



AFRL-RI-RS-TR-2022-078

LEARNING DEEP KOOPMAN OPERATORS FOR DATA-DRIVEN MODEL DISCOVERY OF SYNTHETIC BIOLOGICAL SYSTEMS

UNIVERSITY OF CALIFORNIA

JUNE 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2022-078 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

STEVEN L. DRAGER
Work Unit Manager

/ S /

GREGORY J. HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
JUNE 2022		FINAL TECHNICAL REPORT		START DATE SEPTEMBER 2019	END DATE SEPTEMBER 2021
4. TITLE AND SUBTITLE LEARNING DEEP KOOPMAN OPERATORS FOR DATA-DRIVEN MODEL DISCOVERY OF SYNTHETIC BIOLOGICAL SYSTEMS					
5a. CONTRACT NUMBER FA8750-19-2-0502		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61101E	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER R2VX	
6. AUTHOR(S) Enoch Yeung					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The Regents of the University of California on behalf of its Santa Barbara campus UCSB Office of Research 3227 Cheadle Hall Santa Barbara CA 93106-2050				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RITA DARPA 525 Brooks Road 675 North Randolph Street Rome NY 13441-4505 Arlington VA 22203-2114			10. SPONSOR/MONITOR'S ACRONYM(S) RI		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RI-RS-TR-2022-078
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This effort developed a deep learning framework to discover Koopman network models that map causal dependencies between all measured biocircuit outputs, experimental input parameters, and contextual design parameters. The research discovered dynamic network models of how measured quantities (e.g. multiple fluorescent reporters) and monitored experimental parameters (e.g. OD, temperature, inducer concentration, age of media) are causally related, thus generalizing the concept of dynamical structure functions (and transfer functions) for arbitrary nonlinear systems. These models were used to identify categories and relationships of biological parts or biocircuits with analogous behavior, infer the state of latent variables to generate hypotheses for experimental surprise, predict and evaluate operating envelopes for stable system behavior, and quantitatively predict biocircuit dynamic response as a function of experimental parameters. The methods in this project combined the expressive power, scalability of deep learning algorithms and rigor of Koopman operator theory to discover data-driven dynamical system models for hypothesis generation and biocircuit characterization.					
15. SUBJECT TERMS Koopman operators, data science, data-driven discovery, synthetic biology, systems biology, computational biology, nonlinear systems, dynamical systems, machine learning, deep learning, genomics, transcriptomics, proteomics, biosensing, biosensors, automation					
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR		18. NUMBER OF PAGES 56
19a. NAME OF RESPONSIBLE PERSON STEVEN L. DRAGER					19b. PHONE NUMBER (Include area code) N/A

Table of Contents

List of Figures	ii
List of Tables	iii
1. Summary	1
2. Introduction.....	2
3. Methods, Assumptions, & Procedures.....	4
4. Results & Discussion	5
4.1 Deep dynamic mode decomposition.....	5
4.1.1 Introduction & SD2 program context	5
4.1.2 The Koopman Operator	6
4.1.3 The Deep Dynamic Mode Decomposition Algorithm.....	7
4.1.4 The Emergence of Koopman Basis Functions During Training.....	8
4.1.5 Example: The Glycolysis Network.....	10
4.1.6 Example: Operational Envelope of the Toggle Switch.....	12
4.2 Structured dynamic mode decomposition for analysis of additive, reductionist SD2 strain libraries	14
4.2.1 Summary of Results.....	14
4.2.2 Introduction & SD2 Context.....	14
4.2.3 Structured dynamic mode decomposition.....	16
4.2.4 Quantifying impact of the SD2 NAND gate on host	17
4.3 Data-driven circuit specification for SD2 obstacle course challenges.....	20
4.3.1 Introduction & SD2 Context.....	20
4.3.2 Results.....	22
4.4 The Sensor Scraper for Organophosphate Detection.....	26
4.4.1 Induction of malathion elicits fast host response.....	28
4.4.2 Dynamic mode decomposition uncovers modes of host cell response.....	29
4.4.3 Observability analysis of gene expression.....	32
4.5 The Landing Site Recommender for Genomic Engineering of New Organisms	36
5. Conclusions.....	41
6. References.....	43
List of Symbols, Abbreviations, and Acronyms	50

List of Figures

Figure 1: The Koopman basis functions generated by a 20 layer drop-out ELU neural network.	10
Figure 2: Multi-step prediction using a single time-point on the glycolytic oscillator.....	11
Figure 3: Model predictions and test data of the genetic toggle switch for varying concentrations of ATC and IPTG.....	13
Figure 4: (Left) The performance specification used to evaluate the functionality of the toggle switch at each three-tuple of temperature, IPTG, and ATC in input-condition space. (Right) The operational envelope of the genetic toggle switch is visualized as the aggregate polygonal shape formed by green dots generated using deepDMD model predictions.....	14
Figure 5: Schematic of the bottom-up design of a NAND gate in E. coli.	17
Figure 6: The Koopman operator estimated from structured DMD of the host dynamics in response to arabinose and IPTG induction.	19
Figure 7: The input Koopman operator estimated from structured DMD modeling circuit-to-host interaction of the NAND circuit.	20
Figure 8: An overview of how the KCOT algorithm works.....	23
Figure 9: The reference curves for a two node hysteretic switch designed using the KCOT algorithm.	24
Figure 10: (Top Left) Circuit network topology recommended from first iteration with the Koopman Operator circuit optimization tool (KCOT), featuring mutual positive feedback. (Bottom Left) Circuit network topology recommended from comparative analysis of the original Koopman operator and the realized Koopman operator; a positive activation edge is replaced with a negative repression edge. (Top Right) The reference Koopman operator generated from reference curves (left) and the final realized Koopman operator obtained from genetic circuit engineering and circuit characterization data (right). (Bottom Right) The experimental data (right subpanel) shows overall consistency of the two node switch to varying levels of IPTG , when compared with the original reference curves used to train the reference Koopman operator (left subpanel).	25
Figure 11: Transcriptional genetic sensors underlying the response of transcriptome perturbations can be extracted using data-driven observer design.	28
Figure 12: Dynamic mode decomposition provides an interpretable and predictive model of gene expression dynamics.	32
Figure 13: Gene sampling weights which maximize observability provide a machine learned ranking for extraction of genetic sensing elements.....	34
Figure 14: Biophysically informed landing site recommendation tool.	37
Figure 15: Time-series performance, reporter activity score and growth defects for top 5 and bottom 5 performing strains.....	39
Figure 16: The genomic perturbation is calculated by calculating the 2-norm of the difference between the strain's transcriptomic profile and the wild-type across the entire genome.	40

List of Tables

Table 1: Prediction Accuracy of EDMD vs DeepDMD Koopman Operators on Partially Observed Linear Systems (POLS) and the Glycolytic Pathway 9

Table 2: Sensor gene metadata and sensor transfer curve parameters..... 35

1. Summary

We developed a deep learning framework to discover Koopman network models that map causal dependencies between all measured biocircuit outputs, experimental input parameters, and contextual design parameters. We discovered dynamic network models of how measured quantities (e.g. multiple fluorescent reporters) and monitored experimental parameters (e.g. optical density (OD), temperature, inducer concentration, age of media) are causally related, thus generalizing the concept of dynamical structure functions (and transfer functions) for arbitrary nonlinear systems. These models were used to identify categories and relationships of biological parts or biocircuits with analogous behavior, infer the state of latent variables to generate hypotheses for experimental surprise, predict and evaluate operating envelopes for stable system behavior, and quantitatively predict biocircuit dynamic response as a function of experimental parameters. The methods in our project combined the expressive power, scalability of deep learning algorithms and rigor of Koopman operator theory to discover data-driven dynamical system models for hypothesis generation and biocircuit characterization.

State-of-the-art model discovery methods usually leverage a priori information about model structure. For example, Bayesian hierarchical model identification methods infer model class and parameters based on a priori knowledge of model priors. Compressive sensing algorithms identify input-output and dynamic models based on a pre-defined dictionary of basis functions. State of the art methods for learning Koopman operators rely on dynamic mode decomposition (DMD), which utilize linear models to approximate the infinite-dimensional Koopman operator. Thus, these methods rely on the creativity of a scientist to provide canonical a priori models. This limits their applicability in domains where canonical models are lacking, e.g. synthetic biology, neuroscience, human-machine systems, or social systems.

In the context of synthetic biocircuit design, data-driven discovery of scientific models is hindered by three major technical challenges: 1) learning quantitative relationships between biocircuit variables in the presence of unmodeled dynamics, 2) understanding how those relationships vary as a function of biocircuit context, and 3) converting relationships between design variables and context in biocircuit models to predict stable operational envelopes for biocircuits. Because of these challenges, scientific discovery typically relies on manual or semi-automated collection of data followed by human interpretation of data. Models are viewed as a means for confirming human-generated hypotheses rather than as a means to discover novel scientific hypotheses. Again, this is because models are constructed around first-principles provided by the scientist, rather than from data-driven algorithms.

We developed a data-driven approach to learning network models at a resolution commensurate with the spatial and temporal scale available in data. Instead of trying to fully populate all kinetic parameters for a detailed first-principles model, we developed network models describing only the causal dependencies between measured or known experimental variables. Utilizing this computational framework, the primary deliverables of our project are outlined in the sections to follow.

2. Introduction

Four years ago, deep learning was a rapidly advancing field progressing in parallel to advances in the field of synthetic biology. Synthetic biological design was primarily a manual process — experienced biologists with encyclopedic memory of various biological parts would draw from their collective experience in brainstorming sessions to design genetic circuits. These genetic circuits were often mocked up on a whiteboard, drawn as a graph of interacting nodes and edges, to mirror the logical architecture of silicon circuit counterparts. When the dust cleared from the brainstorming process, sequences representing promoters, ribosome binding sites, coding sequences, and various other specialized phrases would be hand-picked based on positive prior experiences. So, for the most part, synthetic biological circuit design in 2016 was a manual, human-intensive process. There were two notable exceptions to this generalization: Cellular Logic (CELLO) and the Ribosome Binding Site (RBS) calculator.

In April of 2016, a few months before the Synergistic Discovery & Design (SD2) program proposer's day, Alec Nielsen et al. released a paper entitled "Genetic Circuit Design Automation". Alec was a Ph.D. student in the Voigt lab at Massachusetts Institute of Technology. The paper described a new software tool called Cello, an abbreviation of Cellular Logic, which enabled Cello users to write Verilog code and specify their set of parts via a user constraint file (UCF). The specification of the circuit would then result in automatic compilation of biological parts, down to the level of sequence specification, gene orientation, and circuit layout on deoxyribonucleic acid (DNA). Remarkably, CELLO demonstrated a 92% success rate across 60 circuits tested in the original paper.

The second notable exception to manual, synthetic circuit design was the ribosome binding calculator, developed by the Salis Lab at Pennsylvania State University. The ribosome binding calculator utilized the theory of statistical thermodynamics to estimate the binding affinity of a ribosomal protein to the Shine Delgarno sequence of a given ribosome binding site. The RBS calculator was published in 2009, 7 years prior to the inception of SD2, in collaboration with Chris Voigt. The RBS calculator featured a web-based interface, which enabled distributed biological design. To date, over 8,000 synthetic biologists have engineered more than 500,000 genetic systems using parameters aided by the RBS calculator.

With this context, in the spring of 2016, I recall taking a phone call with Dr. Jennifer Roberts, the first program manager and founder of the Defense Advanced Research Projects Agency (DARPA) SD2 program (with great credit to Drs. Elliott and Buckout-White for their work leading this program to the finish line). She asked if I could convey the contributions of machine learning to synthetic biology. In short, I explained that the two fields were utterly independent and unaware of their potential synergy. In my humble view, this was the state of the field and would be the subtext for the next 4 years of collaborative interaction, animated discussion, debate, and intense effort to automate synthetic biological design and analysis for foundry lab work.

One last remark: modeling and analysis has always played a subservient role in biology. All too often, biologists would run their experiments independent of model identification requirements, only to ask that some modeling be added on at the last minute, with short-fuse deadlines to fit the data. But no more. Models rarely drove experimental research; they simply were used to confirm

the already-observed outcomes. To postulate a data-driven design process for synthetic biology, rather than human-driven design, would be a grand leap forward.

Even now, experiments are often conducted independent of modeling, because data curation, engineering, and analysis would be a monumental task in and of itself. The brains that can conduct machine learning analysis are often physically insulated from the pipetting hands that breathe life into new experiments.

Our goal was to merge operator-theoretic methods with deep learning to automate synthetic biological circuit analysis and design. Our key insight was that deep neural networks acted as extraordinarily efficient representations of arbitrary function spaces, which could be used for describing the flow of dynamical systems. At the time, our preliminary results from Koopman operator theory, functional analysis, and dynamical systems suggested that many physical systems, including biological networks, could be described as a time-evolution of nonlinear functions on a defined manifold of physical variables. This report details how that foundational technology gave birth to multiple tools to enable automated data-driven discovery and design in the field of synthetic biology. It is with great gratitude to DARPA and her affiliate organizations that we present our major technical advancements in the DARPA Synergistic Discovery & Design program.

3. Methods, Assumptions, & Procedures

We briefly introduce Koopman operator theory (see [50] for a full discussion); as we will use it throughout this report. Consider a discrete time open-loop nonlinear system of the form:

$$x_{t+1} = f(x_t) \quad (1)$$

where $f: M \subset \mathbb{R}^n \rightarrow M$ is an analytic vector field on the state space. The Koopman operator of (1), $\mathcal{K}: \mathcal{F} \rightarrow \mathcal{F}$, is a linear operator that acts on observable functions $\psi(x_k)$ and propagates them forward in time as:

$$\psi(x_{t+1}) = \mathcal{K}\psi(x_t) \quad (2)$$

Here $\mathcal{F}$ is the space of observable functions that is invariant under the action of $\mathcal{K}$.

Using data-driven approaches, commonly DMD [69] or extended DMD [88], an approximation to the Koopman operator, K , can be computed. The approach taken to compute an approximation to the Koopman operator in both DMD and extended DMD is to solve the following optimization problem:

$$\min_K \left\| \Psi(X_f) - K\Psi(X_p) \right\| \quad (3)$$

where $X_f \equiv [x_1 \ \dots \ x_{N-1}]$, $X_p \equiv [x_2 \ \dots \ x_N]$ are snapshot matrices formed from the discrete-time dynamical system (1) and $\Psi(X) \equiv [\psi_1(x) \ \dots \ \psi_R(x)]$ is the mapping from physical space into the space of observables. DMD is a special case of extended DMD where $\psi(x) = x$. It was shown by Rowley et al. that the approximate Koopman operator obtained from DMD is closely related to a spectral analysis of the linear but infinite-dimensional Koopman operator [65].

