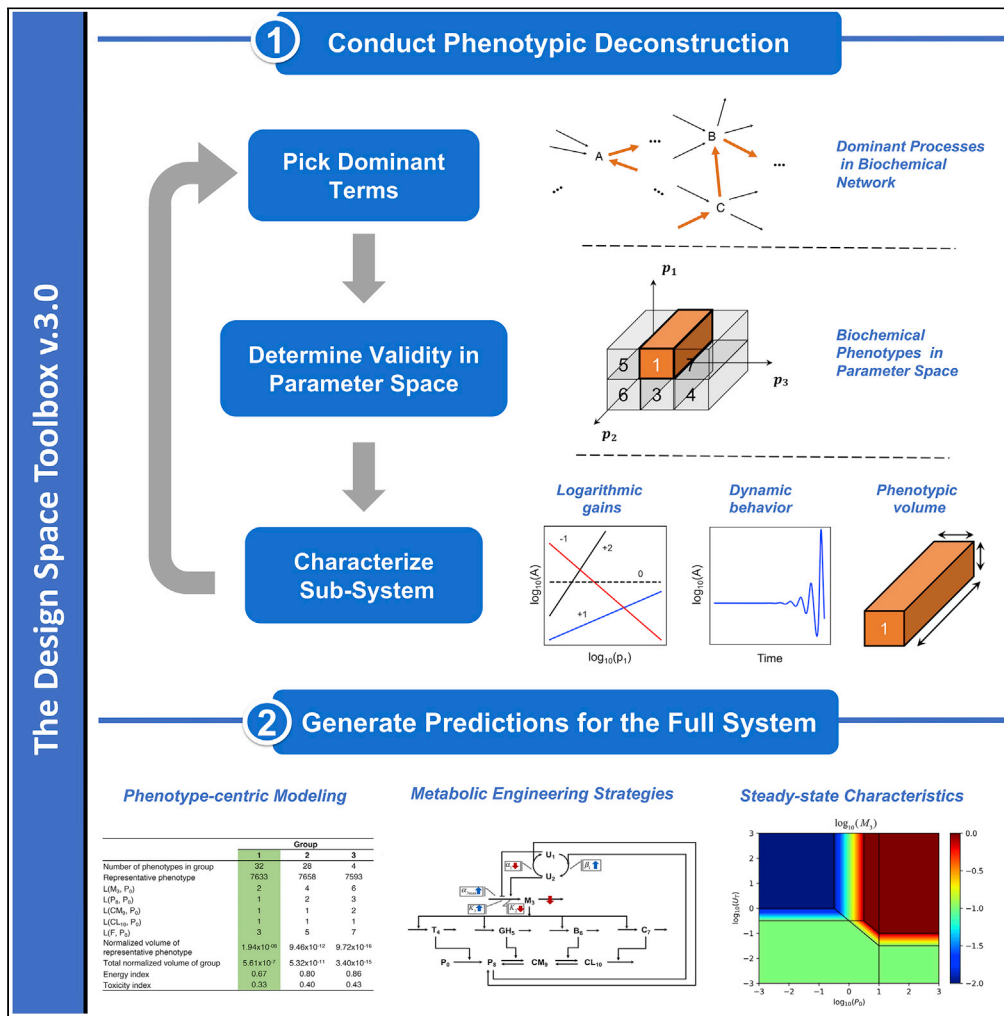


Article

Mechanistic Modeling of Biochemical Systems without *A Priori* Parameter Values Using the Design Space Toolbox v.3.0



Miguel Á. Valderrama-Gómez, Jason G. Lomnitz, Rick A. Fasani, Michael A. Savageau

masavageau@ucdavis.edu

HIGHLIGHTS

DST3 extends the Design Space formalism by identifying additional phenotypes

Additional phenotypes arise from cycles, conservations, and metabolic imbalances

DST3 enables mechanistic modeling without previous knowledge of parameter values

It fully unlocks the potential of the novel phenotype-centric modeling strategy

Valderrama-Gómez et al.,
iScience 23, 101200
June 26, 2020 © 2020 The Author(s).
<https://doi.org/10.1016/j.isci.2020.101200>



Article

Mechanistic Modeling of Biochemical Systems without *A Priori* Parameter Values Using the Design Space Toolbox v.3.0Miguel Á. Valderrama-Gómez,¹ Jason G. Lomnitz,² Rick A. Fasani,³ and Michael A. Savageau^{1,4,5,*}

SUMMARY

Mechanistic models of biochemical systems provide a rigorous description of biological phenomena. They are indispensable for making predictions and elucidating biological design principles. To date, mathematical analysis and characterization of these models encounter a bottleneck consisting of large numbers of unknown parameter values. Here, we introduce the Design Space Toolbox v.3.0 (DST3), a software implementation of the Design Space formalism enabling mechanistic modeling without requiring previous knowledge of parameter values. This is achieved by using a phenotype-centric modeling approach, in which the system is first decomposed into a series of biochemical phenotypes. Parameter values realizing phenotypes of interest are subsequently predicted. DST3 represents the most generally applicable implementation of the Design Space formalism and offers unique advantages over earlier versions. By expanding the Design Space formalism and streamlining its distribution, DST3 represents a valuable tool for elucidating biological design principles and designing novel synthetic circuits.

INTRODUCTION

Mechanistic modeling is the current gold standard for quantitative characterization and understanding of complex biological phenomena. However, mathematical analysis of rigorous mechanistic models is not yet the method of choice to address biological questions owing to a persistent gap in knowledge of the involved parameter values. Within this context, modeling without *a priori* values for the parameters is a radical claim. To see how radical, it helps to contrast the fundamentally new phenotype-centric approach implemented here with the conventional simulation-centric approach based on sampling and simulation (Lomnitz and Savageau, 2016a; Valderrama-Gómez et al., 2018). Both approaches start with a typical qualitative hypothesis or model consisting of (1) the number of molecular elements, (2) their interactions with one another (connections), (3) the signs of their interactions (+/−), and (4) the stoichiometry of the interactions (number of molecules involved). Such a model would have numerous mechanistic parameters (rate constants, binding constants, equilibrium constants) whose values are unknown and difficult to determine.

In the conventional strategy, one samples parameter values, which can be done by randomly drawing values from a uniform distribution within a pre-defined range (Lee et al., 2014), simulates the non-linear differential equations, and compares the results with the corresponding experimentally observed quantitative phenotypes of the system. This is repeated numerous times to select the values that tend to improve the fit between the simulated results and the experimental data (some examples of this methodology include the work by Forger and Peskin (2003), which implemented a coordinate search algorithm to find optimal parameter values; the work by Zhou et al. (2005), which manually tuned parameter values; and the work by Mirsky et al. (2009), which used an evolutionary strategy to find a set of parameters that most closely matched experimental data). The result might be a good fit, one that is marginally acceptable, or no fit at all.

In contrast to this conventional approach, the phenotype-centric approach provides a fundamental advance that is entirely different. It starts with the same qualitative model as the conventional approach.

¹Department of Microbiology & Molecular Genetics, University of California, Davis, CA 95616, USA

²VSP Global, Rancho Cordova, CA 95670, USA

³Agilent Technologies, Santa Clara, CA 95051, USA

⁴Department of Biomedical Engineering, University of California, Davis, CA 95616, USA

⁵Lead Contact

*Correspondence: masavageau@ucdavis.edu
<https://doi.org/10.1016/j.isci.2020.101200>



However, well-established powerful linear analysis methods (Strang, 2005) are used (Savageau, 1969, 1971a, 1971b) to identify and enumerate the entire repertoire of biochemical phenotypes for the model (Savageau et al., 2009). In this first step there is already the ability to discriminate among models/hypothesis; if the experimentally observed phenotype is not present in the repertoire, then the model can be eliminated (Lomnitz and Savageau, 2016b). On the other hand, if the phenotype of interest is present in the repertoire, then the same, powerful, well-established linear analytic methods can be used to predict a full set of parameter values that will realize that phenotype (Fasani and Savageau, 2010; Lomnitz and Savageau, 2016a; Valderrama-Gómez et al., 2018).

Here, we introduce the Design Space Toolbox v.3.0 (DST3) to expand the capabilities of the Design Space formalism (Savageau et al., 2009). This new version builds on previous iterations of the software (DST1: Fasani and Savageau, 2010, implemented the general data structures; DST2: Lomnitz and Savageau, 2016a, added parallelization routines and a graphical user interface) to allow the automatic identification and mathematical characterization of additional biochemical phenotypes arising from critically important under-determined cases.

These special phenotypes emerge from cycles, metabolic imbalances, and conservation constraints present in many biochemical systems. Examples of such systems include but are not limited to metabolic networks containing multiple reversible reactions (Schuster et al., 2000), a common motif leading to cycles; signaling cascades of chemical species with different regulatory states (e.g., achieved by different conformational or phosphorylation states) that are linked by conservation relationships (Haraldsdóttir and Fleming, 2016; Klipp and Liebermeister, 2006); and saturable processes, for which a pathway pool is imbalanced with respect its influx and efflux, thus generating a steady accumulation or depletion of material within the system (Levy, 1999; Dahl et al., 2013).

Besides the expanded computational engine of DST3, its C-library that is now able to handle these biologically important singularities automatically when they appear *individually and simultaneously*, the new version of the Design Space Toolbox (DST3) has further unique features that distinguish it from previous versions. First, the portability and installation of DST3 (Figure S1) is greatly improved and simplified by using Docker (Boettiger, 2015). Second, DST3 offers an improved and more stable IPython-based user interface for users with limited programming experience (refer to Figures S2 and S5–S11). Third, calculating the product of the global tolerances (Coelho et al., 2009) for all parameters in log-coordinates, a proxy for a phenotype's volume in parameter space and for its associated global robustness (Valderrama-Gómez et al., 2018) is now automated within DST3. Fourth, solvers for systems of both ordinary differential (ODE) and differential algebraic (DAE) equations (Petzold, 1983; Andersson et al., 2015) were incorporated, thus allowing a fully integrated dynamic characterization of the Full System. The analysis of the Full System by these completely different methodologies provides an independent means of confirming the results obtained by the Design Space methodology.

The Results section is divided into three sub-sections. The first sub-section reviews briefly key concepts of the Design Space formalism (see Savageau et al., 2009 and Fasani and Savageau, 2010 for a more detailed theoretical treatment) needed to understand the advances described in this work. In the second sub-section, we build on these concepts to develop general mathematical strategies aimed at resolving matrix singularities arising from system topologies containing cycles, moiety conservations, and metabolic imbalances. In the third sub-section, we illustrate the capabilities of DST3 by analyzing a case study, the protocatechuate metabolic system of *Acinetobacter* (Dal et al., 2005; Trautwein and Gerischer, 2001), a biochemical system exhibiting multiple, nested singularities. The Methods section provides details on the software architecture of DST3, the different ways to access its computational capabilities, and the installation instructions via Docker.

RESULTS

Review of Key Concepts

Biochemical systems described by the power-law functions of chemical kinetics and the rational functions of biochemical kinetics can be represented by generalized mass action (GMA) kinetics (Savageau and Voit, 1987) of the form:

$$\begin{aligned} \frac{dX_i}{dt} &= \sum_{k=1}^{P_i} \alpha_{ik} \prod_{j=1}^{n+m} X_j^{g_{ijk}} - \sum_{k=1}^{Q_i} \beta_{ik} \prod_{j=1}^{n+m} X_j^{h_{ijk}} \quad i=1, \dots, n_c \\ 0 &= \sum_{k=1}^{P_i} \alpha_{ik} \prod_{j=1}^{n+m} X_j^{g_{ijk}} - \sum_{k=1}^{Q_i} \beta_{ik} \prod_{j=1}^{n+m} X_j^{h_{ijk}} \quad i=(n_c+1), \dots, n \end{aligned} \quad (\text{Equation 1})$$

where α_{ik} and β_{ik} represent rate constants and g_{ijk} h_{ijk} are kinetic orders. P_i and Q_i are the number of positive and negative terms in the i -th equation, respectively. X_i represent the variables of interest (typically the concentration of a biochemical species) in a system containing a total of n dependent and m independent variables. In general, dependent variables can be split into two groups: chemical variables, for which a differential equation exists, and auxiliary variables, for which algebraic constraints are defined. Each group contains n_c and $n - n_c$ members, respectively. The set n_c of chemical variables has a direct biological meaning and represents all the chemical/biological entities (i.e., enzymes, metabolites, chemical species, mRNA molecules) of a given system. On the other hand, the set $n - n_c$ of auxiliary variables has meaning in the context of recasting the system of ordinary differential equations into its GMA form. Independent variables for which a differential equation or algebraic constraint are not defined are treated as parameters.

For any system, one of the P_i positive terms and one of the Q_i negative terms in Equation 1 will momentarily dominate over the others in each one of the n equations in the system. This gives rise to a so-called *dominant S-System* (Savageau, 1969; Savageau et al., 2009), which can be generically described by Equation 2:

$$\begin{aligned} \frac{dX_i}{dt} &= \alpha_{ip_i} \prod_{j=1}^{n+m} X_j^{g_{ijp_i}} - \beta_{iq_i} \prod_{j=1}^{n+m} X_j^{h_{ijq_i}} \quad i=1, \dots, n_c \\ 0 &= \alpha_{ip_i} \prod_{j=1}^{n+m} X_j^{g_{ijp_i}} - \beta_{iq_i} \prod_{j=1}^{n+m} X_j^{h_{ijq_i}} \quad i=(n_c+1), \dots, n \end{aligned} \quad (\text{Equation 2})$$

with p_i and q_i being the indices of the dominant positive and dominant negative term in the i -th equation, respectively. In steady state, Equation 2 can be combined into a single equation:

$$0 = \alpha_{ip_i} \prod_{j=1}^{n+m} X_j^{g_{ijp_i}} - \beta_{iq_i} \prod_{j=1}^{n+m} X_j^{h_{ijq_i}} \quad i=1, \dots, n \quad (\text{Equation 3})$$

The validity of a dominant S-System in steady state implies certain conditions (Savageau et al., 2009; Fasani and Savageau, 2010), which are represented by inequalities of the form:

$$\alpha_{ip_i} \prod_{j=1}^{n+m} X_j^{g_{ijp_i}} > \alpha_{ik} \prod_{j=1}^{n+m} X_j^{g_{ijk}} \quad \forall k = \{1, 2, 3, \dots, P_i | k \neq p_i\} \quad (\text{Equation 4})$$

$$\beta_{iq_i} \prod_{j=1}^{n+m} X_j^{h_{ijq_i}} > \beta_{ik} \prod_{j=1}^{n+m} X_j^{h_{ijk}} \quad \forall k = \{1, 2, 3, \dots, Q_i | k \neq q_i\}. \quad (\text{Equation 5})$$

Here, k represents indices of corresponding non-dominant terms. Steady-state concentrations of the dependent variables can be obtained in three steps (Savageau, 1969; Fasani and Savageau, 2010). By rearranging Equation 3 and taking logarithms, one obtains Equation 6:

$$\log \alpha_{ip_i} + \sum_{j=1}^{n+m} g_{ijp_i} \log X_j = \log \beta_{iq_i} + \sum_{j=1}^{n+m} h_{ijq_i} \log X_j, \quad (\text{Equation 6})$$

which can be written in matrix form as:

$$Ay = b, \quad (\text{Equation 7})$$

where $y_j = \ln X_j$, $a_{ij} = g_{ijp_i} - h_{ijq_i}$, and $b_i = \ln(\beta_{iq_i} / \alpha_{ip_i})$. In a second step, dependent (y_D) and independent (y_I) variables are split to obtain:

$$A_D y_D = b - A_I y_I. \quad (\text{Equation 8})$$

The vector of dependent concentration variables y_D can be obtained in a third step by matrix operations:

$$y_D = A_D^{-1} b - A_D^{-1} A_I y_I. \quad (\text{Equation 9})$$

The vector of dependent flux variables $\log F_i$ is obtained by matrix multiplication:

$$\log F = Gy. \quad (\text{Equation 10})$$

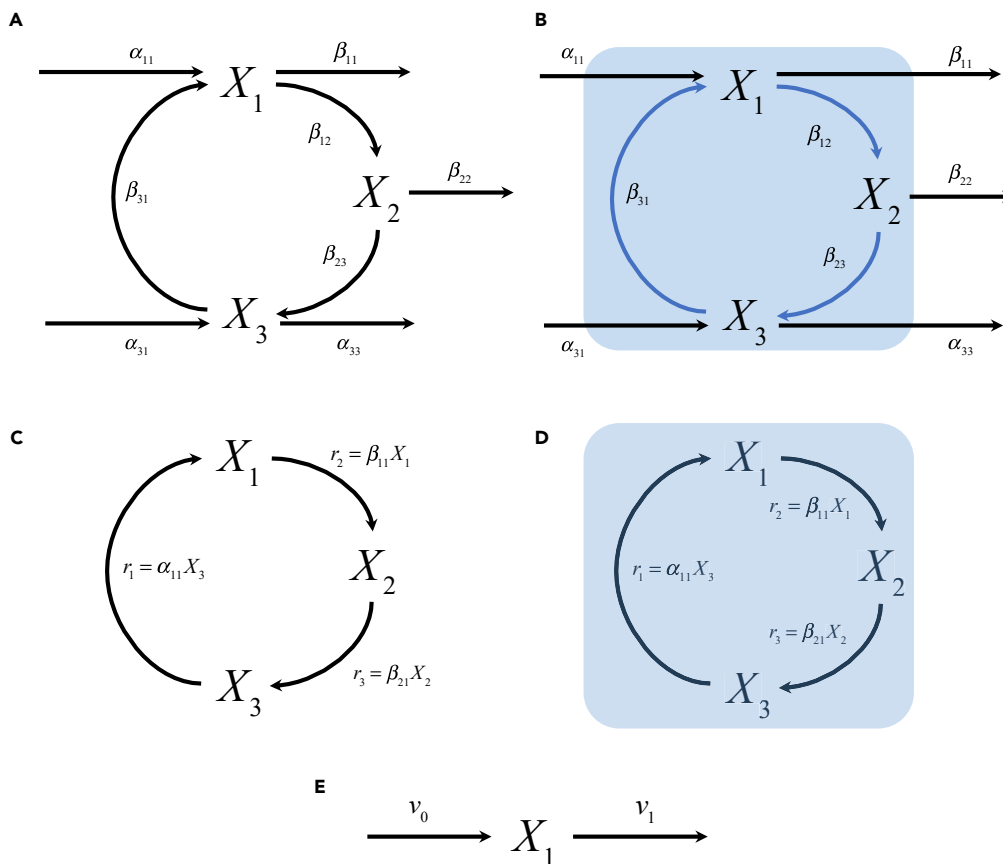


Figure 1. Systems Containing Cycles, Conservation Constraints, and Metabolic Imbalances

- (A) A cycle comprising three metabolic species is shown. Mass action kinetics is used to mathematically describe the flux through each reaction. Rate constants for each one of these reactions are shown.
- (B) If the fluxes represented by the blue arrows dominate over the others, a singularity arises that prevents the system from having a unique steady-state solution. This singularity is resolved by introducing a mass balance equation around a control volume (blue rectangle) containing the cycle.
- (C) Three-component system containing one conservation constraint. Fluxes to and from metabolite pools X_1 , X_2 , and X_3 are mathematically described using mass action kinetics.
- (D) No fluxes enter or leave the control volume (blue rectangle) around metabolites involved in the conservation.
- (E) Metabolic pool with one input and one output flux. Depending on the numerical values of fluxes v_0 and v_1 , the concentration of X_1 can steadily increase ($v_0 > v_1$), decrease ($v_0 < v_1$), or remain unchanged over time ($v_0 = v_1$).

The concept of biochemical phenotype (or simply *phenotype*) is an integral element of the Design Space formalism and will be broadly used throughout this work. A phenotype is defined in the context of a mechanistic mathematical model of a given biological system. The mathematical representation of a biochemical system decomposed into its repertoire of steady-state phenotypes, each given by a set of dominant S-system equations (Equation 3) and associated boundaries, involves a comprehensive integration of information for all the system's concentrations, fluxes, and parameters (Savageau et al., 2009; Fasani and Savageau, 2010). From a biological point of view, most of the mathematical properties of a biochemical phenotype, for instance, its dynamic behavior (discussed in section "Phenotype Analysis Reveals Dynamic Properties of the System") and logarithmic gains (discussed in section "Logarithmic Gains Can Guide the Design of Engineering Strategies") can be experimentally observed and measured, thus rendering biochemical phenotypes a powerful tool for the elucidation of design principles for natural systems and for the design of synthetic networks with novel functionalities.

