Automated refinement and inference of analytical models for metabolic networks

2011 Phys. Biol. 8 055011

(http://iopscience.iop.org/1478-3975/8/5/055011)

Phys. Biol. **8** (2011) 055011 (20pp) doi:10.1088/1478-3975/8/5/055011

Automated refinement and inference of analytical models for metabolic networks

Michael D Schmidt[1], Ravishankar R Vallabhajosyula[2], Jerry W Jenkins[3], Jonathan E Hood[2], Abhishek S Soni[2], John P Wikswo[4] and Hod Lipson[5]

[1] Cornell Computational Systems Laboratory, Cornell University, Ithaca, NY, USA
[2] CFD Research Corporation, Huntsville, AL, USA
[3] HudsonAlpha Institute, Huntsville, AL, USA
[4] Departments of Biomedical Engineering, Molecular Physiology and Biophysics, and Physics and Astronomy, and the Vanderbilt Institute for Integrative Biosystems Research and Education, Vanderbilt University, Nashville, TN, USA
[5] School of Mechanical and Aerospace Engineering and the Department of Computing and Information Science, Cornell University, Ithaca, NY, USA

E-mail: john.wikswo@vanderbilt.edu and hod.lipson@cornell.edu

Received 7 March 2011
Accepted for publication 27 June 2011
Published 10 August 2011
Online at stacks.iop.org/PhysBio/8/055011

Abstract
The reverse engineering of metabolic networks from experimental data is traditionally a labor-intensive task requiring *a priori* systems knowledge. Using a proven model as a test system, we demonstrate an automated method to simplify this process by modifying an existing or related model—suggesting nonlinear terms and structural modifications—or even constructing a new model that agrees with the system's time series observations. In certain cases, this method can identify the full dynamical model from scratch without prior knowledge or structural assumptions. The algorithm selects between multiple candidate models by designing experiments to make their predictions disagree. We performed computational experiments to analyze a nonlinear seven-dimensional model of yeast glycolytic oscillations. This approach corrected mistakes reliably in both approximated and overspecified models. The method performed well to high levels of noise for most states, could identify the correct model *de novo*, and make better predictions than ordinary parametric regression and neural network models. We identified an invariant quantity in the model, which accurately derived kinetics and the numerical sensitivity coefficients of the system. Finally, we compared the system to dynamic flux estimation and discussed the scaling and application of this methodology to automated experiment design and control in biological systems in real time.

S Online supplementary data available from stacks.iop.org/PhysBio/8/055011/mmedia

Introduction

Many remarkable types of behavior in nature arise from complex signaling or metabolic networks. A mathematical description is one way to represent the dynamics of a network that is amenable to human interpretation. Mathematical models also often permit the prediction of new or rare behavior. A predictive network model is essential for external control of complex cellular behavior. However, manually finding a full analytical expression can be an arduous task—particularly in multidimensional biological systems with nonlinear reactions, feedback and oscillations. Models can either be reductionist with the utmost possible detail [1–3], or effective or surrogate models that capture specific dynamics, often using simplified mathematical representations [4–10]. In both cases, the identification of metabolic and signaling network models and their parameters is of pressing practical interest [11, 12].

Active-learning methods applied to the metabolic inverse problem [13] offer the possibility of using a computational system not only to test hypotheses but to generate new ones.

The 'robot scientist' project [14–17] is developing systems that can (1) assemble the background knowledge into a model in the form of a logical graph, (2) use that knowledge to design the best experiment to identify the location of a genetic perturbation on that graph, (3) perform and analyze microbial experiments, and (4) use the results to design the next experiment or genetic modification, and it provides significant savings in cost and time over human or random selection of experiments.

Here we propose a new method that could accelerate the task of model inference by automatically correcting hypothesized models—simplifying terms or proposing new nonlinear structures—to improve the agreement between model and experimental observations. The proposed approach does not require the assumption that the variables being modeled obey power-law or piecewise power-law representations. We also test the ability to model *de novo*, without prior knowledge of the structure of the metabolic system under study. This approach can be applied either to existing time series data, *in silico* simulations of experiments, or wet lab experiments suggested (or controlled) by the algorithm. Finally, we adapt the process to identify an invariant quantity in the system that plays a key role in predicting the system dynamics.

We performed computational experiments to analyze a seven-dimensional nonlinear model of glycolytic oscillation in yeast by perturbing and collecting noisy observations based on a well-characterized target model that could serve as a benchmark for validation of our methodology. Past research [18–20] has demonstrated the construction of a model of a biological system using automated experimental design. Here, we provide new advances into automated modeling: the ability to utilize exceedingly large data sets efficiently with fitness prediction, improved representations of mathematical models in the computational search, and integration of these approaches with automated experiment design. The advances presented here enable the first automated dynamical modeling of a realistic and biologically interesting system, where the system is an order of magnitude more complex and nonlinear, and in the presence of an order of magnitude higher measurement noise than we have previously shown.

A variety of methods has been used to infer gene regulatory networks (GRN) [21, 22], including genetics, biochemistry, molecular biology, and medicine [23]. Most often, preexisting models are used to provide a functional form, and then an optimization technique is used to fit the model parameters. Because of the breadth of data available, much of signaling network inference is based upon high-throughput mRNA microarray data for gene arrays, while metabolic network analysis considers both gene expression and high-throughput mass spectrometry of metabolites [24]. There are various challenges specific to the inference of metabolic networks from such data [25], since metabolism includes not only transcriptional regulation of enzymes, but also the conversion of substrate species with stoichiometric constraints. The computational challenge is exacerbated by the range of metabolic time constants and concentrations, each of which can easily span many orders of magnitude.

While there remain many unsolved problems in the inference of GRN models, metabolic networks surpass many other biological networks in terms of their breadth, detail, quantitative nature and experimental validation. Currently, it is possible to obtain quantitative, dynamic measurements of metabolic concentrations, metabolite fluxes, and genetic modification simultaneously, providing an important connection between the transcriptome/proteome and cellular phenotype [26, 27]. The speed and efficiency with which these data can be acquired suggest that metabolic networks are well suited for automated inference using computer-designed and -controlled wet lab experiments.

The most common mathematical form used to represent a metabolic network is a set of ordinary differential equations (ODEs) that describe the time derivatives of chemical concentrations [28] in the system as a function of its current state. ODEs are amenable to human interpretation because they are deterministic models and explicitly encode causal relationships [29], including feedback loops that are difficult to model using other methods. Terms in the differential equations correspond to reactions occurring in the system based on their connectivity, such as first- and second-order rate laws, power laws and Michaelis–Menten kinetics [30].

Methods such as symbolic regression [31–34] are capable of identifying and adapting differential equations automatically from experimental data [18, 35, 36]; however, substantial challenges remain to scale this approach to the dimensionality and functional complexity necessary for biological applications. We describe how an integrated framework of equations, graph-based symbolic encoding [36], fitness prediction [35, 37] and estimation exploration [18–20] can for the first time provide the degree of symbolic regression required for biological applications. In addition, this symbolic regression approach also allows the extraction of an algebraic expression that remains essentially invariant under all dynamic conditions of the model, shown by means of an example involving the yeast glycolytic oscillatory model. This result has key implications for enabling wet lab automation via computation of kinetics and the related sensitivity coefficients for the observed variable in the model, constructed via symbolic regression with data from biological experiments.

Background

Metabolic modeling

While our approach is generally applicable to a variety of biological networks, we restrict the present analysis to metabolism. Given the breadth of metabolic networks, we find it useful to organize metabolic models into three categories: comprehensive versus localized, static versus dynamic and linear versus nonlinear. All metabolic modeling approaches share these three categories, as shown in table 1. Among these approaches, metabolic control analysis (MCA) examines sensitivity of steady-state metabolic fluxes and concentrations to small perturbations of control variables. This provides some insights into the metabolic

Table 1. Properties of various modeling approaches used to study metabolic systems.

Name of the modeling approach	Linear or nonlinear	Static or dynamic	Localized or comprehensive
Genome-scale modeling using generalized mass action kinetics [38]	Linear	Dynamic	Comprehensive
Flux balance analysis (FBA)	Linear	Static	
Metabolic control analysis (MCA)	Linear	Static	
Dynamic flux balance analysis (dFBA) [40, 41]	Linear	Static	Between localized and comprehensive
Dynamic metabolic control analysis (dMCA) [42–44]	Linear	Static	
Cybernetic modeling [47]	Nonlinear	Dynamic	
Biochemical systems theory (S-Systems approach) [45, 46]	Nonlinear	Dynamic	Localized

inverse problem [38], which involves determining the nature of the equation network that underlies observed behavior using techniques such as reverse engineering or systems identification. MCA analysis further suggests that the metabolome may provide more information about cellular phenotype than the transcriptome, as changes in individual enzymes have little effect on metabolite fluxes but major effects on metabolite concentrations [39]. The propagation of impulse perturbations in such chemical concentrations can also be used to infer kinetic network topology [40, 41]. Genetic programming techniques, previously demonstrated for the reverse engineering of chemical reaction networks from observed time series data, can be used to infer both the topology and numerical parameters of metabolic networks [30, 41]. For example, evolutionary programming produced the best solution for a three-enzyme problem despite its large computation time [42], while gradient methods performed the worst.

Reverse engineering, a deterministic model of a metabolic network, consists of determining both the correct functional *form* of a set of ODEs to describe the system and the proper set of model parameters to fit experimentally collected data to within a given tolerance. The inverse metabolic problem is universally recognized as very hard [14, 39] and is most likely NP complete [21, 43]. A number of approaches, including global nonlinear optimization approaches [44–47], have been developed to solve this problem. Similarly, neural networks have also been used to address the metabolic inverse problem, where the measurable variables are the inputs and the unknown parameters are the outputs. The method is predicated on knowledge of the pathway structure and requires multiple hidden layers in the case of nonlinear kinetics (such as Michaelis–Menten) [43]. Alternatively, a neural network can be used to map the relationships between concentration gradients directly from data. In both cases, the resulting network is used to primarily predict performance, rather than to gain insight into the structure of the underlying system.

Within the past decade, a variety of techniques has been introduced to accelerate the estimation of model parameters [48]. A more daunting challenge is to determine the mathematical representation of the model's system dynamics. This requires incorporation of steady-state solutions and constraints that ensure stoichiometric consistency. A difficulty with the application of such *a priori* constraints is the possibility that the system will deviate significantly from the assumed optimal condition under transient abnormal conditions, such as myocardial ischemia. While this can be overcome with dynamic flux balance analysis (dFBA) [49], the researcher must still choose an appropriate set of constraints to describe the transient behavior of the system, which can be challenging.

In the larger context, it is recognized that traditional metabolic flux analysis must be extended to include functional genomic information [24, 50], particularly since the regulation of metabolic behavior is shared by both metabolic constraints and transcriptional regulation [51]. Further, metabolic modeling and detailed comprehensive models of biological networks increasingly suffer from the identifiability problem [52], where there is an inability to distinguish experimentally between parameter combinations that produce identical measurements. Consequently, additional methods are needed to reduce model complexity of large biochemical systems [28]. We address this need by presenting an approach that can identify local or effective models for nonlinear and dynamic subsets of larger systems. This will aid in exploring the underlying physiology and enable accurate external control of the system [53] and the optimized design of wet lab experiments to address these challenges.