4. Results & Discussion

The major pillars of innovation in this project center on new methods for data-driven discovery and design. Each new method described in this section was developed to address a collaborative research problem in the DARPA Synergistic Discovery and Design program. Each method utilizes our computational framework for Koopman operator theory or a variation on the core concept to extract insight directly from data. Below we briefly recount the SD2 program context that motivated the innovation described and the steps towards innovation and a new technical product.

4.1 Deep dynamic mode decomposition

4.1.1 Introduction & SD2 program context

Biological circuits on the SD2 program exhibited nonlinear dynamics. For example, multi-stable genetic circuits arose as design outcomes of the Duke University team's Database of Signatures of Gene Regulatory Network tool (dsGRN). This tool predicts genetic network topologies that give rise to parametrically robust hysteresis. Logic circuits capable of on and off response levels naturally are better described with nonlinear transfer curves, specifically curves parameterized by nonlinear Hill functions. The challenge is to discover predictive models that capture these nonlinear relationships.

State of the art methods for modeling synthetic biological circuits relies on ground-up assembly of models. Based on their subject matter expertise, researchers intuit and write down a list of all the relevant chemical reactions and construct a deterministic ordinary differential equations (ODE) or stochastic chemical kinetics model for simulating the network's dynamics. The challenge with these approaches is the lack of knowledge about intrinsic parameters; parameters are often substituted with nominal working values and varied over several orders of magnitude. This sort of approach leads to classical over-fitting issues; a model that fits data does not necessarily explain the data!

Unfortunately, biological data usually falls into one of two categories: 1) data that is comprehensive in spatial resolution but limited in temporal resolution, e.g. steady-state 'omics assays or 2) data with high temporal resolution but with poor spatial resolution, e.g. time-lapse assays of a few fluorescent proteins. Further, in biological systems, model identifiability is confounded by model uncertainty, namely unmodeled dynamics and latent variables influencing the biocircuit of interest. These two issues make it very difficult to learn models of full-scale chemical reaction kinetics, detailing kinetics of every reaction at each point in time.

An alternative approach is to construct causal network models that are scaled back in complexity, to match the resolution of the data. This guarantees that the model parameters trained on the data will not be under-constrained, which significantly reduces the risk of overfitting. Specifically, we have shown it is possible to construct a condensed-graph model, with edges that vary as a function of time, latent variables and parameters. In this way, the network structure of the system is dynamic, allowing us to model the action of latent variables on causal relationships between measured variables, but without fully specifying the internal rate parameters of all associated reactions. These models have proven successful at identifying causal relationships between design

variables, as well as hidden relationships caused by biological crosstalk or context effects. The drawback of these methods is that biological systems are nonlinear; these methods use linearization techniques to obtain a local estimate of the causal network model.

4.1.2 The Koopman Operator

In 1931, B. O. Koopman published a paper showing that the evolution of any set of observables on a dynamical system can be expressed through the action of an infinite dimensional linear operator, the Koopman operator [41]. Because the Koopman operator is a canonical representation of any autonomous dynamical system, in principle, its use can bring to bear linear analysis methods on nonlinear systems. The Koopman operator is especially powerful for inferring properties of dynamical systems that are either partially or completely unknown or that are too complex to express using standard methods in analysis. Examples of such systems include biological networks, extremely large physical systems (which are intractable to analyze as white-box models), social networks, cyber-physical communication networks, and distributed computing systems that are subject to varying degrees of uncertainty. For this reason, the Koopman operator has gained attention as an effective tool for data-driven model discovery. The Koopman operator provides a data-driven model for comparing the asymptotic behavior of dynamical systems [50], specifically as a function of its spectra [50, 66].

Various algorithms have extended these methods using dynamic and extended dynamic mode decomposition, both for autonomous and controlled systems [88, 15, 59]. The prevailing method for learning the Koopman operator from data is dynamic mode decomposition [66]. Dynamic mode decomposition is the process of identifying a linear operator from temporally or spatially-linked data, ultimately with the objective of characterizing the spectrum of the operator. There are many variants of dynamic mode decomposition, but the most recent advances in Koopman operator learning have emerged from *extended dynamic mode decomposition* [88]. In extended dynamic mode decomposition, the idea is to lift the set of system observables from its native vector space into a higher dimensional space, usually through a nonlinear transformation [50, 66, 88]. When studying nonlinear systems, the hope is to lift the observables onto a nonlinear manifold, where their trajectories then evolve according to a linear law (the Koopman operator). The set of nonlinear transformations is often referred to as a dictionary, while the choice of what dictionary functions to include is left to the discretion of the scientist or based on *a priori* knowledge of the physical system of interest.

Thus, extended dynamic mode decomposition allows for the study of complex nonlinear phenomena e.g., the spectrum of nonlinear flows, but it is currently limited by a scientist's knowledge or intuition. This inherently makes the learning process manual, since a dictionary must be intuited, implemented, and then evaluated, followed by careful review of the results by the scientist. If the dictionary is not sufficiently complex, then the scientist must engineer prospective functions into the dictionary (see Figure 1).

4.1.3 The Deep Dynamic Mode Decomposition Algorithm

In deep dynamic mode decomposition, we define the dictionary as the output of a deep neural network. The deep neural network can be specified to have any architecture upstream, so long as it takes y_k as an input. The advantage of employing deep neural networks is that they have rich expressivity [61, 9], can be trained automatically, and scale to extremely large networks [10]. With each additional layer, the number of new nonlinear functions grows combinatorially, but only with a linear increase in the model parameters. We setup the deep Koopman operator learning problem as follows. Define the deep dictionary as

$$N_\Psi(x_n)$$

where $N_\Psi(x_n)$ is a neural network of choice. The Koopman operator problem is cast similarly as with extended dynamic mode decomposition:

$$\min_{K, \theta} ||N_\Psi(x_{n+1}, \theta) - KN_\Psi(x_n, \theta)|| + \lambda_1 ||K|| + \lambda_2 ||(\theta)||$$

where

$$\theta = (W_1, \dots, W_d, b_1, \dots, b_d, \dots)$$

are the numerical parameters of the neural network. For example, in the case of a multi-layer deep feedforward residual network,

$$N_\Psi(x_n) = h_d \circ h_{d-1} \circ \dots \circ h_1(x_n)$$

with $h_j(h_{j-1}) = \sigma_j(W_{j-1}h_{j-1} + b_{j-1})$, and $W_j \in \mathbb{R}^{r_j \times r_{j-1}}$. The parameters d, r_j are hyperparameters that are optimized using a combination of dropout and regularization, while the activation function σ_j can be chosen to be one of the many activation functions that result in fast convergence, e.g. the exponential linear unit (ELU), complementary rectified linear unit (cRELU), or the rectified linear unit (RELU).

Given this formulation, we remark it is also possible to initialize a neural network, akin to selecting a randomly parameterized dictionary, and directly solve the dynamic mode decomposition problem. Computing Y_f and Y_p as before, yields

$$Y_f = \left[\begin{array}{c|c|c} N_\Psi(x_{n+1}^{(0)}) & \dots & N_\Psi(x_1^{(0)}) \\ \vdots & \ddots & \vdots \\ N_\Psi(x_{n+1}^{(p)}) & \dots & N_\Psi(x_1^{(p)}) \end{array} \right], \quad Y_p = \left[\begin{array}{c|c|c} N_\Psi(x_n^{(0)}) & \dots & N_\Psi(x_0^{(0)}) \\ \vdots & \ddots & \vdots \\ N_\Psi(x_n^{(p)}) & \dots & N_\Psi(x_0^{(p)}) \end{array} \right].$$

However, we observed in practice that this approach did not produce accurate estimates of the Koopman operator. The key difference between deep Koopman operator learning and dynamic mode decomposition is that in deep Koopman operator learning, the dictionary and the Koopman operator are learned simultaneously. This increases the number of degrees of freedom in the problem, but no more than was initially present. In reality, the dictionary functions *and* the Koopman operator are unknown. Dynamic mode decomposition simplifies the problem by having

the user impose a particular choice of dictionary, followed by direct optimization of the Koopman operator assuming a fixed dictionary. However, both (sets of) variables are usually unknown in data-driven applications.

Finally, we remark that once the Koopman operator is obtained, as with dynamic mode decomposition, an eigendecomposition of K will yield the eigenmodes for the neural network basis functions. The weighted combination of those basis functions will define the Koopman eigenfunctions, with corresponding Koopman eigenvalues in the diagonalization or Jordan form of K .

Which neural network is used ultimately impacts the nature of the dictionary embedding the observables. If the neural network has memory, e.g., a long-short term memory module (LSTM) network or more generally, a generic recurrent neural network, then the dictionary functions can be viewed as time or space varying. Their properties change according to some memory state, which is dynamic from one time point x_n to the next x_{n+1} . If the network has hybrid or bifurcative behavior, e.g. a switching network, then the dictionary functions may have the capacity to model multiple invariant subspaces of the Koopman operator simultaneously. The challenge is knowing when the dictionary functions are trained accurately to recover the true Koopman operator. In reality, unless the system can be Carleman linearized [42], it is difficult to ascertain the complete dictionary. In the next section, we review a collection of several experiments performed to 1) gain insight into how deep Koopman operators learn basis functions and 2) show the ability of deep Koopman operators to learn completely unknown governing laws for a biological network and forecast multiple steps into the future

4.1.4 The Emergence of Koopman Basis Functions During Training

As a first experiment, we trained deep and classical Koopman operators on randomly generated linear systems with partially observed dynamics. When linear systems have partial state output, their Koopman operator is a coarse grained model of the underlying latent dynamics. For discrete time linear systems, a closed form expression for the Koopman operator exists, called the dynamical structure function [33]. The transfer function entries within a dynamical structure function act as linear fractional transformations on the input signals. Estimating the Koopman operator in this scenario requires discovering the basis functions (in the time-domain) for those transfer function entries.

We first performed experiments on a range of small networks, ranging from 4-128 nodes. A subset of the nodes were allocated as “latent” during the experiment. The systems were simulated in Python and time-series data was collected and divided into training and test data. For optimization, we used the AdaGrad algorithm, as implemented in Tensorflow. We tried randomly shuffling and varying batch size; we found that shuffling did not have a significant effect on the performance of the deep Koopman operator. As expected, if batch size became too small, e.g. less than 10 samples per batch, the algorithm would not converge. All training and validation was done using Python Tensorflow 1.0, on an NVIDIA TitanX Pascal or P40 system.

The results of the study are presented in Table 1. The deep Koopman operator had on average 1% one-step training error and one-step test error after 10,000 iterations (see Table 1). Further iterations reduced this error down to 0.1%. In contrast, the extended dynamic mode decomposition (EDMD) Koopman operator averaged over 10% on all network sizes. It was also difficult to

execute regularized extended dynamic mode decomposition on larger dynamical systems. The primary challenge is that the convex optimization routines run out of memory. We tested the algorithm both on a NVIDIA TitanX Pascal graphical processing unit (GPU), as well as an NVIDIA P40 GPU. For moderate sized networks with more than 10 variables, the quadratic and cubic terms induce rapid expansion in the size of the Koopman operator, which ultimately slowed the convex python package (CVXPY) solvers.

Table 1: Prediction Accuracy of EDMD vs DeepDMD Koopman Operators on Partially Observed Linear Systems (POLS) and the Glycolytic Pathway

Network Size				
(POLS)	EDMD Train. Err.	EDMD Test Err.	DeepDMD Train Err.	DeepDMD Test Err.
4	14.5%	14.5%	< 1%	<1%
8	15.10%	15.10%	<1%	<1%
16	14.3%	15.2%	<1%	<1%
32	11.0%	11.8%	<1%	<1%
64	14.5%	16.4%	<1%	<1%
128	12.5%	12.5 %	<1%	<1%
Species Measured	EDMD Train. Err	EDMD Test Err.	DeepDMD Train Err.	DeepDMD Test Err.
7	2.67%	2.5%	0.5%	0.5%
5	4.95	5.2% %	.47%	.66%
3	4.67%	5.05%	.76%	.84%

On the other hand, we were able to compute the deep Koopman operator for increasingly higher dimensional systems. For small systems, a network width of 20 nodes and a depth of 3-5 layers was sufficient. With networks ranging from 10-100 states, we found the deep Koopman operator could be trained if the neural network was sufficiently deep. This may be because the deeper the neural network, the more efficiently it encodes complex nonlinear functions. Once again, the one-step training error and test error were approximately 1%.

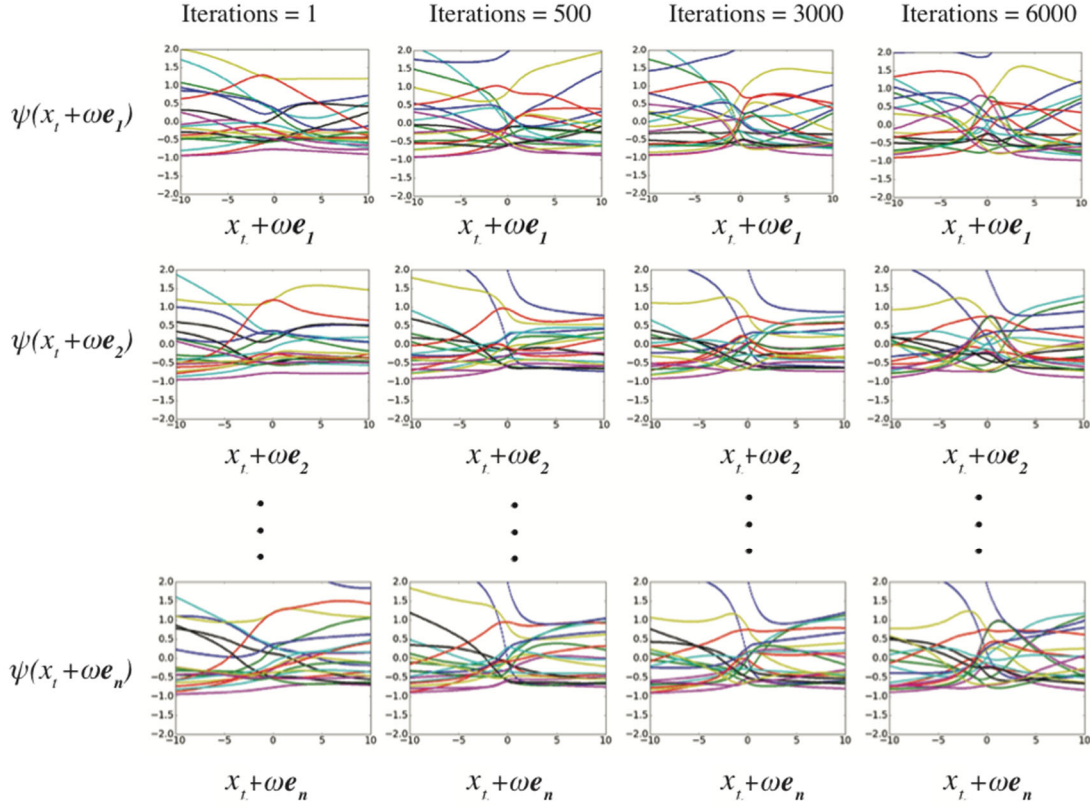


Figure 1: The Koopman basis functions generated by a 20 layer drop-out ELU neural network. The x axis shows the size of the perturbation to a particular dimension of the observable vector y_n and the y axis shows the response of the basis functions.

