool to be deterministic.

As no other tool or workflow yet exists which accomplishes both the tasks of modeling and design of genetic circuits from a library of available bioparts, it is difficult to compare this work against that of other studies. It is of interest to the researchers to confirm the usefulness of the EuGeneCiD and EuGeneCiM tools with *in vivo* tests of circuits built from these modeled results; however, it was determined that such a test is outside the scope of this work.

## STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Symbols used
  - EuGeneCiD problem statement and explanation
  - Constraint equations
  - Constraint equations
  - Designing and modeling genetic circuits
  - Computing, language, and solving resources in implementation
- QUANTIFICATION AND STATISTICAL ANALYSIS
- ADDITIONAL RESOURCES

## SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2021.103000.

## ACKNOWLEDGMENTS

This work has been completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative. The authors gratefully acknowledgefunding from the NSF EPSCoR Center for Root and Rhizobiome Innovation Grant 25-1215-0139-025 at the University of

Nebraska – Lincoln. The authors would also like to acknowledge the contribution of participants of the Young Nebraska Scientists Program (YNSP) who participated in this research including (listing in alphabetical order by family name): Bree Brunsman, Evan Fulton, Ali Keshk, Kareem Keshk, and Molly Nora.

## AUTHOR CONTRIBUTIONS

Conceptualization, W.L.S. and R.S.; Data curation, W.L.S. and A.B.; Formal analysis, W.L.S. and A.B.; Funding Acquisition, R.S.; Investigation, W.L.S. and A. B.; Methodology, W.L.S.; Project administration, R.S.; Resources, R.S.; Software, W.L.S. and A.B.; Supervision, R.S.; Validation, W.L.S.; Visualization, W.L.S.; Writing – original draft, W.L.S. and R.S.; Writing – reviewing & editing – W.L.S., A.B., and R.S.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

## INCLUSION AND DIVERSITY

One or more of the authors of this paper self-identifies as living with a disability. One or more of the authors of this paper self-identifies as a member of the LGBTQ+ community.

Received: May 14, 2021
Revised: July 13, 2021
Accepted: August 12, 2021
Published: September 6, 2021

## REFERENCES

Beyer, P., Al-Babili, S., Ye, X., Lucca, P., Schaub, P., Welsch, R., and Potrykus, I. (2002). Golden Rice: introducing the beta-carotene biosynthesis pathway into rice endosperm by genetic engineering to defeat vitamin A deficiency. J. Nutr. *132*, 506S–510S. http://www.ncbi.nlm.nih.gov/pubmed/11880581.

Borujeni, A.E., Zhang, J., Doosthosseini, H., Nielsen, A.A.K., and Voigt, C.A. (2020). Genetic circuit characterization by inferring RNA polymerase movement and ribosome usage. Nat. Commun. *11*. https://doi.org/10.1038/s41467-020-18630-2.

Chen, Y., Zhang, S., Young, E.M., Jones, T.S., Densmore, D., and Voigt, C.A. (2020). Genetic circuit design automation for yeast. Nat. Microbiol. *5*. https://doi.org/10.1038/s41564-020-0757-2.

Dasika, M.S., and Maranas, C.D. (2008). OptCircuit: an optimization based method for computational design of genetic circuits. BMC Syst. Biol. *2*, 1–19. https://doi.org/10.1186/1752-0509-2-24.

de Felippes, F.F., McHale, M., Doran, R.L., Roden, S., Eamens, A.L., Finnegan, E.J., and Waterhouse, P.M. (2020). The key role of terminators on the expression and post-transcriptional gene silencing of transgenes. Plant J. *104*, 96–112. https://doi.org/10.1111/tpj.14907.

English, M.A., Gayet, R.V., and Collins, J.J. (2021). Designing biological circuits: synthetic biology within the operon model and beyond. Annu. Rev. Biochem. *90*, 1–24. https://doi.org/10.1146/annurev-biochem-013118-111914.

Davidsohn, N., Beal, J., Kiani, S., Adler, A., Yaman, F., Li, Y., Xie, Z., and Weiss, R. (2015). Accurate predictions of genetic circuit behavior from Part Characterization and modular composition. ACS Synth. Biol. *4*, 673–681. https://doi.org/10.1021/sb500263b.

Gonzali, S., Mazzucato, A., and Perata, P. (2009). Purple as a tomato: towards high anthocyanin tomatoes. Trends Plant Sci. *14*, 237–241. https://doi.org/10.1016/j.tplants.2009.02.001.

Hill, A.D., Tomshine, J.R., Weeding, E.M.B., Sotiropoulos, V., and Kaznessis, Y.N. (2008). SynBioSS: the synthetic biology modeling suite. Bioinformatics *24*, 2551–2553. https://doi.org/10.1093/bioinformatics/btn468.

Holland, C.K., and Jez, J.M. (2018). Arabidopsis: the original plant chassis organism. Plant Cell Rep. *37*, 1359–1366. https://doi.org/10.1007/s00299-018-2286-5.

Jacob, J.M., Chinnannan, K., Saratale, R.G., Kumar, S.S., Prabakar, D., Kadirvelu, K., and Pugazhendhi, A. (2018). Biological approaches to tackle heavy metal pollution: a survey of literature. J. Environ. Manage. *217*, 56–70. https://doi.org/10.1016/j.jenvman.2018.03.077.

Khalil, A.S., and Collins, J.J. (2010). Synthetic biology: applications come of age. Nat. Rev. Genet. *11*, 367–379. https://doi.org/10.1038/nrg2775.

Kim, J., and Winfree, E. (2011). Synthetic in vitro transcriptional oscillators. Mol. Syst. Biol. *7*, 1–15. https://doi.org/10.1038/msb.2010.119.

Liu, W., and Stewart, C.N. (2015). Plant synthetic biology. Trends Plant Sci. *20*, 309–317. https://doi.org/10.1016/j.tplants.2015.02.004.

Messing, J. (1998). Plant science in lac: a continuation of using tools from Escherichia coli in studying gene function in heterologous systems. Proc. Natl. Acad. Sci. U S A *95*, 93–94. https://doi.org/10.1073/pnas.95.1.93.

Mortimer, J.C. (2019). Plant synthetic biology could drive a revolution in biofuels and medicine. Exp. Biol. Med. *244*, 323–331. https://doi.org/10.1177/1535370218793890.

Nagaya, S., Kawamura, K., Shinmyo, A., and Kato, K. (2010). The HSP terminator of arabidopsis thaliana increases gene expression in plant cells. Plant Cell Physiol. *51*, 328–332. https://doi.org/10.1093/pcp/pcp188.

Pixley, K.V., Falck-Zepeda, J.B., Giller, K.E., Glenna, L.L., Gould, F., Mallory-Smith, C.A., Stelly, D.M., and Stewart, C.N. (2019). Genome editing, gene drives, and synthetic biology: will they contribute to disease-resistant crops, and who will benefit? Annu. Rev. Phytopathol. *57*, 165–188. https://doi.org/10.1146/annurev-phyto-080417-045954.