Dynamics of biological systems and their invariants

In recent years, the application of concepts from dynamical systems theory to biological systems has led to the development of systems biology [54]. This has enabled the study of complex biochemical systems with nonlinear enzyme kinetics and regulatory feedback loops. Models of such systems can aid in recreating phenomena such as glycolytic oscillations [55] with the help of tools that are also used in many other disciplines [56] to study behavior such as bifurcations and limit cycles. Often, these are identified using a system linearized about its steady state [57] and can lead to the discovery of structures in phase space called attractors [58]. These include stable, unstable and center manifolds, collectively known as invariant manifolds [56]. Similar studies have also been attempted using alternative approaches, such as Fourier transforms, to identify control mechanisms for autonomous oscillations [59]. For example, variations in gene expressions in multistable systems are reduced over the course

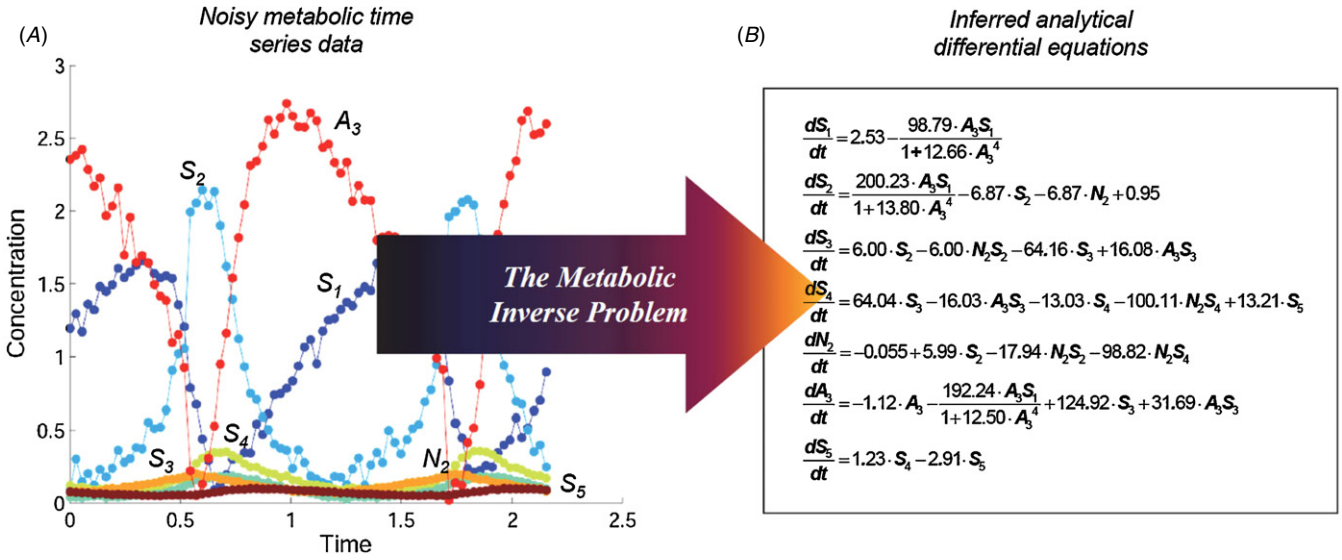


Figure 1. Concept and flow chart for automated model inference. (*A*) Noisy time series data reflecting anaerobic metabolism concentrations over time (presented with a higher-than-required sampling rate). (*B*) Noisy time series data are automatically translated into a set of coupled analytical differential equations without prior knowledge of the system structure.

of time in a process known as canalization, which has recently been investigated by means of invariant manifolds [60]. This is computationally expensive, however, and has therefore led to alternate approximation methods [61, 62] that can improve the performance of many model reduction algorithms [63, 64] and the partitioning of the system into slow and fast components [65]. Identification of these invariant manifolds is therefore of interest for fields such as chemical kinetics [66], as it enables the reduction of a complex model with many variables to one containing only a few. Such advances show the applicability of methods from dynamical systems theory to understanding problems of interest in metabolic networks.

Classical mechanical systems such as the double pendulum can be described in terms of Hamiltonians, Lagrangians and equations of motion which are derived from the physical principles that govern their dynamics. In contrast, biochemical systems were not expected to be governed by similar laws. However, symbolic regression applied to an oscillatory physical system has uncovered such invariant expressions using only observable data [67]. Herein we apply this approach to the model of yeast glycolysis. While the relationship between the derived invariants and the model dynamics is yet to be fully understood, the invariant form yields the equations for the observed variables, which are selected from species amenable to measurement, and their numerical sensitivities.

Methods

Overview of the approach

We first provide a brief overview of the proposed approach for adapting and modeling dynamical models before describing the methods in more detail. The purpose of our approach, illustrated in figure 1, is to identify the differential equations that describe the reactions taking place in the metabolic network by designing numerical experiments and searching for dynamical equations automatically. Initially, we have no data, and the first experiments could either be to observe the system performance for a randomly selected set of initial conditions within the normal bounds of the system, or to make simple observations of nominal stable behavior, such as stable nodes and limit cycles. Later, these experiments are chosen optimally to discriminate among the candidate models in the equation search. We perform experiments in a virtual wet lab, where a glycolysis system is simulated by a black box that outputs noisy time series data (also see methods in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia).

Once we have initial time course data from the system, we then begin searching for each differential equation of the system using symbolic regression [31–34] and fitness prediction [35, 37] (also see methods in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia). Symbolic regression searches both the structure, i.e. analytical form, and the parameters of an equation to fit and explain the data in the most parsimonious way. The second component in our method, fitness prediction, simplifies large data sets to accelerate the equation search while maintaining accurate ranking among all candidate differentials in the search. Our representation of a differential equation in the search is an acyclic graph [36]. This encoding also accelerates the equation search while producing more parsimonious expressions on average in comparison to traditional encodings. We have made software implementing these techniques available online for download at [68].

As many differential equations compete to fit the training data, it is likely that multiple equations are able to explain the behavior in different ways. Given these multiple explanations, we use an estimation exploration algorithm (EEA) to search in parallel for experiments that will maximize disagreement in the predictions of the evolving set of models [18–20].

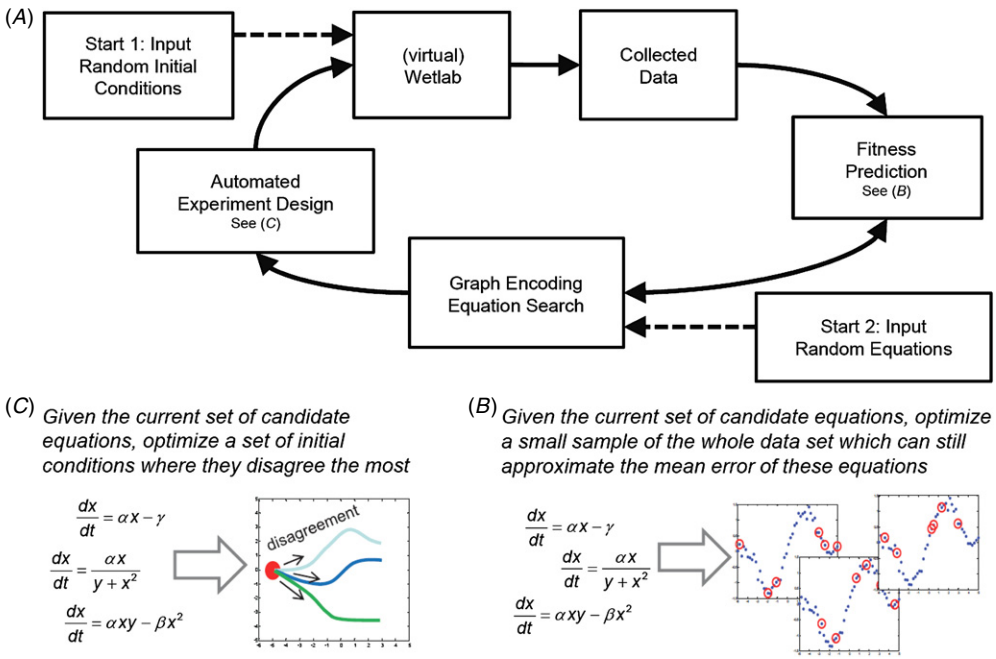


Figure 2. Workflow detailing the steps involved in our approach. (*A*) Method starts with a set of randomly chosen experimental data. Next, fitness predictors are optimized to approximate the complete data set (*B*). Graph encoding is initiated with random equations, and candidate models are then coevolved to optimize to the fitness predictors. The new candidate models are then used to design new experiments (*C*) which maximize disagreement among model predictions to refine their structure, and the cycle is repeated.

For a dynamical system such as glycolysis, we represent an experiment as a set of initial conditions that we apply *in silico* to the black box system so we can observe its transient trajectory and behavior. We dictate the most informative experiment to be the set of initial conditions in which the current population of equations has the highest statistical variance in its predicted dynamics. Periodically, we perform the best experiment on the black box system, collect new data, and once again evolve equations to explain them. We repeat this process until a dominant model emerges. Figure 2 outlines the overall approach, whose steps we now describe in detail.

Symbolic regression and fitness prediction

Genetic programming is a widely studied class of evolutionary algorithms inspired by biological evolution [34] that can be used to search for mathematical models. In a traditional genetic program, an initially random population of individuals evolves iteratively in computer memory to maximize some objective—for example, equations to model experimental data with the lowest squared error. Equations with the highest fitness, i.e. the lowest error, persist in the population to *recombine* (genetic crossover) and *mutate* to replace less-fit individuals.

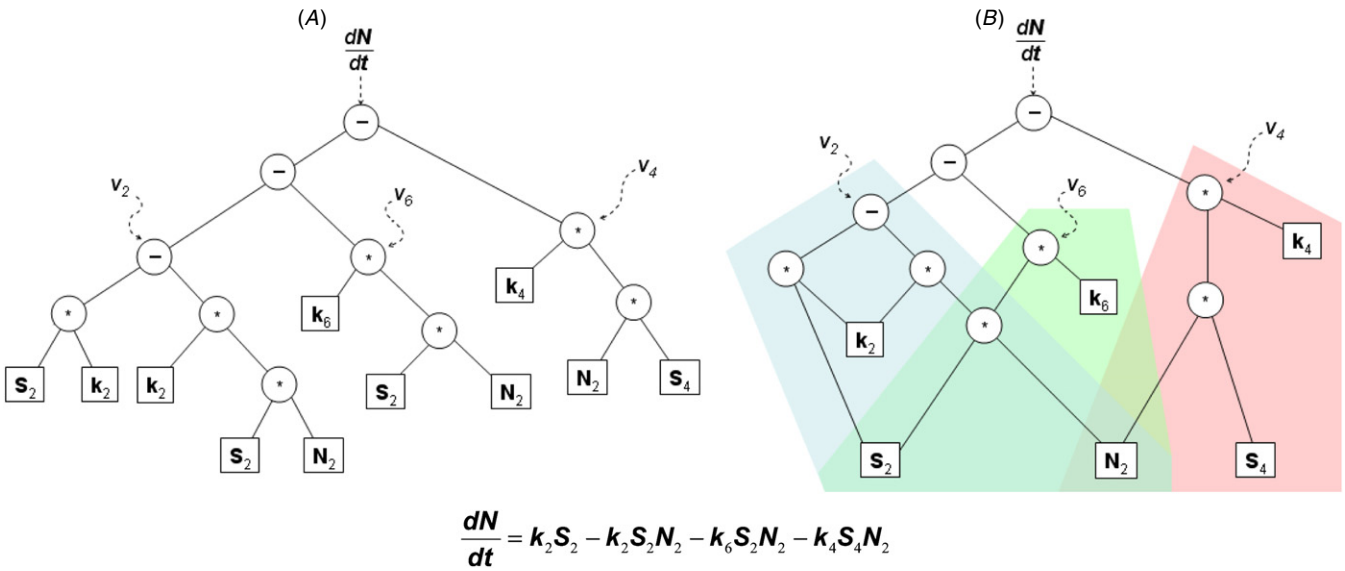
Symbolic regression uses *genetic programming* to evolve (compete) algebraic expressions to explain experimental data [34]. Unlike polynomial (parametric) regression or neural networks which also fit data, symbolic regression searches a space of analytical equations to explain available experimental observations. Symbolic regression composes equations using basic algebraic building blocks with the aim to formulate simpler (e.g., fewer parameters) or more natural expressions (robust to perturbations) that are most likely to correspond to the underlying intrinsic mechanisms of the system.

Symbolic regression compares candidate equations by calculating their residual errors on the experimental data—also known as the equation's *fitness metric*—for example, using square-error or correlation. The scaling of symbolic regression is dominated by the size of the equations that need to be evaluated, which can be exponential in the worst case, rather than the dimensionality of the system [36].

This paper emphasizes that this approach can be made tractable for appreciable systems. In past research, algorithms have used all available data at once to evaluate the fit. However, this metric can be overly stringent and inhibit equations from building intermediate expressions needed for the final model.

Instead, we have developed a method called *fitness prediction* to reduce overall computational cost and to approximate the local search gradient [35, 37]. A fitness predictor is a small subset of all collected data samples that approximates the entire data set. For example, if a data set contains 1000 data points, a fitness predictor could be a list of 10 points from the full data set. If the samples in the predictor are chosen well, measuring fitness on the smaller sample is sufficient to rank and discriminate the candidate equations in the population of equations.

The data subset is adapted within a separate population of fitness predictors (data subsets) that evolves in parallel with the symbolic regression of differential equations. The fitness of a fitness predictor is its ability to approximate error on the full data set for the current population of equations.



$$\frac{dN}{dt} = k_2 S_2 - k_2 S_2 N_2 - k_6 S_2 N_2 - k_4 S_4 N_2$$

Figure 3. Two analytical model representations for NADH in the cell glycolysis model. (*A*) Tree encoding, and (*B*) graph encoding of the same equation. While the tree encoding is simpler to manipulate algorithmically (e.g., alter subexpressions), it requires redundant subtrees and is prone to produce large equations that may not accurately represent the biological system. The graph encoding couples subtrees, thereby biasing equations to preserve simpler shared expressions.

The differential equation and the fitness predictor populations coevolve together throughout the equation search. In the symbolic regression population, the fitness of each equation is calculated using the data samples in the top-ranked predictor at all times. In contrast to standard symbolic regression alone, equations still compete to fit the experimental data, but are free to drift in more trajectories; the predictors adapt with the goal of defeating poor deviations.