To better understand what the deep Koopman operator was learning, we visualized the response of the neural network dictionary functions to basis perturbations along each observable axis. We found that much of the drastic changes in the basis function profile occurs in the first 10,000 iterations of training. During this process, the basis functions initialize as flat constant functions across all points and gradually train to spike, dip, or ramp in response to input signals (see Figure 3). This allows our Koopman dictionary to be updated during training, while simultaneously optimizing the Koopman operator. This may explain why the deep Koopman operator is able to achieve a higher fidelity than the extended DMD Koopman operator.

4.1.5 Example: The Glycolysis Network

The ultimate benchmark of whether or not a Koopman operator has been successfully learned is if the operator is able to predict multiple points into the future, using only a single root time-point. Any small amount of error in the Koopman operator estimate will propagate and compound on itself. Moreover, the ultimate challenge is to learn the Koopman operator on a nonlinear system for which there is no known Carleman linearization or Koopman invariant subspace.

The glycolysis pathway is a fundamental pathway to metabolism in biology. We use the standard model introduced in [27], a seven state nonlinear model with Michaelis-Menten dynamics. It is given as follows:

$$\begin{aligned}
\frac{dS_1}{dt} &= J_0 - \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q}, \\
\frac{dS_2}{dt} &= 2 \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q} - k_2 S_2 (N - S_5) - k_6 S_2 S_5, \\
\frac{dS_3}{dt} &= k_2 S_2 (N - S_5) - k_3 S_3 (A - S_6), \\
\frac{dS_4}{dt} &= k_3 S_3 (A - S_6) - k_4 S_4 S_5 - \kappa (S_4 - S_7), \\
\frac{dS_5}{dt} &= k_2 S_2 (N - S_5) - k_4 S_4 S_5 - k_6 S_2 S_5, \\
\frac{dS_6}{dt} &= -2 \frac{k_1 S_1 S_6}{1 + (S_6/K_1)^q} + 2k_3 S_3 (A - S_6) - k_5 S_6, \\
\frac{dS_7}{dt} &= \mu \kappa (S_4 - S_7) - k S_7,
\end{aligned}$$

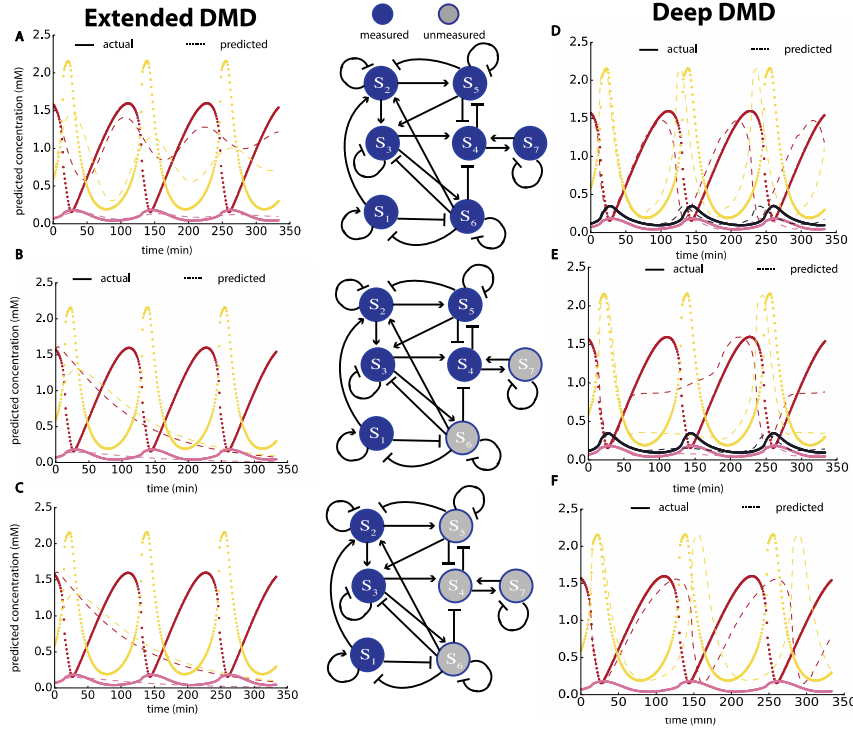


Figure 2: Multi-step prediction using a single time-point on the glycolytic oscillator. Trained using measurements from A) 7 states with EDMD, B) 5 states with EDMD, C) 3 states with EDMD, D) 7 states with our deep dynamic mode decomposition algorithm (DeepDMD), E) 5 states with DeepDMD, F) 3 states with DeepDMD.

The network has 19 edges and 7 nodes, autoregulatory feedback, both positive and negative feedback. We consider a challenging scenario, where the network parameters are tuned to oscillate. As seen in Table 1, the deep Koopman operator achieved less than .1% error on all test cases, whether $n = 7, 5$, or 3 nodes in the glycolysis network were measured. Interestingly, the deep Koopman operator is able to predict multiple time-steps into the future (~ 100 timepoints) with quantitative accuracy and qualitative accuracy for the full duration of the simulation (500 timepoints). These results underscore the capacity for deep dynamic mode decomposition models to fit and predict biological dynamics from SD2 datasets, as we demonstrate in the subsequent sections.

4.1.6 Example: Operational Envelope of the Toggle Switch

An important early synthetic biology research problem in the SD2 program was discovering the operational envelope of a genetic circuit. At the time, no computational or machine learning-based methods existed to predict what the range of conditions would be under which a circuit was operational. Part of the challenge was that the definition of operational or functionality was ill-defined. We examined the problem of defining functionality of the genetic toggle switch, the first digital genetic circuit that launched the field of synthetic biology [32]. Specifically, we evaluated the performance of the toggle switch as a memory-based comparator; how well would it map competing concentrations of the two input molecules, isopropyl- β -thiogalactoside (IPTG) and anhydrotetracycline (ATC), to the appropriate fluorescent state (high GFP versus high RFP). As part of an SD2 data generation effort, we collected data from 96 different pairwise combinations of varying concentrations for IPTG and ATC and characterized the genetic toggle switch's fluorescence in both GFP and RFP channels using single-cell analysis in a flow cytometer. Each sample generated 12 timepoints, with each timepoint generating a bi-variate distribution of gene expression for RFP and GFP. We calculated the first and second moments for each sample and fed this as training data to our deep dynamic mode decomposition algorithm.

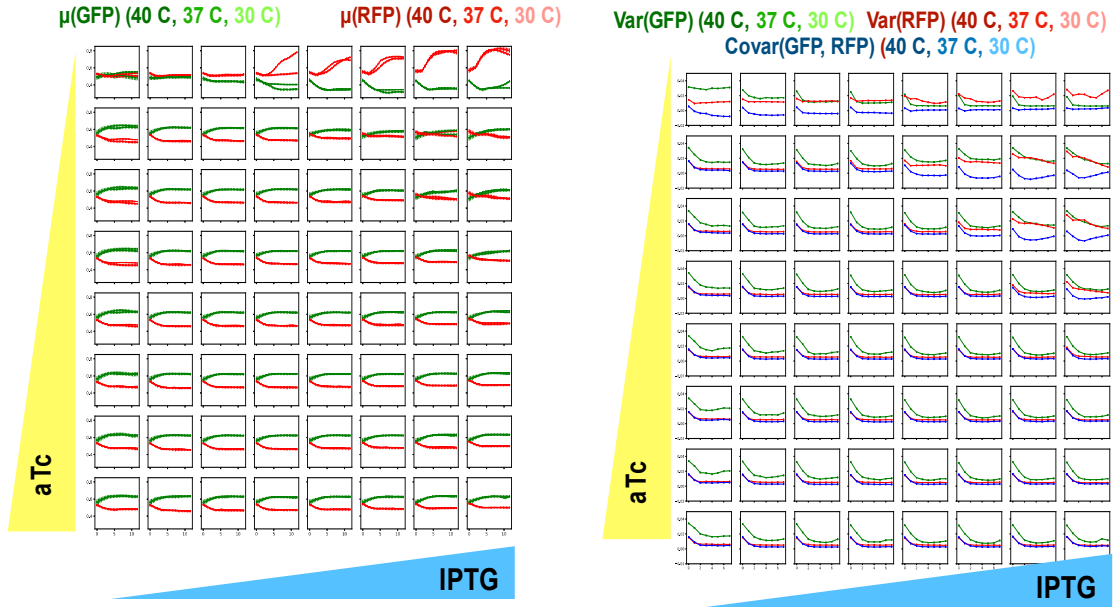


Figure 3: Model predictions and test data of the genetic toggle switch for varying concentrations of ATC and IPTG.

Predictions for the mean vector (left) and covariance matrix (right) of GFP and RFP flow cytometry distributions at each timepoint are plotted as green and red lines, respectively, while the ground-truth test datapoints are superimposed as dots. Four different temperatures were evaluated simultaneously in this experiment, using four different incubators, represented by varying intensity of either the green or red hue of the data.

We modified the deep dynamic mode decomposition algorithm to account for inputs, which in this case modeled the ambient temperature and ambient levels of IPTG and ATC as a three dimensional input. The training data for these models is shown as dotted lines, while the n -step model predictions are overlaid or superimposed as continuous red or green lines in Figure 3. Not only were we able to train a predictive model, we were then able to apply a logical specification of functionality to evaluate whether the genetic circuit performed as expected at each concentration of IPTG and ATC. This allowed us to explore the *operational envelope* or range of conditions, in which the genetic circuit operated correctly, as we varied the putative levels of each input. For each point in 2-dimensional input space, we plot the outcome of functionality as either green for the toggle switch functioning as expected, or red, to indicate circuit failure. We were able to interpolate across the phase space and plot a refined set of operational versus dysfunctional points to generate a first-ever empirical operational envelope of the genetic toggle switch (Figure 4).

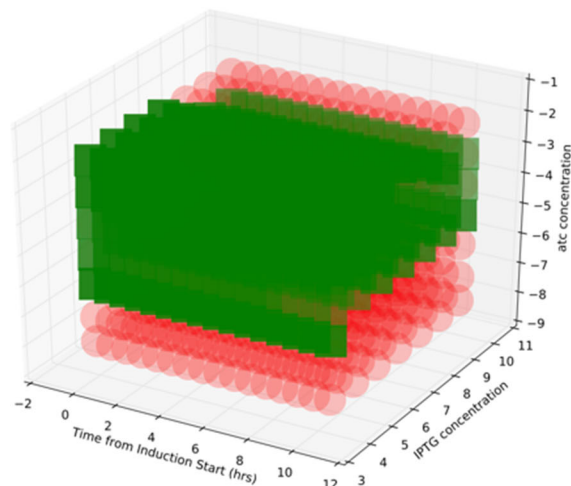
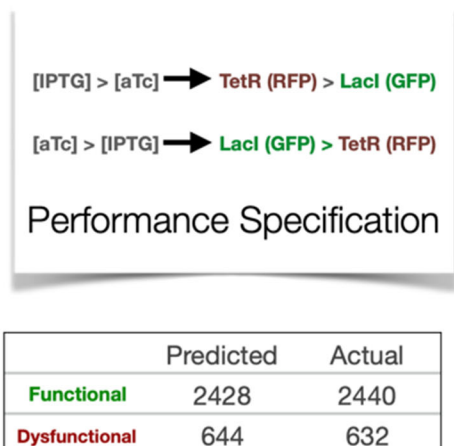


Figure 4: (Left) The performance specification used to evaluate the functionality of the toggle switch at each three-tuple of temperature, IPTG, and ATC in input-condition space. (Right) The operational envelope of the genetic toggle switch is visualized as the aggregate polygonal shape formed by green dots generated using deepDMD model predictions. Red dots indicate input-conditions that result in circuit failure.

4.2 Structured dynamic mode decomposition for analysis of additive, reductionist SD2 strain libraries

4.2.1 Summary of Results

Genetic circuits are designed to implement certain logic in living cells, keeping burden on the host cell minimal. However, manipulating the genome often will have a significant impact for various reasons (usage of the cell machinery to express new genes, toxicity of genes, interactions with native genes, etc.). In this work we utilize Koopman operator theory to construct data-driven models of transcriptomic-level dynamics from noisy and temporally sparse next generation RNA sequencing measurements (RNAseq). We show how Koopman models can be used to quantify impact on genetic circuits. We consider an experimental example, using high-throughput RNAseq measurements collected from wild-type *E. coli*, single gate components transformed in *E. coli*, and a NAND circuit composed from individual gates in *E. coli*, to explore how Koopman subspace functions encode increasing circuit interference on *E. coli* chassis dynamics. The algorithm provides a novel method for quantifying the impact of synthetic biological circuits on host-chassis dynamics.

4.2.2 Introduction & SD2 Context

Synthetic biology is concerned with building useful biological circuits with predictable dynamics. Most genetic circuits depend on the local machinery of a host chassis. Since the host must expend energy in expressing exogenous genes, resources are taken from endogenous processes which can impact the function of both synthetic and native genes. Genetic circuit burden is often explicitly modeled as sequestration of cellular resources or competition for a limited set of binding sites. These models are hypothesis driven, in that they rely on hypotheses or explicit knowledge of the nature of circuit burden.

Functional genetic circuits require a minimal footprint on the host. The higher the burden on the host, the more likely it is that the host mutates out the circuit. Even in cases where synthetic genes leave a minimal footprint on the host, an unpredictable change in dynamic behavior may occur due to the activation of synthetic genes. Rondelez showed that competition between synthetic and native genes have important effects on the global dynamics of the system [64]. Recent studies involving host-circuit interactions looked at cross talk [90] between genetic circuits and host resources for transcription [35], translation [11, 19, 34, 17, 56], and protein degradation [25, 60]. These model-based approaches further our understanding of host-circuit interaction, however the models are often based on biophysical mechanisms that are difficult to validate or observe. Transcriptomics and proteomics resolve the activity of thousands of genes, providing a rich resource for learning models to answer key questions without the need for hypothesis-driven modeling. How can these measurements be leveraged through a data-driven approach to better understand host-circuit interaction and genetic stability?

Spectral methods have been increasingly popular in the data-driven analysis of nonlinear dynamical systems. Recently, researchers working in Koopman operator theory have shown that it is possible to identify and learn the fundamental modes of a nonlinear dynamical system from data [65]. The seminal work by Schmid in developing dynamic mode decomposition (DMD) has led to an enormous growth in the use of Koopman spectral analysis of nonlinear dynamical systems [69]. More recently, learning higher dimensional Koopman operators from data has become computationally tractable, largely due to advances in integrating machine learning and deep learning to generate efficient representations of observable bases [91, 47, 58]. Often in biology and especially in omics measurements, the data are temporally sparse. Sinha and Yeung developed a method for computing the Koopman operator from sparse data [72].