Rizwan, M., Ali, S., Qayyum, M.F., Ok, Y.S., Zia-ur-Rehman, M., Abbas, Z., and Hannan, F. (2017). Use of Maize (Zea mays L.) for phytomanagement of Cd-contaminated soils: a critical review. Environ. Geochem. Health *39*, 259–277. https://doi.org/10.1007/s10653-016-9826-0.

Scheller, L., Schmollack, M., Bertschi, A., Mansouri, M., Saxena, P., and Fussenegger, M. (2020). Phosphoregulated orthogonal signal transduction in mammalian cells. Nat. Commun. *11*. https://doi.org/10.1038/s41467-020-16895-1.

Schroeder, W.L., and Saha, R. (2020). OptFill: a tool for infeasible cycle-free gapfilling of stoichiometric metabolic models. iScience *23*, 100783. https://doi.org/10.1016/j.isci.2019.100783.

Sekara, A., Poniedzialek, M., Ciura, J., and Jedrszczyk, E. (2005). Zinc and copper

accumulation and distribution in the tissues of nine crops: implications for phytoremediation. Polish J. Environ. Stud. *14*, 829–835.

Tan, S.I., and Ng, I.S. (2021). CRISPRi-mediated NIMPLY logic gate for fine-tuning the whole-cell sensing toward simple urine glucose detection. ACS Synth. Biol. *10*, 412–421. https://doi.org/10.1021/acssynbio.1c00014.

Vardhan, K.H., Kumar, P.S., and Panda, R.C. (2019). A review on heavy metal pollution, toxicity and remedial measures: current trends and future perspectives. J. Mol. Liquids *290*, 111197. https://doi.org/10.1016/j.molliq.2019.111197.

Vareda, J.P., Valente, A.J.M., and Durães, L. (2019). Assessment of heavy metal pollution from anthropogenic activities and remediation strategies: a review. J. Environ. Manage. *246*, 101–118. https://doi.org/10.1016/j.jenvman.2019.05.126.

Voigt, C.A. (2020). Synthetic biology 2020–2030: six commercially-available products that are changing our world. Nat. Commun. *11*, 10–15. https://doi.org/10.1038/s41467-020-20122-2.

Wuana, R.A., and Okieimen, F.E. (2010). Phytoremediation potential of maize (Zea mays L.).A review. Afr. Stud.Popul. Health *00*, 275–287. http://www.asopah.org.

Xia, P.F., Ling, H., Foo, J.L., and Chang, M.W. (2019). Synthetic genetic circuits for programmable biological functionalities. Biotechnol. Adv. *37*, 107393. https://doi.org/10.1016/j.biotechadv.2019.04.015.

Zomorrodi, A.R., and Maranas, C.D. (2014). Coarse-grained optimization-driven design and piecewise linear modeling of synthetic genetic circuits. Eur. J. Oper.Res. *237*, 665–676. https://doi.org/10.1016/j.ejor.2014.01.054.

## STAR★METHODS

### KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Deposited data | | |
| GitHub | www.github.com | RRID:SCR_002630 |
| Software and algorithms | | |
| Perl Programming Language (version 5.26 for Unix) | Perl www.perl.org | RRID:SCR_018313 |
| Strawberry Perl version 5.24.0.1 (for Windows) | Strawberry Perl Strawberryperl.com | RRID:SCR_018313 |
| The world-wide-web library for Perl, module 6.39 | LWP Meta CPAN https://metacpan.org/pod/LWP | N/A |
| Comprehensive Perl Archive Network (CPAN) | https://metacpan.org/ | RRID:SCR_007253 |
| Generalized Algebraic Modeling System (GAMS) version 24.7.4 | GAMS Products and Downloads www.gams.com/products/buy-gams/ | RRID:SCR_018312 |
| CPLEX solver version 12.6 | GAMS Products and Downloads www.gams.com/products/buy-gams/ | N/A |
| Other | | |
| Holland Computing Center: Crane Computing Cluster (64 GB RAM, Intel Xenon E5-2670 2.60 GHz processor, 2 CPUs per node) | Holland Computing Center https://hcc.unl.edu/ | N/A |
| ASUSTeKZyphyrus G model laptop computer with Microsoft Windows 10. | Any reasonably up-to-date computer, and alternative OSs, will work for this protocol. | N/A |
| Dell OptiPlex 790 desktop computer with Microsoft Windows 10 Enterprise | Any reasonably up-to-date computer, and alternative OSs, will work for this protocol. | N/A |

### RESOURCE AVAILABILITY

#### Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Rajib Saha (rsaha2@unl.edu).

#### Materials availability

This study did not generate new unique reagents.

#### Data and code availability

The published article does not include all datasets and code generated or analyzed during this study. All datasets and code generated during this study are available at GitHub in the ssbio/EuGeneCiDM repository [https://doi.org/10.5281/zenodo.4762590] or at the following URL github.com/ssbio/EuGeneCiDM.

### METHOD DETAILS

#### Symbols used

This section is provided here to increase clarity of the provided equations which follow. For the purposes of this text, a set is an unordered collection of distinct elements, a parameter is a value which is constant during the solution process whereas the value of a variable is altered by the solver to identify optimal solutions.

*Sets.*

$$A \equiv \textit{set of all molecules}$$
$$P \subset A \equiv \textit{set of promotors}$$
$$J \subset A \equiv \textit{set of transcripts}$$
$$E \subset A \equiv \textit{set of enzymes}$$

$E_d \subseteq E \equiv$ set of enzymes which it is desired for the circuit to respond to

$L \subset A \equiv$ set of ligands

$L_d \subseteq L \equiv$ set of ligands which it is desired for the circuit to respond to (note that this should always contain a none)

$T \subset A \equiv$ set of all terminators

$\mathbb{R} \equiv$ set of real numbers

$\mathbb{R}^+ \equiv$ set of nonnegative, real numbers

$\mathbb{R}^- \equiv$ set of nonpositive, real numbers

$\mathbb{B} \equiv$ binary set, contains only the numbers 1 and 0, e.g. $\mathbb{B} = \{0, 1\}$

$\mathbb{T} \equiv$ trinary set containing only the numbers $-1$, 0, and 1, e.g. $\mathbb{T} = \{-1, 0, 1\}$

*Parameters.*

$\lambda_{eL_1L_2} \in \mathbb{B} \equiv$ input logic matrix value for enzyme $e$ under conditions of ligands $L_1, L_2 \in L_d$ present

$Z_p \in \mathbb{B} \equiv$ normal state of promotor $p \in P$

$\zeta_e \in \mathbb{B} \equiv$ normal state of enzyme $e \in E$

$I_{pa} \in \mathbb{T} \equiv$ Effects of $a \in A$ as a ligand upon the activity of promotor $p \in P$ ($-1$ inhibition, 0 no effect, 1 activation)

$H_{pa} \in \mathbb{R}^+ \equiv$ strength of interaction between promotor $p \in P$ and molecule $a \in A$

$B_{ea} \in \mathbb{T} \equiv$ Effects of $a \in A$ as a ligand upon the activity of enzyme $e \in E$ ($-1$ inhibition, 0 no effect, 1 activation)

$Q_{ea} \in \mathbb{R}^+ \equiv$ strength of interaction between enzyme $e \in E$ and molecule $a \in A$