Strategies for Treating Three Types of Singularities

Earlier versions of the Design Space Toolbox (Fasani and Savageau, 2010; Lomnitz and Savageau, 2016b) have exclusively dealt with cases for which the inverse of the matrix A_D exists (Figure S4). However, a

number of important system topologies—cycles, conservations, metabolic imbalances (Figure 1)—cause the matrix A_D to become singular and effectively limit the utility of earlier versions of the Design Space Toolbox. The ubiquity of such topologies in biochemical networks renders their automatic handling highly relevant in the context of the Design Space formalism.

The mathematical framework presented here to deal with these singularities can be seen as an extension of previous theory with *ad hoc* applications in a more limited context by our group (refer to Savageau, 1969 for metabolic imbalances; Alves and Savageau, 2003 for conservations; Tolla et al., 2015 for cycles). The extension presented in this work generalizes the theory and automates the computational applications thereby expanding the Design Space formalism (Savageau et al., 2009). The strategies are illustrated by means of simple examples treated briefly below with mathematical details given in section “Strategies for Treating Three Types of Singularities” of the Supplemental Information.

Cycles Are Resolved by Considering Global Dominance Equations

Cycles of reactions are a common feature of biochemical systems. They typically have a number of input and output fluxes. A simple example is the fumarate nitrate reduction (FNR) regulator of *Escherichia coli* that exists in a cycle with three forms having one influx and two effluxes (Tolla et al., 2015). The FNR global regulator is responsible for sensing the environment, more specifically the availability of O_2 . It modifies the global gene expression of the cell to adapt its machinery for the transition from aerobic to anaerobic growth. Let us consider the simple system shown in Figure 1A, in which species X_1 , X_2 , and X_3 interact to form a substrate cycle driven far from thermodynamic equilibrium. This example is deliberately selected to focus on the mathematical details of the singularity contained in the kinetic equations and on the strategy that resolves it. We start by setting up equations to describe the change in the concentration of each chemical species over time. Mass action kinetics are used to generate the rate laws describing the flux through each reaction. The resulting expressions are then combined by means of Kirchhoff’s node law to generate balance equations for each metabolite in the system.

The Design Space formalism (Savageau et al., 2009) can be applied to decompose this set of equations into different cases, each having a unique set of dominant terms and being valid within a specific region in parameter space. One such case for this system is case number 27, with case signature [22 11 21]. This signature contains three pairs of indices, one for each equation, indicating the identity of the positive and negative term dominating in each equation. Visual inspection of the matrix A_D for this case reveals the presence of a linear dependency among its rows, and, thus, there is no unique steady-state solution. Nevertheless, the system of algebraic equations is consistent and a solution (or set of solutions) can be found by analyzing global dominance conditions on the influxes and effluxes that describe a mass balance around the cycle present in this system.

Once the extended system has been automatically set up, the Design Space formalism can be applied to identify valid sub-cases that resolve the cyclical case. Table S1 shows S-system equations for each one of the six valid sub-cases generated from the extended system. Note the special form of these equations. S-systems originating from the dominance analysis of the global dominance equation are used to replace the differential equation for the pool with the dominant efflux. For instance, the expression $\alpha_{11} - \beta_{11}X_1$ (obtained when the first positive and first negative term in the global dominance equation are dominant) is used to replace the differential equation for X_1 (refer to sub-case 1 in Table S1). Additionally, this expression is scaled to match the coefficient of the negative term in the Full System (i.e., the original set of equations). Consider for instance the expression $\alpha_{11} - 2\beta_{33}X_3$, which is obtained when the first positive and third negative term in the global dominance equation are dominant. Since the coefficient of the negative term in the original equation is 1, the scaled expression $\frac{1}{2}\alpha_{11} - \beta_{33}X_3$ is used to construct sub-case 3 of Table S1.

Owing to the special way in which terms stemming from the global dominance equation are used to construct equations for sub-cases, a three-digit case signature is introduced. This allows for tracking the origin of terms that make up the S-systems of these sub-cases. In addition to the indices of dominant positive and negative terms contained in the traditional two-digit signature, the extended three-digit signature contains the index of the equation from which its positive dominant term originated. Consider, for instance, the case signature for sub-case 27_5 in Table S1 and its associated S-system. The signature [22 312 21] dictates that differential equations for pools X_1 and X_3 are constructed by picking dominant terms in the traditional way, whereas the differential equation for pool X_2 is made from the first positive term of the *third* equation and the second negative term of the *second* equation.

Conserved Moieties Are Handled by Considering the Total Size of Conserved Pools

Pools of metabolites with constant total concentration, on some timescale, are a common feature of complex metabolic and signaling systems. They involve conserved moieties, which are groups of atoms that remain intact in all reactions of a system. AMP, NAD⁺, and NADP⁺ are prominent examples of conserved moieties in energy metabolism (Haraldsdóttir and Fleming, 2016). Consider the simple system in Figure 1C with three components linked by a conservation relationship. As in the case of the cycles in section “Cycles Are Resolved by Considering Global Dominance Equations,” the presence of conservations causes the matrix A_D to be singular, since the three concentrations are not independent. Nevertheless, steady-state solutions can be obtained by discarding one of the differential equations and adding the algebraic constraint that the sum of the three concentrations must equal their conserved amount.

The analysis of the differential-algebraic system using the Design Space formalism involves the usual generation of cases by picking dominant terms for each of the equations of the system. The three cases that result are shown in Table S2. Note that each case is defined by only two differential equations and one algebraic constraint. In order to capture the special way in which the equations are constructed for each case, the indices of the differential equation being deleted are set to zero in the case signature. Case 3, for instance, in which the differential equation for pool X_3 is missing, has a case signature of [11 11 00 13] to reflect this fact.

Metabolic Imbalances Are Treated by Considering Knife-Edge Conditions

Flux imbalances are frequently encountered in the metabolism of engineered microbial strains (Dahl et al., 2013; George et al., 2014; Alonso-Gutierrez et al., 2017) and in inborn metabolic diseases such as phenylketonuria (Levy, 1999) and maple syrup urine disease (Haymond et al., 1973), often by the excretion of some metabolite. Let us consider the simplest example of a metabolic pool as shown in Figure 1E with one input (v_0) and one output (v_1) flux, where v_0 is a constant and the output flux is described by a Michaelis-Menten rate law $v_1 = (V_M X_1 K_M^{-1}) (1 + X_1 K_M^{-1})^{-1}$. K_M and V_M represent the Michaelis constant and maximal reaction rate, respectively. The change in concentration of metabolite X_1 over time can be described by the generalized mass action system

$$\frac{dX_1}{dt} = v_0 - V_M K_M^{-1} X_1 D^{-1} \quad (\text{Equation 11})$$

$$0 = 1 + K_M^{-1} X_1 - D. \quad (\text{Equation 12})$$

where D represents an auxiliary variable introduced in the recasting process to describe the denominator of the Michaelis-Menten rate law. Let us now consider the equations for the case with signature [11 21]:

$$\frac{dX_1}{dt} = v_0 - V_M K_M^{-1} X_1 D^{-1} \quad (\text{Equation 13})$$

$$0 = K_M^{-1} X_1 - D, \quad (\text{Equation 14})$$

which together with its associated dominance condition

$$K_M^{-1} X_1 > 1, \quad (\text{Equation 15})$$

imply

$$\frac{dX_1}{dt} = v_0 - V_M. \quad (\text{Equation 16})$$

This system does not have a steady-state solution. Indeed, Equation 16 only provides a consistency condition for the concentration of X_1 to remain unchanged over time: $0 = v_0 - V_M$. We will refer to this kind of constraint as a *knife-edge condition*. In general, we are interested in the behavior of the system when knife-edge conditions are not satisfied, i.e., $v_0 \neq V_M$. Violating the knife-edge condition in a specific direction implies an extreme value for X_1 : $X_1 \rightarrow \infty$ or $X_1 \rightarrow 0$. The validity of either situation is assessed by checking the validity of the associated dominance conditions, as shown in Table S3. Taken together, these results indicate that, for the system shown in Figure 1E, the concentration of the pool X_1 will steadily increase over time, i.e., it will *blow up* if the system's parameters fulfill the conditions $K_M^{-1} X_1 > 1$ and $v_0 V_M^{-1} > 1$.

Note that the case with signature [11 11], and associated dominance condition, has a conventional steady-state solution given by $X_1 = v_0 K_M V_M^{-1}$ for $K_M^{-1} X_1 < 1$. The general procedure for treating cases with multiple knife-edge conditions is presented in section “Metabolic Imbalances are Treated by Considering Knife-Edge Conditions” of the Supplemental Information.

Analysis of a Biochemical System Exhibiting Multiple Singularities

For a case study, we have selected the protocatechuate degradation pathway of *Acinetobacter* sp. strain ADP1. Protocatechuate is a chemical compound derived from phenolic species including p-cresol, 4-hydroxybenzoate, and numerous lignin monomers and is converted to β -keto adipate by the pathway under study (Harwood and Parales, 1996). The protocatechuate degradation pathway is one of the two branches of the β -keto adipate pathway and is widely distributed among taxonomically diverse eubacteria and fungi (Harwood and Parales, 1996). The ecological importance of the β -keto adipate pathway resides in its key role in recycling vast amounts of aromatic material in the natural carbon cycle. From an industrial point of view, the protocatechuate degradation pathway plays a central role in the valorization of lignin for the production of fuels, chemicals, and materials (Linger et al., 2014; Salvachua et al., 2015) as well as in the detoxification of environmental pollutants (Harwood and Parales, 1996). There are many microbes capable of these industrially important processes in the context of toxic environmental hydrocarbons. Although they may have similar if not identical pathways for these functions, their genomic architectures exhibit major differences that are not well understood (Jiménez et al., 2002; Harwood and Parales, 1996). One of the simplest of these architectures for the transport and catabolism of protocatechuate is found in *Acinetobacter* sp. strain ADP1. Its genetic system encodes a single polycistronic mRNA for transporter and catabolic enzymes of the protocatechuate specific pathway, as well as several shared enzymes (Trautwein and Gerischer, 2001; Dal et al., 2005). For our purposes, we shall focus only on the protocatechuate specific pathway illustrated in Figure 2A (refer to Table S5 in the Supplemental Information for a summary of symbols used in this section). The system is composed of a signaling cascade, a gene circuit, and a metabolic module. The transcription factor (PcaU) functions as both a repressor (U_1) and an activator (U_2) and has a conserved total concentration of $U_T = U_1 + U_2$. It controls the synthesis of a polycistronic mRNA (M_3) that encodes the transporter, PcaK (T_4), and several enzymes, three of which constitute this pathway: protocatechuate 3,4-dioxygenase, PcaGH (GH_5), β -carboxy-*cis,cis*-muconate cycloisomerase, PcaB (B_6), and 4-carboxymuconolactone decarboxylase, PcaC (C_7). The environmentally supplied substrate protocatechuate (P_0) is transported into the cell where it becomes intracellular protocatechuate (P_8), which is both a metabolic intermediate and the natural inducer for the transcription factor. The following metabolites in the pathway are β -carboxy-*cis,cis*-muconate (CM_9) and γ -carboxymuconolactone (CL_{10}) (Dal et al., 2005).

This system exhibits all three types of singularities discussed in section “Strategies for Treating Three Types of Singularities,” thus making it an excellent example for demonstrating the relevance of our work. It contains a conservation relationship from the transcription factor, cycles from the reversible enzymatic reactions within the metabolic pathway, and blow-ups from imbalances within the same pathway. The dynamics of the system can be described by a set of differential algebraic equations involving 30 parameter values, which can be considered unknown for our purposes here. Refer to Equations S34–S45 in the Supplemental Information for details on the mathematical model and to Tables S5 and S6 for an overview of the symbols used and their biological context.

Filtering the Phenotypic Repertoire for Phenotypes of Interest

Enumerating the phenotypic repertoire of a system is typically the first step in the phenotype-centric modeling strategy (Valderrama-Gómez et al., 2018). Even systems of moderate size can exhibit a surprisingly large number of biochemical phenotypes. Therefore, the second important step is to filter the repertoire for the phenotypes of interest. For example, filtering for cases with 2 eigenvalues with positive real part can be used to identify oscillatory phenotypes (Lomnitz and Savageau, 2014), filtering for cases with 1 eigenvalue with positive real part can be used to identify multi-stability and hysteresis (Fasani and Savageau, 2013), and filtering for a logical function consisting of a pattern of dependent variables that increase, decrease, or remain unchanged in response to a change in an independent variable can be used for model discrimination (Lomnitz and Savageau, 2016a). Since all of the phenotype characteristics can be exported from DST3 to an Excel spreadsheet, many types of user-defined filters can be customized to meet the user’s needs (see part 3 of the tutorial contained within the DST3 Docker image for an example).

Here, we show how one can progressively filter the repertoire of the protocatechuate system to narrow the focus on phenotypes of interest. If we allow for all possibilities, DST3 shows that the system represented in Figure 2A is capable of exhibiting 3,450 valid phenotypes, of a total of 10,368 potential phenotypes (Figure S7). However, we can progressively filter this list automatically to include only those phenotypes of interest. In the context of this case study, we are interested in the steady states of this system that maximizes

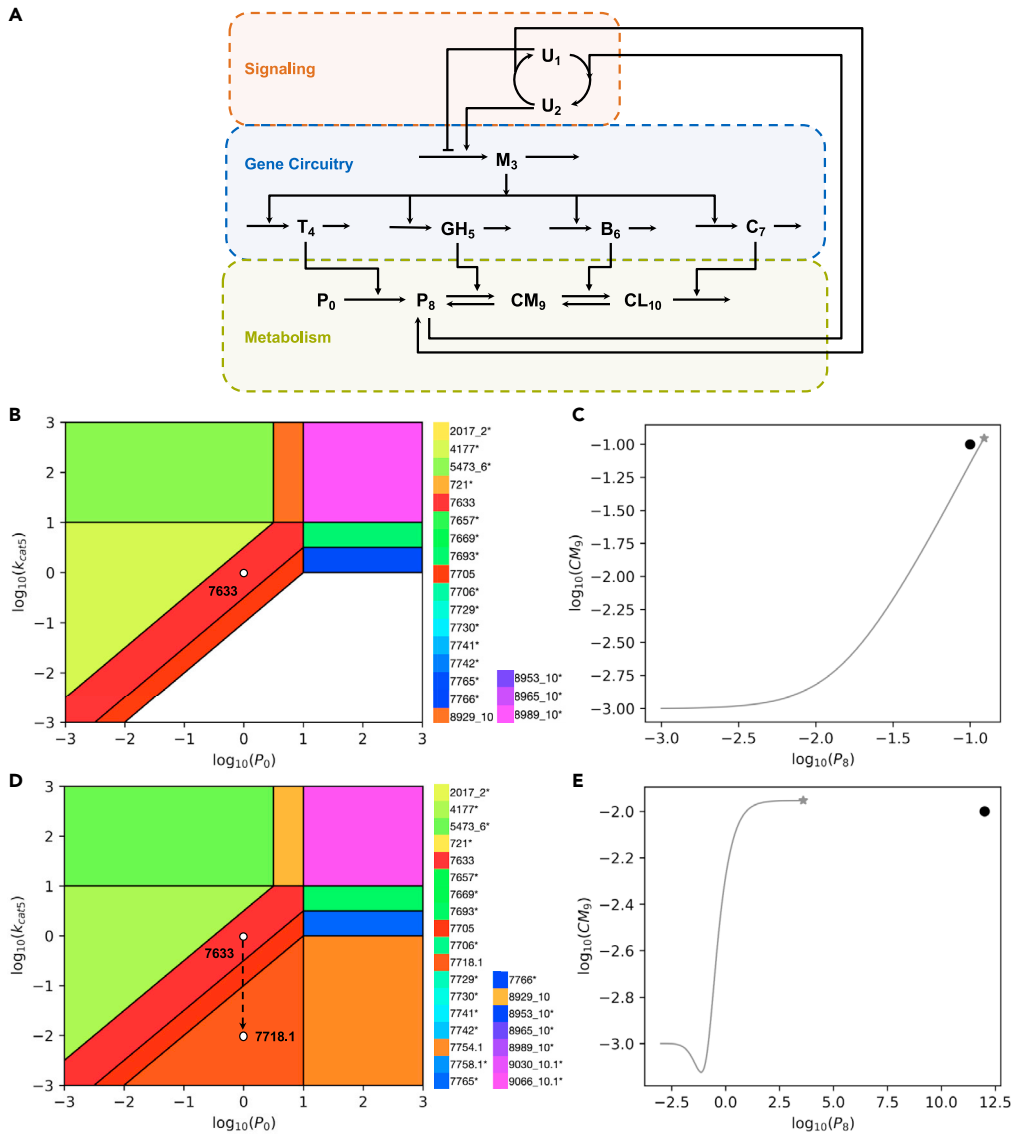


Figure 2. Integrated System Exhibiting Multiple Singularities and its Characterization by DST3

(A) The signaling module responds to the inducer signal P_8 , which stimulates the conversion of the transcription factor from the repressor U_1 into the activator U_2 form. The regulator controls the synthesis of a polycistronic mRNA molecule M_3 , from which four proteins T_4 , GH_5 , B_6 , and C_7 are translated. Transporter T_4 catalyzes the import of metabolite P_8 into the cell from an external pool P_0 . Enzymes GH_5 and B_6 then catalyze the reversible conversion of P_8 into CM_9 and CM_7 into CL_{10} . The last enzyme C_7 catalyzes the conversion of CL_{10} into the end-product β -ketoacid enol-lactone (not shown).

(B) A Design Space Plot around phenotype 7633 is shown. The white dot represents the operating point of the system, which is automatically calculated using the Analyze Case tab of the DST3 user interface. Parameter values predicted for this operating point are: $K_1 = 0.316227766017$, $K_2 = 0.1$, $K_{M0} = 10.0$, $K_{MSf} = 1.0$, $K_{MSr} = 1.0$, $K_{M6f} = 1.0$, $K_{M6r} = 1.0$, $K_{M7} = 1.0$, $K_{eq5} = 10.0$, $K_{eq6} = 1.0$, $U_T = 1.0$, $P_0 = 1.0$, $\alpha_1 = 1.0$, $\alpha_{3\text{basal}} = 0.1$, $\alpha_{3\text{max}} = 1.0$, $\alpha_{3\text{min}} = 0.01$, $\alpha_4 = 1.0$, $\alpha_5 = 1.0$, $\alpha_6 = 1.0$, $\alpha_7 = 10.0$, $\beta_1 = 1.0$, $\beta_3 = 1.0$, $\beta_4 = 1.0$, $\beta_5 = 1.0$, $\beta_6 = 1.0$, $\beta_7 = 1.0$, $k_{cat4} = 1.0$, $k_{cat5} = 1.0$, $k_{cat6} = 1.0$, $k_{cat7} = 1.0$; Kinetic order(s): $m = 2$, $p = 2$; Parametric constraints: $\alpha_{3\text{max}} > \alpha_{3\text{basal}} > \alpha_{3\text{min}}$.

(C) A Trajectories Plot is used to characterize the operating point shown in (B). The solid gray line represents the evolution of the system starting from an initial state in which all concentrations are set to 0.001. The gray star represents the steady state reached by the system after numerical integration for 500 time units. Refer to Table S7 for further details on the numerical solvers used by DST3. The black dot next to the gray star represents the steady state predicted by DST3 for phenotype 7633 using linear algebra.

(D) The operating point of the system has been modified by decreasing k_{cat5} from 1.0 to 0.01 so that it is now contained within the region of the pathological phenotype 7718.1.

Figure 2. Continued

(E) A Trajectories Plot is used to characterize the operating point shown in (D). The temporal behavior of metabolite pools P_8 and CM_7 is shown. CM_7 reaches a steady state of approximately 0.01, whereas the concentration of P_8 continuously increases over time and does not reach a steady state. The gray star represents the state of the system after numerical integration for 50,000 time units. When integrated for a longer period of time, the gray star continues moving further to the right at a constant CM_7 concentration, till it eventually reaches and passes the location of the black dot. Initial conditions are the same as in (C).

the pathway flux, while minimizing the accumulation of toxic intermediates. First, if we filter for phenotypes that are non-pathological by not checking for blow-ups (Figure S5), i.e., that do not have imbalances resulting in concentrations that continuously increase or decrease, then the number of non-pathological phenotypes is 384 and they are all stable (all eigenvalues have a negative real part). Second, if we filter these for phenotypes that respond to changes in the environmental substrate P_0 , by requiring a non-zero logarithmic gain in metabolite concentrations and pathway flux in response to a change in substrate, then there are only 192 responders. Third, if we filter these for phenotypes that are inducible, by requiring a non-zero logarithmic gain in mRNA in response to changes in substrate, then there are only 64 inducible responders. Finally, if we group the inducible phenotypes for specific non-zero logarithmic gains in mRNA, then we find only three values: $L(M_3, P_0) = 2$ with 32 examples, $L(M_3, P_0) = 4$ with 28 examples, and $L(M_3, P_0) = 6$ with 4 examples.