Fitness prediction allows a genetic algorithm to search a wider range of equations by adapting the fitness heuristic and reducing its computational cost. An interesting result [37] shows that symbolic regression is more successful when equations are pressured to explain only key features of the systems at any given time rather than the entire data set at once. The effect of overfitting is actually reduced when using fitness prediction as the predictors adapt (coevolve) to defeat overfitting to the subset [37], and consequently further overfitting. This allows equations to drift from the objective gradient, but the focus adapts with the population to prevent excessive divergence from the intended gradient.

Here, we have introduced the key concepts of symbolic regression and fitness prediction. Additional details and implementation information are available in [37] and the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia, where we provide the algorithm settings. We have developed our own software to perform all computational searches described in this paper, available at [68], but the procedure could be implemented in many ways on different software platforms.

Acyclic graph encoding of the model

The ability to identify an accurate and parsimonious differential equation model using symbolic regression can rely strongly on the *genetic encoding* (e.g., the genotype organization of a symbolic expression within the algorithm) used in the equation search. In particular, the encoding needs to be amenable to crossover and mutation operations of the evolutionary algorithm.

Traditionally, symbolic expressions have been represented as binary-trees, where parent nodes represent algebraic operations such as addition or multiplication, and leaf nodes represent symbolic variables and parameter constants (figure 3(*A*)). While tree encodings are simple and easy to manipulate in an evolutionary search, they are susceptible to producing complex and bloated equations, often resulting in unsuitable models for understanding the underlying system.

Instead of a tree, we use an acyclic graph encoding for symbolic regression that scales well computationally and exploits the shared structures found in differential equations [36]. The acyclic graph encoding represents a symbolic expression similarly to a tree encoding—interpreting parent nodes as mathematical operations such as addition and multiplication and leaf nodes as state variables or parameter constants—however, subexpressions can be reused (figure 3(*B*)). In our implementation, the encoding for the graph is an ordered list of operations much like assembly code: each operation builds up successive subexpressions in the final expression, using any preceding operations and symbolic variables.

The graph encoding takes advantage of redundant subexpressions, such as coupled reactions in metabolic networks. Effectively, a reused subexpression constrains the genotype of the equation, making these subexpressions more stable and less likely to drift to more bloated equivalents and thereby biasing the search against bloated equations and overfitting [36]. In the end, the graph-encoded equations are simpler and faster to evaluate on average, allowing us to scale the modeling procedure to more complex systems. Additional

Table 2. Model variables, the allowed range of initial states for the training data set, and the standard deviation of the limit cycle used to compute the amount of added noise.

Variable	Name	Range	Standard deviation
S_1	Glucose	[0.15, 1.60]	0.4872
S_2	Glyceraldehydes-3-phosphate and dihydroxyacetone phosphate pool	[0.19, 2.16]	0.6263
S_3	1,3-bisphosphoglycerate	[0.04, 0.20]	0.0503
S_4	Cytosolic pyruvate and acetaldehyde pool	[0.10, 0.35]	0.0814
N_2	NADH	[0.08, 0.30]	0.0379
A_3	ATP	[0.14, 2.67]	0.7478
S_5	Extracellular pyruvate and acetaldehyde pool	[0.05, 0.10]	0.0159

details are provided in [36] and the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia.

Automated experimental design

Once the symbolic regression step has evolved a population of candidate equations to fit the current set of training data, there may be several coherent equations that model the data in different ways—particularly in high-dimensional domains with sparse data where many equivalent explanations exist for the simplest behavior. But which of these several mathematical explanations of the system is correct?

The EEA is a method to automatically design a new experiment that can help differentiate the current model candidates and refine their structure [18–20, 69]. The purpose of the EEA is to decipher which model is likely to be correct by searching for experiment settings, initial conditions, perturbations, or procedures that cause current models to disagree most in their predictions.

The EEA designs and conducts the experiments in parallel with the equation search. Different experiments compete in a population to maximize the variance they produce in the current differential equations. Periodically, every 50 000 iterations (or roughly every 10 min), we take the best experiment and perform it on the black box system to collect new data to test predictions made by the current population of differential equations.

Generating data

For our experiments in this paper, we collect data in a virtual wet lab by numerically integrating the glycolysis model from an initial state and recording the state variables over the transient trajectory. The initial state is either randomly chosen (for collecting initial data before modeling) or chosen by the algorithm. For a given initial state, we record the system's state variables every 0.1 min until we have acquired 100 samples.

In our results, we report performance on two data sets: a training data set and a test data set. Only the training data set is used to evolve and select the differential equation models, while the larger test data set is used to help analyze and compare the different modeling methods considered in our results.

We confined all initial states to realistic environments indicated in table 2 for the training data. These constraints are the ranges each metabolite experiences when the system is left alone to oscillate naturally on its stable limit cycle. Therefore, each initial condition is a combination of typical individual metabolite levels found in the metabolic network's natural oscillations. In the case of the test data set, however, the upper-bound constraints were doubled to expand the phase space by a factor of 2^7. The looser constraints used for the test data set measure how well models extrapolate and predict new behavior in locations in phase space distant from those used in the inference of the model.

As part of the black box glycolysis model, we simulate taking physical measurements on a real system by adding normally distributed random noise to each state variable in each time sample. The standard deviation of the random noise added to each state is relative to the standard deviation of the state variable in the system's stable limit cycle. This gives variables with large magnitude oscillations higher noise than variables with smaller magnitudes and makes the noise impact independent of measurement units. We used 10% noise in most experiments, i.e. the ratio between the noise standard deviation and the metabolite standard deviation is 0.1. This is consistent with reports of ∼10% average standard errors of the measured metabolic oscillation amplitudes (with the percentage errors smaller on large-amplitude variables) [70] and added noise amplitudes of 10% used in other metabolic system estimates [48]. We estimated the numerical derivative of the data using Loess locally weighted polynomial fitting [71] with window size of 50 (also see methods and figure S3 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia).

In later experiments, we sweep the amount of noise from 0 to 50% noise. The 10% noise is the first level where identifying the exact system structure becomes nontrivial. After this point, five of the seven states can be modeled exactly up to 30% noise, and four of the seven at 50% noise. It is worth noting that many real biological systems may also contain different types of noise, such as asymmetrical noise, and a modified equation search method could be used in these cases, such as explicitly including symbolic noise sources in the ODE model [72].

Glycolytic oscillation models

For our *in silico* demonstration of this process, we begin with a published numerical model [73, 74] of glycolytic oscillation in yeast for the system upon which our algorithm experiments. Tables 2 and 3 and other tables in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia provide

Table 3. Differential equations describing glycolytic oscillation of the generating model (left panel) and the inferred model from the training data, which had 10% noise (right panel).

Original system	Automatically inferred system
$\frac{dS_1}{dt} = 2.5 - \frac{100*A_3S_1}{1+13.68*A_3^4}$	$\frac{dS_1}{dt} = 2.53 - \frac{98.79\cdot A_3S_1}{1+12.66\cdot A_3^4}$
$\frac{dS_2}{dt} = \frac{200*A_3S_1}{1+13.68*A_3^4} - 6*S_2 - 6*S_2N_2$	$\frac{dS_2}{dt} = \frac{200.23\cdot A_3S_1}{1+13.80\cdot A_3^4} - 6.87\cdot S_2 - 6.87\cdot N_2 + 0.95$
$\frac{dS_3}{dt} = 6*S_2 - 6*N_2S_2 - 64*S_3 + 16*A_3S_3$	$\frac{dS_3}{dt} = 6.00\cdot S_2 - 6.00\cdot N_2S_2 - 64.16\cdot S_3 + 16.08\cdot A_3S_3$
$\frac{dS_4}{dt} = 64*S_3 - 16*A_3S_3 - 13*S_4 - 100*N_2S_4 + 13*S_5$	$\frac{dS_4}{dt} = 64.04\cdot S_3 - 16.03\cdot A_3S_3 - 13.03\cdot S_4 - 100.11\cdot N_2S_4 + 13.21\cdot S_5$
$\frac{dN_2}{dt} = 6*S_2 - 18*N_2S_2 - 100*N_2S_4$	$\frac{dN_2}{dt} = -0.055 + 5.99\cdot S_2 - 17.94\cdot N_2S_2 - 98.82\cdot N_2S_4$
$\frac{dA_3}{dt} = -1.28*A_3 - \frac{200*A_3S_1}{1+13.68*A_3^4} + 128*S_3 + 32*A_3S_3$	$\frac{dA_3}{dt} = -1.12\cdot A_3 - \frac{192.24\cdot A_3S_1}{1+12.50\cdot A_3^4} + 124.92\cdot S_3 + 31.69\cdot A_3S_3$
$\frac{dS_5}{dt} = 1.3*S_4 - 3.1*S_5$	$\frac{dS_5}{dt} = 1.23\cdot S_4 - 2.91\cdot S_5$

details of the models shown in figure 4. In this seven-variable model, the respiratory chain (mitochondrial oxidative phosphorylation) is completely inhibited. The reaction network for this system, shown in figure 4(*A*), contains the main reactions of glycolysis and adjacent reactions producing ethanol and glycerol. We provide additional model details in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia. During the original model development, the complexity of the model was reduced by omitting many of the glycolytic reactions, and by lumping together other reactions, so that several of the model variables denote concentrations of pools of intermediates rather than concentrations of the individual compounds, e.g., the pools of triose phosphates (glyceraldehydes-3-phosphate, dihydroxyacetone phosphate) and pyruvate and acetaldehyde. This simplification has been rigorously justified using a judiciously applied quasi-steady-state approximation [75]. This particular model is capable of reproducing glycolytic oscillations with a period in the range of 0.10 to 12 min and has been used to study the temperature dependence and temperature compensation of yeast glycolytic oscillations [74].

Reverse engineering glycolytic oscillation in yeast

We used the model of glycolytic oscillations in yeast shown in figure 4(*A*) to simulate experimenting on a wet system. Glycolytic oscillation is one of the most common examples of oscillatory behavior at the cellular level and enables a broader understanding of the underlying dynamic processes that lead to rhythmic behavior. Of such systems, anaerobic glucose metabolism in yeast is most commonly studied. In a particular region of parameter space, all of the glycolytic intermediates show oscillatory behavior with a variation in the frequency of oscillation observed across species. In the vicinity of the attractor that is responsible for these oscillations, the system never reaches steady state and hence this behavior cannot be readily analyzed by equilibrium or stoichiometric approaches such as metabolic flux balance analysis [76]. We use this oscillatory system to demonstrate the capability of our approach to infer the equations governing a nonlinear dynamical metabolic system.

Our experiments placed the yeast glucose model (figure 4(*A*) and figure S1 (available at stacks.iop.org/PhysBio/8/055011/mmedia)) in a numerical black box and then allowed our algorithm to conduct *in silico* experiments on this black box. For our studies, we collect data by numerically integrating the differential equations in the black box glycolysis model and adding noise. Initial states, i.e. the initial conditions, are constrained to a specified range, and the initial states for the test data are sampled over a larger volume in state-variable space to determine how well models can extrapolate and predict new behavior.

Our goal is to find the exact differential equations of the unknown system algorithmically. More specifically, we are interested in modeling metabolic networks as a dynamical system—a set of ODEs. In a system of N state variables that we observe experimentally (e.g., extracellular concentrations of glucose (S_1) or NADH (N_2) over time), we must identify N (possibly nonlinear) differential equations. Synthesizing these mathematical models of a dynamical system is the most computationally intensive task in our procedure. We first smooth and then differentiate the observed time series data to produce its derivatives. We then search for the differential equations that reproduce each numerically estimated derivative.

We calculate the numerical time derivative of each variable at each data point in order to compare it with the value of the candidate differential equation explicitly. Accumulating errors of the derivative values avoids the need to integrate each differential equation, which can be computationally expensive [18].

Regression procedure for all methods

During regression for each compared algorithm—symbolic regression nonlinear regression and neural network regression—we track accuracy on both the training and test data sets over time. Only the training data set is used to update the models. By recording the accuracy of the model of the test data set over time, we can analyze later how well the regression procedure identifies models that generalize and extrapolate to data not in the training set (e.g., figure 6).