$V = 1E4 \equiv$ an arbitrarily large number

$\varepsilon = 1E-4 \equiv$ an arbitrarily small number

$\theta_e \in \mathbb{R}^+ \equiv$ concentration threshold at which the enzyme $e \in E$ must be present to be said to be "active"

$\eta_j \in \mathbb{R}^+ \equiv$ translational efficiency of transcript $j \in J$

$F_p \in \mathbb{R}^+ \equiv$ leakiness of a promotor $p \in P$

$G_t \in \mathbb{R}^+ \equiv$ half $-$ life of terminator $t \in T$

$\tau_e \in \mathbb{R}^+ \equiv$ half $-$ life of enzyme $e \in E$

$\sigma_{a_1a_2} \in \mathbb{B} \equiv$ value of 1 if $a_1 \in A$ is the same as $a_2 \in A$ and zero otherwise, identifies equivalent elements

$S_p \in \mathbb{R}^+ \equiv$ strength of promotor $p \in P$

*Variables.*

$\alpha_{pL_1L_2} \in \mathbb{R} \equiv$ integer net effect of all inhibition and activation on a given promotor $p \in P$ under conditions of ligands $L_1, L_2 \in L_d$ present ($>0$ promotor can be active $\leq 0$ promotor cannot be active)

$\alpha^+_{pL_1L_2} \in \mathbb{B} \equiv$ binary net effect of ligands upon promotor $p \in P$ in circuit under ligand conditions $L_1$, $L_2 \in L_d$ (1 promotor can be active, 0 promotor cannot be active)

$\gamma_{eL_1L_2} \in \mathbb{R} \equiv$ integer net effect of all inhibition and ativation on a given enzyme $e \in E$ under ligand conditions $L_1, L_2 \in L_d$ ($>0$ enzyme can be active, $\leq 0$ enzyme cannot)

$\gamma^+_{eL_1L_2} \in \mathbb{B} \equiv$ binary net effect of ligands upon enyme $e \in E$ in circuit under ligand conditions $L_1$, $L_2 \in L_d$ (1 enzyme can be active, 0 enzyme cannot be active)

$\varphi_{jtL_1L_2} \in \mathbb{R}^+ \equiv$ level of transcript $j$ expression under ligand conditions $L_1$, $L_2 \in L_d$

$M_{pjt} \in \mathbb{B} \equiv$ binary variable which creates promotor $p \in P$, transcript $j \in J$, and terminator $t \in T$ triads representing the design (variable in EuGeneCiD, (parameter in EuGeneCiS)

$C_{eL_1L_2} \in \mathbb{R}^+ \equiv$ concentration of enzyme $e \in E$ under under ligand conditions $L_1$, $L_2 \in L_d$

$\xi_{pjtL_1L_2} \in \mathbb{R}^+ \equiv$ deliberate transcription of $j \in J$ transcribed from promotor $p \in P$ and transcript $t \in T$ under ligand conditions $L_1$, $L_2 \in L_d$

$\omega_{eL_1L_2} \in \mathbb{B} \equiv$ determines if enzyme $e \in E$ is produced under ligand conditions $L_1$, $L_2 \in L_d$

$Y_{eL_1L_2} \in \mathbb{B} \equiv$ *binary variable determining if the enzyme* $e \in E$ *has sufficient concentration to be considered active under ligand conditions* $L_1,\ L_2 \in L_d$

$W_{eL_1L_2} \in \mathbb{B} \equiv$ *binary variable determining if enzyme* $e \in E$ *is both at sufficient concentration to be active and that it is not inhibited, in short that it will function underligand conditions* $L_1,\ L_2 \in L_d$

$\kappa_{eL_1L_2} \in \mathbb{B} \equiv$ *binary variable determining if enzyme* $\in E$ *is produced and can be active under ligand conditions* $L_1,\ L_2 \in L_d$

$Z_D \in \mathbb{R} \equiv$ *objective variable for EuGeneCiD*

$Z_M \in \mathbb{R} \equiv$ *objective variable for EuGeneCiM*

$D_{ee_1} \in \mathbb{R}^+ \equiv$ *direct attribution of enzyme e activity to* $e_1$ *through enzyme interactions*

$K_{ee_1} \in \mathbb{R}^+ \equiv$ *direct attribution of enzyme e activity to* $e_1$ *through enzyme* $e_1$ *on triad interactions*

$U_{ee_1e_2} \in \mathbb{R}^+ \equiv$ *attribution of enzyme e activity to* $e_2$ *acting through* $e_1$ *through enzyme interactions*

$U'_{ee_1} \in \mathbb{R}^+ \equiv$ *networked attribution of enzyme e activity to* $e_1$ *acting through other enzymes reflecting direct enzyme − enzyme interactions*

$\chi_{ee_1e_2} \in \mathbb{R}^+ \equiv$ *attribution of enzyme e activity to* $e_2$ *acting through* $e_1$ *through enzyme on triad interactions*

$X_{ee_1} \in \mathbb{R}^+ \equiv$ *networked attribution of enzyme e activity to* $e_1$ *acting through other enzymes reflecting enzyme on triad interactions*

$L_e \in \mathbb{B} \equiv$ *value of 1 if enzyme e is encoded by the genetic circuit design, 0 otherwise*

$\beta_{ee_1} \in \mathbb{B} \equiv$ *value of 1 if enzymes e and* $e_1$ *are encoded by the genetic circuit design, 0 otherwise*

## EuGeneCiD problem statement and explanation

*Objective function.* Objective function (Equation 1)

$$\text{maximize } Z_D = \sum_{e \in E_d} \sum_{L_1 \in L_d} \sum_{L_2 \in L_d} \left[ C_{eL_1L_2} \lambda_{eL_1L_2} - C_{eL_1L_2} \left(1 - \lambda_{eL_1L_2}\right) \right] \quad \text{(Equation 1)}$$

Where $Z_D$ is the objective value, $C_{eL_1L_2}$ is the contraction of enzyme $e$ under conditions with signals $L_1$ and $L_2$ (which includes "none") and $\lambda_{eL_1L_2}$ is the desired phenotype in response to signals $L_1$ and $L_2$ as encoded in the conceptualized logic table (this term is order-dependent). See the STAR Methods section for the full list of symbols and their definitions. This equation, Equation (1), seeks to maximize the responses of the desired enzymes under their desired conditions (in terms of concentration) and minimize the responses of the undesired enzymes under their undesired condition.

Note that in the above equation, the order of set elements matters, e.g., $C_{GFP,Zn^{2+},none}$ is mathematically distinct from $C_{GFP,none,Zn^{2+}}$ though efforts have been made to ensure that they will have the same value. Nonetheless, the issue of combinations (of which there are a total of 8 for any given ligand set in this work, where the set includes the two ligands to which the system should respond as well as "none") affects the objective function. From this, an AND circuit would only have 1 of 8 values of $\lambda_{eL_1L_2}$ with a 1 and the remainder would be 0. Similarly, a NOR circuit would only have a single non-zero value in its order-dependent conceptualization matrix ($\lambda_{eL_1L_2}$). This results in these circuits having unusually low objective values, as most terms are subtractive. The tendency in optimal designs then is to strongly favor designs with minimal expression leakage. Conversely, OR and NAND circuits have only one or two zero values in their order-dependent conceptualization matrix ($\lambda_{eL_1L_2}$), and therefore most terms are additive. Therefore, optimal circuit designs here tend to favor high inducible expression. Therefore, in Figure 7, it is best to not compare objective function values between different conceptualizations, but to only compare within conceptualizations. Depending on the tendencies of circuit design due to the circuit type, more complex circuits could result in lower expression leakage or higher inducible expression, and these complexities cannot be built into small circuits consisting of one or two triads.