Following this initial screening, the DST3 can be used to characterize automatically the inducible responders by comparing them on the basis of three functional criteria: global robustness to a change in phenotype, energy index (maximum flux with minimum production of protein machinery), and toxicity index (maximum flux with minimum accumulation of toxic intermediates). Global robustness is determined by the product of the global tolerances (Coelho et al., 2009) for all of the parameters of the system, which is a proxy for the volume of the phenotype's polytope in the system Design Space—note that the ability to automatically compute phenotypic volumes is exclusive to DST3. We define the energy index as the cost/benefit determined by the ratio of the logarithmic gain in mRNA, which is a proxy for the increased expenditure of energy for protein production, to the logarithmic gain in the pathway flux produced, $energy\ index = L(M_3, P_0)/L(F, P_0)$. The toxicity index is the cost/benefit determined by the ratio of the logarithmic gain in the toxic intermediate, protocatechuate (P_8), to the logarithmic gain in the pathway flux produced, $toxicity\ index = L(P_8, P_0)/L(F, P_0)$.

The results summarized in Table 1 show that 32 of the 64 phenotypes have the best global robustness, best energy index, and best toxicity index. The next 28 phenotypes have intermediate values for these three criteria and the remaining 4 phenotypes have the worst global robustness and worst energy and toxicity indices. There is a clear trade-off revealed by this analysis. The pathway flux can be increased by moving from the phenotypes in the first group to those in the third group, but only by sacrificing global robustness, energy efficiency, and toxicity. Note that generation of the results in Table 1 did not require any kind of simulation or optimization. Rather, it involved the application of linear algebra to analyze phenotype-specific properties, such as logarithmic gains and phenotypic volumes, that were used to compare a given set of biochemical phenotypes using three different criteria: global robustness as well as energy and toxicity indices. After values for these criteria were calculated, the set of phenotypes were distributed into three different groups. Then, these groups were sorted by its total normalized volume (global robustness).

Phenotype Analysis Reveals Dynamic Properties of the System

Let us consider the phenotype numbered 7633 as a representative member of the best class, which could conceivably have been selected in nature. We start our analysis by using DST3 to predict specific values for each one of the 30 parameters required to fully define an operating point of the system within phenotype 7633 along with its phenotypic boundaries. For that, we use the Analyze Case tab (Figure S9A) within the Main Menu of the DST3 user interface. Note that an analogous analysis can be performed for any other phenotype.

Since we have the ability to characterize the Full System exhibiting this phenotype, we can ask if there might be a strategy for further improving its performance (section "Logarithmic Gains Can Guide the Design of Engineering Strategies") or for avoiding dysfunction through rational engineering. An obvious dysfunction occurs when there is a violation of one of the most basic design principles, namely, the maximal velocity of a downstream enzyme should be greater than that of the upstream enzymes in the pathway (Savageau et al.,

	Group		
	1	2	3
Number of phenotypes in group	32	28	4
Representative phenotype	7633	7658	7593
L(M ₃ , P ₀)	2	4	6
L(P ₈ , P ₀)	1	2	3
L(CM ₉ , P ₀)	1	1	2
L(CL ₁₀ , P ₀)	1	1	1
L(F, P ₀)	3	5	7
Normalized volume of representative phenotype	1.94 × 10 ⁻⁸	9.46 × 10 ⁻¹²	9.72 × 10 ⁻¹⁶
Total normalized volume of group	5.61 × 10 ⁻⁷	5.32 × 10 ⁻¹¹	3.40 × 10 ⁻¹⁵
Energy index	0.67	0.80	0.86
Toxicity index	0.33	0.40	0.43

Table 1. Repertoire of 64 Biochemical Phenotypes of the Protocatechuate Pathway that Are Responding to Substrate, Are Stable and Inducible

This table was generated for the system described by Equation S46–S62 of the Supplemental Information using the tab Phenotypic Repertoire of the Main Menu of the user interface of DST3 (Figure S6B) and filtered for desirable phenotypes as described in the main text.

2009). The relation of this pathology to the phenotype 7633 is made evident in a Design Space Plot (Figure 2B) with the turnover number (k_{cat5}) for the enzyme converting intracellular protocatechuate (P_8) to β -carboxy-cis,cis-muconate (CM_9) on the y axis. The relative location of physiologically relevant phenotypes coded by color is shown in Figure 2B. The operating point of the system, which was automatically calculated by DST3, is represented by the white dot within the valid region for phenotype 7633 (red polytope). In this specific view of the Design Space, the white blank region in the lower portion of the plot indicates the location of pathological phenotypes exhibiting metabolic imbalances.

The dynamical behavior of the Full System, when its steady-state operating point is located within the valid region for phenotype 7633, can be studied by simulation using a Trajectories Plot (Figure 2C), one of three types of plots available in the Full System tab of the Main Menu. This plot shows the evolution of the system starting from a set of initial conditions to reach its steady state, which is marked by a gray star in Figure 2C. The black dot in this figure represents the steady-state prediction made by DST3 for phenotype 7633 using linear algebra. The relative position of the black dot and the gray star in the trajectories plot demonstrates the accuracy of DST3 when approximating steady states. The Full System tab also allows dynamic phenotypes (case number and signature) to be reported at each point in the solution when there is a change in dominance conditions; this is currently implemented as an exportable.xlsx file (data not shown). In the specific example of the trajectory shown in Figure 2C, the initial conditions of the system locate its operating point in phenotype 1515. As the concentrations of the pools in the system evolve toward a steady state, the operating point of the system transits through phenotypes 649 and 721 to finally reach its steady state located within phenotype 7633.

DST3 makes it possible to identify the nature of pathological phenotypes. By clicking the “Check for Blowups” option in the construction of the Design Space (Figure S5), we obtain Figure 2D, which has five additional phenotypes displayed (7718.1, 7754.1, 7758.1, 9030_10.1, and 9066_10.1). Note that the ability to identify pathological phenotypes exhibiting metabolic imbalances is unique to DST3. The dynamical nature of the pathological phenotypes contained in the lower portion of the Design Space Plot can be trivially predicted. Decreasing the numerical value of k_{cat5} while keeping all other parameter values constant at the operating point of the system would lead to a steady increase of the toxic intermediate P_8 .

A **Trajectories Plot** (Figure 2E), with k_{cat5} decreased from its nominal value of 1 to 0.01, is used to characterize the dynamics of the Full System when the operating point of the system is located within the region of pathological phenotype **7718.1** (Figure 2D). In Figure 2E, the state of the system after numerical integration using the IDA solver (refer to Table S7) is marked by a gray star, whereas the black dot represents the prediction made by DST3 for phenotype **7718.1** using linear algebra (note that the value of 10^{12} is an arbitrarily large value used to signify an approach to infinity). As expected, the numerical analysis of the Full System indicates a continuous accumulation of the toxic intermediate P_8 , whereas the intermediate CM_7 reaches a steady-state value.

In summary, a comparison of the results obtained by numerical integration of the Full System (Figures 2C and 2E) demonstrates the ability of DST3 to predict steady-state values when they exist and predict the blow-up nature of a variable when it does not have a steady-state solution.

Logarithmic Gains Can Guide the Design of Engineering Strategies

Once a stable and globally robust operating point for a given system has been identified, one might be interested in finding strategies to increase the flux through a specific metabolic pathway or to increase the steady-state concentration of certain intermediate metabolites. Here, we exemplify how an analysis of logarithmic gains can be used to identify such strategies. *Logarithmic gains* are amplification factors relating changes in input signals (independent variables) to the resulting changes in output signals (dependent variables). The term *parameter sensitivity* is used instead of logarithmic gain when the effect of varying a parameter on a dependent variable is analyzed. These parameter (in)sensitivities represent the local robustness of a system, in contrast to the global robustness provided by the volume of the valid region of a phenotype in Design Space. Both logarithmic gains and parameter sensitivities are properties that depend exclusively on the kinetic orders of the system and can be calculated for concentrations or fluxes (Savageau, 1971a). DST3 allows the calculation of logarithmic gains and parameter sensitivities using the tab **Analyze Case** of the **Main Menu** in the user interface (Figure S9A). For simplicity, we will use the term logarithmic gain for both logarithmic gains and parameter sensitivities. Table 2 lists logarithmic gains for the phenotype **7633**, the representative phenotype of the first group of phenotypes with desired properties (Table 1).

A number of engineering strategies are contained in Table 2. For instance, interventions increasing the flux through the metabolic pathway without altering the steady-state concentration of the potentially toxic metabolic intermediate P_8 are identified in the third column of Table 2 and are graphically represented in Figure 3A, where each individual arrow represents a different strategy. Note that all these strategies ultimately lead to an increase in the availability of the mRNA molecule M_3 and can be categorized into two groups. The first group contains strategies that *directly* increase the synthesis (by either increasing α_{3max} , decreasing the binding constant K_2 or increasing K_1) and reduce the degradation (by decreasing the rate constant β_3) of M_3 .

The second group encompasses *indirect* strategies that point at increasing the steady-state concentration of the activator form of the transcription factor U_2 by modifying rate constants (P_8 or M_3). An analogous analysis can be done to identify strategies increasing the steady-state concentration of metabolic intermediates without increasing the pathway flux. We use the **Full System** tab within the **Main Menu** of the user interface of DST3 (Figure S11) to demonstrate the validity of these predictions by means of two **Titration Plots**. In each case, the maximal synthesis rate α_{3max} is increased 100-fold and decreased 10-fold from its nominal operating value of 1 and the effect on the pathway flux F (Figure 3B) and on the steady-state concentration of the toxic metabolic intermediate P_8 (Figure 3C) is computed for the Full System. As predicted by a logarithmic gain of $L(F, \alpha_{3max}) = 1$ for phenotype **7633**, increasing α_{3max} leads to an increase in the steady-state flux through the metabolic pathway in the Full System. On the other hand, and as indicated by a logarithmic gain of $L(P_8, \alpha_{3max}) = 0$, increasing or decreasing α_{3max} has no effect on the steady-state concentration of P_8 in the Full System. Thus, *increasing* α_{3max} , which can be experimentally achieved by engineering the promoter region of the polycistronic mRNA or increasing the copy number of the *pca* operon, from its nominal operating value can be used as a strategy to increase the flux through the metabolic pathway without increasing the steady-state concentration of P_8 .

DST3 Efficiently Locates Experimentally Observed Induction Patterns of the System

Trautwein and Gerischer (2001) experimentally characterized the effect exerted by the regulator PcaU from *Acinetobacter* sp. strain ADP1 on the expression of the *pca* genes encoding the protocatechuate pathway

Module	Parameters	Change	Dependent Variables				
			M_3	P_8	CM_9	CL_{10}	F
Signaling	α_1	↓	-2	0	0	0	-2
	β_1	↑	2	0	0	0	2
Gene Circuitry	K_1	↑	2	0	0	0	2
	α_{3max}	↑	1	0	0	0	1
	K_2	↓	-2	0	0	0	-2
	β_3	↓	-1	0	0	0	-1
	α_4	-	2	1	1	1	3
	β_4	-	-2	-1	-1	-1	-3
	α_5	-	-2	-1	0	0	-2
	β_5	-	2	1	0	0	2
	α_6	-	0	0	-1	0	0
	β_6	-	0	0	1	0	0
	α_7	-	0	0	0	-1	0
	β_7	-	0	0	0	1	0
	Metabolism	k_{cat4}	-	2	1	1	1
K_{M0}		-	-2	-1	-1	-1	-3
P_0		-	2	1	1	1	3
k_{cat5}		-	-2	-1	0	0	-2
K_{M5f}		-	2	1	0	0	2
k_{cat6}		-	0	0	-1	0	0
K_{M6f}		-	0	0	1	0	0
k_{cat7}		-	0	0	0	-1	0
K_{M7}		-	0	0	0	1	0

Table 2. Logarithmic Gains for Phenotype 7633

This table was generated using the tab **Analyze Case** of the **Main Menu** in the DST3 user interface. Shown are logarithmic gains for the steady-state concentration of the mRNA molecule M_3 ; the three pathway intermediates P_8 , CM_9 , and CL_{10} ; and the flux through the metabolic pathway F . Parameters with no effect on any of the variables are not shown. A logarithmic gain of 0 indicates no effect. Strategies for changing parameters that increase the pathway flux (F) while keeping the intracellular protocatechuate concentration (P_8) unaltered are identified by an arrow in the third column. A minus sign indicates the lack of an applicable strategy. Refer to [Tables S5](#) and [S6](#) for the biological context of the symbols used in this table.

studied in this work. Based on expression experiments under different conditions, Trautwein and Gerischer suggested a bifunctional nature of the transcription factor PcaU, concluded that protocatechuate-dependent regulation of the *pca* genes only occurs in the presence of a functional PcaU, and quantitatively characterized the degree of induction of the pathway under different growth conditions. The authors measured a 94-fold induction of the pathway, when the *pca* gene expression was compared between growth on succinate and growth on *p*-hydroxybenzoate—note that *Acinetobacter* sp. strain ADP1 degrades *p*-hydroxybenzoate via protocatechuate, which induces *pca* gene expression by activating the regulator PcaU. On the other hand, succinate or its degradation products do not induce *pca* gene expression. This value represented the highest degree of induction observed by the authors among different growth conditions.

[Trautwein and Gerischer \(2001\)](#) also studied the effect that PcaU had on *pca* gene expression in the absence of the inducer protocatechuate. This analysis involved a wild-type and a mutant strain missing

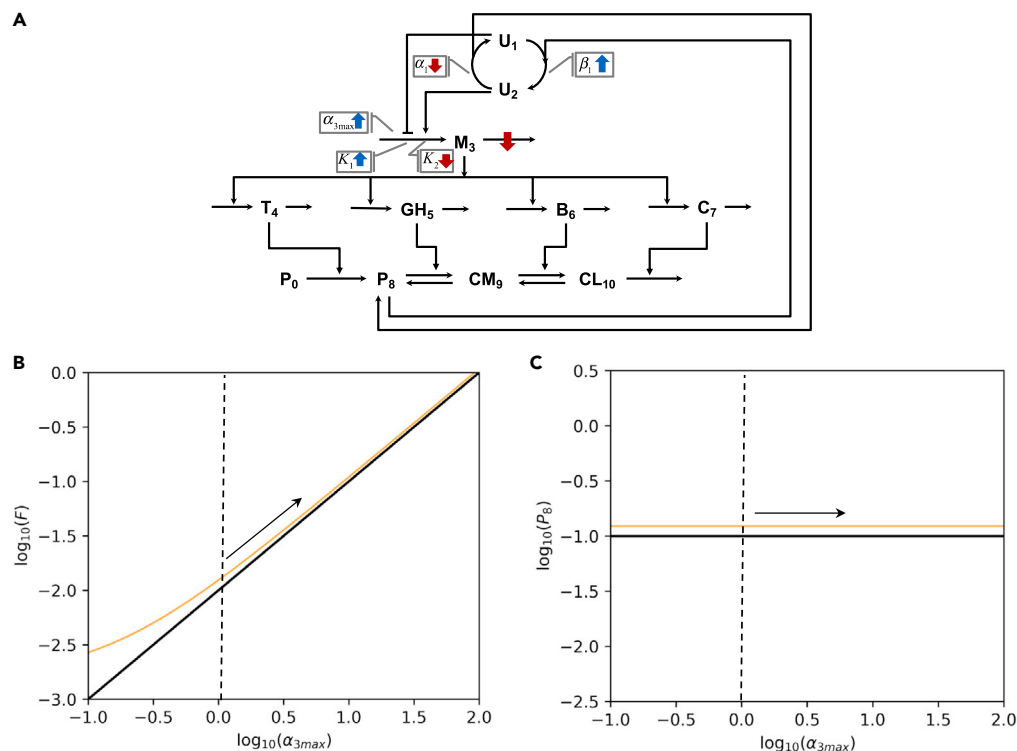


Figure 3. Engineering Strategies for Phenotype 7633

(A) Six different strategies to increase the flux through the metabolic pathway without increasing the steady-state concentration of P_8 are shown. Each colored arrow represents an individual strategy. An arrow placed directly over a synthesis or a degradation flux targets its associated rate constant: α for synthesis and β for degradation. Arrows located within gray boxes usually target kinetic properties of an enzyme or a process. Blue upward arrows symbolize increase, whereas red downward arrows represent decrease. All of the strategies represented in the figure are biologically feasible. However, modifying kinetic properties of a given enzyme or process requires, in most of the cases, a greater experimental effort than modifying its synthesis rate. The effects of perturbing the maximal synthesis rate of the mRNA molecule M_3 (α_{3max}) on the flux through the metabolic pathway F and the steady-state concentration of P_8 are shown in (B) and (C), respectively. The operating point of the system is the one depicted in Figure 2B by the white dot. Vertical dashed lines represent the nominal value of α_{3max} , from which the system is perturbed. The black solid line represents the behavior of phenotype 7633. The generation of each Titration Plot for the Full System involved numerical integration for 100 different values of α_{3max} within the range [0.1 100]. For each point, the system was integrated for 500 time units. To test its stability, the system was integrated for increasing (blue solid line, not shown since covered by orange line) and decreasing α_{3max} values (orange line). Since the system follows the same path when integrated forward and backward, it exhibits mono-stable behavior, as predicted by DST3 based on the number of eigenvalues with positive real part for phenotype 7633. Discrepancies in the location of the solid black and orange lines are due to the simplifications made by the Design Space formalism to generate mathematical expressions for phenotype 7633. However, note that the slope of the black line accurately describes the slope of the orange line.

two-thirds of the *pcaU* gene. After growth on succinate, pyruvate, and acetate, the $\Delta pcaU$ derivative strain expressed between 2- and 3.5-fold higher enzyme levels than the wild-type under the same conditions, thus suggesting that PcaU can act as a repressor in the absence of high concentrations of the inducer protocatechuate.

In section “Filtering the Phenotypic Repertoire for Phenotypes of Interest,” we filtered the phenotypic repertoire exhibited by the network represented in Figure 2A to identify 64 phenotypes of interest that were inducible and responded to changes in the concentration of the inducer protocatechuate. Here, we ask the question whether these phenotypes are able to exhibit induction patterns that match the experimental observations made by Trautwein and Gerischer (2001). Once again, we focus our analysis around phenotype 7633, which is a representative phenotype for the group 1 in Table 1. Figure 4A shows a Design Space plot centered on this phenotype. The x axis corresponds to the logarithm of the concentration of the environmentally supplied substrate protocatechuate P_0 , whereas the y axis corresponds to the logarithm of

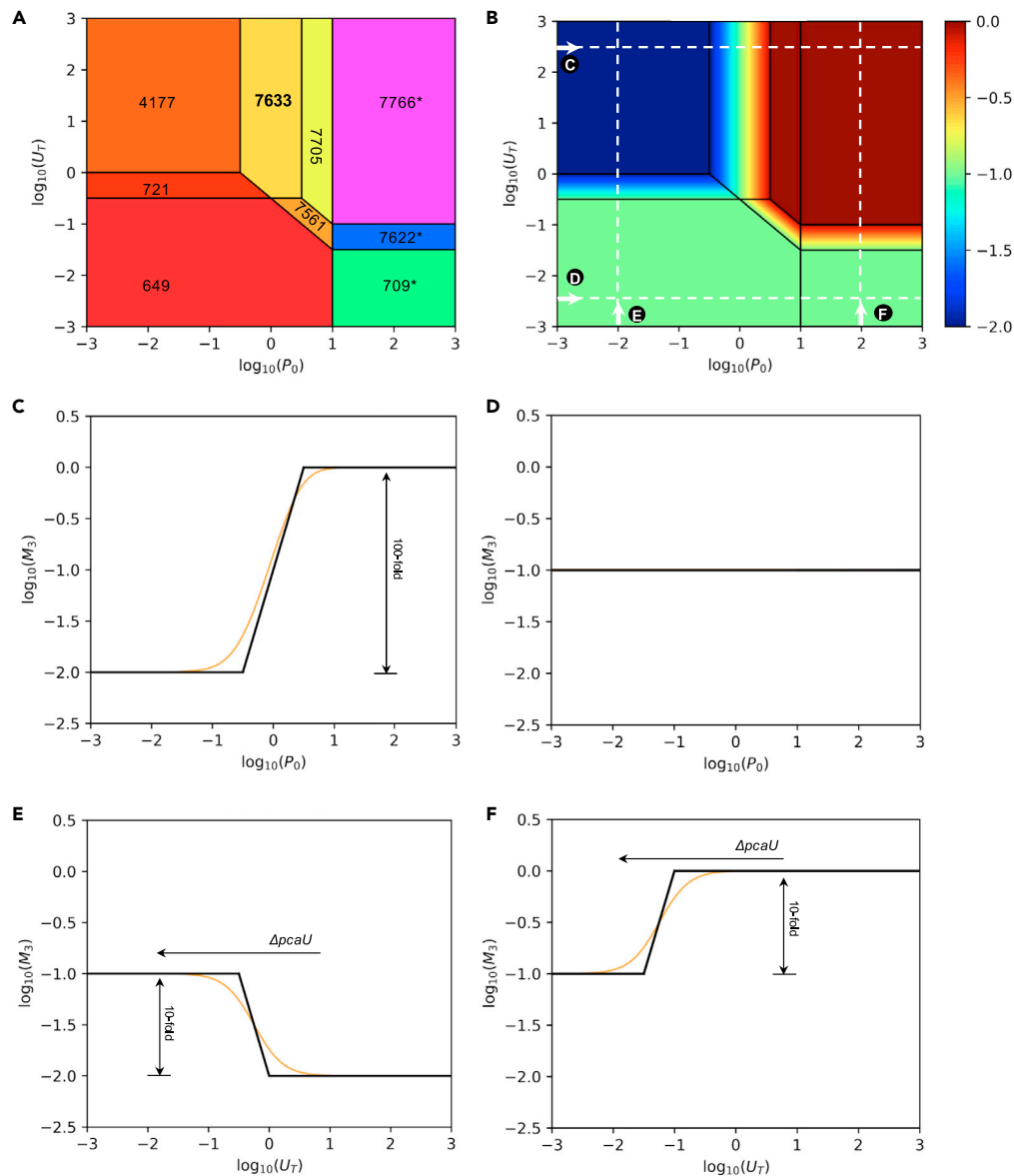


Figure 4. Induction Patterns of the Protocatechuate Pathway Around Phenotype 7633

(A) A Design Space plot around phenotype **7633** is shown. An asterisk next to the phenotype number denotes that at least one boundary condition of the phenotype is on the edge. Parameter values are the same as in [Figure 2A](#).