In nonlinear regression and neural network regression, the training set was constant, with 200 trajectories (random experiments on the black box). In contrast, the symbolic regression algorithm's training set begins with ten trajectories that were produced following ten different, random initial conditions, but subsequently adds new black box trajectories

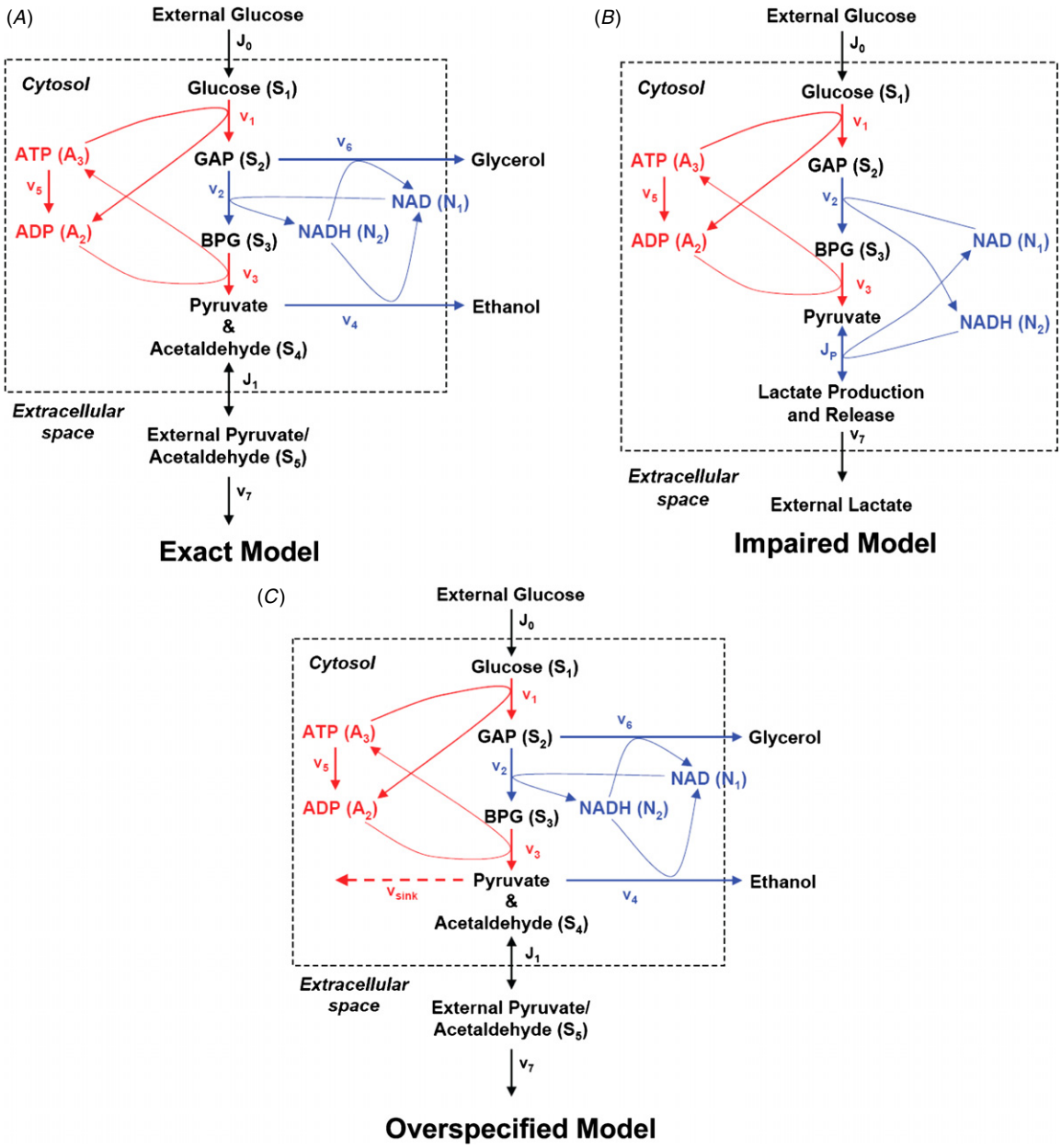


Figure 4. Reaction networks for anaerobic metabolism in a yeast cell. (*A*) Exact model includes membrane transport of glucose and pyruvate/acetaldehyde. Reactions in red involve ATP production/usage, and reactions in blue involve redox species production/usage. (*B*) Impaired model does not produce either glycerol or ethanol. (*C*) Overspecified model has an additional sink for pyruvate/acetaldehyde (S_4).

from initial conditions chosen by the algorithm throughout regression for a maximum of 200 total experiments, with 100 measurements per experiment. Hence such a single black box experiment provides 100 measurements of each variable over the duration of each experiment. As we discussed earlier, each black box experiment is followed by approximately 50 000 EEA iterations conducted during 10 min of real time, an interval that would be consistent with an automated wet lab experiment [53]. For all algorithms, the test data set was held constant. The test data set contained 100 random trajectories. We do not perform early stopping on any of the methods; however, performance trends such as overfitting can be gleaned from the fitness of the test data set using values from the training data set (also see methods and figure S7 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia).

Construction of an invariant expression

The search for a conserved quantity used a similar symbolic regression search as that which produced the dynamical model, but instead of modeling a specific signal, such as a numerical derivative, the search looked for an invariant expression.

Searching for invariant equations, however, is particularly challenging because an error metric is difficult to define [77]. For example, in any data set there are infinitely many trivial equations that satisfy a conserved quantity, such as x–x, or

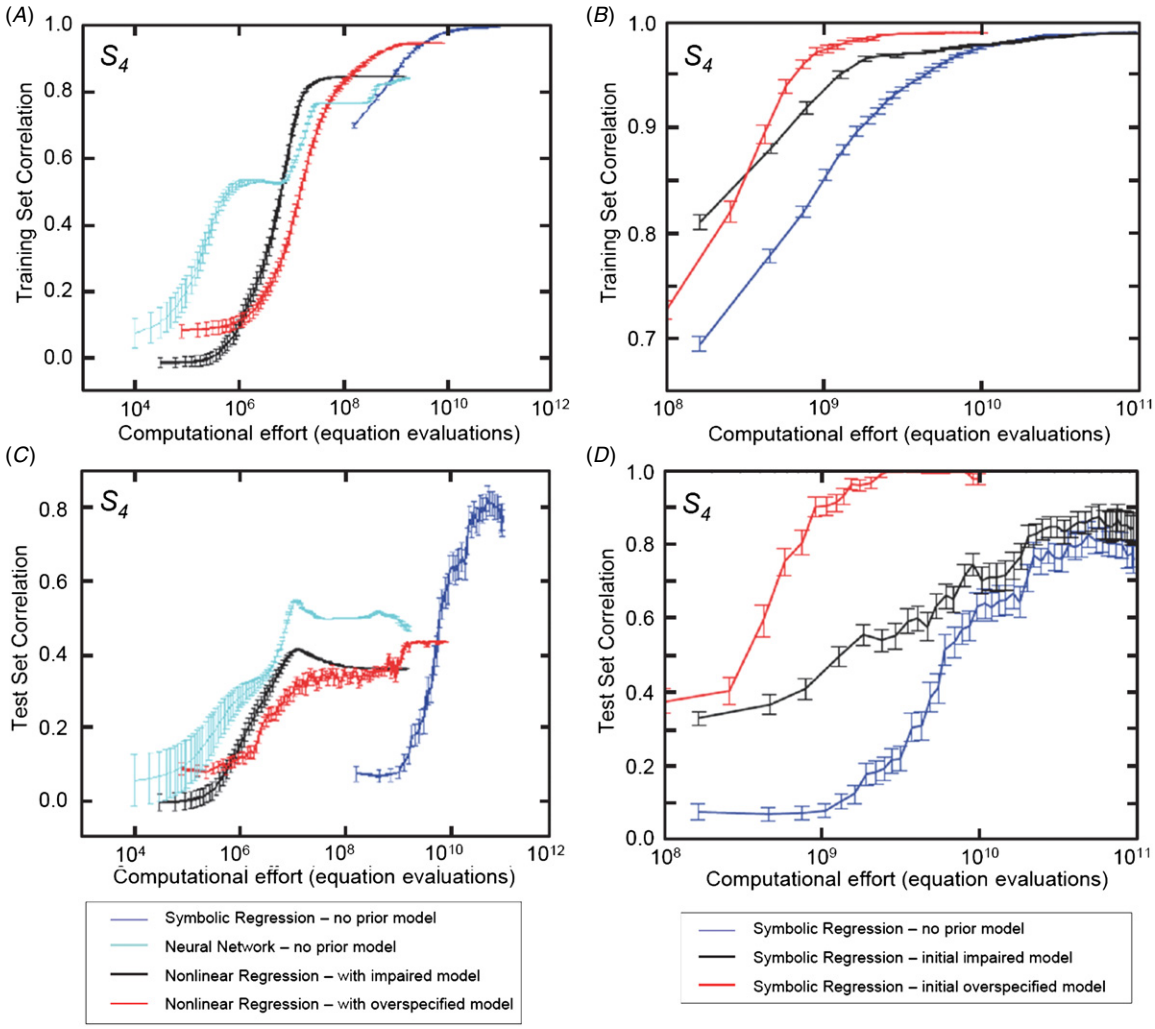


Figure 5. Performance comparison between symbolic, nonlinear and neural network regression. Left: nonlinear parametric regression to the impaired and overspecified models, and neural network regression, as compared to symbolic regression without a prior model. Training data performance (*A*) shows that all algorithms accurately explain the training data. The test data performance (*C*) of the same equations plotted for the training data shows that not all methods extrapolate well to the larger test set. Note that symbolic regression uses more point evaluations in the same amount of running time because it is a parallel search, whereas nonlinear regression and neural network back-propagation use serial updates. Right: performance comparison of symbolic regression when correcting a hypothesized model. (*B*) Blue curves represent the performance of the algorithm to the S_4 equation without any prior model. For the other two pairs of curves, the symbolic regression algorithm was seeded with an incorrect hypothesized model (black = impaired, red = overspecified) and the algorithm had to modify the seeded model to fit the original training data, i.e. to return to the model that was inferred during the progress of the blue curve. The figure shows the performance for the training data (*B*) and the test data (*D*) on the same equations. For both (*A*) and (*B*) results are averaged over 100 trials—error bars represent the standard error.

other more complicated identities that simplify to, or nearly simplify to, a constant. Instead, the invariant search looked for equations that accurately predict implicit derivatives estimated from the data [67, 77]. This requirement identifies invariant expressions that can also derive nontrivial dynamical relations in the data.

We searched for an invariant in the glycolysis system that used the variables S_1, N_2, A_2 and v_1, i.e. glucose, NADH, ADP and the rate of the S_1 reaction (glucose to GAP), respectively, as shown in figure 4(*A*). Invariant equations were rewarded for predicting the implicit derivatives between each pair (such as $\mathrm{d}S_1/\mathrm{d}N_2$, $\mathrm{d}S_1/\mathrm{d}A_3$, etc) and for using all four in the expression (a requirement for our post-analysis). If an expression used less than four of these variables, it was removed from the search population. We selected the most parsimonious invariant equation on the resulting search Pareto frontier (see for example, figure S2 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia).

Finally, we used nonlinear regression on the resulting invariant form to fine-tune the parameter coefficients because the invariants have some freedom in their exact coefficient values. In this step, we solve the invariant equation (e.g., $f(S_1, N_2, A_3, v_1) = 0$) for S_1. We then used Mathematica to tune parameters to predict the S_1 signal. The resulting model is predictive of both the implicit derivatives of the data set and the explicit S_1 signal while still remaining invariant.