Synthetic biological circuit design is often viewed from a reductionist’s perspective. Biological parts are designed and optimized for modularity, so that composition gives rise to predictable behavior. The challenge is that composition, while at times successfully gives rise to predictable *observed* behavior, has an unknown *emergent* effect on the host, and by the principles of feedback, the circuit as well [18].

In this report we develop a completely novel algorithm, *structured DMD*, to complement bottom-up genetic circuit design approaches in synthetic biology. Structured DMD is a purely data-driven model discovery framework that takes advantage of a part-by-part construction process to decouple emergent phenomena from isolated part dynamics. It reduces the total model complexity in the model identification process by adopting a hierarchical approach to identifying components of the model in stages. The decomposition we obtain is additive, due to nice linear mathematical properties endowed by Koopman operators [50]. We showcase our algorithm on a coupled oscillator system, but then consider a real genetic circuit design problem using a NAND gate designed from Tet regulator (TetR) orthologs [75]. Full-state but temporally sparse RNAseq measurements collected from wild-type *E. coli*, single gate components transformed in *E. coli*, and a NAND circuit composed from individual gates in *E. coli* are used to explore how Koopman subspace functions encode increasing circuit interference on *E. coli* chassis dynamics.

4.2.3 Structured dynamic mode decomposition

In this section we, for the first time, introduce structured dynamic mode decomposition (structured DMD). The structured DMD algorithm takes advantage of bottom-up design approaches where an original system is built upon by adding parts layer by layer to achieve complex dynamical behaviors.

The Koopman model (2) can be decomposed into original and added equations, or its parts in the bottom-up design approach, written as

$$\begin{bmatrix} \psi_O(x_{t+1}) \\ \psi_A(x_{t+1}) \end{bmatrix} \equiv \begin{bmatrix} K_{OO} & K_{OA} \\ K_{AO} & K_{AA} \end{bmatrix} \begin{bmatrix} \psi_O(x_t) \\ \psi_A(x_t) \end{bmatrix} \quad (4)$$

where the subscripts O and A correspond to the original and added components. The matrix of Koopman operators in (4) is unknown, but can be obtained using the standard techniques outlined in our Methods section. However, these approaches would not allow the decoupling of the underlying dynamics of the original components from added components. Therefore, it would not be possible to determine the impact that new components have on the original components. With this algorithm, we propose to discover the underlying (original) dynamics directly from data first, and then subsequently learn the interaction dynamics as an additive perturbation in the Koopman model.

If we want to solely understand the impact of added components on the original components of a system, we first learn the original-original interaction dynamics matrix K_{OO} from the original system without added parts. This original system has Koopman model

$$\psi_O(x_{t+1}) = K_{OO} \psi_O(x_t)$$

where K_{OO} is learned through the optimization problem (3). The original-added interaction dynamics matrix K_{OA} can now be learned by viewing it as an added perturbation in the original Koopman model, i.e. the new model is

$$\psi_O(x_{t+1}) = \underbrace{K_{OO}}_{\text{known}} \psi_O(x_t) + K_{OA} \psi_A(x_t).$$

Here we have already learned K_{OO} and of interest to us is K_{OA} which can now be learned directly from data. In this way, we can completely decouple the underlying original dynamics from the effect that any added parts have on the original dynamics.

4.2.4 Quantifying impact of the SD2 NAND gate on host

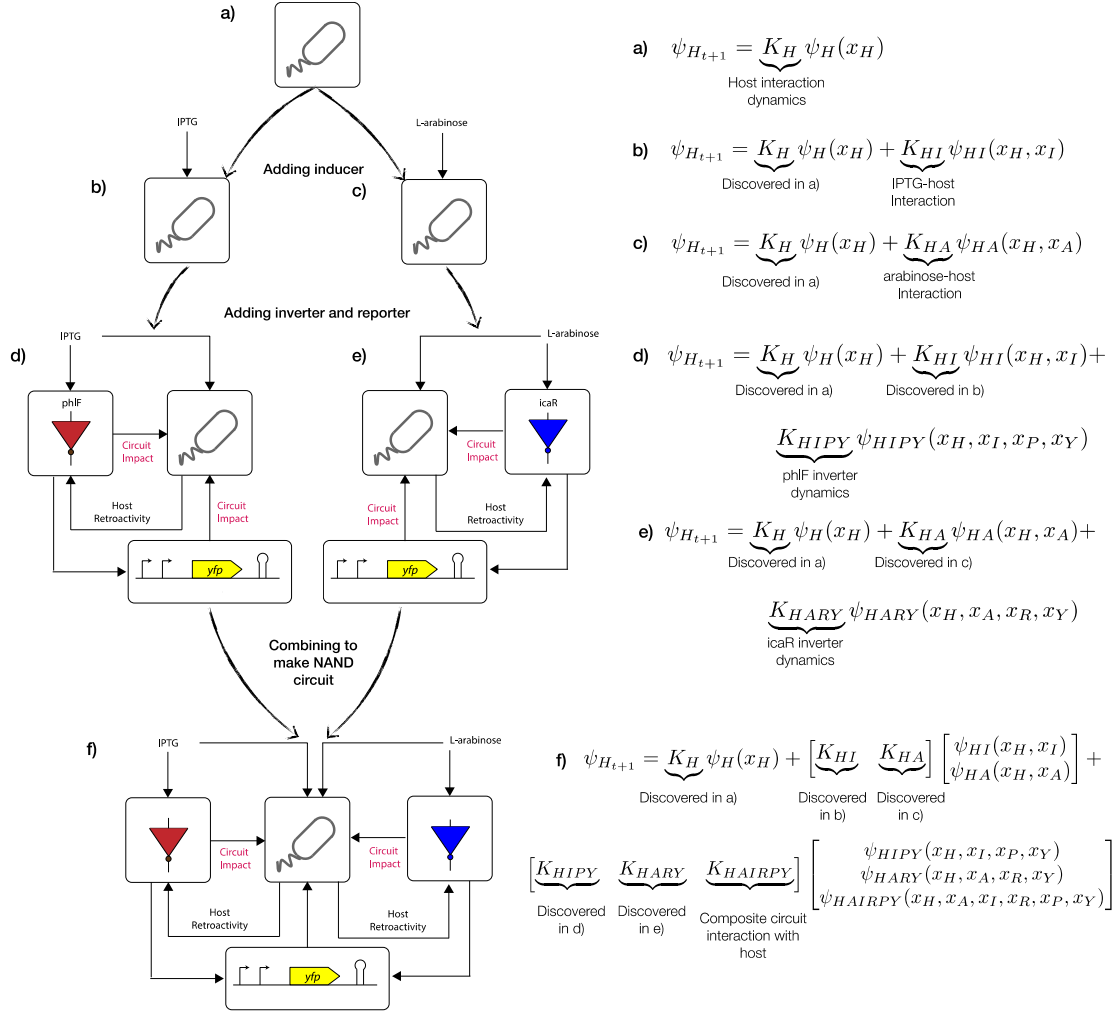


Figure 5: Schematic of the bottom-up design of a NAND gate in *E. coli*.

a) *E. coli*, b) *E. coli* with IPTG input c) *E. coli* with L-arabinose input d) *E. coli* with IPTG input, PhlF gate, and YFP reporter e) *E. coli* with L-arabinose input, IcaR gate, and YFP reporter, f) complete NAND circuit. Under each design iteration is the associated host Koopman model.

First, note that typically RNAseq measurements are sparse in time (two timepoints and four replicates in this case). Sinha and Yeung [72] have addressed the problem of computation of the Koopman operator when the data is sparse. For each data tuple (x_i, x_{i+1}) , the artificial data point $(x_i + \delta x_i, x_{i+1} + \delta x_{i+1})$ is added. Artificial snapshot matrices are formed as:

$$\begin{aligned} X_f &\equiv [x_1 \quad \dots \quad x_{N-1} \quad x_1 + \delta x_1 \quad \dots \quad x_{N-1} + \delta x_{N-1}], \\ X_p &\equiv [x_2 \quad \dots \quad x_N \quad x_2 + \delta x_2 \quad \dots \quad x_N + \delta x_N] \end{aligned}$$

These artificial data points (which are sufficiently small perturbations) are added to the sparse data set to enrich the data. Robust optimization-based techniques are then used to compute the approximate Koopman operator. The optimization problem to be solved is:

$$\min_K \left\| \Psi(X_f) - K\Psi(X_p) \right\|_F + \lambda \left\| K \right\|_F$$

where λ is a regularization parameter.

Figure 5 shows the bottom-up construction of a NAND gate in *E. coli* where each iteration also has an associated host Koopman model (under each schematic). The composite interaction matrix K_{HAIRPY} defines the impact of the NAND gate on the host. To learn this matrix, we first learn the underlying host dynamics K_H in Figure 5a. A heatmap of K_H can be seen in Figure 6. Inducer-host interactions K_{HI} and K_{HA} in Figure 5b and 5c are computed next. The subscripts *I* and *A* correspond to the inducers IPTG and L-arabinose, respectively. PhlF and IcaR inverters are then added along with *yfp* reporters as seen in Figure 5d and 5e. K_{HIPY} and K_{HARY} are learned from these systems where the subscripts *P*, *R*, and *Y* denote the PhlF inverter, IcaR inverter, and *yfp* reporters, respectively. At this stage, we can learn K_{HAIRPY} since all the terms in the Koopman model of Figure 5F are now known. Figure 7 shows a heatmap of K_{HAIRPS} .

The discovered host Koopman model from wild type MG1655K12 *E. coli* reveals a fundamentally antagonistic relationship between the LacI and AraC control modules in the transition from the first time point (log phase) to the second timepoint (stationary phase). We see that arabinose induction activates most of the Ara operon genes, as expected, but simultaneously creates a negative inhibitory effect on LacY, LacZ, and LacI. Conversely, induction with IPTG has a significant downregulating effect on the Ara operon genes, specifically the cluster of genes downstream of the pBAD cluster. This is consistent with prior analysis of the hierarchy of sugar utilization [1]. Crosstalk mechanisms mediated by the catabolite repression protein (CRP) and cyclic adenosine monophosphate (cAMP) pathway prioritize Lac operon activity over Ara operon activity when both sugars are present. While IPTG is not a sugar, it acts as a structural analog and thus induces the same diauxic response. We have verified this phenomena, using a data-driven approach, using only two timepoints and four biological replicates from noisy RNAseq measurements.

The import of this finding is that the NAND circuit inherently activates the diauxic response mechanism to its advantage. Since PhlF and IcaR are designed and intended to act independently, mutual repression of their underlying host machinery results in stronger underlying XOR logic and mutual coupling. That is, activation with arabinose will result in indirect repression of the lactose operon, and vice-versa.

Finally, we discovered that IcaR gene expression induces a positive-feedback loop with the AraBAD cluster. This in turn, results in elevated IcaR expression, which induces cytotoxicity. When calculating the Frobenius norm, as a total sum measure of circuit-to-host impact, we found that arabinose induction in the host had an impact of $\frac{\|K_{HA}\|}{p \times q} = 4 \times 10^{-7}$ (*p* and *q* are the dimensions of K_{HA}), while induction of the IcaR component had an impact of $\frac{\|K_{HIPY}\|}{r \times s} = 2 \times 10^{-2}$, nearly 5 orders of magnitude greater. Even though PhlF and IcaR had comparable per-term gain over the 429 genes we analyzed, IcaR impacted 420 genes, while PhlF only impacted

269 genes. When analyzing DNA sequencing data, we found that the IcaR gene had been deleted from the NAND circuit on the genome; the disparity in gain and widespread influence between the IcaR and the other circuit components provides a hypothesis for IcaR mutation. The IcaR part imposes considerable widespread perturbation on host genes, which is evidence of cytotoxicity leading to mutation.

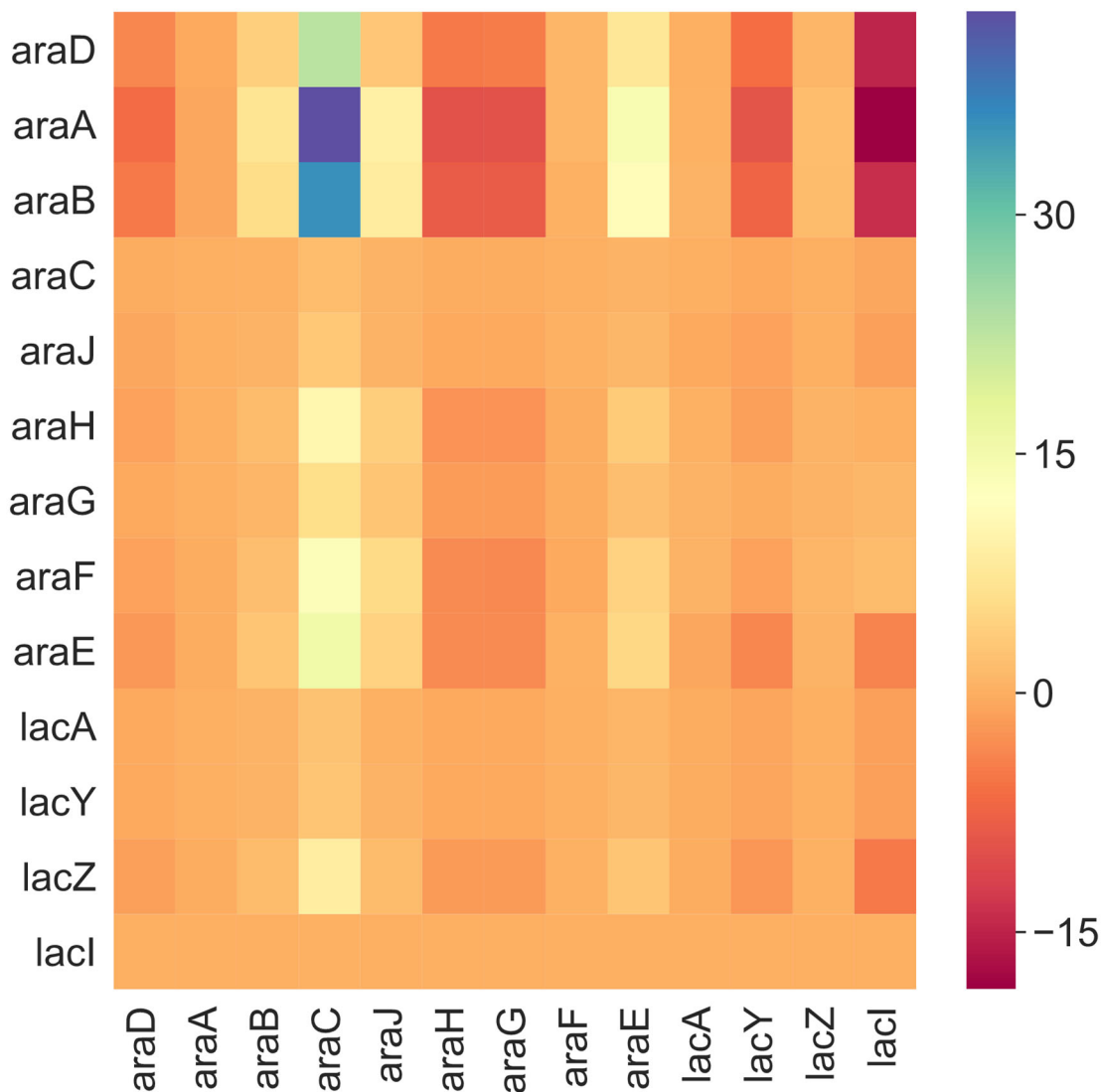


Figure 6: The Koopman operator estimated from structured DMD of the host dynamics in response to arabinose and IPTG induction. The color scale represents likely causal interaction; positive causal interaction represented by positive values and negative causal interaction represented by negative values.