(B) The logarithm of the steady-state concentration of the mRNA molecule M_3 , a proxy for the expression level of the protocatechuate pathway, is shown as a heatmap. These concentrations were calculated using S -system equations for each one of the phenotypes shown in (A). This plot was created through the **Create Plot** tab of the **Main Menu** of DST3 ([Figure S10](#)). As mentioned before, the calculation of steady states within the Design Space formalism only involves matrix operations, rather than numerical integration of the Full System with ODE or DAE solvers. Additionally, note that (B) represents an overview of the potential expression patterns exhibited by the protocatechuate pathway as a function of the external inducer concentration P_0 and the level of the regulator U_T . Four specific induction patterns (denoted by the white dashed lines in [B]) are visualized by means of titration plots.

(C–F) (C) and (D) represent titration plots at two different constant levels of U_T ($\log_{10}U_T = +2.5$ and -2.5 , respectively). On the other hand, (E) and (F) represent titration plots at two different constant levels of the environmentally supplied inducer protocatechuate ($\log_{10}(P_0) = -2.0$ and $+2.0$, respectively). All titration plots were generated using the parameter values of [Figure 2B](#) and show the fold change between the lowest and highest level of the concentration of the mRNA molecule M_3 . Black solid lines in each titration plot represent the behavior of specific phenotypes of the system.

Figure 4. Continued

The generation of each titration plot for the Full System (orange lines) involved numerical integration for 100 different values of the variable shown in the x axis P_0 or U_T , within the range [0.001 1000]. For each step within this range, the system was integrated for 5,000 time units.

the total amount of the regulator PcaU present in the system (U_T). Parameter values are the same as in Figure 2B.

To obtain a global view of the expression pattern exhibited by the system around phenotype **7633**, we generated a **Steady State Concentration Plot** (Figure 4B) of the mRNA molecule M_3 . This plot has the same axes as Figure 4A but shows the logarithm of the concentration of the mRNA molecule M_3 as a heatmap. Solid black lines in Figure 4B represent boundaries between phenotypes, whose identity can be obtained from Figure 4A. Detailed induction patterns were extracted from Figure 4B by a series of two **Titration Plots** at different constant levels of the regulator PcaU (Figures 4C and 4D) and two **Titration Plots** at constant levels of P_0 (Figures 4E and 4F). The parametric trajectories followed by each one of these titration plots are denoted by the white dashed lines in Figure 4B.

Taken together, these four titration plots correctly capture key experimental observations made by Trautwein and Gerischer (2001). Figure 4C along with Figure 4D clearly demonstrates that the protocatechuate-dependent regulation of the *pca* genes only occurs in the presence of a sufficiently high concentration of the regulator PcaU (U_T). If the concentration of PcaU is too low, or if the regulator is not present in a functional form, increasing the concentration of protocatechuate (P_0) will not lead to a change from the basal state of the system, as shown in Figure 4D by a constant steady-state concentration of the mRNA molecule M_3 . However, if U_T is increased to a sufficiently high value, a 100-fold *pca* gene induction by P_0 is observed (Figure 4C). Interestingly, this extent of induction is in agreement with the maximal induction factor experimentally observed by Trautwein and Gerischer (2001).

Evidence for the repressor activity of PcaU at a low protocatechuate level is shown in Figure 4E by an increase in the steady-state concentration of the mRNA molecule M_3 as the total amount of PcaU (U_T) is decreased. Interestingly, a calculated 10-fold repression is in the same order of magnitude as the experimentally observed 2- to 3.5-fold repression. Note that, at a high concentration of the inducer, decreasing the total amount of PcaU (U_T) leads to a decrease in the steady-state concentration of the mRNA molecule M_3 , thus reinforcing the observation that PcaU acts as a repressor only at low concentrations of the inducer protocatechuate.

The ability of the underlying mathematical model to correctly describe experimentally observed induction patterns is not completely surprising, partly because the potential for these features was incorporated during the construction of the model. Rather, the information extracted from Figures 4C–4F confirms the suitability of the model for the mathematical analysis performed in section “[Logarithmic Gains Can Guide the Design of Engineering Strategies](#).” On the other hand, what is indeed surprising is the ability of the Design Space toolbox to quickly identify and locate regions of a 30-dimensional parameter space exhibiting desired properties without using any *a priori* knowledge of the involved parameters.

DISCUSSION

The Design Space Toolbox v.3.0 offers a variety of advantages over its predecessor version. By distributing the software via a Docker image, the installation process of DST3 is reduced to installing Docker itself. All necessary software dependencies and configurations are already contained within the Docker image, so that users can focus on the actual application of the tools to the analysis of biochemical systems. To improve software usability, DST3 comes with a more stable user interface. By integrating ODE and DAE solvers into the user interface (see Table S7 for further details), it is now possible to directly test independently the accuracy of predictions made by the Design Space formalism from within DST3. Additionally, the capabilities of the computational engine of the Design Space Toolbox were extended. DST3 is now able to analyze biochemical systems containing multiple, nested singularities, something that was out of the reach of previous versions of the toolbox. We demonstrated the utility of DST3 by analyzing a case study of an integrated biochemical system consisting of a signaling cascade, a gene circuit, and a metabolic pathway. The system’s topology encoded a cycle, a conservation relationship, and the potential to exhibit blow-up behavior.

We applied a recently developed phenotype-centric modeling strategy (Lomnitz and Savageau, 2015; Valderrama-Gómez et al., 2018) to identify a stable and globally robust operating point of the system. This process involved listing the phenotypic repertoire and filtering it for phenotypes of interest. From a total of 3,450 valid phenotypes, 384 were found to be non-pathological, 192 of these physiological phenotypes were responsive to changes in the concentration of environmental substrate protocatechuate (P_0), and 64 of the responder phenotypes were found to be inducible. The latter fell into three groups based on the steepness of the induction characteristic: $L(M_3, P_0) = 2$ with 32 phenotypes, $L(M_3, P_0) = 4$ with 28, and $L(M_3, P_0) = 6$ with 4. When compared on the bases of three criteria, global robustness, energy efficiency, and toxicity, the first group was best and the third group was worst. An analysis of the volume of the 64 phenotypes with desired properties revealed that their combined volume only accounted for $5.61 \times 10^{-5}\%$ of the total volume of all non-pathological phenotypes identified by DST3 (see Table 1). When pathological phenotypes were considered (phenotypes exhibiting a blow-up behavior), this value decreased to $2.93 \times 10^{-22}\%$. This suggests that desirable phenotypes will have to be actively selected for by nature, since the vast majority of parameter values chosen at random (increased entropy) would produce few desirable phenotypes.

These figures highlight the power of DST3 and the phenotype-centric modeling strategy it enables. Finding the reported operating point for the representative phenotype **7633** and characterizing its robustness and associated boundaries in a 30-dimensional parameter space by means of parameter sampling would have been computationally expensive and impractical. Indeed, current methods based on the ensemble modeling approach (Tran et al., 2008) for robustness analysis (Lee et al., 2014) involve computationally expensive dense parameter sampling and numerical integration by ODE solvers for stability assessment. These approaches require a long computational time for large model ensembles, and they do not allow for a rigorous identification of stability boundaries. On the other hand, the Design Space formalism decomposes the parameter space into a set of polytopes, biochemical phenotypes, whose boundaries and properties are well defined. DST3 not only identifies these phenotypes but also allows the automatic prediction of nominal parameter sets for their realization. This greatly facilitates deterministic simulations of the Full System (Forger and Peskin, 2003; Zhou et al., 2005; Mirsky et al., 2009), which require parameter values, as demonstrated in Figures 2C and 2E, Figures 3B and 3C, and Figures 4C–4F. Similarly, stochastic simulations, which also require parameter values for propensity functions (Drawert et al., 2016), can benefit from the innovations offered by DST3.

DST3 predictions regarding steady states, stability, and blow-up behavior were accurate, as demonstrated by time course, titration, and trajectory plots generated for the Full System. By finding strategies to increase the flux through the metabolic pathway of the system without increasing the steady-state concentration of an intermediate metabolite, we aimed at showing a glimpse of the potential that the Design Space formalism has to offer to the field of *rational* Metabolic Engineering (Bailey, 1991). Further potential applications relate to the ability of DST3 to correctly identify and characterize blow-up phenotypes, which are commonly found in metabolic systems. Often, in the process of strain development, intermediate strains are generated, in which a given intermediate metabolite excessively accumulates or is totally consumed, thus generating a metabolic imbalance within the cell. This decreases strain fitness and can ultimately lead to cellular death (Dahl et al., 2013; George et al., 2014; Alonso-Gutierrez et al., 2017). DST3 is able to identify regions in the parameter space leading to metabolic imbalances and to provide clues to rectify these phenotypes. For instance, consider the operating point of the system shown in Figure 2D, which is located within the blow-up phenotype **7718.1**. Inspection of the Design Space plot around this phenotype indicates that increasing the value of k_{cat5} to values larger than 0.1 would place the operating point of the system within phenotype **7705**, **7633**, **4,177** or **5,473_6**, all of which exhibit a stable, non-pathological steady state. The specific location of the operating point within any of these phenotypes, and the resulting behavior, will depend on the extent of the increase. Alternative, and more practical strategies to rectify the pathological behavior of an operating point located within phenotype **7718.1** include increasing the amount of the PcaGH enzyme by cloning its gene sequence on a controllable plasmid or engineering its ribosomal binding site.

The application of mechanistic models for the identification of metabolic engineering strategies has been rather limited. This has been mainly caused by a lack of knowledge of associated parameter values. As a consequence, constraint-based modeling has been the method of choice applied to rationally guide metabolic engineering strategies (Valderrama-Gómez et al., 2017). By enabling a mechanistic, phenotype-

centric modeling strategy not dependent on parameter values, the Design Space formalism and associated toolbox offers enormous potential for the field of metabolic engineering.

Elucidating biological design principles is another important area for application of the Design Space formalism that was not explored in this work owing to space limitations. In the Design Space, boundaries delimiting biochemical phenotypes are linear functions of the system's parameters in logarithmic coordinates. Thus, design principles can be readily identified in the form of mathematical inequalities involving the parameters of the system. These ideas were applied by [Fasani and Savageau \(2013\)](#) to study properties of toxin-antitoxin systems, which have been linked with the medically relevant persister phenotype exhibited by certain bacterial strains. The study revealed factors affecting the frequency of persisters in the population, such as the overall number of toxin-antitoxin modules and the size and position of the bistable region, a property emerging from the system's architecture.

There are many examples of systems that appear to perform the same function, and yet they exhibit radically different genomic architectures, the reasons for which are poorly understood. An example is provided by the protocatechuate degradation pathway studied in this work. It is one of the two branches of the β -keto adipate pathway, a chromosomally encoded convergent pathway for aromatic compound degradation that is widely distributed in soil bacteria and fungi. Enzyme studies suggest that the pathway is highly conserved in diverse bacteria; however, its regulation and gene organization differ greatly ([Harwood and Parales, 1996](#)). For instance, the pathway genes from *Pseudomonas aeruginosa* and *P. syringae* are arranged in three and four different clusters, respectively. By contrast, all genes are arranged in a single cluster in *Acinetobacter* sp. ADP1 (studied in this work) and in *P. fluorescens* ([Jiménez et al., 2002](#)). It has been suggested that evolutionary processes have shaped moldable aspects of the β -keto adipate pathway to optimally serve diverse lifestyles of bacteria ([Harwood and Parales, 1996](#)). DST3 could be used to compare and contrast inherent aspects of each system, such as its dynamic properties, induction characteristics, and trade-offs regarding energy and toxicity, thus potentially allowing the elucidation of underlying design principles used by nature to create the alternative genomic architectures observed in organisms with different environments and lifestyles.

Limitations of the Study

Even moderate-sized systems are capable of exhibiting a large number of biochemical phenotypes. The ability to filter the repertoire for specific phenotypes is limited by currently available filtering criteria: hysteretic switches, limit cycle oscillations, global robustness, and logical patterns of system-wide increasing and decreasing responses to a given input stimulus. Although we are developing additional criteria, users may need to develop criteria to characterize phenotypes specifically of interest to them. Although the rigorous definition of boundaries between phenotypes is of primary focus in the novel phenotype-centric modeling strategy, three limitations have their origin in the mathematical transformations used to decompose the Full System into a finite set of dominant biochemical phenotypes (represented by S-Systems) and to mathematically analyze them in the logarithmic space. (1) Numerical accuracy, which becomes undefined at boundaries between phenotypes in parameter space where there is no clear pattern of dominance. We denote the extreme case when the operating point of the system is directly located on a phenotypic boundary by a phenotype number with an asterisk. This limitation can be easily overcome by placing the operating point of the system far away from a phenotypic boundary. The Design Space Toolbox provides necessary functionalities to modify the operating point of the system accordingly (see the tutorial contained in the Docker image under /Tutorials/Tutorial_DST3). (2) The logarithmic transformation employed to linearize S-System equations causes trivial solutions to be ignored. In such a solution, at least one of the dependent variables has a steady-state value of zero. We are currently implementing computational routines that address this limitation. (3) The description of some processes, such as the membrane potential, require mathematical functions (e.g., exponential functions) that cannot be transformed trivially into the GMA form, which in the Design Space formalism is used to represent the Full System. An approximation by rational functions is necessary, and in most cases generating the right approximation requires some expertise. We will be addressing this topic in a future study.

Resource Availability

Lead Contact

Further information and requests for resources should be directed to and will be fulfilled by the Lead Contact, Michael A. Savageau (masavageau@ucdavis.edu).

Materials Availability

This study did not generate new unique reagents.

Data and Code Availability

The Docker images used by DST3 are freely available at <https://hub.docker.com/r/savageau/dst3>.

METHODS

All methods can be found in the accompanying [Transparent Methods supplemental file](#).

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at <https://doi.org/10.1016/j.isci.2020.101200>.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the US National Science Foundation, Grant number MCB 1716833.

AUTHOR CONTRIBUTIONS

Conceptualization, M.A.S., R.A.F., J.G.L., and M.A.V.-G.; Methodology, M.A.S. and M.A.V.-G.; Software, J.G.L. and M.A.V.-G.; Validation, M.A.V.-G. and M.A.S.; Investigation, M.A.S. and M.A.V.-G.; Writing, M.A.V.-G. and M.A.S.; Supervision, M.A.S.; Funding Acquisition, M.A.S.

DECLARATION OF INTERESTS

The authors declare no conflict of interests.

Received: January 21, 2020

Revised: May 4, 2020

Accepted: May 21, 2020

Published: June 26, 2020

REFERENCES

- Alonso-Gutierrez, J., Koma, D., Hu, Q., Yang, Y., Chan, L.J.G., Petzold, C.J., Adams, P.D., Vickers, C.E., Nielsen, L.K., Keasling, J.D., and Lee, T.S. (2017). Toward industrial production of isoprenoids in *Escherichia coli*: lessons learned from CRISPR-Cas9 based optimization of a chromosomally integrated mevalonate pathway. *Biotechnol. Bioeng.* **115**, 1000–1013.
- Alves, R., and Savageau, M.A. (2003). Comparative analysis of prototype two-component systems with either bifunctional or monofunctional sensors: differences in molecular structure and physiological function. *Mol. Microbiol.* **48**, 25–51.
- Andersson, C., Führer, C., and Åkesson, J. (2015). Assimulo: a unified framework for ODE solvers. *Math. Comput. Simul.* **116**, 26–43.
- Bailey, J.E. (1991). Toward a science of metabolic engineering. *Science* **252**, 1668–1675.
- Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Oper. Syst. Rev.* **49**, 71–79.
- Coelho, P.M., Salvador, A., and Savageau, M.A. (2009). Quantifying global tolerance of biochemical systems: design implications for moiety-transfer cycles. *PLoS Comput. Biol.* **5**, e1000319.
- Dahl, H.R., Zhang, F., Alonso-Gutierrez, J., Baidoo, E., Bath, T.S., Redding-Johanson, A.M., Petzold, C.J., Mukhopadhyay, A., Lee, T.S., Adams, P.D., and Keasling, J.D. (2013). Engineering dynamic pathway regulation using stress-response promoters. *Nat. Biotechnol.* **31**, 1039–1046.
- Dal, S., Trautwein, G., and Gerischer, U. (2005). Transcriptional organization of genes for protocatechuate and quinate degradation from *Acinetobacter* sp. strain ADP1. *Appl. Environ. Microbiol.* **71**, 1025–1034.
- Drawert, B., Hellander, A., Bales, B., Banerjee, D., Bellesia, G., Daigle, B.J., Douglas, G., Gu, M., Gupta, A., Hellander, S., et al. (2016). Stochastic simulation service: bridging the gap between the computational expert and the biologist. *PLoS Comput. Biol.* **12**, e1005220.
- Fasani, R.A., and Savageau, M.A. (2010). Automated construction and analysis of the design space for biochemical systems. *Bioinformatics* **26**, 2601–2609.
- Fasani, R.A., and Savageau, M.A. (2013). Molecular mechanisms of multiple toxin-antitoxin systems are coordinated to govern the persister phenotype. *Proc. Natl. Acad. Sci. U.S.A.* **110**, 2528–2537.
- Forger, D.B., and Peskin, C.S. (2003). A detailed predictive model of the mammalian circadian clock. *Proc. Natl. Acad. Sci. U.S.A.* **100**, 14806–14811.
- George, K.W., Chen, A., Jain, A., Bath, T.S., Baidoo, E.E.K., Wang, G., Adams, P.D., Petzold, C.J., Keasling, J.D., and Lee, T.S. (2014). Correlation analysis of targeted proteins and metabolites to assess and engineer microbial isopentenol production. *Biotechnol. Bioeng.* **111**, 1648–1658.
- Harwood, C.S., and Parales, R.E. (1996). The β -ketoadipate pathway and the biology of self-identity. *Annu. Rev. Microbiol.* **50**, 553–590.
- Haraldsdóttir, H.S., and Fleming, R.M.T. (2016). Identification of conserved moieties in metabolic networks by graph theoretical analysis of atom transition networks. *PLoS Comput. Biol.* **12**, e1004999.
- Haymond, M.W., Karl, I.E., Feigin, R.D., De Vito, D., and Pagliara, A.S. (1973). Hypoglycemia and maple syrup urine disease: defective gluconeogenesis. *Pediatr. Res.* **7**, 500–508.
- Jiménez, J.I., Miñambres, B., García, J.L., and Díaz, E. (2002). Genomic analysis of the aromatic catabolic pathways from *Pseudomonas putida* KT2440. *Environ. Microbiol.* **4**, 824–841.