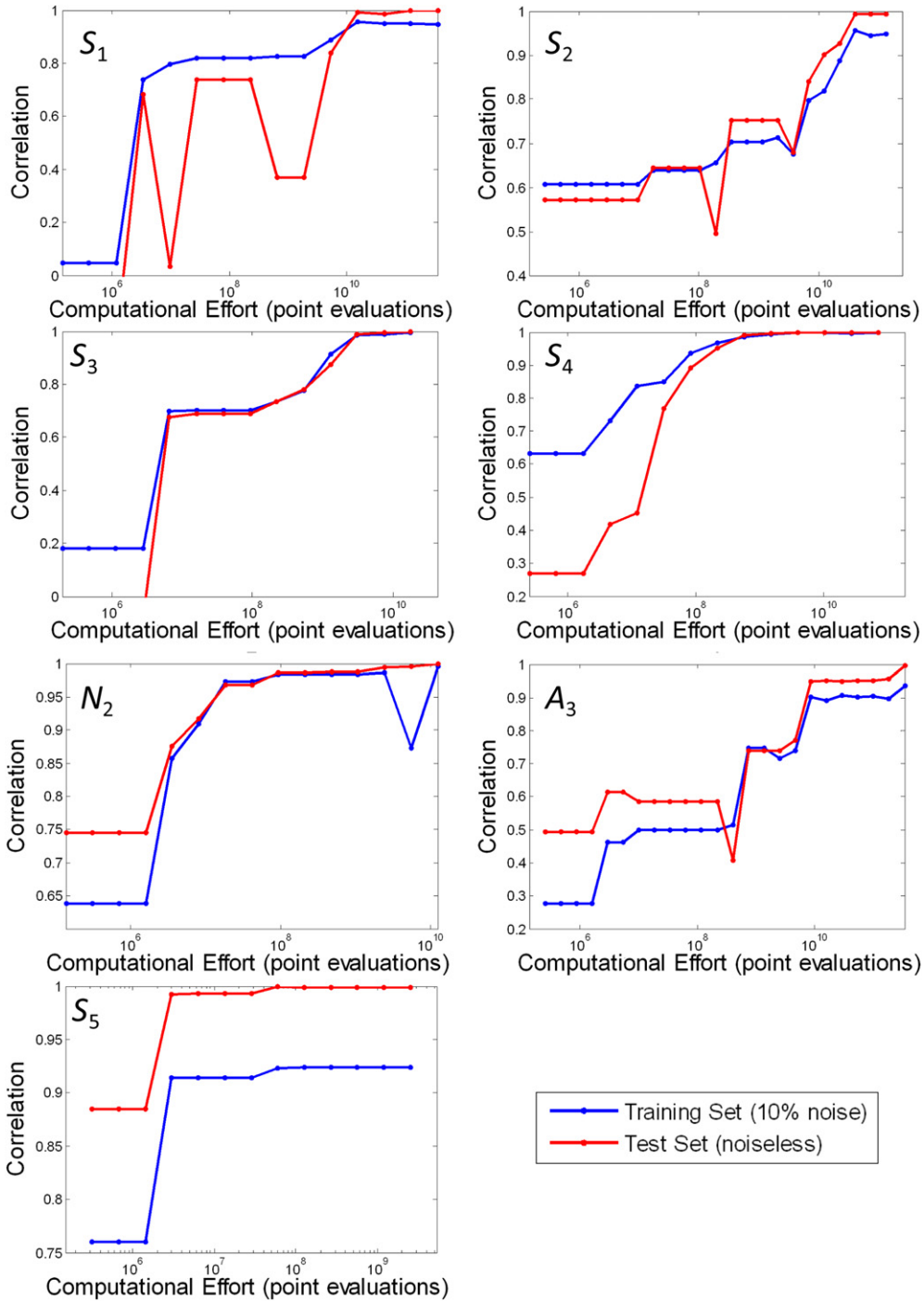


Figure 6. Fit to the data of the highest ranked equation during regression for each glycolysis variable. The blue series show the correlation coefficient to the training data, and the red to the test data. The training data contain 10% noise while the test data have none. The test data contain a larger range of allowed state variables (i.e. sampled with weaker constraints to explore a variable phase space that is 2^7 times larger than for the training set) to determine whether the model identified from the training data can extrapolate and predict new behavior. The variables are identified in figure 4 and table 2.

Results

We used the proposed automated modeling procedure to analyze the seven-dimensional model of glycolytic oscillations in yeast. We tested the ability of the algorithm to adapt and correct a partially correct hypothesized model chosen by the experimenter to fit the exact system, i.e. to augment expert modeling. We tested the regression of the entire system *de novo* (without prior knowledge of the system structure) and compared predicted results with those obtained from nonlinear parametric regression and fitting to neural networks. Finally, we modified the procedure to identify an invariant quantity of the biochemical system that is distinct from any conserved moieties that may exist in the model. The invariant also

predicts kinetics of an observed variable of the system (in our case v_1) which in turn allows for computation of the numerical sensitivities to a high degree of accuracy.

Correcting hypothesized models

The first experiment is to adapt or improve upon an existing hypothesized model, given a hypothesized differential equation. For example, the algorithm can modify a partially correct model by altering its existing structure and terms, pruning unnecessary terms, or synthesizing new terms to identify the exact intrinsic model. While methods for simplifying models already exist [52], these techniques do not compose new terms in a model or correct erroneous terms. Using automated data acquisition, the algorithm can also design experiments to differentiate multiple hypothesized models and test their correctness where they disagree most.

In order to use the algorithm to refine a given model, we first initialize its search population by seeding the initial population with the chosen models. Effectively, this biases the algorithm to reuse the structure of the given model, but does not restrict the algorithm from making large alterations. We first test the impact of this seeding procedure to correctly model the glycolysis S_4 differential equation using the impaired and overspecified models. (The hypothesized differential equations for S_4 in this experiment are given in table S5 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia.)

There are two possible impacts that seeding a hypothesized model can have on symbolic regression. In the best case, the seeded model may be very close and the algorithm only needs to make minor adjustments to converge to the intrinsic model. Alternatively, the seeded model may be absurd, in which case there is no benefit and the algorithm evolves *de novo*. Figures 5(*B*) and (*D*) exemplify these two types of symbolic regression behavior for the S_4 differential equation.

As shown in figures 5(*B*) and (*D*), in the overspecified case, the seeding has improved by a factor of 10 both the speed to regress and the reliability of the results in comparison to regression without a prior model. In effect, the algorithm has corrected the model by removing the sink term and fixing the differences in parameters. In the impaired model case, the effect is much less dramatic. The algorithm must construct the two missing terms. However, the seeding improves the reliability of convergence on average. There is also the possibility that the initial seeding might trap the algorithm in a local error minimum from which the present algorithm cannot escape. Similar problems have been encountered and addressed in conventional nonlinear regression schemes by, for example, choosing random data points at some distance from the regressed data to ensure that the solution is in fact a global rather than local minimum. Similar approaches could be taken with our method, although it is unclear whether this would provide any advantage to *ab initio* regression.

Modeling without a prior model

When modeling without a prior model, we do not assume an initial hypothesis model. The initial equations in the search are randomly initialized; therefore, the algorithm's search for each ODE equation starts at a random point. Table S4 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia provides an example of how the equation for S_2 evolves during regression. We conducted ten independent trials to collect data and model each equation (table 3) corresponding to figure 4(*A*). We show in figure 6 the runs that reached the highest performance on the training data (blue). Additionally, at each level of computation we measured performance of the same equations on the test data (shown in red). Variables S_3 (1,3-bisphosphoglycerate), S_4 (cytosolic pyruvate and acetaldehyde pool), N_2 and S_5 were the fastest equations to infer, and their performance curves gradually converge monotonically during regression. Equations for S_1, S_2 (glyceraldehydes-3-phosphate and dihydroxyacetone phosphate pool) and A_3, which have the most nonlinear structure, show performance that is more rugged. Dips in the training data performance indicate that data from a new black box experiment revealed dynamics that were not in the current data set (or perhaps underemphasized). Such dips tend to precede large improvements in performance.

The equations with the best fits to the *training data* obtained from the model in figure 4(*A*) for the ten trials are shown in table 3. The automatically inferred equations are nearly identical to the black box generating numerical model. Some slight differences remain: most notably, the parameters are inexact, which results in a slight mass imbalance, and one nonlinear term is approximated by a linear term in the S_2 equation. Specifically in our *ab initio* model inference, the S_2 equation approximates the $N_2 * S_2$ term and adds a constant term (0.9467) to the ODE in table 3. The $N_2 * S_2$ term comes from the balance of v_2 and v_6, where v_2 is the conversion of S_2 to S_3 and v_6 is the loss of S_2 to glycerol production. Both the v_6 and v_2 fluxes are NAD dependent, which gives rise to the $N_2 * S_2$ term through a simple application of mass action kinetics. The decoupled $N_2 * S_2$ dependence is now represented as a linear combination of $(N_2 + S_2)$. For the N_2 equation, the combined action of v_2 and v_6 is properly inferred. However, there is (once again) a small constant term (-0.0549) in the N_2 ODE that compensates for the fact that the NAD pool is not being strictly conserved.

Figure 7 shows integration of the inferred model compared to the exact model (figure 4(*A*)), with the differences in frequency being the results of the small mass imbalances that are compensated by the additional terms. Despite these small differences, figure 7 shows the same behavior as the original system. Since symbolic regression does not have any inbuilt 'chemical logic', it is unable to recognize and constrain reaction rate expressions that appear in multiple ODEs. The ramifications of this are twofold: first, the inferred model incurs small mass imbalances within the system that manifest themselves as a carbon loss or a source term in the energetic pools (ATP and NADH); second, the inferred model compensates for the imbalances by adding compensatory terms.

Identification of the simplest equation, S_5 (external pyruvate/acetaldehyde), without *a priori* information required approximately 1 min for $\sim 3 \times 10^6$ evaluations for

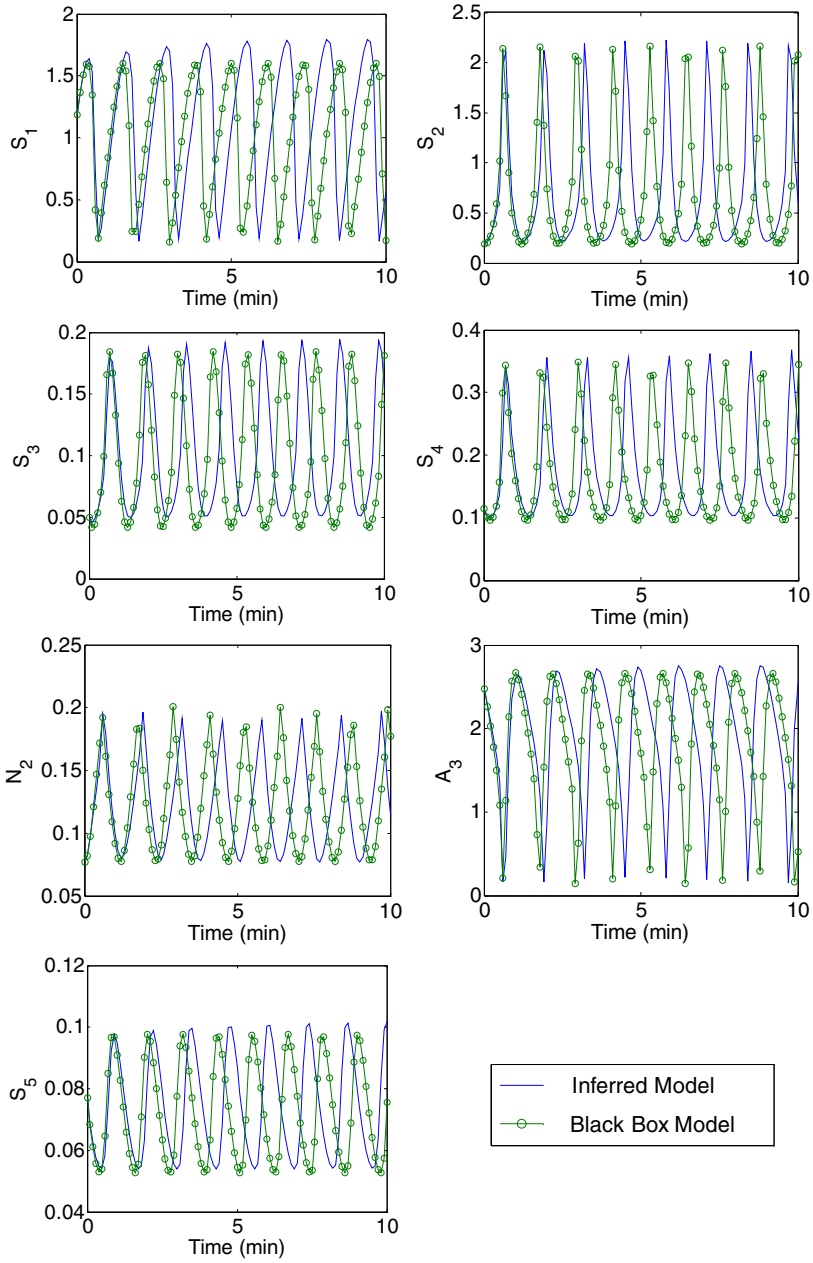


Figure 7. Integration over time of the exact black box and inferred models. The inferred model shown in table 3 differs from the exact model by a slight mass imbalance. Integrated over 10 min, the inferred model captures the same behavior. While small differences in derivative values tend to accumulate during integration, the inferred model captures the integrated behavior remarkably well. The inferred model predicts early behavior accurately and exhibits the same qualitative dynamics later in time, differing only slightly in the phase.

ab initio modeling, and ∼1 model/experiment/evolution cycle. In contrast, the time to regress the most complex differential equations in the glycolysis model, A_3 (ATP), was approximately 1–2 h, and involved $\sim 4 \times 10^{11}$ point evaluations on four workstations (eight 2.4 GHz cores), representing ∼200 model/experiment evolution cycles and $\sim 2 \times 10^9$ point evaluations/cycle.

Comparison to established methods

Searching the space of symbolic differential equations is unique in that it does not require prior information about the system or a prior model. Hence, it is difficult to compare this model inference process with the process of historical development of an existing metabolic model. To demonstrate the capabilities of this approach to specify a compact representation of a model, as might be needed for prediction and control of metabolism in a bioreactor [53], we have compared our approach with two relevant methods: nonlinear regression and neural network regression.