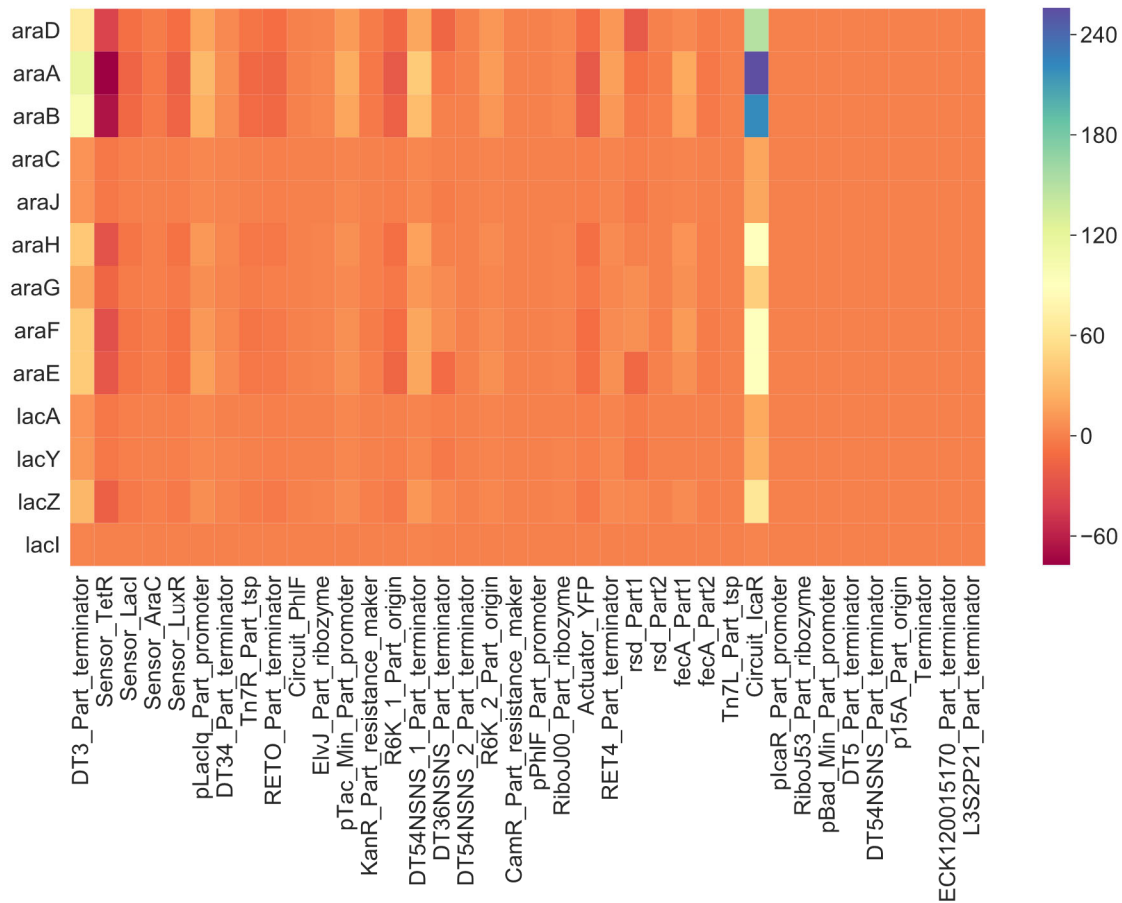


Figure 7: The input Koopman operator estimated from structured DMD modeling circuit-to-host interaction of the NAND circuit.

The color scale represents likely causal interaction; positive causal interaction represented by positive values and negative causal interaction represented by negative values.

4.3 Data-driven circuit specification for SD2 obstacle course challenges

4.3.1 Introduction & SD2 Context

Since the advent of systems-level thinking of biological networks, the genetic circuits and motifs that constitute them have received significant attention [80, 3]. These emergent constituent elements achieve specific computational objectives like positive or negative feedback, autoregulation, bistability etc [51]. Synthetic versions of these components developed recently permit the biomanufacturing industry to extend and optimize existing faculties and incorporate novel phenotypes at pico-watt levels of power consumption [13, 77]. This is however, an expensive, time-consuming and difficult process because of the highly nonlinear nature of biological networks and lack of standardization of parts. There is a growing need for automated or computational approaches for designing, engineering and optimizing genetic circuits. These would streamline the design process by narrowing down the different designs to test. Many tools do achieve this in an open loop fashion [31, 55] without using prior performance data for iterating and improvement.

As in any engineering process, iteration is necessary to improve existing designs. Efficiently closing the Design-Build-Test loop for synthetic genetic circuits and automating it is key to this. The Design-Build-Test-Learn (DBTL) cycle is an indispensable, ubiquitous process in engineering. They systematically improve the design performance after each iteration. It is the norm in fields like aerospace and automotive engineering. This philosophy of iterative closed-loop thinking is now being extended to synthetic biology [87, 16, 57]. This is important because, even upon noticing a performance discrepancy, it is difficult to determine what component of a large network should be modified to better the performance. Closing the DBTL cycle will automate genetic circuit revision in a biophysically informed way.

Circuit design via first principal physical approaches is not sufficient to scale the Design-Build-Test process to commercial levels. Biomanufacturing is a rapidly growing industry that heavily relies on synthetic biology and its promise of programmable biology. It discovers sustainable solutions to serious global issues like environment clean-up, climate change, food-security [18]. Designing genetic circuits on an ad-hoc basis is unsuitable for a rapidly growing biomanufacturing market. The lack of automation makes it difficult for circuit designers to scale synthetic biology to commercial levels in a cost-effective way.

State-of-the-art methods of improving genetic circuit design involve topology-based intuition and modeling component-to-component interactions via ordinary differential equations and Hill functions. However, like in many different kinds of dynamical systems like fluid or solid mechanics, it is no longer feasible to rely on an ansatz and recommend changes to it based on intuition. The scale of data collected is so vast that we need methods of learning the biophysics of a system via data-driven methods.

Genetic circuit design in the past has been topology-centric. Systems-level research efforts have revealed several non-trivial network motifs that emerged out of evolution like feedforward loops, autoregulatory loops and repression loops. In the past, circuit design was based on the intuitive understanding of how these motifs worked individually and together. This involved a lot of trial-and-error and was time-consuming. It increases the design space of genetic circuits to have the user specify the desired behavior as time-series data.

Deep learning accelerates the discovery of novel biological function and exploration of feasible design space in comparison to previous methods. Genetic circuits need to exhibit specific time-series behavior from different nodes. Data-driven snapshot based methods are great tools to study such systems. Dynamic Mode Decomposition (DMD) has emerged as a powerful method of learning important physical structures in dynamical systems and has already been applied towards synthetic biology. There are numerous modifications of DMD. Extended Dynamic Mode Decomposition is one that improves the performance of DMD by enriching the state vector with physically relevant observable functions to ensure Koopman invariance or closure.

Manual selection of these observables is cumbersome and can be automated via even more modern methods like deepDMD. DeepDMD utilizes a deep neural network to recommend biophysically realizable observable hill functions. The interactions specified by the observables between the circuit nodes translate to tangible circuit design in terms of what components to use. The control deep dynamic mode decomposition algorithm (cDeepDMD) uses user-specified performance metrics or circuit characterization data to recommend these hypotheses and modifications. Thus, cDeepDMD closes the DBTL cycle.

4.3.2 Results

The deep learning-driven Koopman circuit optimization tool (cdeepKCOT) allows users to draw their desired circuit behavior freehand. The algorithm is meant to be deployed as a GUI application run on tablets, smart phones and computers. Users will be able to use an electronic pencil to whiteboard time-series performance curves. An imaging feature is also available to photograph these on a plain surface and define an origin. The algorithm will require users to input all the time-series data for each node for all conditions of interest. The algorithm then smooths the data and transforms the data to ensure each time-point has a uniquely associated value before the training process begins with cDeepKCOT.

The cDeepKCOT algorithm arranges the data received from the acquisition function into snapshot matrices used for fitting the Koopman operator. The DeepDMD framework that fits the data learns several observable functions. The number of observable functions to be learned can be specified manually in the deep dictionary size variable. The deepDMD algorithm tries several different observable functions to achieve the goal of minimizing the 2-norm error of the Koopman input equation. Thus, the algorithm seeks the optimal values of the observables and the Koopman operator simultaneously.

The cDeepKCOT algorithm identifies a large number of observable functions. This gives a fine-grained approximation of the governing equations of the system. Many of the observable functions are qualitatively similar in that they can be classified as logistic, hill or radial basis functions. The difference between them being minor variations in the parameters used to fit them. This is expected as, under quasi-steady states, the rate of transcription from each node could have hill function dynamics. Hence, the Lie derivatives from these observables could have Hill function-like features. Based on a Euclidean metric to determine the difference between the learned functions, a smaller subset of functions is selected. This is important because it narrows down the number of design hypotheses and circuit topologies to build and test. The modification of the Koopman operator after this down-selection maintains closure and learns the dynamics of the system albeit at a coarse-grained, but biophysically possible level. The parameters determined by these final hill functions recommend the circuit components to purchase and assemble.

As an example, we started with a time-series curve that is important from a signals perspective, but quite rare in nature. We used a 2-node circuit which generates square waves from both nodes. The data (Figure 8 left) was acquired using the drawing tool and cDeepKCOT learned the Koopman operator (Figure 8 middle). The n-step prediction metrics for various hyperparameters is shown in (Figure 8 right). The optimum number of layers found was 11, using ReLU activation functions. The initial number of activation functions was 20 and the Euclidean winnowing factor reduced it to 4. The n-step prediction from the learning process is shown in Figure 8. The run took 35 minutes using 1 GPU (V100 Titan). Our analysis here shows that complex dynamical behavior can be realized using deepDMD algorithms; our future work after this project will investigate sparsification approaches for finding efficient biophysical realizations of these genetic circuits suggested by deep learning.

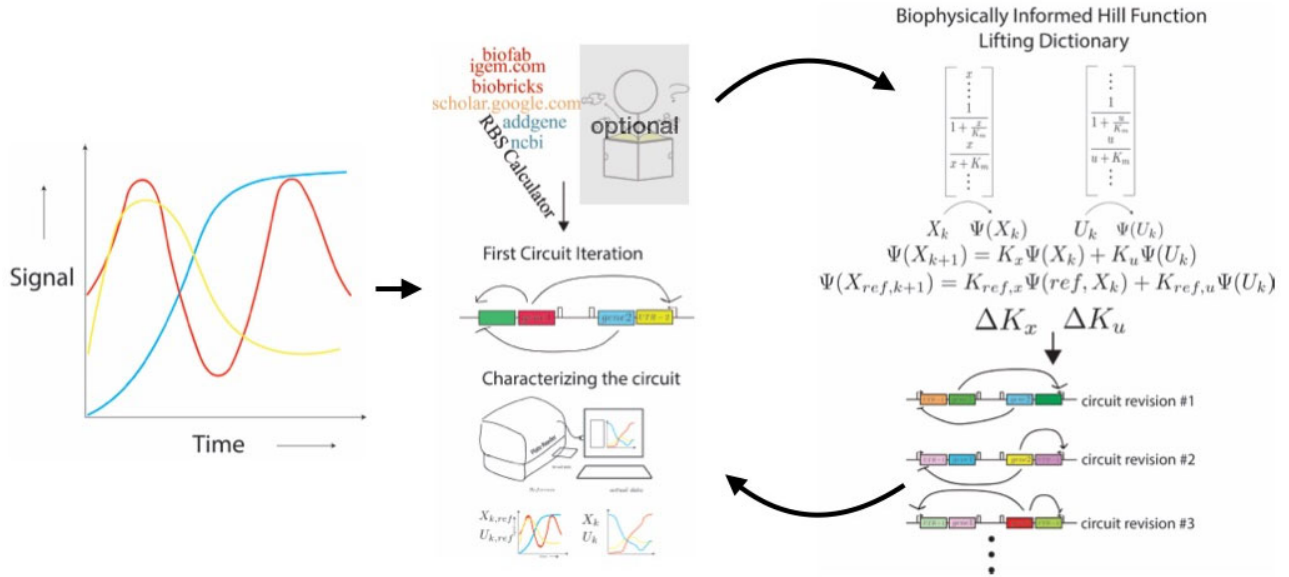


Figure 8: An overview of how the KCOT algorithm works. (Left) The user specifies desired output circuit dynamics. (Middle) Extended or deep dynamic mode decomposition generates a candidate circuit model and network topology to realize; the user tunes the gain of the genetic parts predicted by dynamic mode decomposition using standard sequence-to-function tools like the RBS calculator. (Right) The user generates data from the circuit; either experimental or simulated and revises the design by using dynamic mode decomposition to evaluate a data-driven model. Discrepancies between the reference model and the data-driven model direct the revision process.

To demonstrate cDeepKCOT’s ability to revise circuit topologies, we designed from user-specified reference curves, a 2-node hysteretic switch. The topology of this 2-node hysteretic switch was derived by specifying a series of reference curves shown in Figure 9.