- Klipp, E., and Liebermeister, W. (2006). Mathematical modeling of intracellular signaling pathways. *BMC Neurosci.* 7, S10.
- Lee, Y., Lafontaine Rivera, J.G., and Liao, J.C. (2014). Ensemble modeling for robustness analysis in engineering non-native metabolic pathways. *Metab. Eng.* 25, 63–71.
- Levy, H.L. (1999). Phenylketonuria: old disease, new approach to treatment. *Proc. Natl. Acad. Sci. U S A* 96, 1811–1813.
- Linger, J.G., Vardon, D.R., Guarnieri, M.T., Karp, E.M., Hunsinger, G.B., Franden, M.A., Johnson, C.W., Chupka, G., Strathmann, T.J., Pienkos, P.T., and Beckham, G.T. (2014). Lignin valorization through integrated biological funneling and chemical catalysis. *Proc. Natl. Acad. Sci. U S A* 111, 12013–12018.
- Lomnitz, J.G., and Savageau, M.A. (2014). Strategy revealing phenotypic differences among synthetic oscillator designs. *ACS Synth. Biol.* 3, 686–701.
- Lomnitz, J.G., and Savageau, M.A. (2015). Elucidating the genotype-phenotype map by automatic enumeration and analysis of the phenotypic repertoire. *Npj. Syst. Biol. Appl.* 1, 15003.
- Lomnitz, J.G., and Savageau, M.A. (2016a). Rapid discrimination among putative mechanistic models of biochemical systems. *Sci. Rep.* 6, 32375.
- Lomnitz, J.G., and Savageau, M.A. (2016b). Design Space Toolbox V2: automated software enabling a novel phenotype-centric modeling strategy for natural and synthetic biological systems. *Front. Genet.* 7, 118.
- Mirsky, H.P., Liu, A.C., Welsh, D.K., Kay, S.A., and Doyle, F.J., III (2009). A model of the cell-autonomous mammalian circadian clock. *Proc. Natl. Acad. Sci. U S A* 106, 11107–11112.
- Petzold, L.R. (1983). Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *Siam. J. Sci. Stat. Comput.* 4, 136–148.
- Salvachua, D., Karp, E.M., Nimlos, C.T., Vardon, D.R., and Beckham, G.T. (2015). Towards lignin consolidated bioprocessing: simultaneous lignin depolymerization and product generation by bacteria. *Green. Chem.* 17, 4951–4967.
- Savageau, M.A. (1969). Biochemical systems analysis: II. The steady-state solutions for an n-pool system using a power-law approximation. *J. Theor. Biol.* 25, 370–379.
- Savageau, M.A. (1971a). Concepts relating the behavior of biochemical systems to their underlying molecular properties. *Arch. Biochem. Biophys.* 145, 612–621.
- Savageau, M.A. (1971b). Parameter sensitivity as a criterion for evaluating and comparing the performance of biochemical systems. *Nature* 229, 542–544.
- Savageau, M.A., and Voit, E.O. (1987). Recasting nonlinear differential equations as S-systems: a canonical nonlinear form. *Math. Biosci.* 87, 83–115.
- Savageau, M.A., Coelho, P.M.B.M., Fasani, R.A., Tolla, D.A., and Salvador, A. (2009). Phenotypes and tolerances in the design space of biochemical systems. *Proc. Natl. Acad. Sci. U.S.A* 106, 6435–6440.
- Schuster, S., Fell, D.A., and Dandekar, T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat. Biotechnol.* 18, 326–332.
- Strang, G. (2005). *Linear Algebra and its Applications*, 4th edition (Cengage Learning).
- Tolla, D.A., Kiley, P.J., Lomnitz, J.G., and Savageau, M.A. (2015). Design principles of a conditional futile cycle exploited for regulation. *Mol. Biosyst.* 11, 1841–1849.
- Tran, L.M., Rizk, M.L., and Liao, J.C. (2008). Ensemble modeling of metabolic networks. *Biophys. J.* 95, 5606–5617.
- Trautwein, G., and Gerischer, U. (2001). Effects exerted by transcriptional regulator PcaU from *Acinetobacter* sp. strain ADP1. *J. Bacteriol.* 183, 873–881.
- Valderrama-Gómez, M.A., Kreitmayer, S., Wolf, S., Marin-Sanguino, A., and Kremling, A. (2017). Application of theoretical methods to increase succinate production in engineered strains. *Bioproc. Biosyst. Eng.* 40, 479–497.
- Valderrama-Gómez, M.A., Parales, R.E., and Savageau, M.A. (2018). Phenotype-centric modeling for elucidation of biological design principles. *J. Theor. Biol.* 455, 281–292.
- Zhou, L., Salem, J.E., Sidel, G.M., Stainley, W.C., and Cabrera, M.E. (2005). Mechanistic model of cardiac energy metabolism predicts localization of glycolysis to cytosolic subdomain during ischemia. *Am. J. Physiol. Heart Circ. Physiol.* 288, H2400–H2411.

iScience, Volume 23

Supplemental Information

Mechanistic Modeling of Biochemical Systems

without *A Priori* Parameter Values

Using the Design Space Toolbox v.3.0

Miguel Á. Valderrama-Gómez, Jason G. Lomnitz, Rick A. Fasani, and Michael A. Savageau

Supplemental Figures

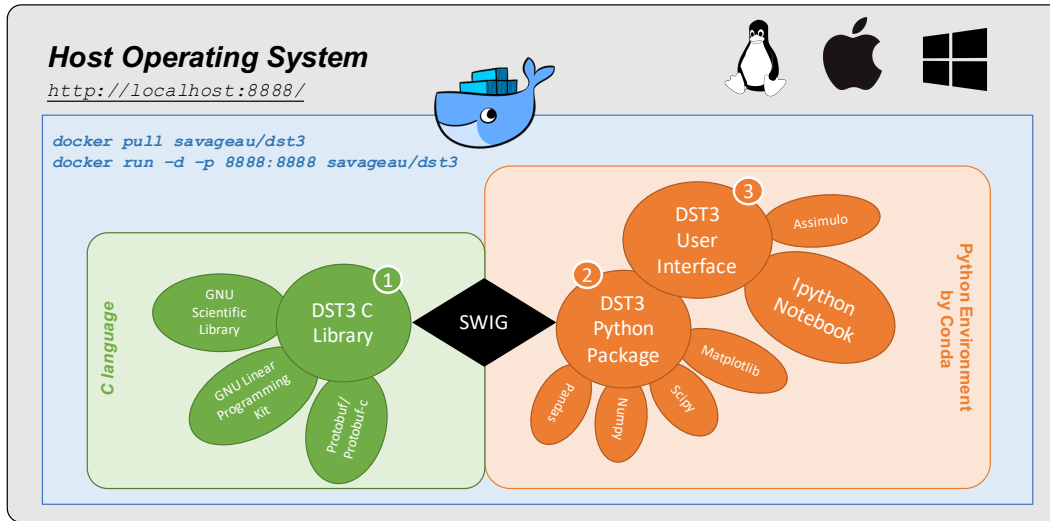


Figure S1. Architecture of DST3 and Its Interaction with the Host Operating System via Docker. Related to Figures 2, 3 and 4. The three components of DST3 are integrated in a layered fashion. The *C library* is the heart of DST3. It leverages the GNU Scientific Library (GSL) to perform numerical computations, specifically matrix operations. A customized version of the GNU Linear Programming Kit (GLPK) is used to solve linear programming problems within DST3. Google protocol buffers (Protobuf/Protobuf-c) are used to write and read data. Access to the C library from Python is provided by SWIG, which stands for Simplified Wrapper and Interface Generator. This allows the creation of a *DST3 Python Package*. The *DST3 User Interface* is based on widgets provided by the IPython Notebook. The Python environment is managed by Conda. The standard Docker Image for DST3 is *savageau/dst3*. Advanced users might prefer *savageau/dst3:python3*, which comes with a DST3 Python Package for Python 3.7.3.

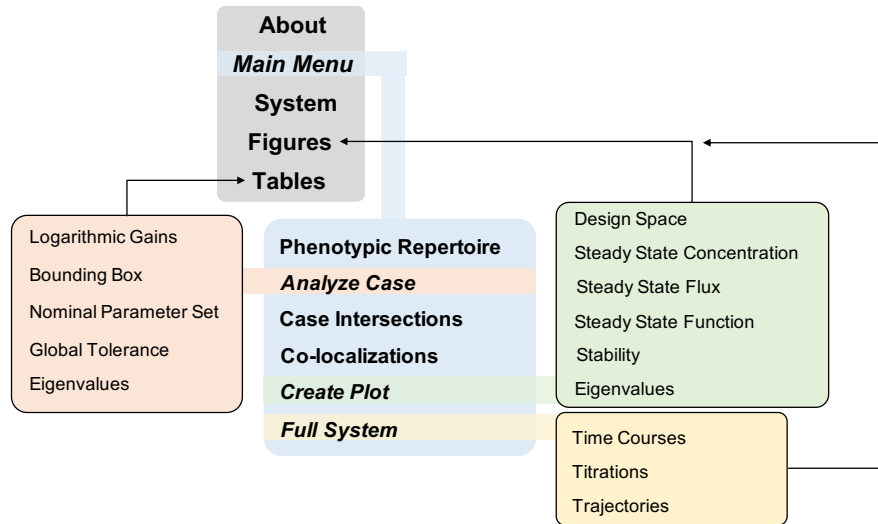


Figure S2. Overview of Menus and Windows Comprising the IPython-based User Interface of DST3. Related to Figures 2, 3 and 4 as well as Table 1 and 2. The user interface consists of a collection of tabs, buttons and text fields that facilitate access to computational tools contained in the DST3 C library. Data can be saved to and loaded from `.dsipy` files. Tables generated from the menu *Phenotypic Repertoire* and *Full System* can be exported to `.xlsx` files for further analysis. Additionally, parameter values can be loaded from tables contained in files with the same extension. The User Interface of DST3 is built on legacy code inherited from DST2. Its portability is guaranteed through the virtualization technology offered by Docker.

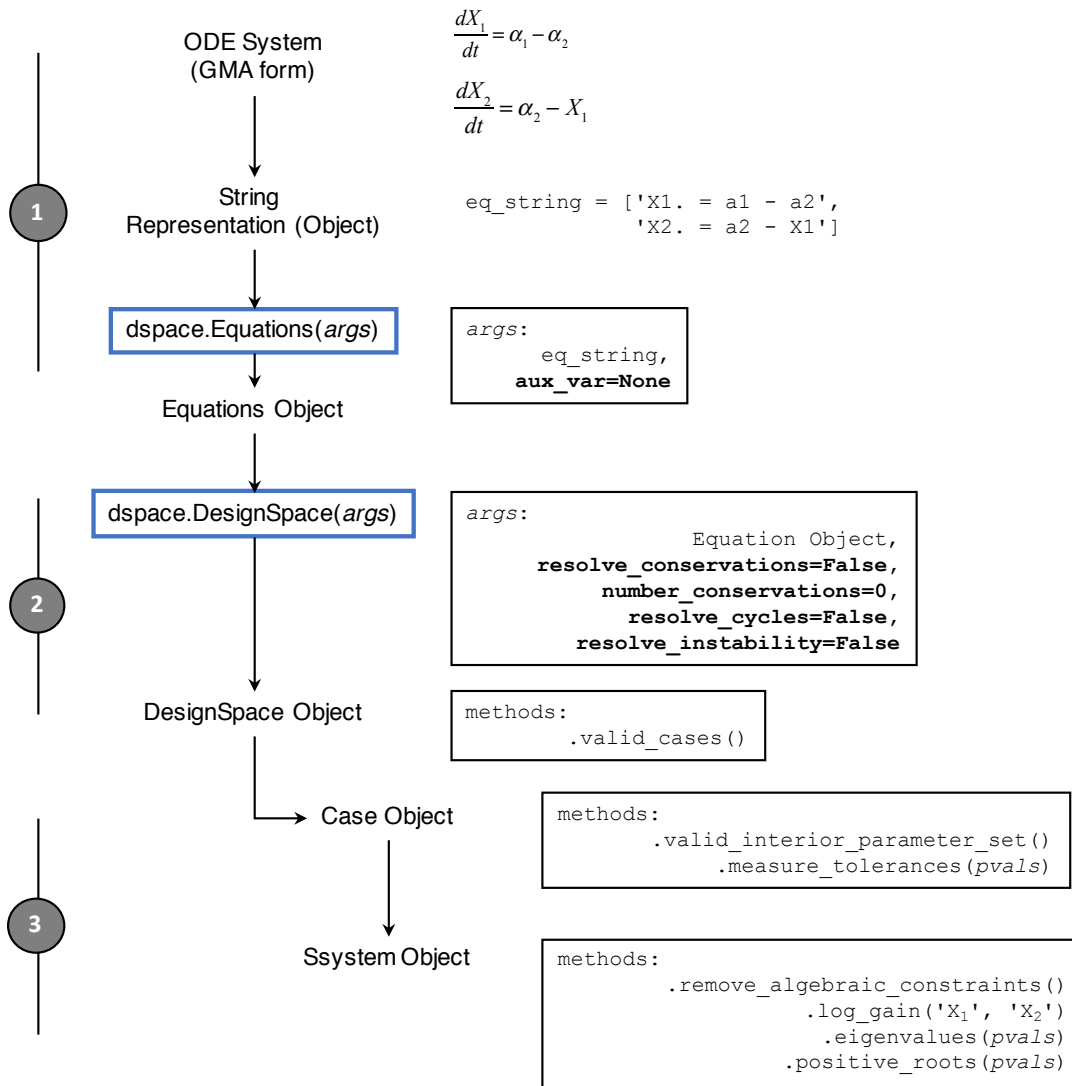


Figure S3. *Phenotypic Deconstruction of a Biochemical System Using the DST3 Python Module.* Related to Figure 1. Three steps are involved in the computational Design Space analysis of any biochemical system. First, an `Equations` Python object is generated by means of the class `dspace.Equations`. This process involves recasting ordinary differential equations describing the systems' dynamics into the GMA form, followed by a further transformation into a list of strings according to syntax rules described in the main text. Auxiliary variables stemming from the recasting process or introduced by conservation constraints need to be declared explicitly using the key argument `aux_var`. In a second step, the `Equations` object is passed to the class `dspace.DesignSpace` along with necessary key arguments to inform the computational engine about the presence of conservations, cycles or metabolic imbalances. The output of this second step is a `DesignSpace` object. Methods associated with this object allow, among other things, the generation of a list of strings containing identifiers of valid cases. In a third step, the `DesignSpace` object can be used to generate `Case` objects using valid case identifiers as input. Each `Case` object contains a respective `SSystem` object. Methods associated with these two objects allow a comprehensive characterization of each valid case, which includes, but is by no means limited to, calculation of interior parameter sets, logarithmic gains, determination of dynamical stability, etc. Refer to Lomnitz and Savageau (2016) and to the documentation contained in the Docker Image of DST3 for more details and usage examples of the module `dspace`.

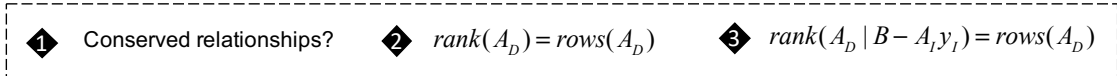
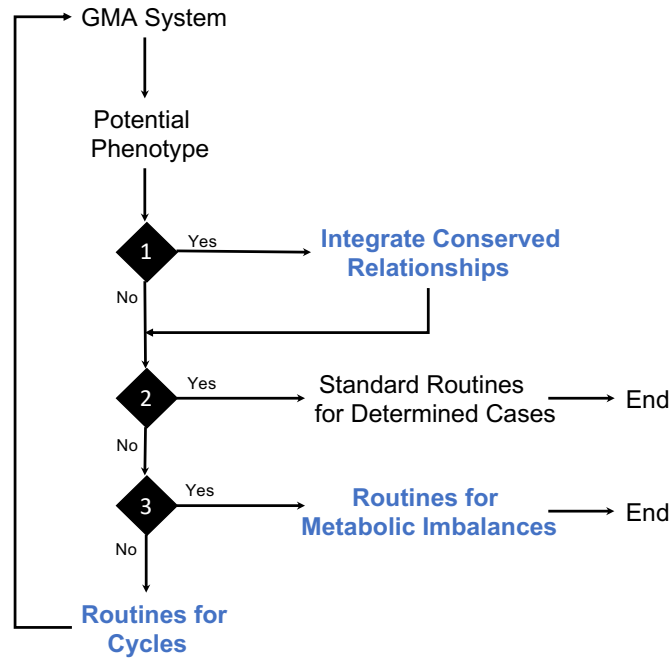


Figure S4. Overview of the computational workflow implemented by DST3. Related to Figure 2. Three main checkpoints define the set of computational algorithms applied to analyze a potential phenotype of a given system. Routines to handle under-determined cases (conserved relationships, metabolic imbalances and cycles) are exclusive to DST3. Refer to Section 2 of the Supplementary Information for details on the mathematical treatment.

× About Main Menu

* Name

Version

* Equations

$$U1. = a1*U2 - b1*P8*U1$$

$$U2. = b1*P8*U1 - a1*U2$$

$$M3. = a3basal*(D3^{n-1}) + a3min*(K1^{n-1})*(U1^n)*(D3^{n-1}) + a3max*(K2^{n-p})*(U2^p)*(D3^{n-1}) - b3*M3$$

$$T4. = a4*M3 - b4*T4$$

$$GH5. = a5*M3 - b5*GH5$$

$$B6. = a6*M3 - b6*B6$$

$$C7. = a7*M3 - b7*C7$$

$$P8. = T4*kcat4*(KM0^{n-1})*P0*(D4^{n-1}) + GH5*kcat5*(KM5f^{n-1})*(Keq5^{n-1})*CM9*(D5^{n-1}) - GH5*kcat5*(KM5f^{n-1})*P8*(D5^{n-1})$$

$$CM9. = GH5*kcat5*(KM5f^{n-1})*P8*(D5^{n-1}) + B6*kcat6*(KM6f^{n-1})*(Keq6^{n-1})*CL10*(D6^{n-1}) - GH5*kcat5*(KM5f^{n-1})*(Keq5^{n-1})*CM9*(D5^{n-1}) - B6*kcat6*(KM6f^{n-1})*CM9*(D6^{n-1})$$

$$CL10. = B6*kcat6*(KM6f^{n-1})*CM9*(D6^{n-1}) - B6*kcat6*(KM6f^{n-1})*(Keq6^{n-1})*CL10*(D6^{n-1}) - C7*kcat7*(KM7^{n-1})*CL10*(D7^{n-1})$$

$$F. = C7*kcat7*(KM7^{n-1})*CL10*(D7^{n-1}) - F$$

$$0 = 1 + (K1^{n-1})*(U1^n) + (K2^{n-p})*(U2^p) - D3$$

$$0 = 1 + (KM0^{n-1})*P0 - D4$$

$$0 = 1 + (KM5f^{n-1})*P8 + (KM5r^{n-1})*CM9 - D5$$

$$0 = 1 + (KM6f^{n-1})*CM9 + (KM6r^{n-1})*CL10 - D6$$

$$0 = 1 + (KM7^{n-1})*CL10 - D7$$

Auxiliary Variables

Conservations 1

Architectural Constraints

Parameters

Kinetic Orders

Additional Options

Check for Cycles 2

Check for Blowups 3

Check for Co-dominance

Figure S5. *Initial Input Window of the DST3 User Interface.* Related to Figures 2, 3 and 4. Relevant information defining the biochemical system under study such as its name, equations, auxiliary variables, parametric constraints and kinetic orders can be passed to the computational engine of DST3 through this window. The conservations field (highlighted by the number 1 in the blue circle), as well as the checkboxes for cycles (highlighted by the number 2 in the blue circle) and metabolic imbalances (e.g., blowing phenotypes, highlighted by the number 3 in the blue circle) can be used to control computational routines used by DST3 to analyze biochemical phenotypes.

A x About Main Menu System Figures Tables

This software is a still under active development and may contain a number of bugs and issues. **If you found a new bug in the stable or experimental version of the design space toolbox; or would like to request an enhancement that has not yet been reported, please report it via email at mvalderramagomez@ucdavis.edu** You are using the Design Space Toolbox V3 (Python Notebook-based widget, a graphical interface for the Design Space Toolbox V3 originally created by Jason Lornitz in the laboratory of Michael A. Savageau and further developed by Miguel A. Valderrama-Gomez

This software has been created for the analysis of mathematical models of biochemical systems. This library deconstructs a model into a series of sub-models that can be analyzed by applying numerical and symbolic methods.

To begin, please enter the information in the required fields (marked by an *******) and optional fields. The Name field is used to save and load a model workspace, which includes the system equations, tables and figures. The primary input into the widget are the system equations, represented by a list of strings.