Neural networks are recognized as being useful for predicting a time series when the underlying mechanism is unknown or is too complex to be easily represented, or noisy data limit the analysis [44]. Such numerical models are less amenable to human interpretation, but can model and predict data similar to that used for training. In the

neural network regression, we use a 1024-neuron hidden layer network mapping the seven state variables to their seven respective time derivatives. The output layer consists of linear perceptrons. We use standard back-propagation to train the network and bias-node weights.

In nonlinear regression, a preexisting mathematical model is chosen to be fitted to the data. The selected preexisting model is assumed to closely relate to the actual, underlying model of the system but may have a slightly different structure—it may be missing key terms, or have unnecessary terms, or incorrect terms. Regressing data to the wrong model may provide a low-error fit, but with parameter values that are in fact quite different from the actual ones because of the adjustments required to fit the wrong model to the data. We fit the nonlinear regression models via gradient-descent, with initial parameters given by their theoretical values (see table S3 of the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia).

For the nonlinear parametric regression comparison, we chose two slightly different pre-existing glycolysis models, also shown in figures 4(*B*) and (*C*)—one that is a simplification of the exact model (the 'impaired' model) and one that is more complex in that it replaces some dynamics with an additional sink term (the 'overspecified' model). The impaired model is produced by eliminating the glycerol (v_6) and ethanol (v_4) production and having the NADH → NAD recycle occur with the production of lactate from pyruvate. This essentially converts the yeast model into that of a mammalian cell. The overspecified model is produced by relaxing the assumption of no carbon loss to other cellular synthetic processes in the yeast model (fatty acid biosynthesis, amino acid production, etc). This is accomplished by adding a carbon sink term (v_{sink}) to the pyruvate pool, whose rate is primarily controlled by the presence of ATP.

On average, all three algorithms model the training data equally well, but some do not generalize well when the regression results are applied to the broader test data set that was not used for training (figure 5(*C*)). It is clear that nonlinear regression of both related models can explain a substantial amount of the S_4 dynamics, particularly within the training data. However, extrapolation to the wider domain of the test data only reaches correlation of approximately one-half. Similarly, the neural network accurately models the training data, but as shown by the early dip in correlation, it significantly overfits before converging, possibly due to the added noise. Additional details are provided in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia.

Invariant-derived model results

Finally, we tested the ability to find an invariant quantity directly from the data. We modified our procedure by changing the fitness metric from the error in predicting the rate of change of each variable to the error in estimating implicit derivatives between pairs of variables in the system [67]. We chose the equation with the lowest magnitude (most invariant) in the resulting models as closest to a true invariant of the system.

This analysis was carried out for three different temperatures of 280, 284 and 293 K, as it is known that the period of yeast glycolytic oscillations increases with temperature. An invariant with the same form was extracted for all three temperatures, and contained three coefficients. This invariant had the form $I_{\text{Temp}} = A_3\ S_1 - (v_1\ ({A_3}^4\ (k_1 - k_2\ N_2) + k_3))$ where k_1, k_2 and k_3 are coefficients of the terms in the invariant that vary with temperature. These coefficients are listed to the right of each panel in figure 8 for all three temperatures, and the time courses of the corresponding invariants are shown in figures 8(*A*)–(*C*), respectively. It can be seen that the invariants, computed by solving the equation $f(S_1, N_2, A_3, v_1) = 0$ for S_1, have small magnitudes relative to the oscillations of the variables themselves in figure 7.

As a demonstration of the utility of the invariant, the variable v_1 can be obtained by solving the invariant expression for v_1, which has been plotted versus time in figures 9(*a*), (*c*) and (*e*), respectively, along with the corresponding v_1 from the analytical model. The error between the invariant-derived and analytical model v_1 is shown in figures 9(*b*), (*d*) and (*f*), respectively (see also figure S8 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia). Numerical sensitivities can be obtained from the invariant expression by differentiating with respect to the species of interest. We computed sensitivities of v_1 with respect to A_3 and N_2 using the invariants for the three temperatures, which have been plotted in figure 10 along with the corresponding difference with respect to the actual model-derived sensitivities (see also figures S9 and S10 in the supporting information available at stacks.iop.org/PhysBio/8/055011/mmedia).

This result shows that searching for invariants directly can closely model the kinetics of the observed variable and its associated numerical sensitivities without access to any additional internal model structure. It could also be useful to use an existing dynamical system from which to then identify an invariant expression to derive the kinetics and sensitivities, yielding a compact representation of the model's dynamics.

Modeling limitations

Modeling any dynamical system requires some baseline assumptions, such as the set of state variables with which to model the system [78]. This is a form of implicit prior knowledge, whereas we ordinarily refer to prior knowledge as higher level assumptions, such as known model relationships or structures. We therefore implicitly assume that the system's key dynamics can be accurately derived and represented by the choice of variables.

While our present implementation does not require chemical mass balance other than what is provided by the ODEs, mass-balance logic could be built into future implementations, albeit with some performance costs. For example, one could assume the current best model and use it to test for constraint violations. Models and experiments could then be penalized or rejected accordingly, focusing the search on the valid equations as shown in [48]. Alternatively, it may be possible to enforce chemical logic [79, 80] by providing

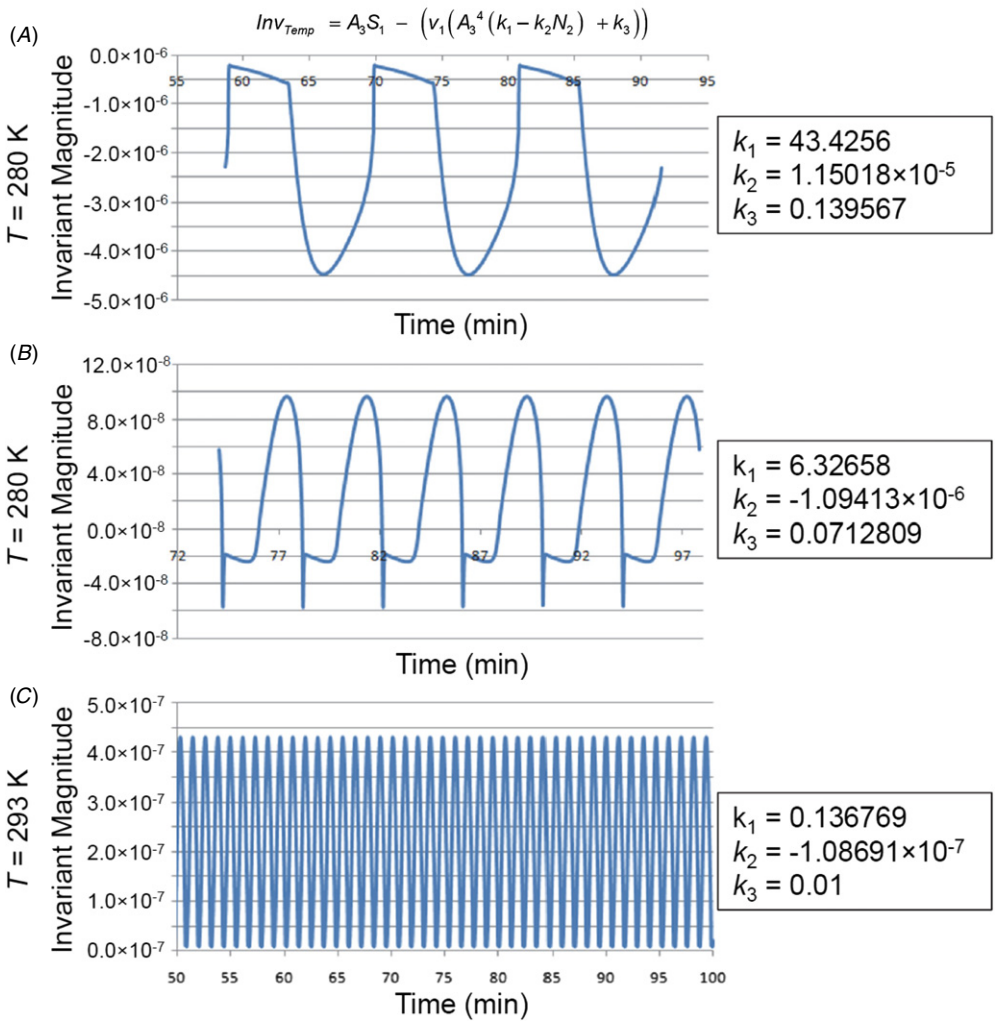


Figure 8. Description of the invariant expression obtained by means of symbolic regression at three temperatures. All invariants are equations of the form at the top of the figure. To the right are shown the coefficients of the invariant that was constructed for each temperature using data from the published model. The graphs show the invariant plotted as a function of time for the three temperatures. Note that the invariant oscillates with a period whose length equals that of the model, and has very small magnitude (10^{-6} to 10^{-8} as shown on the left-side scale) relative to the scale of the oscillations of the system variables (figure 7). Numerical experiments have shown that the invariant is sensitive to the number of significant digits included, implying that accuracy can be improved by allowing the symbolic regression algorithm to run for a longer time, leading to an invariant with smaller magnitudes that are even closer to zero.

the SRA algorithm with an appropriate chemical operator set rather than just the four algebraic operations we allowed.

A potential difficulty when modeling the dynamics of biological systems arises when only a few relevant states in the system can be observed directly. In such cases, these variables must be estimated or inferred in parallel. For example, missing state variables could be represented by one or more parameters that vary with time (e.g., a Taylor series). Missing states can also be estimated, or sometimes exactly derived, by the available observables—as is done in inferential sensing [81]. Finally, missing states could be encoded directly into the model as latent variables. In this case, an initial condition and a corresponding equation could be included for each latent variable in the model search, and this variable would then be simulated using the observable data to produce the missing time series. If the missing states are ignored completely, the inferred dynamical models will necessarily be higher-level approximations of the system's dynamics. These are important areas of future research; in this paper, however, we focus on the case where we can observe, or have already accurately estimated, the relevant states.

Finally, searching for a dynamical model from scratch can be computationally intensive. A previous study [36] showed that the performance is dominated by the size of the target expression rather than the dimensionality of the problem. Therefore, systems with an increasing number of state variables take linearly more computation effort (an additional search for each additional state variable). However, modeling systems with extremely dense and complex equations (e.g., equations that are difficult to write or even read manually) could require substantially more computational effort with the depth of the expression. Finally, in cases where the algorithm could not deduce exact equations in detail, it can still provide an approximate high-level model of the major dynamics., i.e. an effective or surrogate model that could be used for controlling the system [53].

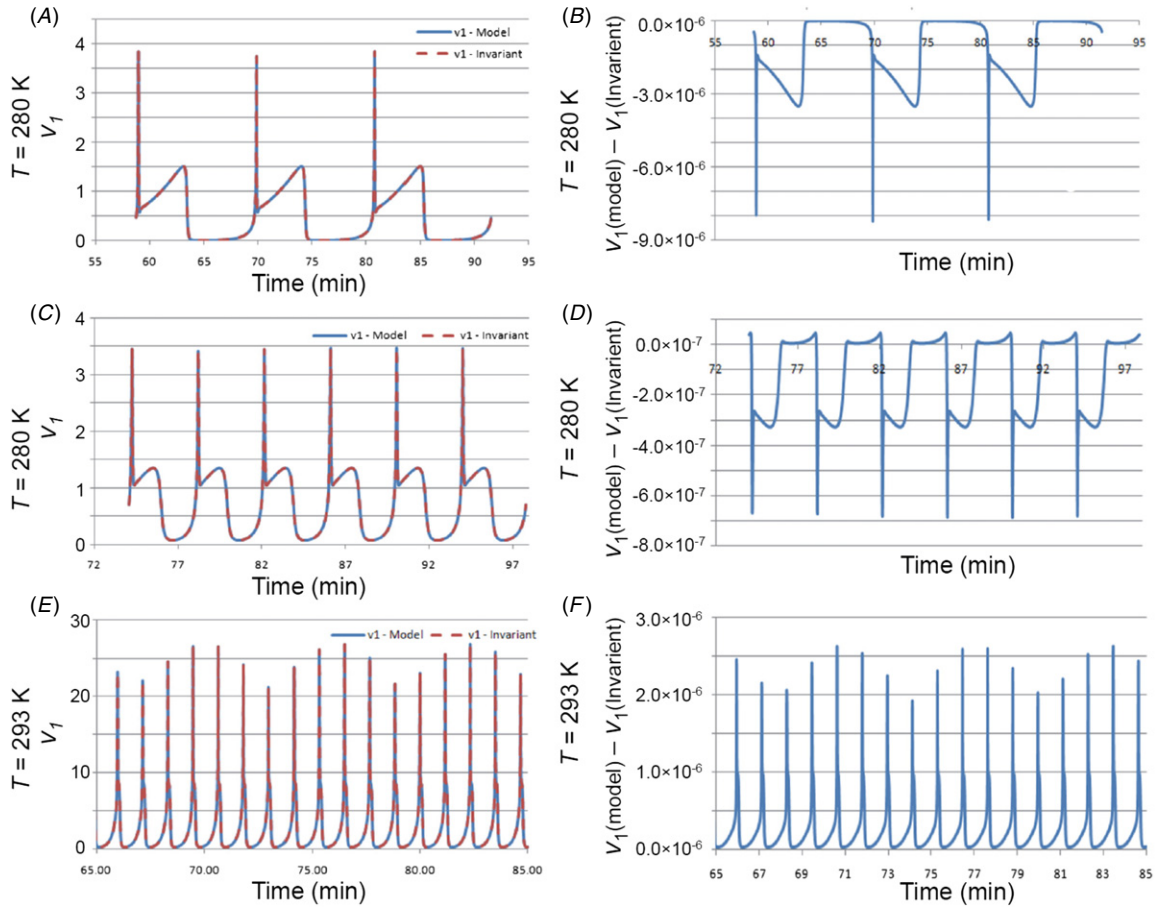


Figure 9. Comparison of the glucose reaction rate obtained from the invariant and the analytical model. v_1 is plotted versus time and compared with the v_1 obtained from the analytical model for each of the temperatures considered (panels *A*, *C* and *E*). Panels *B*, *D* and *F* show the difference between v_1 derived from the invariant and the v_1 derived from the model. Note that the amplitude of the difference is 10^{-5} to 10^{-6} smaller than that of either v_1.