## Constraint equations

*Circuit size limitations.* Circuit size limitations are defined in Equations 2, 3, 4, and 5.These equations limit the number of:

1) Maximum number of copies of a single promotor which can be used in the circuit design ($N_{p,max}$), Equation (2).
2) Maximum number of copies of a single transcript which can be used in the circuit design ($N_{j,max}$), Equation (3).
3) Maximum number of copies of a single terminator which can be used in the circuit design ($N_{t,max}$), Equation (4).
4) Total number of promotors, transcripts, and terminator triads which the circuit design can use ($N_{circuit,max}$), Equation (5).

$$\sum_{j\in J}\sum_{t\in T} M_{pjt} \leq N_{p,max} \qquad \forall\ p\in P \qquad \text{(Equation 2)}$$

$$\sum_{p\in P}\sum_{t\in T} M_{pjt} \leq N_{j,max} \qquad \forall\ j\in J \qquad \text{(Equation 3)}$$

$$\sum_{p\in P}\sum_{j\in J} M_{pjt} \leq N_{t,max} \qquad \forall\ t\in T \qquad \text{(Equation 4)}$$

$$\sum_{j\in J}\sum_{p\in P}\sum_{t\in T} M_{pjt} \leq N_{circuit,max} \qquad \text{(Equation 5)}$$

Note that by the nature of the variables used (e.g., $M_{pjt}$ being binary), only one copy of any given triad may be present in the designed circuit. However, any number of promotor/transcript, promotor/terminator, and transcript/terminator pairs may be repeated. This is important to later constraints. It should be noted that $N_{circuit,max}$ is set to 1 in the first attempt to solve EuGeneCiD and incremented by 1 each time no solution is found or the problem is deemed infeasible. In this way, the simplest circuit designs possible are identified and precluded from future solutions so that each solution is the simplest possible (Equations 2, 3, 4, and 5).

*Promotor state under conditions.* These equations determine if a promotor is active under the given conditions of ligand 1 and/or/nor 2 being present. Equations perform as follows (Equations 6, 7, and 8):

1) Determines the net effect of (by term): i) promotor normal state, ii) activation or inhibition by enzymes produced by the circuit, iii) inhibition or activation by ligand $L_1$, iv) inhibition or activation by ligand $L_2$, v) prevent duplicate activation/inhibition if $L_1$ and $L_2$. Equation (6).
2) Ensures that if $\alpha_{pL_dL_{d1}} > 0$ then $\alpha^{+}_{pL_1L_2} = 1$, and if $\alpha_{pL_dL_{d1}} \leq 0$ then $\alpha^{+}_{pL_1L_2} = 0$. Equations (7) and (8).

$$\alpha_{pL_1L_2} = Z_p + \sum_{e\in E}\left[W_{eL_1L_2} I_{pe} H_{pe}\right] + I_{pL_1} H_{pL_1} + I_{pL_2} H_{pL_2} - I_{pL_1}\sigma_{L_1L_2} H_{pL_1} \qquad \forall\ p\in P; L_1, L_2\in L_d \qquad \text{(Equation 6)}$$

$$\alpha_{pL_1L_2} \geq -V\left(1-\alpha^{+}_{pL_1L_2}\right) + \varepsilon\alpha^{+}_{pL_1L_2} \qquad \forall\ p\in P; L_1, L_2\in L_d \qquad \text{(Equation 7)}$$

$$\alpha_{pL_1L_2} \leq V\alpha^{+}_{pL_1L_2} \qquad \forall\ p\in P; L_1, L_2\in L_d \qquad \text{(Equation 8)}$$

*Transcription under conditions.* These equations determine if and to what extent transcript $j$ is intentionally transcribed from promotor $p$ under ligand $L_1$ and $L_2$ conditions ($\xi_{pjtL_1L_2}$). The following equations accomplish the following:

1) A transcript cannot be transcribed from a given promotor unless the promotor and transcript are paired in the circuit design.
2) Transcription will not occur unless the promotor is "on".
3) All three constraints are equivalent to: $\xi_{pjtL_dL_{d1}} = S_p M_{pjt}\alpha^{+}_{pL_1L_2}$, Equations (9), (10), and (11).

$$\xi_{pjtL_1L_2} \leq S_p M_{pjt} \qquad \forall\ p\in P; j\in J; t\in T; L_1, L_2\in L_d \qquad \text{(Equation 9)}$$

$$\xi_{pjtL_1L_2} \leq S_p\alpha^{+}_{pL_1L_2} \qquad \forall\ p\in P; j\in J; t\in T; L_1, L_2\in L_d \qquad \text{(Equation 10)}$$

$$\xi_{pjtL_1L_2} \geq S_p\left(M_{pjt} + \alpha^{+}_{pL_1L_2} - 1\right) \qquad \forall\ p\in P; j\in J; t\in T; L_1, L_2\in L_d \qquad \text{(Equation 11)}$$

The following equations determine the transcript level ($\varphi_{jL_dL_{d1}}$) as the sum of positive effects on the transcript level, including deliberate ($\xi_{pjL_1L_2}$) and leaky ($M_{pjt}F_p$) transcription. This is scaled by a half-life-based amount of RNA degradation to simulate the fact that degradation occurs and factors this into circuit design (Equations 9, 10, 11, and 12).

$$\varphi_{jtL_1L_2} = \sum_{p\in P}\left[\left(\xi_{pjL_1L_2} + M_{pjt}F_p\right)\left(0.5^{\left(\frac{1}{\sigma_t+\varepsilon}\right)}\right)\right] \qquad \forall\ j\in J; t\in T; L_1, L_2\in L_d \qquad \text{(Equation 12)}$$

*Translation under conditions.* The following equation determines the enzyme concentration level ($C_{eL_dL_{d1}}$) as the sum of effects on the enzyme concentration level ($C_{eL_dL_{d1}}$), Equation (17), reduced by a half-life-based enzyme degradation multiplicative factor (Equations 13, 14, 15, 16, and 17).

$$C_{eL_1L_2} = \sum_{j\in J}\left[\left(\rho_{je}\eta_j\varphi_{jL_1L_2}\right)\left(0.5^{\frac{1}{\bar{R}e+\varepsilon}}\right)\right] \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 13)}$$

The following equations determine if the enzyme is being produced $\omega_{eL_1L_2} = 1$ if produced and zero otherwise.

$$\omega_{eL_1L_2} \le VC_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 14)}$$
$$\omega_{eL_1L_2} \ge \varepsilon C_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 15)}$$

The following equations, Equations (18) and (19), determine if the concentration of the enzyme is at sufficient levels ($\theta_e$) to say that the enzyme could be active, $C^+_{eL_1L_2} = 1$ if sufficient concentration, zero otherwise.

$$(\theta_e+\varepsilon)C^+_{eL_1L_2} \le C_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 16)}$$
$$C_{eL_1L_2} \le (V-(\theta_e-\varepsilon))C^+_{eL_1L_2} + (\theta_e-\varepsilon) \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 17)}$$

*Enzyme regulation and activity under conditions.* Determine the net effect of ligands on the enzyme ($\gamma_{eL_dL_{d1}}$) to determine if the protein is active or inactive due to the present ligands ($\delta_{eL_dL_{d1}}$, concentration incorporated through interaction strength $Q_{eL_{d1}}$) (Equations 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, and 28).