The equations must satisfy the following rules:

1. Each equation has to be explicitly stated as:
 - 1.1 A differential equation, where the "." operator denotes the derivative with respect to time.
 - 1.2 An algebraic constraint, where the left hand side is either a variable or a mathematical expression. Auxiliary variables associated with the constraint must be explicitly defined (unless the left-hand side is the auxiliary variable).
2. Multiplication is represented by the "*" operator.
3. Powers are represented by the "^" operator.
4. Architectural constraints are defined as inequalities, where both sides of the inequality are products of power-laws.

B x About Main Menu System Figures Tables

Actions

Phenotypic Repertoire Analyze Case Case Intersections Co-localizations Create Plot Full System

All
 Valid
 User Specified

User specified cases

Cases indicated by signature?


Biological constraints:

Show signature?

Show parent cases?

Export Phenotypic Repertoire (.xlsx)

Add column

Column 3 Log-Gain 

Log-gain (dependent variable) U1

Log-gain (independent variable) a1

Validity

Is Blowing

Volume

Normalized Volume

eigenvalues w/ positive real part

Log-Gain

Remove column
 Add column filter
 Done

Options

Edit Symbols
 Edit Parameters
 Save Data

Figure S6. *About and Main Menu Tabs of the DST3 User Interface.* Related to Table 1. **A.** The *About* tab provides general information about DST3, its developers as well as an email address to report bugs. Additionally, syntax rules to be followed for the string representation of the mathematical model are provided. **B.** The *Main Menu* tab provides access to the main functionalities of DST3, which are contained in six different action tabs: Phenotypic Repertoire, Analyze Case, Case Intersections, Co-localizations, Create Plot and Full System. The *Phenotypic Repertoire* tab is shown in this panel. It allows listing and filtering the phenotypic repertoire of the model according to different user-defined criteria (volume, logarithmic gains, etc.). Refer to the tutorial contained within the DST3 Docker Image under **/Tutorials/Tutorial_DST3/Part1_DST3_User_Interface** for additional details on each one of these six action tabs.

× About Main Menu System **Figures** Tables

Name: Protocatechuate_Pathway
Number of Cases: 10368
System Signature: [1111311111111111112221211312131312112]

Equations:

Typeset Equations?

$$\begin{aligned}
 \dot{U}_1 &= \alpha_1 U_2 - \beta_1 U_1 P_8 \\
 \dot{U}_2 &= \beta_1 U_1 P_8 - \alpha_1 U_2 \\
 \dot{M}_3 &= \alpha_{3\text{basal}} D_3^{-1} + \alpha_{3\text{min}} K_1^{-2} U_1^2 D_3^{-1} + \alpha_{3\text{max}} K_2^{-2} U_2^2 D_3^{-1} - \beta_3 M_3 \\
 \dot{T}_4 &= \alpha_4 M_3 - \beta_4 T_4 \\
 \dot{G}H_5 &= \alpha_5 M_3 - \beta_5 G H_5 \\
 \dot{B}_6 &= \alpha_6 M_3 - \beta_6 B_6 \\
 \dot{C}_7 &= \alpha_7 M_3 - \beta_7 C_7 \\
 \dot{P}_8 &= k_{cat4} K_{M0}^{-1} P_0 T_4 D_4^{-1} + k_{cat5} K_{M5f}^{-1} K_{eq5}^{-1} G H_5 C M_9 D_5^{-1} - k_{cat5} K_{M5f}^{-1} G H_5 P_8 D_5^{-1} \\
 \dot{C}M_9 &= k_{cat5} K_{M5f}^{-1} G H_5 P_8 D_5^{-1} + k_{cat6} K_{M6f}^{-1} K_{eq6}^{-1} B_6 C L_{10} D_6^{-1} - k_{cat5} K_{M5f}^{-1} K_{eq5}^{-1} G H_5 C M_9 D_5^{-1} - k_{cat6} K_{M6f}^{-1} B_6 C M_9 D_6^{-1} \\
 \dot{C}L_{10} &= k_{cat6} K_{M6f}^{-1} B_6 C M_9 D_6^{-1} - k_{cat6} K_{M6f}^{-1} K_{eq6}^{-1} B_6 C L_{10} D_6^{-1} - k_{cat7} K_{M7}^{-1} C_7 C L_{10} D_7^{-1} \\
 \dot{F} &= k_{cat7} K_{M7}^{-1} C_7 C L_{10} D_7^{-1} - F \\
 D_3 &= 1 + K_1^{-2} U_1^2 + K_2^{-2} U_2^2 \\
 D_4 &= 1 + K_{M0}^{-1} P_0 \\
 D_5 &= 1 + K_{M5f}^{-1} P_8 + K_{M5r}^{-1} C M_9 \\
 D_6 &= 1 + K_{M6f}^{-1} C M_9 + K_{M6r}^{-1} C L_{10} \\
 D_7 &= 1 + K_{M7}^{-1} C L_{10} \\
 0 &= U_T - U_1 - U_2
 \end{aligned}$$

Figure S7. System Tab of the DST3 User Interface. Related to Figure 2 and Table 1. General information of the system under analysis such as the maximal number of biochemical phenotypes it can exhibit, its signature and its equations are provided within the System tab.

A × About Main Menu System **Figures** Tables

Delete Selected Figures

Figures that will not be saved:

B × About Main Menu System **Figures** Tables

Delete Selected Tables

Figure S8. Figures and Tables Tab of the DST3 User Interface. Related to Figures 2, 3 and 4 and to Table 2. The Figures (A) and Tables (B) tabs contain elements saved in the current working session or loaded from data files. It is possible to visualize elements and delete both figures and tables from these tabs.

A × About Main Menu System Figures Tables

Actions

Phenotypic Repertoire Analyze Case Case Intersections Co-localizations Create Plot Full System

Cases indicated by signature?

* Analysis for case:

Biological constraints:

Create Analysis

Options

Edit Symbols

Edit Parameters

Save Data

B **Actions**

Phenotypic Repertoire Analyze Case Case Intersections Co-localizations Create Plot Full System

* Cases to intersect:

Cases indicated by signature?

Biological constraints:

Create Case Intersection

C **Actions**

Phenotypic Repertoire Analyze Case Case Intersections Co-localizations Create Plot Full System

* Cases to co-localize:

Cases indicated by signature?

* Slice variables:

Biological constraints:

Create Co-Localization

Figure S9. *Analyze Case*, *Case Intersections* and *Co-localizations* tabs of the *DST3* User Interface. Related to Table 2. The *Analyze Case* tab (**A**) allows the generation of a comprehensive analysis of a given phenotype specified by its case number. A central functionality provided by this action tab is the automatic calculation of parameter values for the realization of the specific biochemical phenotype. This set of parameters allows a more directed exploration and analysis of the parameter space and the behavior of the system using the tabs *Create Plot* (Fig. S10) and *Full System* (Fig. S11). The *Case Intersections* (**B**) and *Co-localizations* (**C**) tabs are used to study the relative position of a given set of phenotypes in the parameter space.

[About](#)
[Main Menu](#)
[System](#)
[Figures](#)
[Tables](#)

Actions

[Phenotypic Repertoire](#)
[Analyze Case](#)
[Case Intersections](#)
[Co-localizations](#)
[Create Plot](#)
[Full System](#)

* X-Axis

* Y-Axis

* Plot Type

* X-Min

* X-Max

* Y-Min

* Y-Max

Center Axes

Draw Boundaries

Title

Caption

Only Cases

Split Colorbar

Add Plot

Options

Edit Symbols

Edit Parameters

Save Data

Design Space (interactive)

Design Space

Steady State Concentration

Steady State Flux

Steady State Function

Stability


Eigenvalues

Figure S10. *Create Plot Tab of the DST3 User Interface.* Related to Figure 2, 3 and 4. The *Create Plot* tab allows the generation of seven different types of plots, shown in the blue box. These plots allow the visualization of various properties of the biochemical phenotypes of the system under study, such as their spatial arrangement in the parameter space, their stability properties and their steady states.

[About](#) [Main Menu](#) [System](#) [Figures](#) [Tables](#)

Actions

[Phenotypic Repertoire](#) [Analyze Case](#) [Case Intersections](#) [Co-localizations](#) [Create Plot](#) [Full System](#)

Plot Type: 

Time Courses
 Titrations
 Trajectories

X-Axis:

Y-Axis:

Secondary Y-Axis:

Initial Time Point:

Final Time Point:

Number of Points for Time:

Initial Concentrations

GH5:

CM9:

F:

T4:

U1:

CL10:

U2:

B6:

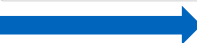
M3:

C7:

P8:

Coordinates:

Calculate Response Time For SS Function:

DAE Solver: 

IDA
 RADAU5

Title:

Caption:

Options

Figure S11. Full System Tab of the DST3 User Interface. Related to Figures 2, 3 and 4. The Full System tab allows the characterization of the dynamic properties of the mechanistic model under analysis using numerical solvers (ODE and DAE). Refer to Table S7 for an overview of the solvers available in DST3. Three different types of plots can be generated in this tab: Time Courses, Titrations and Trajectories plots.

Supplemental Tables

Table S1. *Cyclic Cases, their Signatures and Case Numbers.* Related to Figure 1. Resolving the singularity contained in case 27 (see Fig. 1B) involves a Design Space analysis of the sub-system described by Eqs. S13-S16. This analysis generates six sub-cases, each one of which is generated as dictated by a *three-digit signature*. Note that the reference system of equations from which dominant terms are picked according to the case signature is the Full System (Eqs. S1-S3) and not the sub-system used to resolve the singularity (Eqs. S13-S16). The case numbers for the valid sub-cases have the parent case number with an underscore followed by a number associated with the sub-case; e.g., 27_3.

Sub-case 27_1 [111 11 21]	Sub-case 27_2 [22 112 21]	Sub-case 27_3 [22 11 112]
$\frac{dX_1}{dt} = \alpha_{11} - \beta_{11}X_1$	$\frac{dX_1}{dt} = 2\beta_{31}X_3 - 2\beta_{12}X_1^2$	$\frac{dX_1}{dt} = 2\beta_{31}X_3 - 2\beta_{12}X_1^2$
$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2$	$\frac{dX_2}{dt} = \frac{1}{2}\alpha_{11} - \beta_{22}X_2$	$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2$
$\frac{dX_3}{dt} = \beta_{23}X_2 - \beta_{31}X_3$	$\frac{dX_3}{dt} = \beta_{23}X_2 - \beta_{31}X_3$	$\frac{dX_3}{dt} = \frac{1}{2}\alpha_{11} - \beta_{33}X_3$
Sub-case 27_4 [311 11 21]	Sub-case 27_5 [22 312 21]	Sub-case 27_6 [22 11 312]
$\frac{dX_1}{dt} = 2\alpha_{31} - \beta_{11}X_1$	$\frac{dX_1}{dt} = 2\beta_{31}X_3 - 2\beta_{12}X_1^2$	$\frac{dX_1}{dt} = 2\beta_{31}X_3 - 2\beta_{12}X_1^2$
$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2$	$\frac{dX_2}{dt} = \alpha_{31} - \beta_{22}X_2$	$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2$
$\frac{dX_3}{dt} = \beta_{23}X_2 - \beta_{31}X_3$	$\frac{dX_3}{dt} = \beta_{23}X_2 - \beta_{31}X_3$	$\frac{dX_3}{dt} = \alpha_{31} - \beta_{33}X_3$

Table S2. Conserved Cases and their Signatures. Related to Figure 1. The synthetic network shown in Fig. 1C can be decomposed into three different cases by applying the Design Space formalism. Dominance analysis on the conservation constraint, i.e., Eq. S25, gives rise to each of these cases. Note that each case lacks n_{cr} differential equations, when compared with the Full System (Eqs. S22-S25). The missing differential equation in each case is identified by a pair of zeros in the case signature.

Case 1 [00 11 11 11]	Case 2 [11 00 11 12]	Case 3 [11 11 00 13]
$\frac{dX_2}{dt} = \beta_{11}X_1 - \beta_{21}X_2$	$\frac{dX_1}{dt} = \alpha_{11}X_3 - \beta_{11}X_1$	$\frac{dX_1}{dt} = \alpha_{11}X_3 - \beta_{11}X_1$
$\frac{dX_3}{dt} = \beta_{21}X_2 - \alpha_{11}X_3$	$\frac{dX_3}{dt} = \beta_{21}X_2 - \alpha_{11}X_3$	$\frac{dX_2}{dt} = \beta_{11}X_1 - \beta_{21}X_2$
$0 = CR_1 - X_1$	$0 = CR_1 - X_2$	$0 = CR_1 - X_3$

Table S3. Blow-Up Cases and their Case Numbers for a Single Knife-Edge Condition. Related to Figure 1. Each sub-case is defined by the violation of its knife-edge condition in one of two possible directions. Sub-case 2.1 is valid because $X_1 \rightarrow \infty$ fulfills the associated dominance condition $K_M^{-1}X_1 > I$. The same is not true for sub-case 2.2 because $X_1 \rightarrow 0$ does not satisfy this dominance condition. Computationally determining the validity of each sub-case involves the solution of a linear program, as described by Fasani and Savageau (2010); the only difference being the incorporation of additional inequalities to account for the violation of the knife-edge conditions and associated extreme values for the chemical pools. Since the linear program is formulated in logarithmic coordinates, a value of $\log X_1 = 12$ or $\log X_1 = -12$ is used when $X_1 \rightarrow \infty$ or $X_1 \rightarrow 0$, respectively. The case numbers for the valid sub-cases have the parent case number with a period followed by a number associated with the sub-case. The total number of sub-cases corresponds to $2^{n_{knife}}$, where n_{knife} refers to the number of knife-edge conditions in the system.

	Sub-case 2.1	Sub-case 2.2
Violation of knife-edge	$v_0 > V_M$	$v_0 < V_M$
Implied extreme value	$X_1 \rightarrow \infty$	$X_1 \rightarrow 0$
Dominance condition	$\infty \cdot K_M^{-1} > I$	$0 \cdot K_M^{-1} > I$
Validity	Valid	Not valid

Table S4. *Blow-Up Cases for Two Knife-Edge Conditions.* Related to Figure 1. Since $n_{knife} = 2$, there is a total of four sub-cases to analyze. Only sub-cases 2 and 3 are valid. Sub-cases 1 and 2 exhibit an inconsistent violation of their knife-edge conditions. Sub-case 1.1 for instance, dictates $\alpha_1 > \alpha_2$ and $\alpha_2 > X_1$, which implies $X_1 \rightarrow \infty$ and $X_2 \rightarrow \infty$. By replacing extreme values into these inequalities, one obtains $\alpha_2 > \infty$, which cannot be fulfilled, rendering sub-case 1.1 invalid. Dominance conditions do not exist for this example because the Full System being analyzed is already an S-System.

	Sub-case 1.1	Sub-case 1.2	Sub-case 1.3	Sub-case 1.4
Violation of knife-edge	$\alpha_1 > \alpha_2$ $\alpha_2 > X_1$	$\alpha_1 < \alpha_2$ $\alpha_2 > X_1$	$\alpha_1 > \alpha_2$ $\alpha_2 < X_1$	$\alpha_1 < \alpha_2$ $\alpha_2 < X_1$
Implied extreme values	$X_1 \rightarrow \infty$ $X_2 \rightarrow \infty$	$X_1 \rightarrow 0$ $X_2 \rightarrow \infty$	$X_1 \rightarrow \infty$ $X_2 \rightarrow 0$	$X_1 \rightarrow 0$ $X_2 \rightarrow 0$
Dominance conditions	None	None	None	None
Validity	Not valid	Valid	Valid	Not valid

Table S5. List of Symbols Used in the Mathematical Formulation of the Protocatechuate Degradation Pathway. Related to Figure 2. The biological identity and context of symbols used to represent pools modeled in Eqs. S34-S44 is provided.

Module	Symbol	Name or Function	Description	
Signaling	U₁	Repressor form of the transcription factor PcaU	PcaU belongs to the IclR family of transcriptional regulators (PobR subfamily) and carries a helix-turn-helix motif at the N terminus. It acts as an activator in the presence of the inducer protocatechuate and as a repressor in its absence (Trautwein and Gerischer, 2001)	
	U₂	Activator form of the transcription factor PcaU		
Gene Circuitry	M₃	Polycistronic mRNA molecule	It encodes <i>pca</i> genes for the degradation of protocatechuate (<i>pcaBKCHG</i>)	
	T₄	4-Hydroxybenzoate transporter	This transporter protein, encoded by the gene <i>pcaK</i> , is also able to transport protocatechuate into the cell.	
	GH₅	Protocatechuate 3,4-dioxygenase	Dimer protein. The α sub-unit is encoded by <i>pcaG</i> , while the β sub-unit is encoded by <i>pcaH</i> . It catalyzes the reversible conversion of protocatechuate to β -carboxy-cis,cis-muconate.	
	B₆	β -carboxy-cis,cis-muconate cycloisomerase	This protein is encoded by the gene <i>pcaB</i> . It catalyzes the reversible conversion of β -carboxy-cis,cis-muconate to γ -carboxymuconolactone.	
	C₇	4-carboxymuconolactone decarboxylase	This protein is encoded by the gene <i>pcaC</i> . It catalyzes the conversion of γ -carboxymuconolactone to β -ketoadipate enol-lactone	
	Metabolism	P₀	Environmentally supplied protocatechuate	Protocatechuate is a (toxic) chemical compound derived from phenolic species including p-cresol, 4-hydroxybenzoate and numerous lignin monomers and is converted to β -ketoadipate by the pathway under study. Intracellular protocatechuate induces the transcription of <i>pca</i> genes via PcaU.
		P₈	Intracellular protocatechuate	
CM₉		β -carboxy-cis,cis-muconate	Intermediate metabolites of the protocatechuate degradation pathway	
CL₁₀		γ -carboxymuconolactone		

Table S6. List of Parameters Involved in Modeling the Pathway for Protocatechuate Degradation. Related to Figure 2. The biological meaning of each one of the 30 parameters used in Eqs. S34-S45 is provided.

Module	Parameter	Biological Meaning
Signaling	α_1	Bimolecular rate constant for formation of U_1 from U_2
	β_1	Monomolecular rate constant for decay of U_1 to U_2
	U_T	Total concentration of the transcription factor PcaU ($U_1 + U_2$)
Gene Circuitry	α_{3basal}	Rate constant for the basal transcription rate of the mRNA molecule M_3
	α_{3min}	Rate constant for the minimum transcription rate of the mRNA molecule M_3 (under repression by U_1)
	α_{3max}	Rate constant for the maximum transcription rate of the mRNA molecule M_3 (under activation by U_2)
	K_1	Binding constant of U_1 for the promoter of the mRNA molecule M_3
	K_2	Binding constant of U_2 for the promoter of the mRNA molecule M_3
	β_3	Rate constant for the degradation of the mRNA molecule M_3
	α_4	Rate constant for the translation of the protein T_4 from the mRNA molecule M_3
	β_4	Rate constant for the degradation of the protein T_4
	α_5	Rate constant for the translation of the protein GH_5 from the mRNA molecule M_3
	β_5	Rate constant for the degradation of the protein GH_5
	α_6	Rate constant for the translation of the protein B_6 from the mRNA molecule M_3
	β_6	Rate constant for the degradation of the protein B_6
	α_7	Rate constant for the translation of the protein C_7 from the mRNA molecule M_3
	β_7	Rate constant for the degradation of the protein C_7
Metabolism	P_0	Concentration of the environmentally supplied protocatechuate
	k_{cat4}	Turnover number of the enzyme T_4
	K_{M0}	Michaelis Menten constant of enzyme T_4
	k_{cat5}	Turnover number of the enzyme GH_5
	K_{M5f}	Michaelis Menten constant of enzyme GH_5 for the forward reaction
	K_{eq5}	Equilibrium constant for the reaction catalyzed by GH_5
	K_{M5r}	Michaelis Menten constant of enzyme GH_5 for the reverse reaction
	k_{cat6}	Turnover number of the enzyme B_6
	K_{M6f}	Michaelis Menten constant of enzyme B_6 for the forward reaction
	K_{eq6}	Equilibrium constant for the reaction catalyzed by B_6

K_{M6r}	Michaelis Menten constant of enzyme B ₆ for the reverse reaction
k_{cat7}	Turnover number of the enzyme C ₇
K_{M7}	Michaelis Menten constant of enzyme C ₇

Table S7. Numerical Solvers Available in DST3. Related to Figures 2, 3 and 4. The Full System tab of the DST3 user interface (Fig. S11) offers access to three different numerical solvers: a solver of ordinary differential equations (ODE) and two solvers of differential algebraic equations (DAE). When the Full System contains a conservation relationship in the form of an algebraic constraint, one of the two DAE solvers is used to numerically integrate the Full System. In this work, all numerical integrations reported were done using the IDA solver of the [Sundials solver suite](#). The step size was 10 steps per time unit for all analyses. Default values for other relevant parameters are listed below.