Discussion

Our experiments tested the proposed method to modify a related or hypothesized mathematical model—suggesting nonlinear terms or modifications—to make it agree with data, or to propose complete dynamical models without a prior model. The modeling process searches the space of symbolic differential equations using symbolic regression and fitness prediction for building mathematical equations and an EEA for designing new *in silico* or even wet lab experiments to test and refine candidate models. While this method is general enough to be applied to many types of modeling problems, we focus on metabolic networks because they are known to be difficult to model with conventional methods due to their dimensionality, feedback loops, and rich dynamics, yet should be well suited for automated wet lab experimentation.

Our *in silico* experiments showed that seeding symbolic regression with a hypothesized model or closely related model can significantly improve the ability and speed to find the exact model for the unknown metabolic network. In contrast to other techniques for model reduction, this approach can also expand a nonlinear model to include features not present in the hypothesized or baseline model. Similarly, the approach can be used to identify a set of initial conditions that could be used to discriminate between two models, i.e. design experiments that can test hypotheses and correct subtle differences using experimental data to identify how a particular metabolic system differs from one described by an established model.

We also demonstrated that our approach could identify the complete set of differential equations without a prior model. All equations were identified reliably with up to 10% observation noise, and five of the seven were identified reliably with up to 50% noise. It is worth noting that this system also represents one of the largest and most realistic experiments where the set of ODEs was identified automatically from data. While the understanding of nonlinear interactions of glycolytic oscillations in yeast has required several years of analysis [73, 74], our results indicate that the proposed method could help accelerate this task.

We also compared the ability of our symbolic regression-based approach to predict future trajectories of the system with the nonlinear regression of two approximate glycolysis models, neural network regression, and the proposed method. While each algorithm modeled the training data well, the proposed method found models that extrapolated accurately to the test data set better than the ordinary regression and neural network methods. This suggests that the method could also be

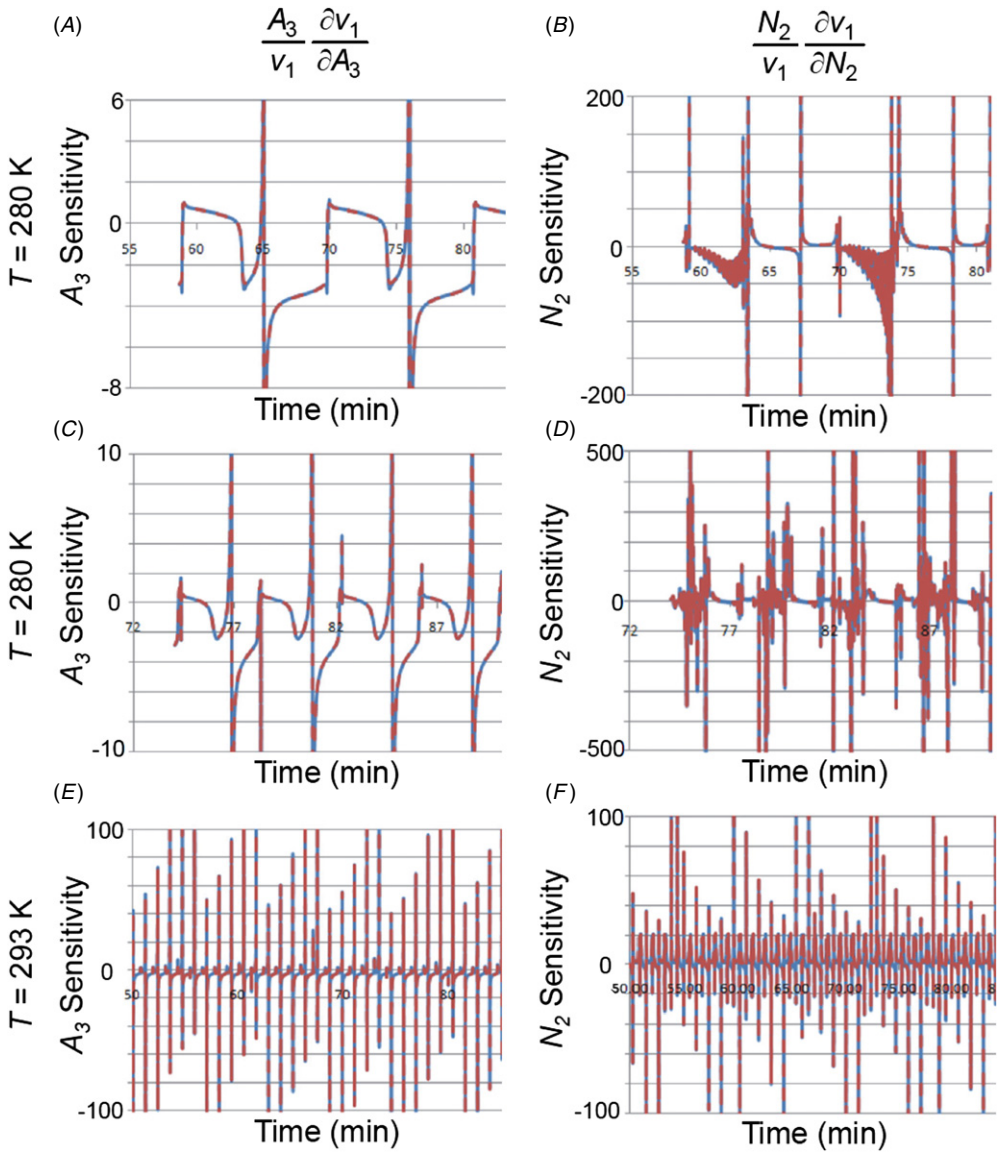


Figure 10. Invariant-determined sensitivity of v_1 with respect to A_3 and N_2. The invariant was used to compute the sensitivity of v_1 with respect to A_3 (equation at the top of the left column) and N_2 (equation at the top of right column) for 280 K (panels *A* and *B*); for 284 K (*C* and *D*); and for 293 K (*E* and *F*).

beneficial for numerical simulations in addition to analytical analysis.

Finally, we showed that the symbolic regression approach can be applied to find an invariant expression for the metabolic network. The invariant of the biochemical network identified by the algorithm can be considered to be a closed object embedded in the multidimensional phase space of the model variables. The projection of this surface onto the subspace of the observed variables accurately derived its kinetics and respective sensitivity coefficients. We have yet to identify formally the relationship between the invariants that we derive and invariant manifolds [56, 59–66]. Similarly, we have not yet determined whether the modeling process for the entire system could be accelerated by first searching for a single invariant equation.

We are currently developing the substantial experimental hardware and algorithms required to allow the EEA to conduct wet lab experiments on yeast and other biochemical systems [53, 82]. A fascinating possibility would be to add near-real-time measurements of the transcriptome into robot metabolic scientists by measuring the expression of selected genes through optical detection of green fluorescent protein (GFP) markers. This would be particularly important when studying systems that demonstrate rapid coupling between metabolic dynamics and gene expression [83, 84]. Expansion of the robot scientists' techniques to large numbers of genes would require, for example, chemostats capable of providing a steady supply of cells for mRNA analysis, and an mRNA identification technique that was both rapid and required only small numbers of cells. A major challenge will be to infer long-term metabolic network dynamics, where metabolic changes are driven by complex GRN and the transcriptional factors that they control, some of which are themselves metabolites, and neither the active network topology nor the network parameters are fixed

[85]. An issue would be whether in a wet lab environment the algorithm can infer a model faster than the system itself evolves, i.e., whether the biological system can be treated as quasistatic during model inference.

Acknowledgments

The authors acknowledge support from the Cornell IGERT Program in Nonlinear Systems, the National Science Foundation grant 0941561 (CDI), National Institutes of Health grants U01AI061223, R01-HL58241-11 and 1RC2DA028981-01, Defense Threat Reduction Agency grant HDTRA1-09-1-0013, the National Academies Keck Futures Initiative, the Vanderbilt Institute for Integrative Biosystems Research and Education, CFD Research Corporation and the Simons Center for Systems Biology at the Institute for Advanced Study. We thank Allison Price and Don Berry for their editorial and bibliographic assistance, and the Institute for Advanced Study for providing an environment conducive to the drafting of this manuscript.

References

[1] Barik D, Baumann W T, Paul M R, Novak B and Tyson J J 2010 A model of yeast cell-cycle regulation based on multisite phosphorylation *Mol. Syst. Biol.* **6** 405

[2] Chen K C, Calzone L, Csikasz-Nagy A, Cross F R, Novak B and Tyson J J 2004 Integrative analysis of cell cycle control in budding yeast *Mol. Biol. Cell* **15** 3841–62

[3] Kuttykrishnan S, Sabina J, Langton L L, Johnston M and Brent M R 2010 A quantitative model of glucose signaling in yeast reveals an incoherent feed forward loop leading to a specific, transient pulse of transcription *Proc. Natl Acad. Sci.* **107** 16743–8

[4] Tyson J J, Chen K C and Novak B 2003 Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell *Curr. Opin. Cell Biol.* **15** 221–31

[5] Voit E O 1991 *Canonical Nonlinear Modeling: S-System Approach to Understanding Complexity* (New York: Van Nostrand Reinhold)

[6] Savageau M A 2009 *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology.* Reprint of the Original Edition Published in 1976 (Reading, MA: Addison-Wesley)

[7] Soni A S, Jenkins J W and Sundaram S S 2008 Determination of critical network interactions: an augmented Boolean pseudo-dynamics approach *IET Syst. Biol.* **2** 55–63

[8] Margolin A A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, la Favera R and Califano A 2006 ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context *BMC Bioinform.* **7** Suppl. 1 S7

[9] Bel G, Munsky B and Nemenman I 2010 The simplicity of completion time distributions for common complex biochemical processes *Phys. Biol.* **7** 016003

[10] Khoo MCK 2000 *Physiological Control Systems: Analysis, Simulation, and Estimation* (New York: IEEE Press)

[11] Stolovitzky G and Califano A 2007 *Reverse Engineering Biological Networks: Opportunities and Challenges in Computational Methods for Pathway Inference* (Boston, MA: Blackwell)

[12] Chou I C and Voit E O 2009 Recent developments in parameter estimation and structure identification of biochemical and genomic systems *Math. Biosci.* **219** 57–83

[13] Bryant C H, Muggleton S H, Oliver S G, Kell D B, Reiser P and King R D 2001 Combining inductive logic programming, active learning and robotics to discover the function of genes *Electron. Artic. Comput. Inf. Sci.* **6** http://www.ep.liu.se/ea/cis/2001-012/

[14] Kell D B 2006 Metabolomics, modelling and machine learning in systems biology—towards an understanding of the languages of cells *FEBS J.* **273** 873–94 (Delivered on 3 July 2005 at the *30th FEBS Congress and 9th IUBMB Conf. (Budapest)*)

[15] King R D, Whelan K E, Jones F M, Reiser P G K, Bryant C H, Muggleton S H, Kell D B and Oliver S G 2004 Functional genomic hypothesis generation and experimentation by a robot scientist *Nature* **427** 247–52

[16] O'Hagan S, Dunn W B, Brown M, Knowles J D and Kell D B 2005 Closed-loop, multiobjective optimization of analytical instrumentation: gas chromatography/time-of-flight mass spectrometry of the metabolomes of human serum and of yeast fermentations *Anal. Chem.* **77** 290–303