1) Sum of the effects of present ligands and enzymes on the possibility of enzyme *e* being able to be activated ($\gamma_{eL_dL_{d1}}$), Equation (18).
2) Determine net effect of activation/inhibition on the enzyme ($\delta_{eL_dL_{d1}}$) Equations (19) and (20).

$$\gamma_{eL_1L_2} = \zeta_e + \sum_{e_1\in E}\left(W_{e_1L_1L_2}B_{ee_1}Q_{ee_1}\right) + B_{eL_1}Q_{eL_1} + B_{eL_2}Q_{eL_2} - B_{eL_1}Q_{eL_1}\sigma_{L_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 18)}$$

$$\gamma_{eL_1L_2} \ge -V\left(1-\gamma^+_{eL_1L_2}\right) + \varepsilon\gamma^+_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 19)}$$
$$\gamma_{eL_1L_2} \le V\gamma^+_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 20)}$$

Determine if the protein is both produced and can be active. These three constraints, Equations (21), (22), and (23), are equivalent to $\kappa_{eL_dL_{d1}} = \omega_{eL_dL_{d1}}\delta_{eL_dL_{d1}}$ (this works because all the variables are binary).

$$\kappa_{eL_1L_2} \le \omega_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 21)}$$
$$\kappa_{eL_1L_2} \le \gamma^+_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 22)}$$
$$\kappa_{eL_1L_2} \ge \omega_{eL_1L_2} + \gamma^+_{eL_1L_2} - 1 \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 23)}$$

Determine if the protein is produced, active, and at sufficient concentration for it to function. These three constraints, Equations (24), (25), and (26), are equivalent to $W_{eL_dL_{d1}} = \kappa_{eL_dL_{d1}}Y_{eL_dL_{d1}}$ (this works because all the variables are binary).

$$W_{eL_1L_2} \le \kappa_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 24)}$$
$$W_{eL_1L_2} \le C^+_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 25)}$$
$$W_{eL_1L_2} \ge \kappa_{eL_1L_2} + C^+_{eL_1L_2} - 1 \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 26)}$$

Force the logic table to be true in Equation (27).

$$W_{e_d L_1 L_2} = \lambda_{e_d L_1 L_2} \qquad \forall\ e_d \in E_d;\ L_1, L_2 \in L_d \qquad \text{(Equation 27)}$$

*Attribution of enzyme activity to given conditions under conditions.* Given all these equations, it is not guaranteed that the circuit produced thus far will truly respond to the input ligands. One persistent issue with the formulation to this point is that a Bistable Orthogonal Design (BOD) can be returned which is independent of the input ligands and the optimization solver will simply choose the appropriate state to appear to meet the logic table. This causes a circuit which appears to the solver to meet design criteria, but in fact does not because it does not respond to ligand conditions. This issue is addressed through what we are choosing to call the attribution constraints. These constraints are created to determine what changes the activity of a protein in a given genetic circuit (e.g., what is the change attributable to?). This is done with several stages of equations (Equations 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, and 50).

*Set 1: Determine if a particular enzyme pair is encoded.* These equations are used to determine if a particular enzyme is encoded (encoded in the binary $L_e$). This is important in that an enzyme has no attribution from other enzymes and is not attributable to other enzymes (Equations 28, 29, 30, 31, and 32).

$$L_e \geq \varepsilon \sum_{p \in P} \sum_{j \in J} \sum_{t \in T} \left[ M_{pjt} \rho_{je} \right] \qquad \forall\ e \in E \qquad \text{(Equation 28)}$$

$$L_e \leq V \sum_{p \in P} \sum_{j \in J} \sum_{t \in T} \left[ M_{pjt} \rho_{je} \right] \qquad \forall\ e \in E \qquad \text{(Equation 29)}$$

Note that this is formulated as such to allow for multiple transcript copies in a given circuit design. Next, a determination is made as to whether enzyme pairs are encoded (encoded in the binary $\beta_{ee_1}$); attribution cannot exist between enzymes.

$$\beta_{ee_1} \leq L_e \qquad \forall\ e \in E \qquad \text{(Equation 30)}$$

$$\beta_{ee_1} \leq L_{e_1} \qquad \forall\ e_1 \in E \qquad \text{(Equation 31)}$$

$$\beta_{ee_1} \geq L_e + L_{e_1} + 1 \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 32)}$$

*Set 2: Determine if a particular enzyme affects another enzyme's expression.* Next, we determine the effect of one enzyme upon the expression of another, through various means. First, through directly affecting enzyme activity (effect of $e_1$ upon $e$). Note that the variable $D_{ee_1}$ is restricted to be strictly non-negative (Equations 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, and 47).

$$D_{ee_1} = \left| B_{ee_1} \right| \beta_{ee_1} \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 33)}$$

Note that the above is linear because $B_{ee_1}$ is a parameter. It was discovered during debugging procedures that attempting to track the sign of attributions can lead to numerical issues (such as an attribution canceling itself out, but still existing); therefore, only the fact of attribution is determined using absolute values. The next group of equations determines the effect of $e_1$ upon $e$ through controlling the triad expressing $e$. Note that the variable $K_{ee_1}$ is restricted to be strictly non-negative.

$$K_{ee_1} \leq V \beta_{ee_1} \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 34)}$$

$$K_{ee_1} \leq \sum_{p \in P} \sum_{j \in J} \left[ \left| I_{pe_1} \right| * \sum_{t \in T} M_{pjt} \rho_{je} \right] + V\left(1 - \beta_{ee_1}\right) \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 35)}$$

$$K_{ee_1} \geq \sum_{p \in P} \sum_{j \in J} \left[ \left| I_{pe_1} \right| * \sum_{t \in T} M_{pjt} \rho_{je} \right] - V\left(1 - \beta_{ee_1}\right) \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 36)}$$

In combination with the domain of $K_{ee_1}$, $K_{ee_1} = 0$ if $\beta_{ee_1} = 0$, and $K_{ee_1} = \sum_{p \in P} \sum_{j \in J} \left[ \left| I_{pe_1} \right| * \sum_{t \in T} M_{pjt} \rho_{je} \right]$ otherwise.