	Numerical Solver		
	IDA (Sundials)	RADAU5	ODEINT
Problem Type	DAE	DAE	ODE
Available in Python Package	Assimulo (Andersson et al., 2015)	Assimulo (Andersson et al., 2015)	Scipy
Integration method	Variable-order, variable-coefficient backward differentiation formulas in fixed-leading-coefficient form. The method order varies between 1 and 5	Runge-Kutta method of order 5	Adams methods for non-stiff problems (orders 1-12). Backward differentiation formulas (orders 1-5 with modified Newton iteration) for stiff problems
Absolute Tolerance	1.00E-06	1.00E-06	1.49E-08
Relative Tolerance	1.00E-06	1.00E-06	1.49E-08
Maximal number of steps for each integration point in time	1000	1000	Solver-determined

1. Transparent Methods

The theoretical foundation of the Design Space formalism was laid back in the 70's (Savageau, 1969; Savageau, 1971a; Savageau, 1971b; Savageau, 1979). This early work introduced the concept of S-systems and their mathematical characterization regarding dynamic stability of steady states, logarithmic gains for signal amplification, and parameter sensitivities for local robustness. Recently, these concepts were integrated to describe a generic approach to the construction of the *Design Space*, a structured parameter space in which qualitatively distinct biochemical phenotypes can be identified, counted and located (Savageau et al., 2009). Linking regions of the parameter space with biochemical phenotypes has allowed the elucidation of design principles and the introduction of a radically new phenotype-centric modeling strategy (Lomnitz and Savageau, 2016; Valderrama-Gómez and Savageau, 2018). Over the last decade, two computational implementations of the Design Space formalism have been developed. Rick Fasani first introduced the Design Space Toolbox for MATLAB (DST1), a formal software implementation automating key steps of this methodology. Fasani's contribution included an elegant mathematical description of the Design Space and a detailed explanation of its construction (Fasani and Savageau, 2010). Later, Jason Lomnitz introduced the Design Space Toolbox V2 (DST2) (Lomnitz and Savageau, 2016). DST2 consisted of a collection of tools comprised of a stand-alone library, written in the C language, that implements its own symbolic algebra engine and leverages open-source compiled libraries for linear algebra and linear optimization (via the GLPK library). By using multi-threaded concurrent algorithms to speed up calculations, DST2 took advantage of the parallelizable nature of the Design Space approach by analyzing each biochemical phenotype of the system independently.

Here, we show how various computational tools are integrated to create DST3. We go on to explain how Docker images and containers can be used to access DST3 on virtually any operating system. Then, we briefly describe two components of DST3 that can be used to access the computational capabilities of DST3: its user interface and its Python module. Refer to lpython-notebooks contained in the Docker image under **/Tutorials/Tutorial_DST3** for a detailed description of the user interface.

1.1 Design Space Toolbox v.3.0

Innovations contained in DST3 aim at improving three key aspects of the software: utility, usability and portability. By further developing the C library of DST3 to allow for the automatic identification and mathematical characterization of various types of singularities, we increased the scope of systems that can be analyzed by DST3, thus improving its *utility*. By enhancing stability and functionality of the IPython-based user interface of DST3, we increased software *usability* for users with limited programming knowledge. For advanced users, we generated a python module that runs on python 3.7.3 and can be integrated into customized programs. Due to its various external software dependencies, DST2 suffered from a limited portability. We address this issue by distributing DST3 via Docker. This effectively renders Docker the only software dependency necessary to run DST3 and guarantees *portability* across major operating systems.

1.1.1 Technology overview

Three main components make up DST3: A C library, a Python package and a user interface (Fig. S1). All three components are interconnected, with the C library being the computational engine that performs most of the numerical analyses. The Python package was designed to provide high-level access to the C library, making further software development simpler and faster. The user interface was built using IPython widgets and accesses the C library through the Python module.

Four steps are required to install and access all components of DST3:

1. Install Docker on your operating system. Refer to <https://savageaulab.wordpress.com/installing-docker/> for instructions.
2. Download the latest DST3 image by typing the following command in a Terminal or Prompt Window:

```
docker pull savageau/dst3
```

This will download the latest stable version of DST3 running on Python 2.7.3, for which a user interface is available. For advanced users, a Docker image of DST3 running on Python 3.7.3 (only the Python module is available) can be downloaded instead by typing:

```
docker pull savageau/dst3:python3
```

3. Start a Docker container to access DST3 by typing the following command in a Terminal or Prompt window:

```
docker run -d -p 8888:8888 savageau/dst3
```

That command will create a container without read and write privileges in the host computer. Files created within the container will be lost after the container is stopped. In order to grant read and write privileges, the following command should be used instead:

```
docker run -d -p 8888:8888 --mount  
type=bind,source=/Users,target=/Documents/host savageau/dst3
```

Windows users should use:

```
docker run -d -p 8888:8888 --mount  
type=bind,source=//c/Users,target=/Documents/host savageau/dst3
```

4. Access DST3 by opening the following address on any internet browser:

<http://localhost:8888/>

Windows users can refer to further the instructions contained in the webpage:

<https://savageaulab.wordpress.com/docker-image-for-the-design-space-toolbox-v3/>

1.1.2 DST3 User Interface

DST3 comes with an updated and more stable Ipython-based user interface. Fig. S2 presents a hierarchical overview of the different menus available. The gray box represents the initial menu, from which the functionality of DST3 can be accessed. The **About** menu (Fig. S6A) provides general information about the software, including its version, developers and an option to report bugs. Syntax rules for equations are also contained in this menu. Analyses supported by DST3 can be accessed from the **Main Menu** (Fig. S6B). Results, in the form of figures and tables, are managed by the **Figures** (Fig. S8A) and **Tables** menus (Fig. S8B), respectively. The **System** tab (Fig. S7) contains general information about the specific set of differential equations subject to analysis. It includes the name of the system, the total number of potential cases and its system signature.

Action tabs contained in **Main Menu** include the following:

Phenotypic Repertoire: This window allows the user to list and filter biochemical phenotypes – cases – according to their validity, case signature, log-gain values, number of eigenvalues with

positive real part and volume (Fig. S6B). The resulting phenotypic repertoire can be exported to a **.xlsx** file or saved into the **Tables** menu if desired.

Analyze Case: This tab creates the full analysis of a case referenced by its case number or case signature (Fig. S9A). This analysis includes the S-system equations that mathematically define the case, conditions that need to be fulfilled for its validity, its steady state solution – if it exists – and boundary constraints. For cases with a steady state solution, logarithmic gains for dependent variables with respect to independent variables and parameters are calculated and reported. For valid cases, a bounding box for the corresponding high-dimensional polytope is also provided. Additionally, it is possible to estimate a set of parameter values located within this high-dimensional polytope through linear programming. Global tolerances can be calculated from that nominal parameter set or from any other parameter set within the polytope by calculating respective lower and upper bounds for each parameter. The **Analyze Case** window also provides eigenvalues for the respective S-system. Note that individual tables generated by the **Analyze Case** tab can be saved into the **Tables** menu.

Case Intersections: For a given set of cases, this tab indicates if there is a region in parameter space where the cases overlap (Fig. S9B). If this region exist, global tolerances are reported for a parameter set located within the intersecting polytope.

Co-localizations: For a given set of cases and so-called slice variables, this tab indicates if regions of validity for each case exist within the given slice (Fig. S9C). If the co-localization is valid, global tolerances are reported. Additionally, Design Space plots can be generated to visualize the co-localization of the cases.

Create Plot: This window allows the generation of multiple plots, which are useful in characterizing the Design Space and its predictions for stability, steady state concentrations, fluxes, etc. (Fig. S10). Environmentally determined independent variables and genetically determined parameters are typically plotted on the x- and y-axes, and the selected phenotypic characteristics are plotted as a heat map on the z-axis. Even though an explorative characterization of the Design Space is possible via the *interactive* Design Space plot, functionalities contained in the **Create Plot** window are more effective when used to characterize a specific region of interest in the parameter space. This region can be usually found by combining

functionalities of the **Phenotypic Repertoire** and **Analyze Case** windows. Plots generated by this window can be saved in the menu **Figures**.

Full System: This window allows the characterization of the temporal response of the original set of differential equations under study (Full System), as well as the analysis of its stability properties (Time Course, Titration and Trajectory Plots). This characterization is useful to assess the accuracy of predictions made using the S-system approximations (Fig. S11). Analysis of systems of differential algebraic equations (DAE) is performed through the python package Assimulo (Andersson et al., 2015), while systems of ordinary differential equations (ODE) are numerically solved by the ODEINT solver of the Python package Scipy. Refer to Table S7 for further details. Results generated by this window can be saved in the menu **Figures** or exported to **.xlsx** files for further analysis.

1.1.3 DST3 Python Module

The capability of the DST3 C library to identify special phenotypes can be enabled by changing default values of arguments passed to classes `Equations` and `DesignSpace`. Both classes are contained in the DST3 Python Package `dSPACE` and play a central role in generating Python objects involved in any computational Design Space analysis, as shown in Fig. S3. The input to the class `Equations` consists of a machine-readable string representation of the system of equations describing the dynamics of the network. A string representation that can be parsed by the computational engine of DST3 should be compliant with following syntax rules:

1. Each equation has to be explicitly stated as:
 - a. A differential equation, where the "." operator denotes the derivative with respect to time.
 - b. An algebraic constraint, where the left-hand side is either a variable or a mathematical expression. Auxiliary variables associated with the constraint must be explicitly defined (unless the left-hand side is the auxiliary variable). Algebraic constraints can be used to represent conservation relationships. In that case, they should be placed after regular algebraic constraints. Additionally, they should follow the form dictated by Eq. S21. When using the Python module, associated auxiliary variables should be explicitly declared as 'Xci', with $i = 1, \dots, n_{cr}$. This

definition is not necessary when using the DST3 user interface. In any case, variable names 'Xci', with $i=1,\dots,n_{cr}$ are reserved for the computational engine and should not be contained in the system of equations defined by the user.

2. Multiplication is represented by the "*" operator.
3. Powers are represented by the "^" operator.
4. Architectural constraints are defined as inequalities, where both sides of the inequality are products of power-laws.

In order to exemplify the generation of valid machine-readable string representations and the usage of the DST3 Python module, we calculate valid cases for three different synthetic biochemical systems, each one exhibiting a different type of singularity. For each system, five lines of Python code are presented and discussed. Computational steps involve in each case:

1. Importing the DST3 python module `dspace`,
2. Defining a string representation for the system,
3. Generating an `Equations` object,
4. Generating a `DesignSpace` object,
5. Generating a list of valid cases.

Cycles

The synthetic network under analysis is described by Eqs. S1-S3. Generating a string representation of this system is straightforward and results in a list of three strings, one for each differential equation, as shown in [line 2](#) of the snippet below.

```
1 import dspace
2 eqs_str = ['X1. = a11 + 2*b31*X3 - b11*X1 - 2*b12*(X1^2)',
            'X2. = b12*(X1^2) - b23*X2 - b22*X2',
            'X3. = a31 + b23*X2 - b31*X3 - b33*X3']
3 equations = dspace.Equations(eqs_str)
4 ds = dspace.DesignSpace(equations, resolve_cycles=True)
5 ds.valid_cases()
```

```
[out] ['1', '2', '3', '4', '5', '6', '7', '8', '10', '12', '13', '14', '15', '16', '17', '18', '21', '22', '26', '27_1', '27_2', '27_3', '27_4', '27_5', '27_6', '29', '30']
```

Since the system does not contain any auxiliary variable, the object `eq_string` is passed as the sole positional argument to the class `dspace.Equations()` to generate the `Equations` object, which is stored in the variable `equations`. In order to identify and resolve the cycle encoded within this system, the key argument `resolve_cycles` is set to `True` and passed along with the `equations` object to the class `dspace.DesignSpace()` to generate a `DesignSpace` object which is stored in the variable `ds`. A list of valid cases is generated through the method `valid_cases()` of the `ds` object. Note that cases `27_1`, `27_2`, ..., `27_6` result from resolving the cyclical case 27 (refer to Table S1).

Conservations

As discussed before, the system described by Eqs. S22-S25. contains a conservation constraint among its constituent pools, as defined by Eq. S25. According to the DST3 syntax rules, this conservation relationship can be explicitly defined as an algebraic constraint and should be placed in the last position of the string representation of the system (see [line 2](#) of the snippet below). Since the Python module is being used to analyze this system, an associated auxiliary variable needs to be explicitly defined, as shown in [line 3](#).

```
1 import dspace
2 eqs_str = ['X1. = a11*X3 - b11*X1',
            'X2. = b11*X1 - b21*X2',
            'X3. = b21*X2 - a11*X3',
            '0 = CR1 - X1 - X2 - X3 ']
3 equations = dspace.Equations(eqs_str,
                               auxiliary_variables=['Xc1'])
4 ds = dspace.DesignSpace(eq, resolve_conservations=True,
                          number_conservations=1)
5 ds.valid_cases()
[out] ['1', '2', '3']
```

The computational engine of DST3 is informed about the conservation constraint by using two key arguments: `number_conservations=1` and `resolve_conservations=True`. Valid cases can be printed by means of the method `valid_cases()` of the `ds` object.

Metabolic Imbalances

The last example consists of a system containing one differential equation and one algebraic constraint, as defined by Eqs. S26-S27. For certain parameter values, this system has the potential to exhibit a blow-up for variable X_1 . The class `dspace.Equations` is informed about the algebraic constraint contained in the string representation by using the key argument `auxiliary_variables=['D']`. Similarly, the DST3 computational engine is configured to check for blow-ups by using the key argument `resolve_instability=True`, as shown in the snippet below:

```
1 import dspace
2 eqs_str = ['X1. = Vo - Vm*(D^-1)*K^(-1)*X1',
            '0 = 1 + (K^-1)*X1 - D ']
3 equations = dspace.Equations(eqs_str,
                               auxiliary_variables=['D'])
4 ds = dspace.DesignSpace(equations, resolve_instability=True)
5 ds.valid_cases()
[out] ['1', '2.1']
```

Valid cases can be customarily printed using the method `valid_cases()` of the `ds` object. Note that the case identifier 2.1 refers to the sub-case 2.1 of Table S3.

1.2 Data and Code Availability

The Docker images used by DST3 are freely available at <https://hub.docker.com/r/savageau/dst3>

1.3 Additional Resources

A tutorial is available as various IPython notebooks within the DST3 docker image under `/Tutorials/Tutorial_DST3`

2. Strategies for Treating Three Types of Singularities

2.1 Cycles are resolved by considering global dominance equations.

We start by setting up Eqs. S1 to S3 to describe the change in time for the concentration of each chemical species shown in Fig. 1A in the main text. Mass action kinetics are used to generate rate laws describing the flux through each reaction of the network. The resulting expressions are then combined by means of Kirchhoff's node law to generate balance equations for each metabolite in the network.

$$\frac{dX_1}{dt} = \alpha_{11} + 2\beta_{31}X_3 - \beta_{11}X_1 - 2\beta_{12}X_1^2 \quad (\text{S1})$$

$$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2 - \beta_{22}X_2 \quad (\text{S2})$$

$$\frac{dX_3}{dt} = \alpha_{31} + \beta_{23}X_2 - \beta_{31}X_3 - \beta_{33}X_3. \quad (\text{S3})$$

The Design Space formalism can be applied to decompose this set of equations into different cases, each having a unique set of dominant terms and being valid within a specific region in parameter space. Eqs. S4 – S6 represent one of those cases, which is defined by the case signature [22 11 21] and the case number 27.

$$\frac{dX_1}{dt} = 2\beta_{31}X_3 - 2\beta_{12}X_1^2 \quad (\text{S4})$$

$$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2 \quad (\text{S5})$$

$$\frac{dX_3}{dt} = \beta_{23}X_2 - \beta_{31}X_3. \quad (\text{S6})$$

Necessary conditions for these terms to be dominant are described by Eqs. S7 – S11:

$$2\alpha_{11}^{-1}\beta_{31}X_3 > 1 \quad (\text{S7})$$

$$2\beta_{11}^{-1}\beta_{12}X_1 > 1 \quad (\text{S8})$$

$$\beta_{23}\beta_{22}^{-1} > 1 \quad (\text{S9})$$

$$\beta_{23}\alpha_{31}^{-1}X_2 > 1 \quad (\text{S10})$$

$$\beta_{31}\beta_{33}^{-1} > 1 . \quad (\text{S11})$$

It should be noted that the notation used in the computational algorithms differs from the conventional notation in the main text, where parameters (b -vector) and independent variables (y_I vector) are separate, since both are treated here as independent variables (y_I vector). The b -vector, which conventionally represented ratios of parameters, is now used to represent only stoichiometry values. This facilitates the computational matrix operations.

By setting the left-hand side of Eqs. S4-S6 to zero, taking logarithms and rearranging, one obtains Eq. S12, which exhibits the form of Eq. 8 in the main text:

$$\begin{pmatrix} -2 & 0 & 1 \\ 2 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} y_D = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix} y_I , \quad (\text{S12})$$

with $y_D^T = \left(\log X_1 \quad \log X_2 \quad \log X_3 \right)$ and $y_I^T = \left(\log \beta_{12} \quad \log \beta_{31} \quad \log \beta_{23} \right)$.

Through this Supplemental Material, vectors A_D^T and y_I are defined as in the first version of the Design Space Toolbox (DST1, Fasani and Savageau, 2010). Visual inspection of the matrix A_D reveals the presence of a linear dependency among its rows: $\text{row}_1(A_D) = -\text{row}_2(A_D) - \text{row}_3(A_D)$. This causes matrix A_D to be rank deficient [$\text{rank}(A_D) = 2$] and prevents the computation of a unique solution for y_D . In this case, $\text{rank}(A_D | B - A_I y_I) < \text{rows}(A_D)$, where $\text{rank}(A_D | B - A_I y_I)$ refers to the rank of the augmented matrix that results from concatenating the matrix A_D and the vector $B - A_I y_I$ (Strang, 2005), and the system of algebraic equations has the potential to contain a single or multiple cycles (Fig. S4). Its solution(s) can be found by applying the Design Space

formalism to an extended set of equations. The modification of the system consists in adding a so-called *global dominance* equation, which describes a mass balance around each cycle present in the system. The following steps are involved in generating the extended set of equations:

1. Identify set(s) of *cyclical variables* C by computing the null space of matrix A_D . Cyclical variables are characterized by non-zero row entries in the null space matrix. Note that the number of cycles contained in A_D is defined by the number of columns of the null space: $col(Null(A_D))$. For the specific case being analyzed one obtains: $[Null(A_D)]^T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$, meaning that the system contains one single cycle, with the set of cyclical variables $C = \{ X_1, X_2, X_3 \}$.
2. Set up a *global dominance* equation for each cycle. This equation is a mathematical representation of a mass balance around a given cycle in steady state. In the case of the network considered in Fig. 1A, the global dominance equation will be a function of fluxes entering and leaving the control volume delimited by the blue rectangle in Fig. 1B, i.e, fluxes governed by rate constants $\alpha_{11}, \alpha_{31}, \beta_{11}, \beta_{22}$ and α_{33} . Global dominance equations are constructed by weighting those fluxes using coefficients obtained from the null space of matrix A_D^T . In this specific case one obtains: $[Null(A_D^T)]^T = [1 \ 2 \ 2]$, which yields the following global dominance equation: $0 = \alpha_{11} - \beta_{11}X_1 - 2X_2\beta_{22} + 2\alpha_{31} - 2X_3\beta_{33}$.
3. Generate extended sub-system by introducing the global dominance equation(s). Eqs. S13-S16 along with the conditions defined by Eqs. S7-S11 define the extended sub-system that should be used to resolve the cyclical case generated by Eqs. S4-S6.

$$\frac{dX_1}{dt} = 2\beta_{31}X_3 - 2\beta_{12}X_1^2 \quad (\text{S13})$$

$$\frac{dX_2}{dt} = \beta_{12}X_1^2 - \beta_{23}X_2 \quad (\text{S14})$$

$$\frac{dX_3}{dt} = \beta_{23}X_2 - \beta_{31}X_3 \quad (\text{S15})$$

$$0 = \alpha_{11} + 2\alpha_{31} - \beta_{11}X_1 - 2\beta_{22}X_2 - 2\beta_{33}X_3. \quad (\text{S16})$$

DST3 implements steps 1 to 3 in a recursive fashion to resolve multiple and nested cycles. Once the extended sub-system has been set up, the Design Space formalism can be applied to identify valid sub-cases resolving the cyclical case. Table S1 shows S-system equations for each one of the six valid sub-cases generated from the extended system. Note the special form of these equations. S-systems originating from a dominance analysis on the global dominance equation (Eq. S16) are used to replace the differential equation for the pool with the dominant efflux. For instance, the expression $\alpha_{11} - \beta_{11}X_1$, which is obtained when the first positive and first negative term in Eq. S16 are dominant, is used to replace the differential equation for X_1 (refer to sub-case 27_1 in Table S1). Additionally, this expression is scaled to match the stoichiometric coefficient of the negative term in the Full System (i.e., the original set of equations). Consider for instance the expression $\alpha_{11} - 2\beta_{33}X_3$, which is obtained when the first positive and third negative term of Eq. S16 are dominant. Since the original stoichiometric coefficient of the negative term is 1 (see Eq. S3), the scaled expression $\frac{1}{2}\alpha_{11} - \beta_{33}X_3$ is used to construct sub-case 27_3 of Table S1.