[17] King R D *et al* 2009 The automation of science *Science* **324** 85–9

[18] Bongard J and Lipson H 2007 Automated reverse engineering of nonlinear dynamical systems *Proc. Natl Acad. Sci.* **104** 9943–8

[19] Bongard J C and Lipson H 2005 Nonlinear system identification using coevolution of models and tests *IEEE Trans. Evol. Comput.* **9** 361–84

[20] Zykov V, Bongard J and Lipson H 2005 Co-evolutionary variance can guide physical testing in evolutionary system identification *NASA/DoD Conf. on Evolvable Hardware* ed J D Lohn (Los Alamitos, CA: IEEE Computer Society Press) pp 213–20

[21] Styczynski M P and Stephanopoulos G 2005 Overview of computational methods for the inference of gene regulatory networks *Comput. Chem. Eng.* **29** 519–34

[22] Gardner T S, di Bernardo D, Lorenz D and Collins J J 2003 Inferring genetic networks and identifying compound mode of action via expression profiling *Science* **301** 102–5

[23] Levine A J, Hu W, Feng Z and Gil G 2007 Reconstructing signal transduction pathways: challenges and opportunities *Ann. NY Acad. Sci.* **1115** 32–50

[24] Nielsen J and Oliver S 2005 The next wave in metabolome analysis *Trends Biotechnol.* **23** 544–6

[25] Nemenman I, Escola G S, Hlavacek W S, Unkefer P J, Unkefer C J and Wall M E 2007 Reconstruction of metabolic networks from high-throughput metabolite profiling data: *in silico* analysis of red blood cell metabolism *Ann. NY Acad. Sci.* **1115** 102–15

[26] Kauffman K J, Pajerowski J D, Jamshidi N, Palsson B O and Edwards J S 2002 Description and analysis of metabolic connectivity and dynamics in the human red blood cell *Biophys. J.* **83** 646–62

[27] Ni T C and Savageau M A 1996 Model assessment and refinement using strategies from biochemical systems theory: application to metabolism in human red blood cells *J. Theor. Biol.* **179** 329–68

[28] Vallabhajosyula R R and Sauro H M 2006 Complexity reduction of biochemical networks *Proc. 2006 Winter Simulation Conf. (Monterey, CA, USA)* ed L F Perrone, F P Wieland, J Liu, B G Lawson, D M Nicol and R M Fujimoto (Piscataway, NJ: IEEE) pp 1690–7

[29] Bansal M, Belcastro V, Ambesi-Impiombato A and di Bernardo D 2007 How to infer gene networks from expression profiles *Mol. Syst. Biol.* **3** 78

[30] Koza J R 2001 Reverse engineering of metabolic pathways from observed data using genetic programming *Pacific Symp. on Biocomputing Proc.* (River Edge, NJ: World Scientific) pp 434–45

[31] Agusto D A and Barbosa H J C 2000 Symbolic regression via genetic programming *6th Brazilian Symp. on Neural Networks* ed C H C Ribeiro and F M G Franca (Los Alamitos, CA: IEEE Computer Society Press) pp 173–8

[32] Duffy J and Engle-Warnick J 2002 Using symbolic regression to infer strategies from experimental data *Evolutionary Computation in Economics and Finance* ed S H Chen (New York: Physica) pp 61–84

[33] Hoai N X, McKay R I, Essam D and Chau R 2002 Solving the symbolic regression problem with tree-adjunct grammar guided genetic programming: the comparative results *Proc. 2002 World Congress on Computational Intelligence, WCCI* (Piscataway, NJ: IEEE) pp 1326–31

[34] Koza J R 1992 *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (Cambridge, MA: MIT Press)

[35] Schmidt M and Lipson H 2007 Coevolving fitness models for accelerating evolution and reducing evaluations *Genetic Programming Theory and Practice IV* ed R Riolo, T Soule and B Worzel (New York, NY: Springer) pp 113–30

[36] Schmidt M and Lipson H 2007 Comparison of tree and graph encodings as function of problem complexity *Proc. 9th Annu. Conf. on Genetic and Evolutionary Computation* ed D Thierens (New York: ACM Press) pp 1674–9

[37] Schmidt M D and Lipson H 2008 Coevolution of fitness predictors *IEEE Trans. Evol. Comput.* **12** 736–49

[38] Fell D A 1992 Metabolic control analysis—a survey of its theoretical and experimental development *Biochem. J.* **286** 313–30

[39] Kell D B 2004 Metabolomics and systems biology: making sense of the soup *Curr. Opin. Microbiol.* **7** 296–307

[40] Vance W, Arkin A and Ross J 2002 Determination of causal connectivities of species in reaction networks *Proc. Natl Acad. Sci.* **99** 5816–21

[41] Ross J 2003 New approaches to the deduction of complex reaction mechanisms *Acc. Chem. Res.* **36** 839–47

[42] Mendes P 2001 Modeling large biological systems from functional genomic data: parameter estimation *Foundations of Systems Biology* ed H Kitano (Cambridge, MA: MIT Press) pp 163–86

[43] Mendes P and Kell D B 1996 On the analysis of the inverse problem of metabolic pathways using artificial neural networks *BioSystems* **38** 15–28

[44] Crampin E J, Schnell S and McSharry P E 2004 Mathematical and computational techniques to deduce complex biochemical reaction mechanisms *Progr. Biophys. Mol. Biol.* **86** 77–112

[45] Beard D A, Qian H and Bassingthwaighte J B 2004 Stoichiometric foundation of large-scale biochemical system analysis *Modelling in Molecular Biology* ed G Ciobanu and G Rozenberg (Berlin: Springer) pp 1–19

[46] Mendes P and Kell D B 1998 Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation *Bioinformatics* **14** 869–83

[47] Moles C G, Mendes P and Banga J R 2003 Parameter estimation in biochemical pathways: a comparison of global optimization methods *Genome Res.* **13** 2467–74

[48] Goel G, Chou I C and Voit E O 2008 System estimation from metabolic time-series data *Bioinformatics* **24** 2505–11

[49] Luo R Y, Liao S, Tao G Y, Li Y Y, Zeng S Q, Li Y X and Luo Q M 2006 Dynamic analysis of optimality in myocardial energy metabolism under normal and ischemic conditions *Mol. Syst. Biol.* **2** 2006.0031

[50] Cakir T, Patil K R, Onsan Z I, Ulgen K O, Kirdar B and Nielsen J 2006 Integration of metabolome data with metabolic networks reveals reporter reactions *Mol. Syst. Biol.* **2** 50

[51] Shlomi T, Eisenberg Y, Sharan R and Ruppin E 2007 A genome-scale computational study of the interplay between transcriptional regulation and metabolism *Mol. Syst. Biol.* **3** 101

[52] Schmidt H, Madsen M F, Dano S and Cedersund G 2008 Complexity reduction of biochemical rate expressions *Bioinformatics* **24** 848–54

[53] LeDuc P R, Messner W C and Wikswo J P 2011 How do control-based approaches enter into biology? *Annu. Rev. Biomed. Engr.* Review in Advance doi:10.1146/annurev-bioeng-071910–124651

[54] Westerhoff H V and Palsson B O 2004 The evolution of molecular biology into systems biology *Nat. Biotechnol.* **22** 1249–52

[55] Goldbeter A 2002 Computational approaches to cellular rhythms *Nature* **420** 238–45

[56] Guckenheimer J and Holmes P 1997 *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (New York: Springer)

[57] Chandra F A, Buzi G and Doyle J C 2009 Linear control analysis of the autocatalytic glycolysis system *American Control Conf., 2009* (St Louis, MO: IEEE) pp 319–24

[58] Hirsch M W, Smale S and Devaney R L 2004 *Differential Equations, Dynamical Systems, and an Introduction to Chaos* (San Diego, CA: Academic Press)

[59] Reijenga K A, Westerhoff H V, Kholodenko B N and Snoep J L 2002 Control analysis for autonomously oscillating biochemical networks *Biophys. J.* **82** 99–108

[60] Manu S S, Spirov A V, Gursky V V, Janssens H, Kim A R, Radulescu O, Vanario-Alonso C E, Sharp D H, Samsonova M and Reinitz J 2009 Canalization of gene expression and domain shifts in the drosophila blastoderm by dynamical attractors *PLoS Comput. Biol.* **5** e1000303

[61] Guckenheimer J and Vladimirsky A 2004 A fast method for approximating invariant manifolds *SIAM J. Appl. Dyn. Syst.* **3** 232–60

[62] Gorban A N, Karlin I V and Zinovyev A Y 2004 Constructive methods of invariant manifolds for kinetic problems *Phys. Rep.* **396** 197–403

[63] Roussel M R and Fraser S J 2001 Invariant manifold methods for metabolic model reduction *Chaos: Interdiscip. J. Nonlinear Sci.* **11** 196–206

[64] Chiavazzo E, Gorban A N and Karlin I V 2007 Comparison of invariant manifolds for model reduction in chemical kinetics *Commun. Comput. Phys.* **2** 964–92

[65] Gorban A N and Karlin I V 2005 *Invariant Manifolds for Physical and Chemical Kinetics* (Berlin: Springer)

[66] Gorban A N and Karlin I V 2003 Method of invariant manifold for chemical kinetics *Chem. Eng. Sci.* **58** 4751–68

[67] Schmidt M and Lipson H 2009 Distilling free-form natural laws from experimental data *Science* **324** 81–5

[68] Schmidt M D and Lipson H 2010 Eureqa: search your data for hidden mathematical relationships http://ccsl.mae.cornell.edu/outreach

[69] Schmidt M D and Lipson H 2006 Actively probing and modeling users in interactive coevolution *GECCO 2006 Genetic and Evolutionary Computation Conf. (Seattle, Washington, USA, 8–12 July 2006)* ed M Keijzer (New York: Association for Computing Machinery) pp 385–6

[70] Richard P, Teusink B, Hemker M B, VanDam K and Westerhoff H V 1996 Sustained oscillations in free-energy state and hexose phosphates in yeast *Yeast* **12** 731–40

[71] Cleveland W S and Devlin S J 1988 Locally weighted regression—an approach to regression analysis by local fitting *J. Am. Stat. Assoc.* **83** 596–610

[72] Schmidt M D and Lipson H 2007 Learning noise *Genetic and Evolutionary Computation Conf.* (New York: ACM) pp 1680–5

[73] Wolf J and Heinrich R 2000 Effect of cellular interaction on glycolytic oscillations in yeast: a theoretical investigation *Biochem. J.* **345** 321–34

[74] Ruoff P, Christensen M K, Wolf J and Heinrich R 2003 Temperature dependency and temperature compensation in a model of yeast glycolytic oscillations *Biophys. Chem.* **106** 179–92

[75] Heinrich R, Rapoport S M and Rapoport T A 1977 Metabolic-regulation and mathematical-models *Prog. Biophys. Mol. Biol.* **32** 1–82

[76] Varma A and Palsson B O 1994 Metabolic flux balancing: basic concepts, scientific and practical use *Nat. Biotechnol.* **12** 994–8

[77] Schmidt M and Lipson H 2009 Symbolic regression of implicit equations *Genetic Programming Theory and Practice VII* ed R Riolo, U-M O'Reilly and T McConaghy (New York: Springer) pp 73–85

[78] McMillen D, Kopell N, Hasty J and Collins J J 2002 Synchronizing genetic relaxation oscillators by intercell signaling *Proc. Natl Acad. Sci.* **99** 679–84

[79] Ramakrishnan N and Bhalla U S 2008 Memory switches in chemical reaction space *PLoS Comput. Biol.* **4** e1000122

[80] Ramakrishnan N, Bhalla U S and Tyson J J 2009 Computing with proteins *Computer* **42** 47–56

[81] Na M G, Hwang I J and Lee Y J 2006 Inferential sensing and monitoring for feedwater flow rate in pressurized water reactors *IEEE Trans. Nucl. Sci.* **53** 2335–42

[82] Enders J R, Marasco C C, Kole A, Nguyen B, Sundarapandian S, Seale K T, Wikswo J P and Mclean J A 2010 Towards monitoring real-time cellular response using an integrated microfluidics-MALDI/NESI-ion mobility-mass spectrometry platform *IET Syst. Biol.* **4** 416–27

[83] Stern S, Dror T, Stolovicki E, Brenner N and Braun E 2007 Genome-wide transcriptional plasticity underlies cellular adaptation to novel challenge *Mol. Syst. Biol.* **3** 106

[84] Kresnowati M T A P, van Winden W A, Almering M J H, ten Pierick A, Ras C, Knijnenburg T A, ran-Lapujade P, Pronk J T, Heijnen J J and Daran J M 2006 When transcriptome meets metabolome: fast cellular responses of yeast to sudden relief of glucose limitation *Mol. Syst. Biol.* **2** 49

[85] Patil K R and Nielsen J 2005 Uncovering transcriptional regulation of metabolism by using metabolic network topology *Proc. Natl Acad. Sci.* **102** 2685–9