Next, the effect of one enzyme ($e_2$) upon another enzyme ($e$) through another enzyme ($e_1$). This passing of attribution might be through direct enzyme effects ($D_{ee_1}$) or through the effect of one enzyme upon the triad of another ($K_{ee_1}$). The variable $\nu'_{e_1 e_2}$ below is a binary variable noting if there is attribution of enzyme $e_2$ upon enzyme $e_1$ (e.g., $e_2$ in some way affects the activity of $e_1$).

$$U_{ee_1 e_2} \leq V \beta_{ee_1} \qquad \forall\ e, e_1, e_2 \in E \qquad \text{(Equation 37)}$$

$$U_{ee_1 e_2} \leq \left| B_{ee_1} \right| \nu'_{e_1 e_2} + V\left(1 - \beta_{ee_1}\right) \qquad \forall\ e, e_1, e_2 \in E \qquad \text{(Equation 38)}$$

$$U_{ee_1 e_2} \geq \left| B_{ee_1} \right| \nu'_{e_1 e_2} - V\left(1 - \beta_{ee_1}\right) \qquad \forall\ e, e_1, e_2 \in E \qquad \text{(Equation 39)}$$

This can then be condensed into the variable $U'_{ee_1}$ which removes the middle enzyme:

$$U'_{ee_1} = \sum_{e_2 \in E} \left[ U_{ee_2e_1} \left(1 - \sigma_{ee_1} \sigma_{e_1e_2}\right)\right] \qquad \forall\ e, e_1, e_2 \in E \qquad \text{(Equation 40)}$$

Therefore, $U'_{ee_1}$ represents the indirect attribution of $e_1$ to the activity of $e$ through direct attributions. This allows any number of intermediates between two enzymes to still count toward attribution due to the effects of networking. Note that the $(1 - \sigma_{ee_1}\sigma_{e_1e_2})$ term prevents an enzyme attributing to itself through itself. This prevents a potential self-referential problem which occurs with the definition of $\nu'_{ee_1}$. It should be noted that $U'_{ee_1}$ tracks only enzyme-enzyme interaction networks. Similarly, $X_{ee_1}$ will track enzyme attribution networks through effects on enzyme triads, though due to the need to track triads the formulation is necessarily more complex. Together, $U'_{ee_1}$ and $X_{ee_1}$ allow for full networked tracking of attribution through any number of intermediary enzymes and regulatory mechanisms.

$$\chi_{ee_1pjt} \le VM_{pjt} \qquad \forall\ e, e_1 \in E; p \in P; j \in J; t \in T \qquad \text{(Equation 41)}$$

$$\chi_{ee_1pjt} \le \sum_{e_2 \in E} \left[ \left| I_{pe_2} \right| \nu'_{e_2e_1} \rho_{je} \right] + V\left(1 - M_{pjt}\right) \qquad \forall\ e, e_1 \in E; p \in P; j \in J; t \in T \qquad \text{(Equation 42)}$$

$$\chi_{ee_1pjt} \ge \sum_{e_2 \in E} \left[ \left| I_{pe_2} \right| \nu'_{e_2e_1} \rho_{je} \right] - V\left(1 - M_{pjt}\right) \qquad \forall\ e, e_1 \in E; p \in P; j \in J; t \in T \qquad \text{(Equation 43)}$$

$$X_{ee_1} = \sum_{p \in P} \sum_{j \in J} \sum_{t \in T} \left[ \chi_{ee_1pjt} \right] \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 44)}$$

Now that the direct ($D_{ee_1}$ and $K_{ee_1}$) and networked ($U'_{ee_1}$ and $X_{ee_1}$) attribution variables have been determined, the total attribution can be determined.

$$\nu_{ee_1} = D_{ee_1} + K_{ee_1} + U'_{ee_1} + X_{ee_1} \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 45)}$$

Note that $\nu_{ee_1}$ is a nonnegative variable, since $D_{ee_1}$, $K_{ee_1}$, $U'_{ee_1}$, and $X_{ee_1}$ are all nonnegative values which may have values greater than 1 depending on the definitions of $I_{pa}$ (for $p \in P$ and $a \in A$) and $B_{ea}$ (for $e \in E$ and $a \in A$). For instance, in some cases it is useful to have values greater than 1 in $I_{pa}$ or $B_{ea}$ to indicate that some effectors are stronger than others. Due to the need for referencing total attribution within the network attribution variables ($U'_{ee_1}$ and $X_{ee_1}$, which themselves are part of the total attribution) there arises an issue related to the use of multiplication. If a value other than zero or one is used in calculating the total attribution's effect on the network attribution variables, attributions which influence each other could quickly increase in magnitude through recursion. Another potential issue is the possibility that if total attributions are not equal in magnitude, this could result in solution infeasibility as the two attributions cannot exist together. Therefore, there is a need to transform the non-negative $\nu_{ee_1}$ into the binary $\nu'_{ee_1}$ so that multiplicative identity Equations 38, 39, 42, and 43 might apply and bypass both these issues. Therefore, $\nu'_{ee_1}$ is a binary which is determined using the following constraints.

$$\nu_{ee_1} \ge \nu'_{ee_1} \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 46)}$$

$$\nu_{ee_1} \le V\nu'_{ee_1} \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 47)}$$

*Set 3: Preventing self-controlling enzymes.* Now that attribution of one enzyme to another can be determined ($\nu'_{ee_1}$), we have used this variable to prevent an enzyme from directly or indirectly controlling its own expression (which can lead to BODs). This can be prevented by ensuring that there is no self-attribution (Equations 48 and 49).

$$\nu'_{ee_1} \ge \sigma_{ee_1} - 1 \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 48)}$$

$$\nu'_{ee_1} \le 1 - \sigma_{ee_1} \qquad \forall\ e, e_1 \in E \qquad \text{(Equation 49)}$$

*Set 4: Prevent the addition of meaningless bioparts.* The above equations prevent self-attribution and BODs, but do not prevent the addition of meaningless triads to a solution. It was found during development that the addition of meaningless triads was one way for a solution to be reported again at larger circuit sizes. This can be relatively easily fixed with a single equation, which ensures that any encoded enzyme affects circuit reporter enzymes (Equation 50).

$$L_e \le \sum_{e_d \in E_d} \left[ \nu'_{e_de} \right] + E^{val}_{d,e} \qquad \forall\ e \in E \qquad \text{(Equation 50)}$$

where $E^{val}_{d,e} = 1$ if $e$ is a member of the set $E_d$ and $E^{val}_{d,e} = 0$ otherwise. This ensures that each encoded enzyme in some way influences the activity of at least one reporter enzyme or is itself a reporter enzyme.