2.2 Conserved moieties are handled by considering the total size of conserved pools.

Consider the simple system in Fig. 1C in the main text with three components linked by a conservation relationship. Mass balance equations can be set up for each metabolite using the rate laws shown in Fig. 1C:

$$\frac{dX_1}{dt} = r_1 - r_2 \quad (\text{S17})$$

$$\frac{dX_2}{dt} = r_2 - r_3 \quad (\text{S18})$$

$$\frac{dX_3}{dt} = r_3 - r_1. \quad (\text{S19})$$

Balance Eqs. S17-S19 can be compactly expressed in matrix form to yield Eq. S20:

$$\frac{dX}{dt} = Sr, \quad (\text{S20})$$

with S being the stoichiometric matrix and r a vector of rate laws describing the flux through each reaction. The number of conservations n_{cr} within the system is given by:

$$n_{cr} = \text{col}(\text{Null}(S^T)), \text{ which equals 1 for the system under consideration.}$$

Conservation relationships can be mathematically described as linear dependencies among metabolite pools. They can be expressed in matrix form as:

$$0 = CR - \text{Null}(S^T)^T \times X. \quad (\text{S21})$$

CR is a vector of independent variables and CR_i expresses the total pool size of each conservation. X represents a vector of concentration pools. Applying Eq. S21 to the system defined by Eqs. S17-S19 yields: $0 = CR_1 - X_1 - X_2 - X_3$. DST3 is able to handle biochemical systems with single or multiple conservation relationships. For the system depicted in Fig. 1C, Eqs. S22-S25 represents an appropriate set of equations that can be analyzed by DST3 given the conservation relationship provided:

$$\frac{dX_1}{dt} = \alpha_{11}X_3 - \beta_{11}X_1 \quad (\text{S22})$$

$$\frac{dX_2}{dt} = \beta_{11}X_1 - \beta_{21}X_2 \quad (\text{S23})$$

$$\frac{dX_3}{dt} = \beta_{21}X_2 - \alpha_{11}X_3 \quad (\text{S24})$$

$$0 = CR_1 - X_1 - X_2 - X_3 \quad (\text{S25})$$

The analysis of this system using the Design Space formalism involves the usual generation of cases by picking dominant terms for each of the equations. To handle the singularity generated by the conservation encoded in the system, DST3 eliminates the differential equation(s) corresponding to the dominant negative term of each conservation relationship. This generates cases with S-systems that are deficient in n_{cr} differential equations. Table S2 contains three cases that result from applying the Design Space formalism to the system defined by Eqs S22-S25. Note that each case is defined by only two differential equations and one algebraic constraint. In order to capture this special way of constructing the S-system equations, the indices of the differential equation being deleted are set to zero in the case signature. Case 3 for instance,

in which the differential equation for pool X_3 is missing, has a case signature of [11 11 00 13] to reflect this fact.

Note the similarity between the system topology of the cyclical case 27 (Fig. 1B) and the conserved system of this section (Fig. 1C). In both instances, metabolites X_1 , X_2 and X_3 interact in a cyclical fashion to introduce a linear dependency among their pools that renders their A_D matrix singular. For cyclical cases (e.g. case 27), this dependency is eliminated by means of a global dominance equation, which represents a mass balance around the cycle. This strategy cannot be applied for conserved moieties, because they are not synthesized, degraded or exchanged with the environment (Haraldsdóttir and Fleming, 2016). From a mass balance perspective, this implies that fluxes entering or leaving the conservation do not exist, as exemplified in Fig. 1D. The singularity is thus eliminated by replacing the differential equation for one of the metabolites involved in the conservation by an algebraic constraint that contains an additional independent parameter (CR_i).

2.3 Metabolic Imbalances are treated by considering knife-edge conditions

Given the system shown in Fig. 1E and described by Eqs. 11-12 in the main text, consider the equations for case 2 with signature [11 21]:

$$\frac{dX_1}{dt} = v_0 - V_M D^{-1} K_M^{-1} X_1 \quad (\text{S26})$$

$$0 = K_M^{-1} X_1 - D, \quad (\text{S27})$$

and its associated dominance condition:

$$K_M^{-1} X_1 > 1. \quad (\text{S28})$$

Substituting the algebraic constraint (Eq. S27) into the differential equation for X_1 (Eq. S26) yields the following dynamical system:

$$\frac{dX_1}{dt} = v_0 - V_M. \quad (\text{S29})$$

Setting the left-hand side of this equation to zero, taking logarithms of both sides and rearranging in matrix notation analogously to Eq. 8 in the main text results in:

$$[0]y_D = [0] - \begin{bmatrix} 1 & -1 \end{bmatrix} y_I, \quad (\text{S30})$$

with $y_D = [\log X_1]$ and $y_I^T = \begin{bmatrix} \log v_0 & \log V_M \end{bmatrix}$. Since $\text{rank}(A_D | B - A_I y_I) = \text{rows}(A_D)$, where $\text{rank}(A_D | B - A_I y_I)$ refers to the rank of the augmented matrix that results from concatenating the matrix A_D and the vector $B - A_I y_I$ (Strang, 2005), the system does not have a steady state solution but has rather the potential to contain a metabolic imbalance (Fig. S4). Indeed, Eq. S29 only provides a consistency condition for the concentration X_I to remain unchanged over time: $0 = v_0 - V_M$. We will refer to this kind of constraint as a *knife-edge condition*. In general, we are interested in the behavior of the system when knife-edge conditions are not satisfied, i.e., $v_0 \neq V_M$. Violating the knife-edge condition in a specific direction implies an extreme value for X_I : $X_I \rightarrow \infty$ or $X_I \rightarrow 0$. The validity of either situation is assessed by checking the validity of the associated dominance conditions, as shown in Table S3. Taken together, these results indicate that for the system shown in Fig. 1E, the concentration of the pool X_I will steadily increase over time, i.e., it will *blow up* if variables fulfill the conditions $K_M^{-1} X_1 > 1$ and $v_0 V_M^{-1} > 1$.

To introduce further concepts required to establish a general framework for the treatment of cases for which $\text{rank}(A_D | B - A_I y_I) = \text{rows}(A_D)$, consider the system described by Eqs. S31-S32:

$$\frac{dX_1}{dt} = \alpha_1 - \alpha_2 \quad (\text{S31})$$

$$\frac{dX_2}{dt} = \alpha_2 - X_1, \quad (\text{S32})$$

which when rearranged and expressed in matrix notation to resemble the form of Eq. 8 in the main text yields:

$$\begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} y_D = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} y_I, \quad (\text{S33})$$

with $y_D^T = \begin{bmatrix} \log X_1 & \log X_2 \end{bmatrix}$ and $y_I^T = \begin{bmatrix} \log \alpha_1 & \log \alpha_2 \end{bmatrix}$. Since the matrix A_D is singular and $\text{rank}(A_D | B - A_I y_I) = \text{rows}(A_D) = 2$, the system does not have a valid steady state solution. It rather has the potential to exhibit a metabolic imbalance. The structure of the matrix A_D indicates the existence of two knife-edge conditions, which give rise to four sub-cases, as shown in Table S4. In general, the number of subcases to be tested equals $2^{n_{\text{knife}}}$, with n_{knife} the number of knife edges present in the system. Out of four possible sub-cases, only sub-case 2 and sub-case 3 are valid because of the way in which they violate both knife-edge conditions is consistent. The fact that the matrix A_D has one degree of freedom, $\text{row}(A_D) - \text{rank}(A_D) = 2 - 1 = 1$, implies a relationship between the two knife edge conditions. Indeed, violating the knife-edge condition $\alpha_1 = \alpha_2$ in either direction sets an extreme value for X_1 . This in turn dictates the way in which the second knife edge $\alpha_2 = X_1$ will be violated, thus setting an extreme value for X_2 . Since the structure of the matrix A_D prevents X_2 from being calculated via matrix operations, after an extreme value for X_1 has been set we opt for testing the validity of each possible sub-system via linear programming.

More generally, three steps are involved in resolving cases for which

$$\text{rank}(A_D) < \text{rank}(A_D | B - A_I y_I):$$

1. Merge auxiliary variables into differential equations to obtain dynamical systems without algebraic constraints.
2. Identify the number and identity of knife-edge conditions as a function of the degrees of freedom (n_{freedom}) of matrix A_D and the number of zeros in its diagonal (n_{zeros}):

- a. If $n_{freedom}$ is equal to n_{zeros} , then $n_{knife} = n_{freedom}$ and knife edge conditions correspond to balance equations for pools with a zero entry in the diagonal of A_D
 - b. If $n_{freedom} < n_{zeros}$, then $n_{knife} = n_{zeros}$ and knife edge conditions correspond to balance equations for pools with a zero entry in the diagonal A_D
 - c. If $n_{freedom} > n_{zeros}$, then $n_{knife} = n_{freedom}$ and knife edge conditions correspond to balance equations for n_{knife} randomly selected pools.
3. Construct and test the validity of $2^{n_{knife}}$ different linear programs, each violating knife edge conditions in a unique way. Note that the test for validity should include dominance conditions associated with the case under analysis.

3. Analysis of a Biochemical System Exhibiting Multiple Singularities

3.1 Differential Equations for the Biochemical System

Transcription factor:

$$\frac{dU_1}{dt} = \alpha_1 U_2 - \beta_1 P_8 U_1 \quad (\text{S34})$$

$$\frac{dU_2}{dt} = \beta_1 P_8 U_1 - \alpha_1 U_2 \quad (\text{S35})$$

$$0 = U_T - U_1 - U_2 \quad (\text{S36})$$

Transcriptional unit:

$$\frac{dM_3}{dt} = \frac{\alpha_{3basal} + \alpha_{3min} \left(\frac{U_1}{K_1} \right)^n + \alpha_{3max} \left(\frac{U_2}{K_2} \right)^p}{1 + \left(\frac{U_1}{K_1} \right)^n + \left(\frac{U_2}{K_2} \right)^p} - \beta_3 M_3 \quad (\text{S37})$$

Protein synthesis:

$$\frac{dT_4}{dt} = \alpha_4 M_3 - \beta_4 T_4 \quad (\text{S38})$$

$$\frac{dGH_5}{dt} = \alpha_5 M_3 - \beta_5 GH_5 \quad (\text{S39})$$

$$\frac{dB_6}{dt} = \alpha_6 M_3 - \beta_6 B_6 \quad (\text{S40})$$

$$\frac{dC_7}{dt} = \alpha_7 M_3 - \beta_7 C_7 \quad (\text{S41})$$

Enzymatic reactions:

$$\frac{dP_8}{dt} = \frac{T_4 k_{cat4} \left(\frac{P_0}{K_{M0}} \right) + \alpha_1 U_2 - \frac{GH_5 k_{cat5} \left(\frac{P_8}{K_{M5f}} \right) \left[1 - \frac{1}{K_{eq5}} \left(\frac{CM_9}{P_8} \right) \right]}{1 + \left(\frac{P_0}{K_{M0}} \right) + \left(\frac{P_8}{K_{M5f}} \right) + \left(\frac{CM_9}{K_{M5r}} \right)} - \beta_8 P_8 - \beta_1 P_8 U_1 \quad (\text{S42})$$

$$\frac{dCM_9}{dt} = \frac{GH_5 k_{cat5} \left(\frac{P_8}{K_{M5f}} \right) \left[1 - \frac{1}{K_{eq5}} \left(\frac{CM_9}{P_8} \right) \right]}{1 + \left(\frac{P_8}{K_{M5f}} \right) + \left(\frac{CM_9}{K_{M5r}} \right)} - \frac{B_6 k_{cat6} \left(\frac{CM_9}{M_{M6f}} \right) \left[1 - \frac{1}{K_{eq6}} \left(\frac{CL_{10}}{CM_9} \right) \right]}{1 + \left(\frac{CM_9}{M_{M6f}} \right) + \left(\frac{CL_{10}}{K_{M6r}} \right)} \quad (\text{S43})$$

$$\frac{dCL_{10}}{dt} = \frac{B_6 k_{cat6} \left(\frac{CM_9}{M_{M6f}} \right) \left[1 - \frac{1}{K_{eq6}} \left(\frac{CL_{10}}{CM_9} \right) \right]}{1 + \left(\frac{CM_9}{M_{M6f}} \right) + \left(\frac{CL_{10}}{K_{M6r}} \right)} - \frac{C_7 k_{cat7} \left(\frac{CL_{10}}{K_{M7}} \right)}{1 + \left(\frac{CL_{10}}{K_{M7}} \right)} \quad (\text{S44})$$

Pathway flux:

$$F = \frac{C_7 k_{cat7} \left(\frac{CL_{10}}{K_{M7}} \right)}{1 + \left(\frac{CL_{10}}{K_{M7}} \right)} \quad (\text{S45})$$

3.2 GMA Equations for the Biochemical System

Equations S34 to S45 can be recast into their GMA form to yield 10 differential equations, 6 algebraic constraints and 1 conservation relationship as shown in Eqs. S46 to S62. Note that the

terms $+\alpha_1 U_2$ and $-\beta_1 P_8 U_1$ were omitted from the differential equation for the pool P_8 (Eq. S53). This is because in a biological context, these fluxes will never be dominant fluxes defining the dynamics of the pool P_8 .

Differential equations:

$$\frac{dU_1}{dt} = \alpha_1 U_2 - \beta_1 P_8 U_1 \quad (\text{S46})$$

$$\frac{dU_2}{dt} = \beta_1 P_8 U_1 - \alpha_1 U_2 \quad (\text{S47})$$

$$\frac{dM_3}{dt} = \alpha_{3\text{basal}} D_3^{-1} + \alpha_{3\text{min}} K_1^{-n} U_1^n D_3^{-1} + \alpha_{3\text{max}} K_2^{-p} U_2^p D_3^{-1} - \beta_3 M_3 \quad (\text{S48})$$

$$\frac{dT_4}{dt} = \alpha_4 M_3 - \beta_4 T_4 \quad (\text{S49})$$

$$\frac{dGH_5}{dt} = \alpha_5 M_3 - \beta_5 GH_5 \quad (\text{S50})$$

$$\frac{dB_6}{dt} = \alpha_6 M_3 - \beta_6 B_6 \quad (\text{S51})$$

$$\frac{dC_7}{dt} = \alpha_7 M_3 - \beta_7 C_7 \quad (\text{S52})$$

$$\begin{aligned} \frac{dP_8}{dt} = & T_4 k_{\text{cat}4} K_{M0}^{-1} P_0 D_4^{-1} + GH_5 k_{\text{cat}5} K_{M5f}^{-1} K_{\text{eq}5}^{-1} CM_9 D_5^{-1} \\ & - GH_5 k_{\text{cat}5} K_{M5f}^{-1} P_8 D_5^{-1} - \beta_8 P_8 \end{aligned} \quad (\text{S53})$$

$$\begin{aligned} \frac{dCM_9}{dt} = & GH_5 k_{\text{cat}5} K_{M5f}^{-1} P_8 D_5^{-1} + B_6 k_{\text{cat}6} K_{M6f}^{-1} K_{\text{eq}5}^{-1} CL_{10} D_6^{-1} \\ & - GH_5 k_{\text{cat}5} K_{M5f}^{-1} K_{\text{eq}5}^{-1} CM_9 D_5^{-1} - B_6 k_{\text{cat}6} K_{M6f}^{-1} CM_9 D_6^{-1} \end{aligned} \quad (\text{S54})$$

$$\begin{aligned} \frac{dCL_{10}}{dt} = & B_6 k_{\text{cat}6} K_{M6f}^{-1} CM_9 D_6^{-1} \\ & - B_6 k_{\text{cat}6} K_{M6f}^{-1} K_{\text{eq}5}^{-1} CL_{10} D_6^{-1} - C_7 k_{\text{cat}7} K_{M7}^{-1} CL_{10} D_7^{-1} \end{aligned} \quad (\text{S55})$$

Algebraic constraints:

$$0 = C_7 k_{\text{cat}7} K_{M7}^{-1} CL_{10} D_7^{-1} - F \quad (\text{S56})$$

$$0 = 1 + K_1^{-n} U_1^n + K_2^{-p} U_2^p - D_3 \quad (\text{S57})$$

$$0 = 1 + K_{M0}^{-1} P_0 - D_4 \quad (\text{S58})$$

$$0 = 1 + K_{M5f}^{-1} P_8 + K_{M5r}^{-1} CM_9 - D_5 \quad (\text{S59})$$

$$0 = 1 + K_{M6f}^{-1} CM_9 + K_{M6r}^{-1} CL_{10} - D_6 \quad (\text{S60})$$

$$0 = 1 + K_{M7}^{-1} CL_{10} - D_7 \quad (\text{S61})$$

Conservation relationship

$$0 = U_T - U_1 - U_2. \quad (\text{S62})$$

The cooperativity (Hill number) for transcription factor binding is represented by the kinetic orders n and p . The combination of transcriptional repression and activation requires the constraints: $\alpha_{3\min} < \alpha_{3\text{basal}} < \alpha_{3\max}$. The α and β parameters are rate constants for synthesis and degradation processes, respectively. The k_{cat} and K_M parameters used in the metabolic pathway represent turnover numbers and Michaelis constants, respectively. Auxiliary variables D_3 to D_7 replace the denominators in the biochemical kinetic expressions during the process of recasting the original ODE system into the equivalent GMA system. Refer to Tables S5 and S6 for the biological context of symbols used in Eqs. S46 to S62. The 17 equations (Eqs. S46-S62) contain a total of 30 parameters (Table S6). As shown in the main text, no previous knowledge (other than the constraints noted above) for any of these parameter values is required by DST3 to characterize the dynamical behavior of the system and identify a robust operating point for the system.

Supplemental References

- Andersson, C., Führer, C., and Åkesson, J. (2015). Assimulo: A unified framework for ODE solvers. *Math. Comput. Simul.* *116*, 26–43.
- Fasani, R.A., and Savageau, M.A. (2010). Automated construction and analysis of the design space for biochemical systems. *Bioinformatics* *26*, 2601–2609.
- Haraldsdóttir, H.S., and Fleming, R.M.T. (2016). Identification of conserved moieties in metabolic networks by graph theoretical analysis of atom transition networks. *PLoS Comput. Biol.* *12*, e1004999.
- Lomnitz, J.G., and Savageau, M.A. (2016). Design Space Toolbox V2: Automated software enabling a novel phenotype-centric modeling strategy for natural and synthetic biological systems. *Front. Genet.* *7*, 118.
- Savageau, M.A. (1969). Biochemical systems analysis: II. The steady-state solutions for an n-pool system using a power-law approximation. *J. Theor. Biol.* *25*, 370–379.
- Savageau, M.A. (1971a). Concepts relating the behavior of biochemical systems to their underlying molecular properties. *Arch. Biochem. Biophys.* *145*, 612–621.
- Savageau, M.A. (1971b). Parameter sensitivity as a criterion for evaluating and comparing the performance of biochemical systems. *Nature* *229*, 542–544.
- Savageau, M.A. (1979) Growth of complex systems can be related to the properties of their underlying determinants. *Proc. Natl. Acad. Sci. USA.* *76*, 5413–5417.
- Savageau, M.A., Coelho, P.M.B.M., Fasani, R.A., Tolla, D.A., and Salvador, A. (2009). Phenotypes and tolerances in the design space of biochemical systems. *Proc. Natl. Acad. Sci. U.S.A.* *106*, 6435–6440.
- Strang, G. (2005). *Linear Algebra and Its Applications*, 4th edition, (Cengage Learning).
- Valderrama-Gómez, M.A., Parales, R.E., and Savageau, M.A. (2018). Phenotype-centric modeling for elucidation of biological design principles. *J. Theor. Biol.* *455*, 281–292.