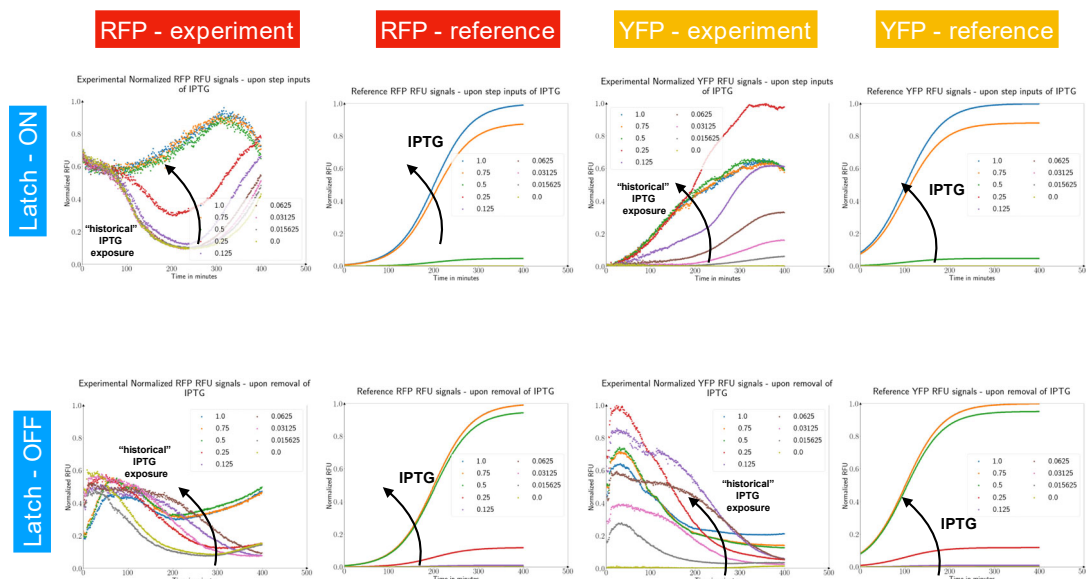


Figure 9: The reference curves for a two node hysteretic switch designed using the KCOT algorithm.

The output traces were generated analytically by defining logistic ramp functions to specify transient behavior of our hysteretic switch. On the left is shown the desired YFP output of the two node switch, while the right shows the RFP output of the two node switch.

We designed a genetic two node version of this circuit, based on the topology recommended by the KCOT tool. The suggested topology to realize the reference dynamic response to varying IPTG levels shown in Figure 10, featured one activation loop, with positive autoregulatory loops on both nodes and a repression loop on the other end. We designed a genetic circuit to realize this topology, using standard synthetic biological parts available in the SD2 program. After generating plate reader data for this circuit, we trained the Koopman operator of the actual system and found that the realized genetic circuit topology was broadly able to recapitulate the reference curve response to varying levels of IPTG. Moreover, the reference Koopman operator (generated from the reference curves) and the revised Koopman operator for this circuit appeared to match very closely, when inspecting both matrices side-by-side, see Figure 10.

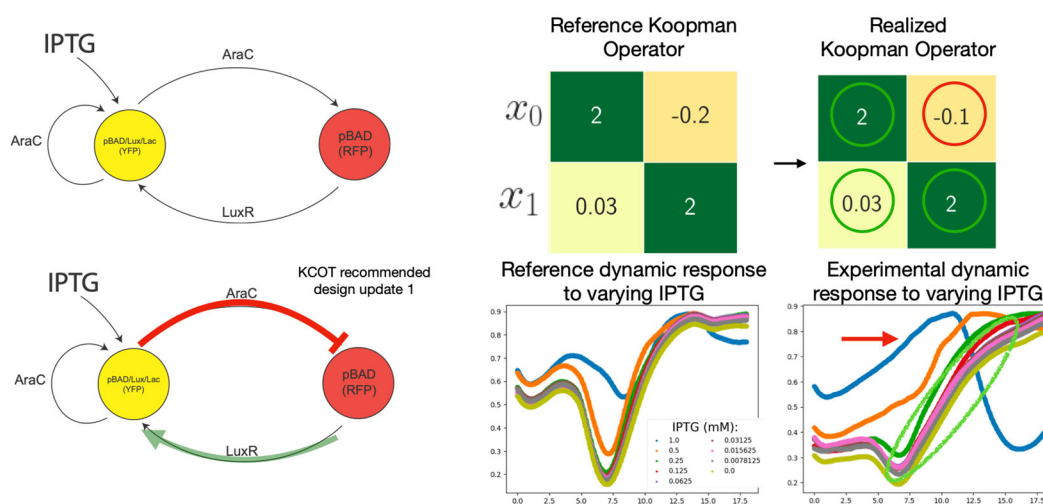


Figure 10: (Top Left) Circuit network topology recommended from first iteration with the Koopman Operator circuit optimization tool (KCOT), featuring mutual positive feedback. (Bottom Left) Circuit network topology recommended from comparative analysis of the original Koopman operator and the realized Koopman operator; a positive activation edge is replaced with a negative repression edge. (Top Right) The reference Koopman operator generated from reference curves (left) and the final realized Koopman operator obtained from genetic circuit engineering and circuit characterization data (right). (Bottom Right) The experimental data (right subpanel) shows overall consistency of the two node switch to varying levels of IPTG, when compared with the original reference curves used to train the reference Koopman operator (left subpanel).

At the beginning of the SD2 program, there was no extant capability for going from a whiteboard with a synthetic biologist, specifying how a circuit would behave by drawing the behavior by hand, and then determining what circuit topology and design would realize this behavior. The KCOT algorithm has bridged this gap entirely, by combining the expressive power of Koopman operators and the numerical efficiency of deep dynamic mode decomposition and extended dynamic mode decomposition algorithms. Now, we can specify a biological behavior for a circuit that is not yet realized or conceived; our algorithm ingests this as coarse grained data and then infers candidate networks that will realize this behavior, which lead to automatic design and build steps in a Foundry lab or directed genetic engineering in a benchtop lab setting.

4.4 The Sensor Scraper for Organophosphate Detection

The expansion of the genetic toolkit for circuit design remains an ongoing challenge as substantial effort is required to mine, design, characterize, and optimize biological parts [71, 52, 39, 49, 45]. A majority of the literature in this field has focused on optimizing and characterizing existing biological parts for genetic circuit design with less attention placed on mining biological parts. This has resulted in much needed insulation and biological orthogonalization strategies [26] for mitigating inadvertent intra-circuit and inter-circuit-host interactions. Moreover, programmatic tools have been developed to automate the design of genetic circuits that implement logical operations using a set of well-characterized parts in model organisms [48, 55, 20, 69]. However, since biological parts and circuits are characterized and optimized within a single model organism and often not evaluated in application relevant organisms, there is no guarantee that the parts can be “taken off the shelf” for use in engineering novel host organisms. A pivot of focus to mining biological parts from novel host organisms would naturally lend itself to aiding in the iterative design approach that the field has largely embraced.

Transcriptional genetic sensors are a class of biological components that control the activity of promoters [84] and have been used to construct whole-cell (living) biosensors [12, 53, 74]. A large portion of transcriptional sensors rely on transcription factor-promoter pairs [67] and have been used in whole-cell biosensing for detection of heavy metals [86], pesticides and herbicides [22, 44, 83], waterborne pathogens [92], disease biomarkers [6, 28], and many more applications discussed in [85]. Since microbes are found in virtually all terrestrial environments, one could imagine that there would be no shortage of transcriptional genetic sensors for novel sensing applications. However, given a novel analyte of interest, a set of biological parts to iterate over for transcriptional sensor design generally does not exist.

The transcriptional activity of an organism can be measured through RNA sequencing (RNA-seq) to produce a snapshot of the bulk cell state subject to intrinsic and/or extrinsic perturbations. The typical approach for identifying upregulated and downregulated genes across experimental conditions is to apply differential expression analysis [46, 63]. A major pitfall with differential expression analysis is its lack of statistical power when faced with a sparse number of biological replicates. That is to say that the false positive rate increases drastically when only a small number of biological replicates are available [70] as is often the case due to the costliness of RNA-seq. A related issue arises in that one must sacrifice time points for biological replicates, reducing the fidelity of the dynamical process being studied. As most biological processes are dynamic, time-series data are essential for accurate modeling of these processes. Furthermore, differential expression analysis provides no information beyond which genes are upregulated/downregulated.

A typical RNA-seq dataset contains hundreds to tens of thousands of genes; despite that, a subset of genes, which we call *target genes*, are typically sufficient for representing the underlying biological variation in the dataset. This observation is explained by the fact that variations in many genes are not due to the biological process of interest [62] and that many genes have correlated expression levels [82]. The task of identifying a subset of the state (genes) which recapitulate the entire state (transcriptome/cell state) and explain the variations of interest has long been studied in the field of dynamics and controls in the form of optimal filtering and sensor placement [5, 21]. In the context of dynamic transcriptional networks, sensor placement is concerned with inferring the underlying cell state based on minimal measurements and the given transcriptional dynamics; this

introduces the concept of observability of a dynamical system [38]. Measures or degrees of observability for linear dynamical systems are captured in the observability gramian [43], which directly informs the amount of energy that measurements of the system produce. To the best of our knowledge, measures of observability have not been applied to genetic networks to identify target genes or genetic sensors.

Overall, a systematic approach for identifying genetic sensors from RNA-seq datasets is still an open and challenging issue. In this work, we develop a machine learning methodology to extract numerous endogenous biological sensors for analytes of interest from time-series gene expression data (Figure 11). Our approach consists of three key steps, each of which is depicted in the middle panel of Figure 11. The first step involves learning the temporal evolution of gene expression by adapting dynamic mode decomposition (DMD) [69, 66, 81] to describe the temporal response of cells to a perturbation using only a sparse set of dynamic modes. Beyond the scope of sensor identification, we show how the dynamic modes can be utilized to cluster genes by their temporal response. The second step involves assigning sampling weights to each gene such that the observability of the learned system is maximized [38, 7, 36]. The sampling weights provide a machine learned ranking of the genes based on their contribution to a measure of energy of the system, and using this ranking, target genes may be selected. To ensure the ranking is identifying genes which can recapitulate the cell state, the third and final step is to select the p genes which maximally contribute to the observability of the system and perform cell state reconstruction using the output of the first two steps. To close the Build-Measure-Learn loop, we use our method to generate a set of 15 synthetic genetic sensors for the pesticide malathion [30, 2, 29], an organophosphate commonly used for insect control, in the bacterium *Pseudomonas fluorescens* SBW25. The transcriptional sensors play distinct biological roles in their host and exhibit unique malathion response curves. Our method uses no prior knowledge of genes involved in malathion sensing or metabolism. Moreover, we use no data source beyond RNA-seq, thereby providing a cost and computationally efficient approach for transcriptional sensor identification.

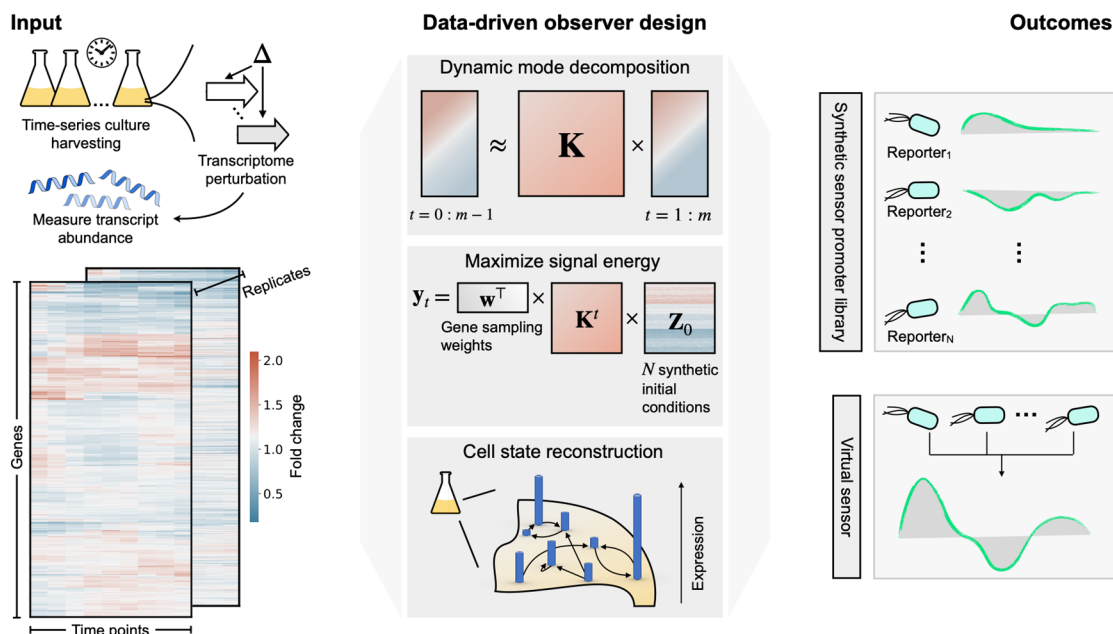


Figure 11: Transcriptional genetic sensors underlying the response of transcriptome perturbations can be extracted using data-driven observer design.

Bulk RNA sequencing (RNA-seq) measures transcript abundance over time following transcriptome perturbations. Our method starts by applying dynamic mode decomposition (DMD) to fold change response to discover dynamic modes which govern the evolution of the cell state. The dynamic modes are used to design a state observer (gene sampling weights) that maximize the signal energy (observability) of the learned system. Measurements from a subset of genes (target genes) informed by the gene sampling weights are then used to reconstruct the cell state. Our method returns: 1) A dynamics matrix describing how expression of gene i at time k is impacted by gene j and 2) gene sampling weights. The outcome, demonstrated in this work, is a library of synthetic sensor promoters (reporter) that are used to detect an analyte of interest. Since each genetic reporter has a unique response to the same perturbation, the library can be artificially fused to produce a purely virtual sensor for enhanced reporting.

4.4.1 Induction of malathion elicits fast host response.

Transcriptional activation and repression of the soil microbe *Pseudomonas fluorescens* SBW25 was induced by malathion at a molar concentration of 1.83 mM (460 ng/ μ L), a moderate amount that can typically be found in streams and ground water after recent pesticide use based on studies done in the United States, Malaysia, China, Japan, and India [89, 76]. Malathion is an organophosphorus synthetic insecticide used mainly in agricultural settings [54] while SBW25 is a strain of bacteria that colonizes soil, water, and plant surface environments [40]. This makes the soil-dwelling strain a prime candidate for identification of transcriptional genetic sensors for the detection of malathion.

We made use of a custom-built vacuum manifold that enables rapid harvesting and instantaneous freezing of cell cultures to arrest transcriptional dynamics. Following malathion induction, cells were harvested at 10 minute intervals for 80 minutes, obtaining a total of 9 time points across two biological replicates that were sequenced. To identify target genes for malathion induction and

subsequently build synthetic sensors, we also collected samples from a cell culture that was not induced with malathion.

RNA sequencing (RNA-seq) provides a snapshot of the entire transcriptome i.e. the presence and quantity of RNA in a sample at a given moment in time. In this work, we examine the fold change response given by first normalizing the raw counts to obtain transcripts per million (TPM) [23] followed by calculating the fold change of the malathion condition with respect to the negative control. By analyzing the time average of the fold change response of each gene, we found that a majority of the protein-coding genes were unperturbed by malathion. There are approximately 6000 known genes in the SBW25 genome, of which 10% were kept for modeling and analysis due to their variation across conditions and relatively high abundance. Specifically, we find that the fold change response of the malathion condition with respect to the negative control shows that 15% of the genes had a time average fold change less than 0.95, 32% had an average fold change greater than 1.05, while the remaining genes had a time average fold change between 0.95 and 1.05. This is consistent with the accepted common knowledge that most genes are not differentially expressed in any one experimental condition [46].