*Speed boosting constraints.* The following constraints should be implicitly true given all of the previous constraints, yet it was discovered, as with the OptFill tool (Schroeder and Saha, 2020), that explicitly defining implicit relationships can result in quicker solution times. The following relationship where explicitly defined (Equations 51, 52, 53, 54, 55, and 56):

1) Equation 51 ensures that all response enzymes are encoded in the genetic circuit.
2) Equations 52, 53, and 54 ensures that no enzyme has activity unless encoded in the genetic circuit.
3) Equations 55 and 56 ensure that no enzyme has concentration unless encoded in the genetic circuit.

$$E_{d,e}^{val} \leq L_e \qquad \forall\ e \in E; L_1, L_2 \in L_d \qquad \text{(Equation 51)}$$
$$W_{eL_1L_2} \leq L_e \qquad \forall\ e \in E; L_1, L_2 \in L_d \qquad \text{(Equation 52)}$$
$$\kappa_{eL_1L_2} \leq L_e \qquad \forall\ e \in E; L_1, L_2 \in L_d \qquad \text{(Equation 53)}$$
$$\omega_{eL_1L_2} \leq L_e \qquad \forall\ e \in E; L_1, L_2 \in L_d \qquad \text{(Equation 54)}$$
$$C_{eL_1L_2} \leq L_e \qquad \forall\ e \in E; L_1, L_2 \in L_d \qquad \text{(Equation 55)}$$
$$C_{eL_1L_2}^{+} \leq L_e \qquad \forall\ e \in E; L_1, L_2 \in L_d \qquad \text{(Equation 56)}$$

*EuGeneCiM problem statement and explanation.* While the EuGeneCiM formulation is based upon that of EuGeneCiD, it is markedly less complex due to three factors: i) the design is already known, so $M_{pjt}$ becomes a parameters as opposed to a variable; ii) the design is already complete, attribution need not be tracked; and iii) the transcript an enzyme levels at the current time point are those produced at previous time point(s) and EuGeneCiM is simply solving for the production rate of enzymes and transcripts for the current time point.

*Objective function.* Objective function (Equation 57)

The selected objective function is to maximize the production of proteins

$$maximize\ Z_M = \sum_{e \in E} \sum_{L_1 \in L_d} \sum_{L_2 \in L_d} \left[C_{eL_1L_2}\right] \qquad \text{(Equation 57)}$$

Note that the objective function is largely unimportant however, as the constraint equations which follow are generally equality constraints, some of which lack variables.

## Constraint equations

*Determining the level of transcript production.* The first set of constraint equations determine the level of transcript production. First, the activity of the promotor under each condition set is evaluated in the same manner as in EuGeneCiD (Equations 6, 7, and 8 and 58):

$$\alpha_{pL_1L_2} = Z_p + \sum_{e \in E} \left[W_{eL_1L_2} I_{pe} H_{pe}\right] + I_{pL_1} H_{pL_1} + I_{pL_2} H_{pL_2} - I_{pL_1} \sigma_{L_1L_2} H_{pL_1} \qquad \forall\ p \in P; L_1, L_2 \in L_d \qquad \text{(Equation 6)}$$
$$\alpha_{pL_1L_2} \geq -V\left(1 - \alpha_{pL_1L_2}^{+}\right) + \varepsilon\alpha_{pL_1L_2}^{+} \qquad \forall\ p \in P; L_1, L_2 \in L_d \qquad \text{(Equation 7)}$$
$$\alpha_{pL_1L_2} \leq V\alpha_{pL_1L_2}^{+} \qquad \forall\ p \in P; L_1, L_2 \in L_d \qquad \text{(Equation 8)}$$

Then, the level of transcript production under each condition can be evaluated, similar to as is done in Equations (9), (10), (11), and (12) with two distinct simplifications: i) as $M_{pjt}$ is a parameter, the linearization of $S_p M_{pjt} \alpha_{pL_1L_2}^{+}$ accomplished in Equations (9), (10), and (11)is no longer needed, and is substituted directly into Equation (12) and ii) degradation of RNA is handled in another programmatic step between the time points, rather than at a single time point as in EuGeneCiD, therefore this is not included.

$$\varphi_{jtL_1L_2}^{prod} = \sum_{p \in P} \left[\left(S_p M_{pjt} \alpha_{pL_1L_2}^{+} + M_{pjt} F_p\right)\right] \qquad \forall\ j \in J; t \in T; L_1, L_2 \in L_d \qquad \text{(Equation 58)}$$

Note that the superscript *prod* is added to $\varphi_{jtL_1L_2}^{prod}$ to indicated that this is the transcript production at the current time point. This is an important distinction as the transcript carried over from the previous time point is denoted $\varphi_{jtL_1L_2}^{t_{n-1}}$ and is used to calculate the protein production at time $t_n$. This arrangement allows for the simulation of the delay between triad activation and transcript production, as well as between

transcript production and enzyme expression. Also, note that the identity of the terminator is tracked in $\varphi^{t_n}_{jtL_1L_2}$ as the terminator determines the half-life of its associated transcript.

*Determining the level of protein production.* As mentioned, the amount of protein produced at time $t_n$ is calculated from the amount of transcript carried over from the previous time point $t_{n-1}$. This is calculated in the following equation, which is analogous to Equation (13) without the degradation term (Equation 59).

$$C^{prod}_{eL_1L_2} = \sum_{j\in J}\left[\rho_{je}\eta_j\sum_{t\in T}\varphi^{t_{n-1}}_{jtL_1L_2}\right] \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 59)}$$

Note that $C^{prod}_{eL_1L_2}$ represents to protein production at time $t_n$, and that the activity of those proteins is determined by the carry-over from the previous time point, $C^{t_{n-1}}_{eL_1L_2}$.

*Determining the activity of the proteins.* Using the carry-over protein concentration, $C^{t_{n-1}}_{eL_1L_2}$, the activity of the enzyme is calculated in the same way as in EuGeneCiD and utilizing the same equations. These equations are restated here, see the symbols used section for symbol definitions (Equations 18, 19, 20, 21, 22, 23, 24, 25 and 26).

$$\gamma_{eL_1L_2} = \zeta_e + \sum_{e_1\in E}\left(W_{e_1L_1L_2}B_{ee_1}Q_{ee_1}\right) + B_{eL_1}Q_{eL_1} + B_{eL_2}Q_{eL_2} - B_{eL_1}Q_{eL_1}\sigma_{L_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d$$

(Equation 18)

$$\gamma_{eL_1L_2} \geq -V\left(1-\gamma^{+}_{eL_1L_2}\right) + \varepsilon\gamma^{+}_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 19)}$$

$$\gamma_{eL_1L_2} \leq V\gamma^{+}_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 20)}$$

$$\kappa_{eL_1L_2} \leq \omega_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 21)}$$

$$\kappa_{eL_1L_2} \leq \gamma^{+}_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 22)}$$

$$\kappa_{eL_1L_2} \geq \omega_{eL_1L_2} + \gamma^{+}_{eL_1L_2} - 1 \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 23)}$$

$$W_{eL_1L_2} \leq \kappa_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 24)}$$

$$W_{eL_1L_2} \leq C^{+}_{eL_1L_2} \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 25)}$$

$$W_{eL_1L_2} \geq \kappa_{eL_1L_2} + C^{+}_{eL_1L_2} - 1 \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 26)}$$

*Modeling degradation of transcripts and enzyme.* Between time points, and attempted solutions of EuGeneCiM, degradation of the bioparts are calculated as follows:

$$\varphi^{t_n}_{jtL_1L_2} = \left(\varphi^{prod}_{jtL_1L_2} + \varphi^{t_{n-1}}_{jtL_1L_2}\right)\left(0.5^{\left(\frac{1}{G_t/1+\varepsilon}\right)}\right) \qquad \forall\ j\in J; t\in T; L_1, L_2\in L_d \qquad \text{(Equation 60)}$$

$$C^{t_n}_{eL_1L_2} = \left(C^{prod}_{eL_1L_2} + C^{t_{n-1}}_{eL_1L_2}\right)\left(0.5^{\left(\frac{1}{R_e/2+\varepsilon}\right)}\right) \qquad \forall\ e\in E; L_1, L_2\in L_d \qquad \text{(Equation 61)}$$

Note that there is one major difference in the degradation terms of Equations (60) and (61): the half-lives are reduced by half in EuGeneCiM compared to EuGeneCiD. This in attempt to reconcile the differences between EuGeneCiD and EuGeneCiM when considering the cumulative effects of dynamic modeling. This occurs because, while EuGeneCiD accounts for a single time point and EuGeneCiM accounts for several, the enzyme and transcript accumulations in EuGeneCiM were generally one or two order of magnitude higher than that predicted in EuGeneCiD. This was an issue because the same concentration thresholds existed for enzyme activity, and therefore resulted in no enzyme being in an "off" state after sufficient time in EuGeneCiM. This fix reduces the half-life of transcripts and enzymes, resulting in closer parity in concentration and modeling of circuit designs while minimizing the number of parameters perturbed.

*Other important aspects of EuGeneCiM formulation.* Constraints not included in the formulation can be as important as those which are and can serve to highlight the function of the problem. Specifically, no constraints are included which force the provided conceptualization (in the form of a logic table) to be true. This is for two reasons. The first is that, in solving in a point by point manner, there will inevitably be time points in which the logic table is not true, particularly due to the delays between transcription and

translation built into the tool. Secondly, this allows EuGeneCiM to be a screening process to remove any designs which function differently when no longer optimizing for desired behavior or when considering dynamic behavior.

### Designing and modeling genetic circuits

See Figure 4 for a visual representation of the overall workflow and to specifically illustrate how the EuGeneCiD and EuGeneCiM formulations fit into this workflow. This work began with the conceptualization of synthetic biology interventions. For the purposes of demonstrating these design and modeling tools, simple circuit conceptualizations were selected, namely the two input circuits of AND, NIMPLY, HALF ADDER, NAND, NOR, XNOR, and XOR. Note that logic gates will be capitalized throughout this text to avoid confusion. These particular conceptualizations were chosen because they are easy to represent in logic table format, and well-known, and often studied in the context of genetic circuits (particularly NOR and NIMPLY) (Borujeni et al., 2020) (Tan and Ng, 2021). A library of bioparts (consisting of promotors, transcripts, terminators, and proteins) was then selected which were i) native to *Arabidopsis* (particularly promotors), ii) demonstrated to be functional in synthetic biology applications in *Arabidopsis*, or iii) were from related plant species which we judged were likely to function in synthetic biology applications. Note that when a particular biopart had different expression or regulation patterns at different stages in growth or in different tissues, the pattern related to seedling root was selected. These parts are described in detail in Table S2 These two items, conceptualizations and the bioparts library, are then appropriately formatted as input files utilizing a Perl script (included in the associated GitHub at github.com/ssbio/EuGeneCiDM) which reads a database file appended with the desired circuit logic, example is provided in Table S3 with the full set used here in the associated GitHub at github.com/ssbio/EuGeneCiDM), and writes the input files accordingly. EuGeneCiD was implemented in the Generalized Algebraic Modeling System (GAMS) language and run using the CPLEX solver. At this point, the workflow diverts to several possible outcomes. First, EuGeneCiD found no designs of the appropriate size, indicated by no solution or an "integer infeasible" model status. This is addressed by incrementing the allowed model size by one, provided the maximum allowable circuit design size has not been exceeded, and re-attempting to solve EuGeneCiD. The second possibility is that EuGeneCiD found a potential design which fits the current criteria. This design will be the output of EuGeneCiD and the input of EuGeneCiM. EuGeneCiM then simulates the designed circuit, beginning at time point zero with no initial concentration of any enzyme or transcript. EuGeneCiM will return, as an output, the relative production of enzymes and transcripts at the given time point. The concentration of enzymes at the current time point is reduced according to the half-life characteristics of the enzyme or transcript terminator, and the newly produced amount of each is added to this value as the carry over to the next time point. EuGeneCiM is then solved for the next time point, and the process is repeated until all time points have a solution. From this, the dynamic behavior of the designed circuit may be plotted as a visual representation of the circuit simulation. This can be done through an additional Perl script (included in the associated GitHub at github.com/ssbio/EuGeneCiDM). The cycle of design (through EuGeneCiD) and simulation (through EuGeneCiM) continues until case two occurs. The final possible outcome of EuGeneCiD is that no designs of the appropriate size can be found, and that incrementing the size would result in exceeding the maximum allowable circuit design size (here, ten triads). In this case, it will be concluded that there are no further designs, and the design and simulation results should be manually screened to pick the most promising design candidate(s). The example given here is a set of Cd/Cu responsive AND circuit from which is selected solution number 41, which has the highest objective value.

### Computing, language, and solving resources in implementation

This study has produced several unique software codes in the form of GAMS or Perl programming languages/tools. For implementing and solving EuGeneCiD and EuGeneCiM the Generalized Algebraic Modeling System (GAMS), version 24.7.4 was used in conjunction with the CPLEX solver version 12.6. Scripts which automate certain tasks utilize Perl version 5.26 for Unix or Strawberry Perl 5.24.0.1 for Windows. The code provided is compatible with both versions. The main workflow (previously described) was implemented on the Holland Computing Center Crane Cluster and allowed to run for at most seven days (168 hours) before being terminated. CPLEX solver settings used are included in the associated GitHub at github.com/ssbio/EuGeneCiDM.

## QUANTIFICATION AND STATISTICAL ANALYSIS

Many values used in the definition of the bioparts in the database used were defined through manual quantification of quantitative data. For promotors, normal state was determined by literature evidence (either

normally on or off). Strength and leakiness were determined, when possible, from western or northern blot images, with strong expression being given a value of 5 and no expression being given a value of 1. In some cases, the fold change in expression of a gene associated with a given promotor was known under induced cases. In these cases, the ratio or strength to leakiness was adjusted to reflect these known expression changes. Inducer and repressor identities were identified using literature evidence. The base strength of induction or repression was set to 1; however, if some ligand showed greater activation or repression than another, a value of 2 was assigned to model a greater effect on the activity of that particular promotor. For transcripts, the transcriptional efficiency can represent various design elements of the gene, codon optimization for instance, which can change the speed or efficiency of translation of the gene. A value of 0 would indicate that the gene cannot be translated and a value of 3 would indicate efficient translation. In this work, there was no such adjusting of the translational properties of the genes; therefore, a base value of 2 was assumed for all translational efficiencies. A small set of three terminators were identified from Nagaya et al. (2010) and the relative half-lives of these terminators were determined as follows. The scale used was from 0 representing near instant of mRNA to 3 representing slow degradation of associated mRNA. Based on Nagaya et al. (2010) values of associated mRNA half-life for each terminator was quantified. For enzymes, the default state was determined from literature. The default expression and half-life were assumed to be 5 and 2, respectively. These values were changed if literature evidence was found to warrant the need to adjust these values. For instance, cI was noted as being rapidly degraded in registry of standard biological parts, and therefore given a shorter half-life.

## ADDITIONAL RESOURCES

This work accompanies a protocol paper for ease of replication, in addition to allowing others to utilize the EuGeneCiD/S tools for their own studies. This protocol can be found in the journal STAR Protocols.