Given our goal of extracting salient sensors from time-series gene expression data, we first model the dynamical process that is driven by the step input of malathion on the SBW25 transcriptome. Specifically, we apply dynamic mode decomposition (DMD) to approximate the fold change response with a sparse set of dynamic modes. In what follows, we demonstrate how DMD can accurately describe gene expression dynamics by decomposing the time-series gene expression into temporally relevant patterns.

4.4.2 Dynamic mode decomposition uncovers modes of host cell response.

The dataset described in the previous section presents a challenge for the statistical modeling techniques commonly used for target gene selection (e.g. differential expression analysis), which typically require three or more replicates to achieve acceptable statistics while six replicates is the minimum recommended number [70]. For this reason, we adopt tools developed in dynamical systems and control theory to i) learn a model of the dynamics of the fold change response and ii) design a target gene identification approach which uses the learned model to rank the genes based on their contribution to the observability of the learned dynamical system (Figure 11).

Dynamic mode decomposition (DMD) is a time-series dimensionality reduction algorithm that was developed in the fluid dynamics community to extract coherent structures and reconstruct dynamical systems from high-dimensional data [69]. Recently, several groups have adapted and applied DMD to biological systems in various contexts [14, 8, 73, 79, 37], choosing DMD for its ability to reproduce dynamic data over traditionally static methods such as principal component [4] or independent component analysis [78].

To uncover the diverse modes of the host cell response to malathion induction, we performed (exact) DMD [81] on the transcriptomic dataset. Specifically, we perform DMD on the fold change, defined as the expression of a gene in the malathion condition divided by the expression of a gene in the negative control, after standard normalization (z-scoring) across the time dimension and the two conditions. We will later clarify the rationale behind this choice of state in this section and the next.

DMD decomposes a gene expression matrix (genes $\times$ time points) into dynamic modes, each of which is associated with a damped (or driven) sinusoidal behavior in time. The reconstruction of the step response of the fold change dynamics is schematically represented in Figure 12a. The heatmap $\mathbf{V}$ represents the matrix of 10 learned dynamic modes, each of which has rate of growth or decay and oscillation frequency given by a single corresponding DMD eigenvalue in $\mathbf{\Lambda}$, and mode amplitude given in $\mathbf{b}$. As they are complex-valued, the magnitude and phase of each DMD mode, eigenvalue, and amplitude is visualized separately. The magnitude of each DMD mode represents gene-wise coherent activation while the phase represents the shift of the damped (or driven) modes relative to each other.

We uncover three dynamically distinct modes of response DMD analysis from the *Pseudomonas fluorescens* SBW25 transcriptome response to malathion. The modes of dynamic response are characterized by the corresponding DMD eigenvalue, λ , and all of the eigenvalues are plotted relative to the unit circle in Figure 12b. The first type of mode that we recover are stable modes of which six of the ten modes are of this type. These modes are characterized by eigenvalues which are inside the unit circle, meaning that the magnitude of λ is less than 1. The majority of modes having stable dynamics is sensible as most genes did not have a response to the step input of malathion and therefore have fold change near 1.0 for all times (near zero after standardization). Stable modes that have eigenvalues nearer to the unit circle are capturing those uninhibited genes, while stable modes that are nearer to the origin are capturing genes which are initially further from the origin and converge to zero over time, i.e. they exhibit strong relative decay in their fold change. The second type of mode we uncover are oscillatory modes, of which there are eight. Eigenvalues with nonzero imaginary part give rise to these oscillatory modes and always come in complex conjugate pairs (if the data are real-valued). Each pair of complex-valued modes then describe a fixed frequency of oscillation, and each gene's dynamics can be reconstructed from one or more of these frequencies. The third and final type of response is an unstable response characterized by eigenvalues whose magnitude is larger than 1.0. As many of the genes were upregulated in response to malathion, this change is reflected in the fold change increasing from the neutral fold change of 1.0. If a gene has this behavior near later times, the DMD model will require this unstable mode to be able to capture the seemingly unstable dynamics. This is because DMD is essentially learning a linear state-space representation of the fold change response and a linear system can only exhibit three types of limiting behaviors, i) convergence to the origin (stable), ii) periodic orbits, and iii) divergence to infinity (unstable). Therefore, for the reconstruction accuracy to be minimized, unstable DMD eigenvalues are necessary.

Though the dynamic modes are not parsimonious, we show that they can successfully group genes into interpretable clusters that qualitatively describe how a gene's expression evolves in time. Figure 12c (upper) shows the evolution of three dynamic modes (scaled by their respective eigenvalues raised to the current time point, $\lambda_i^t b_i$): modes 1, 2, and 6. Mode 1 corresponds to a stable invariant direction in gene space; if expression of all the genes coincided exactly in the

direction of mode 1, their gene expression would be given by mode 1 dynamics alone. Mode 1 has temporal dynamics that are characterized by zero oscillations (verified by the corresponding eigenvalue in Figure 12b) and increasing mode amplitude. We call this mode the (stable) divergent response mode. The genes which are most influenced by mode 1 are those which correspond to elements of the first column of $\mathbf{V}$ that have the largest magnitude. The fold change of the 10 genes which are most influenced by the divergent response mode are visualized in Figure 12c (lower). This approach to clustering by temporal response given by mode 1 has identified genes which diverge from a fold change of 1.0 in a stable manner. Since the contribution of every mode to each gene's reconstruction is significant, it is seen that the genes identified by mode 1 indeed show oscillations, even though mode 1 is non-oscillatory. We refer to mode 2 as the unstable response mode because the dynamics of mode 2 are given by an eigenvalue that has magnitude only slightly larger than 1.0. Performing the temporal clustering using the unstable response mode identifies many genes that indeed exhibit unstable response to malathion induction. Just after malathion induction, these genes are clustered near fold change of 1.0, however, from 70 minutes on, their expression sharply rises or decays from 1.0. This unstable response mode also identifies genes which oscillate with a large frequency of oscillation. The final mode that is representative of the transcriptomic dataset is mode 6, the "no response" mode. We call it this because the genes which are clustered together by mode 6 do not show a response to malathion induction beyond noise. Mode 6 has small amplitude, therefore if a gene is well reconstructed by this mode, it cannot deviate far from fold change of 1.0.

The model of the transcriptomic dynamics response to a step input of malathion that we have learned using DMD has been shown to be interpretable, clustering genes with distinct temporal responses. To instill confidence in the model, we measure the accuracy of reconstruction using the coefficient of determination, R^2 as the metric. The R^2 is computed by feeding an initial condition (the gene expression at time $t = 0$) to the model and then predicting all the subsequent time points (an eight-step prediction) (Figure 12d). We emphasize that this is distinct from measuring model accuracy by computing a one-step prediction for each time point, which gives very little information about the dynamic process that has been captured. Our approach to measuring model accuracy fully captures the dynamic process with an R^2 of 0.93. This result implies that the set of 10 DMD modes, eigenvalues, and amplitudes are all that is necessary to describe the dynamics of the malathion response in the window of time that we have sampled the transcriptome. Because of the instabilities present in the model, we cannot accurately forecast the response further into the future. Figure 12d visualizing the DMD model predictions for five randomly selected genes in the transcriptome. These five genes each exhibit distinct responses to malathion, and each are very well captured by our DMD model. One key takeaway is that gene expression dynamics sampled at the resolution of minutes can be well approximated by a linear dynamical system, i.e. by a set of exponentially shrinking/growing modes. This allows us to apply linear systems techniques to design a target gene sampling [7].

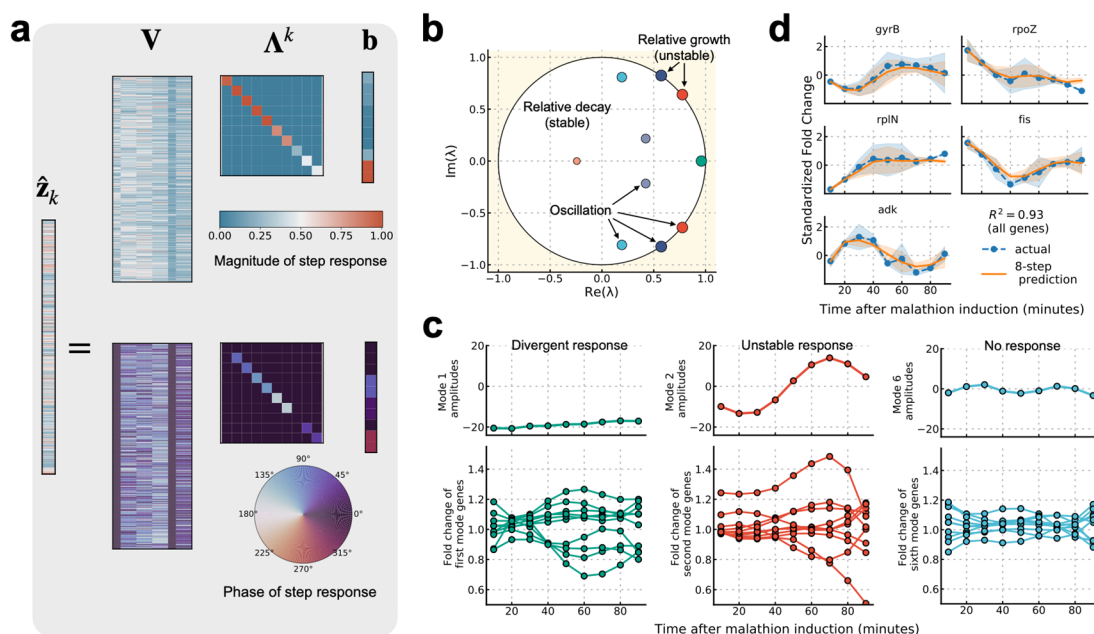


Figure 12: Dynamic mode decomposition provides an interpretable and predictive model of gene expression dynamics.

(a) DMD reconstruction of the fold change dynamics subject to a step input of malathion. (b) The DMD spectrum reveals the growth, decay, and oscillation of each of the 10 dynamic modes that comprise the transcriptomic dataset. (c) The amplitudes of modes 1, 2, and 6 are visualized (upper) along with the 10 genes whose dynamics are most impacted by each of the modes (lower). The color of each mode indicates which eigenvalue it corresponds with. (d) The eight-step prediction is visualized for five random genes in the transcriptomic dataset.

4.4.3 Observability analysis of gene expression.

Our aim in this section is to identify target genes that i) are sensitive to malathion induction, and ii) recapitulate the entire transcriptome. To do this, we make use of the concept of observability of a dynamical system [38]. The approach we take is to maximize the information content of the target genes, meaning that they alone can be used to describe what the entire transcriptome dynamics are. The details of our approach are described in the Methods section. Briefly, we maximize the observability of the dynamical system learned via DMD. This produces a sampling vector, $\mathbf{w}$ where the i th element of the vector is the relative sampling weight of information that is obtained by sampling gene i . We can then rank the genes by their weights in $\mathbf{w}$ to provide a machine learning ranked approach for extraction of transcriptional genetic sensors for malathion.

Figure 13a displays the mean of the sampling vector computed for the transcriptomic dataset and the learned DMD model. The error bars show the sample standard deviation of the sampling weights across 10 runs of the optimization routine to compute $\mathbf{w}$. The mean sampling weights have been normalized to take values between 0 and 1 and the height of each of the sampling weights correspond to the relative information obtained by sampling gene i . By relative information we mean that if $w_i > w_j$, then sampling gene i will allow for better reconstruction of the entire transcriptome than if gene j were sampled; this is quantified by observability. It can be seen that

many of the genes contribute insignificantly or not at all to the reconstruction of the transcriptome. The remaining genes have some significant contribution to the observability and their sampling weights are distributed from 0 to 1.

Figure 13b demonstrates how sampling more genes and for more time points, T , results in increasing reconstruction accuracy. The reconstruction accuracy is taken to be the Pearson correlation between the reconstructed initial condition, $\hat{\mathbf{z}}_0$, and the actual initial condition $\mathbf{z}_0$. The genes are sampled by rank, and five genes are sampled at once to evaluate the reconstruction accuracy. The five genes with maximum sampling weights are sampled first, then the next five with maximum sampling weights are added. We can see the trend, for all sampling time points, is that the more genes that are sampled, the higher the reconstruction accuracy. However, the reconstruction accuracy asymptotes at distinct accuracies for a given number of sampling times T . Sampling the top five genes and reconstructing the initial condition with a single time point results in a Pearson correlation of 0.1, whereas with 10 time points the Pearson correlation is 0.8. This significant result further motivates the problem of target gene selection. The ability to study the dynamics of a few genes with fine temporal resolution can greatly increase the knowledge of the entire system.

Using our method, we selected genes from the top (upregulated genes) and the bottom (downregulated genes) of the ranking to design genetic sensors with. The selected genes and their associated locus tag are given in Table 2 and their gene expression dynamics in transcripts per million (TPM) are shown in Figure 13c along with their percentage ranking (0 being the highest rank). As a test of our methodology, the genes were selected to coarsely cover the ranking space, selecting more genes with high ranking. Moreover, genes were rejected if they belonged to an operon where another gene in the operon was already selected. As we will explain in the next section, it is actually the promoter of the entire operon that we utilize for sensing. Of the 15 sensors, 12 are predicted to be activated by malathion (upregulated) and 3 are predicted to be repressed by malathion (downregulated).

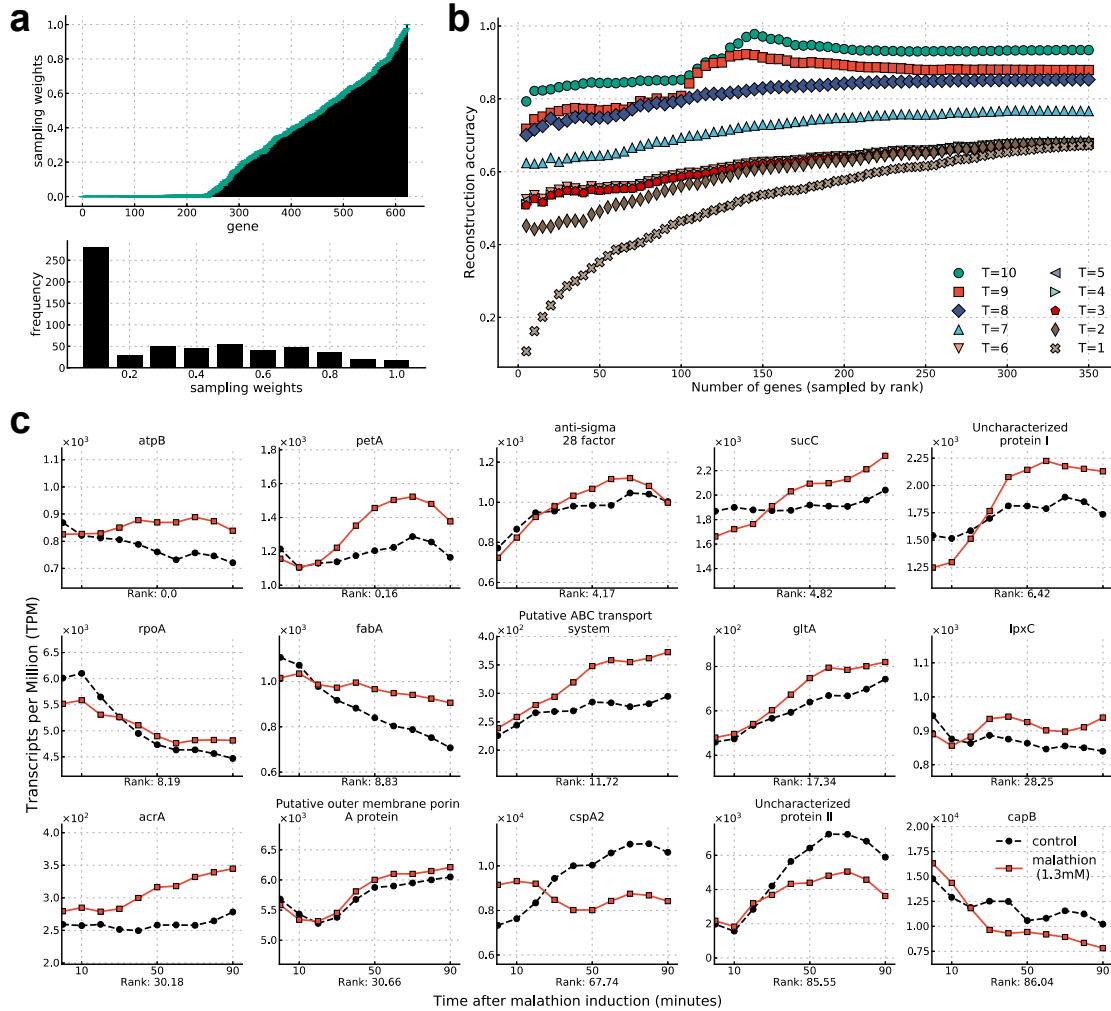


Figure 13: Gene sampling weights which maximize observability provide a machine learned ranking for extraction of genetic sensing elements.
 (a) The weights in the sampling vector w are sorted and plotted in the upper vignette. The error bars correspond to the sample standard deviation across 10 outcomes with random initializations of the optimization routine to obtain w . The lower vignette shows the counts of genes that are sampled with a weight between zero and one. (b) Reconstruction accuracy (Pearson correlation) between the estimated initial condition $\hat{z}_0$ and the actual z_0 is plotted for number of sampled time points $T=1$ to $T=10$. Each data point is obtained by sampling genes by rank (the amount sampled is given on the x-axis), generating outputs for T time points, and then estimating the initial condition. (c) The 15 genes selected from the proposed ranking (by contribution to observability). The label on each x-axis indicates the percentage rank (out of 623 genes) with zero being the highest rank.

There is a diverse set of molecular functions and biological processes that the selected sensor genes are involved in. Table 2 shows the molecular functions, obtained from the Gene Ontology (GO) database [24], for each of the selected genes. We find that the diversity of molecular functions among the genes is high.

Table 2: Sensor gene metadata and sensor transfer curve parameters.

Malathion reporter	Locus tag	Molecular function	Act./Rep.	y _{min}	y _{max}	K _M	n
atpB	PFLU_6124	• proton-transporting ATP synthase activity, rotational mechanism	Activated	1467	1783	0.6	4.5
petA	PFLU_0841	• 2 iron, 2 sulfur cluster binding, • metal ion binding • ubiquinol-cytochrome-c reductase activity	Activated	853	1125	1.4	2.4
anti-sigma 28 factor	PFLU_4736	• Negative regulator of flagellin synthesis	Activated	339	535	0.7	1.5
sucC	PFLU_1823	• ATP binding • magnesium ion binding • succinate-CoA ligase activity	Activated	257	337	0.4	21.6
Uncharacterized protein I	PFLU_3761		Activated	2465	3110	0.5	2.7
rpoA	PFLU_5502	• DNA binding • protein dimerization activity • DNA-directed 5'-3' RNA polymerase activity	Activated	1256	1542	0.9	3.0
fabA	PFLU_1836	• dehydratase activity • isomerase activity	Activated	292	373	0.2	1.1
Putative ABC transport protein	PFLU_0376	• ligand-gated ion channel activity	Activated	584	1083	1.0	2.0
gltA	PFLU_1815	• citrate (Si)-synthase activity	Activated	238	458	0.9	1.9
lpxC	PFLU_0953	• metal ion binding • deacetylase activity	Activated	1017	2418	0.4	8.7
acrA	PFLU_1380	• transmembrane transporter activity	Activated	354	682	0.9	2.9
Putative outer membrane porin A protein	PFLU_4612	• porin activity	Activated	642	1172	0.6	1.5
cspA2	PFLU_4150	• major cold shock protein	Activated	706	1186	1.5	5.3
capB	PFLU_1302A	• cold shock protein	Repressed	9616	10543	1.0	8.6
Uncharacterized protein II	PFLU_1358		Repressed	1073	3387	0.3	1.9

4.5 The Landing Site Recommender for Genomic Engineering of New Organisms

Plasmids and genomic inserts are the workhorses of genetic circuit delivery. Genomic inserts are reliably inherited. The cell's replication machinery ensures that all correctly inserted genetic circuits will be present in future generations. Depending on environmental stress factors and the compatibility of the insert, the cell would also ensure minimum mutation of the circuit. Thus, for predictable and robust operation of genetic circuits in both research and commercial settings, genomic insertion or recombineering is the way forward. The landing sites into which these insertions take place are crucial and need to be vetted carefully to minimize compositional context effects.

The genome is a highly mechanically active system. The time-scale of transcription is small. Depending on the phase of growth, thousands of genes are being transcribed. At any given time, they would be performing different transcriptional events like initiation, elongation, termination etc. Every transcriptional event changes the biophysics of the genome. Torsional events contribute to compositional contexts where the orientation of the genes can locally interfere with each other's operation. This is also true for synthetically introduced genetic circuits. The activity of the circuit will be influenced by the compositional context imposed by the neighbouring genes. To ensure circuits perform as designed, they need to be insulated correctly.

Also, the operation of the circuit would introduce a host burden on its own - both transcriptionally and mechanically. The compositional context introduced by the orientation of genes in the genetic circuit will influence the activity of neighbouring genes. This effect should also be minimized so that recombineering does not perturb the normal operation of the genome. The choice of landing sites for recombineering has been an open question for several years. Many efforts have been made trying to make landing site insertion orthogonal to genome activity. However, most efforts have been circuit-centric. Recombineering should take into account the effect on the genome as well as on the circuit. Some methods include inserting large pieces of insulating DNA before and after the genetic circuit as insulators.

As seen in the previous section, the choice of landing sites has not been biophysically informed. The mechanics of the genome plays an important role in gene transcription. RNA polymerase needs to unwind the double helix before initiation of transcription. The more relaxed the DNA is, the easier it is for transcription initiation. If the genome is too tightly wound, it might be energetically unfeasible for initiation to occur. Since torsion plays such a critical role in transcription, the cell has genes that actively monitor and regulate it. These are the topoisomerases and gyrases that wind and relax the DNA respectively. Thus, landing sites should be selected such that insertion of the circuit has minimal torsional perturbation locally and globally (see Figure 14).

Considering the DNA as a mechanical loop, every torsional event that occurs on its surface gets transmitted to a specific length scale of base pairs. Torsional events during transcription include initiation, elongation and termination. It can be argued that once the RNA polymerase completes transcription, it rewinds the helix as it detaches itself from the DNA. However, since DNA is not an ideally elastic material, some of the tension remains in the system. It is the build-up of these minor additions to tension that winds the helix. If tension continues to build up and is not relieved by the supercoiling regulatory proteins, recombineering into such a genome would result in sudden changes in genomic torsional stress. Hence, a criterion for selection of landing sites should be sites of minimum torsional tension.

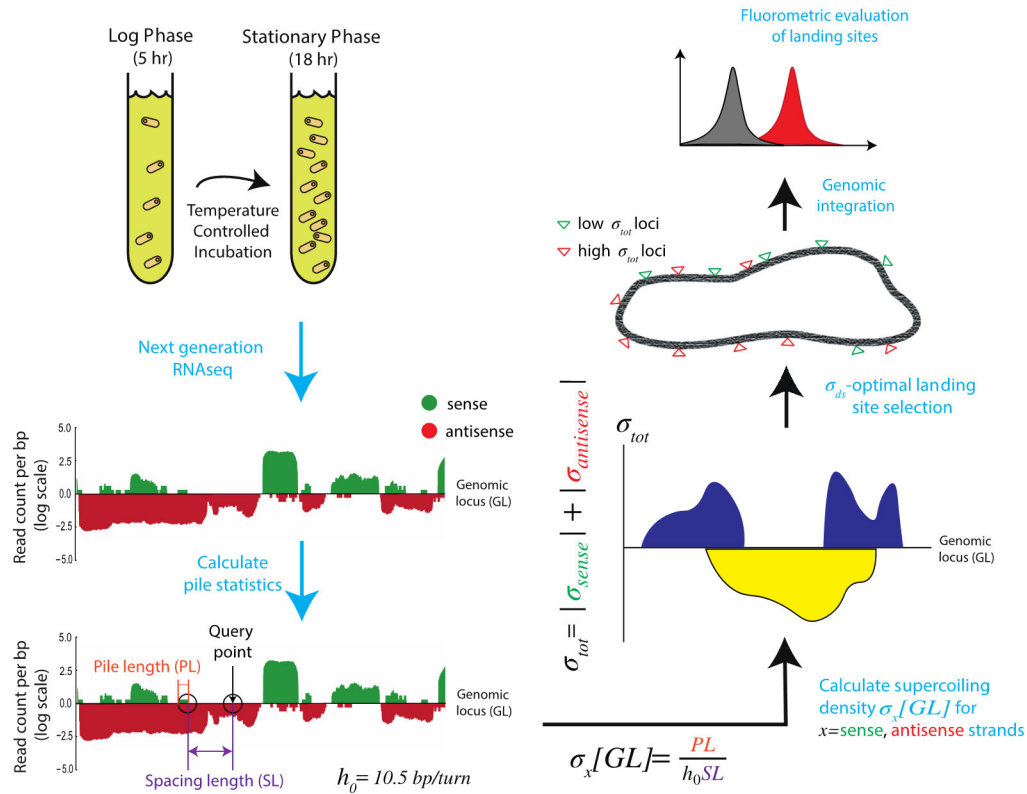


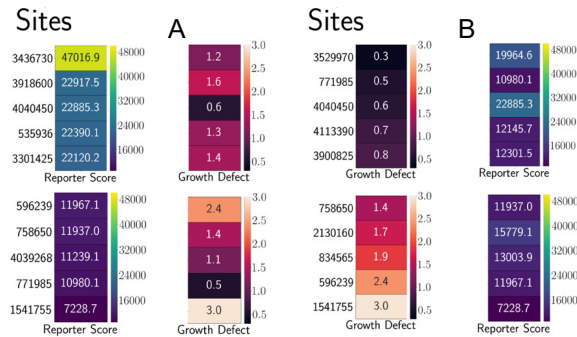
Figure 14: Biophysically informed landing site recommendation tool.

Strains isolated from the environment are cultured into log and stationary phases. After collecting the transcriptomic profile, the superhelical density was calculated for each basepair using the number of supercoils generated by a transcription pile and dividing it by the distance between the point on the transcriptional pile.

To test the activity of different landing sites, we selected 42 different landing sites for the bacterium *Bacillus subtilis* Marburg 168. The initial hypothesis for this selection was that superhelical density is an indicator of superhelical tension. We hypothesized that collecting next generation RNAseq data at both log phase and stationary phase growth conditions, with at least four to eight biological replicates, would reveal whether A) expression of our integrated report cassette was consistent and B) impact on the neighboring genes' transcriptional activity was significant. RNAseq data is typically analyzed using transcripts per million base pair reads (TPM) or fragments per million base pair reads (FPKM) analysis, namely normalized read count analysis of individual genes aligned to an existing reference genome.

We considered an analysis approach that was more fundamental, namely calculation and visualization of RNA transcript levels mapped to each locus on a reference genome. This approach benefits from being a data-driven approach, so that if an annotated genome is not available (e.g., when working with a novel chassis organism), we can still proceed to link supercoiling density predictions to observed outcomes. Further, this approach allows us to visualize long tracts of the contiguous RNA transcription, a phenomenon that is effectively visualized with transcriptomic data. Specifically, the transcriptomic data of the strain mapped to the sequenced genome of the same strain defines a pile diagram, where the read count for each base pair in the transcriptome is visualized contiguously from left to right. These pile diagrams can then be used to calculate superhelical density. The superhelical densities from both strands were calculated then as a ratio of the active transcriptional activity in a given region of the genome, normalized by the base pairs per turn of DNA in standard B-form double stranded DNA. Thus, we obtain a measure of superhelical density for both the sense and anti-sense strand of the genome. The sum of the absolute values of each of these densities is defined as a metric of genomic tension, a function that maps a basepair locus to an estimated aggregate genomic tension number. We evaluated this metric function across the entire genome and selected the top 20 and bottom 20 sites, where top sites were ranked for their low genomic tension metric and vice versa.

The final library prepared according to this metric contained 42 strains. These were characterized using time-series plate reader, flow cytometry and transcriptomic profiling for 21 strains, due to costs and bandwidth limitations of the Foundry lab. The results from these experiments are summarized in Figure 15. The data suggests that there is a wide bimodal distribution in the activity of the reporters at different sites. A subset of the strains exhibited high expression, visible even in bulk culture measurements under a spectrophotometer, while another subset exhibited a lower expression profile. The top expressing sites were almost twice as bright as the low expression strains. Remarkably, most genomically integrated strains do not produce enough fluorescent reporter to be detected under a spectrophotometer. For example, the same genetic cassette integrated at a standard site by the Foundry lab, did not produce any visible fluorescence and did not register a flow cytometry distribution discernible from background fluorescent noise of a wild-type strain. Our designed sites from Landing Site Recommender all produced salient fluorescence in both plate reader and flow cytometry experiments.



$$\text{Growth Defect} : \sqrt{\sum_{t=1}^{960} \left(\text{OD}_{t, \text{strain, avg}} - \text{OD}_{t, \text{NC, avg}} \right)^2}$$

The high expression strains (left) are almost two times as strong as the low expression strains. Both the reporter activity and growth defects need to be considered when selecting the landing site for insertion.

Approved for public release: distribution unlimited.

Sites Ranked by Genomic Perturbation

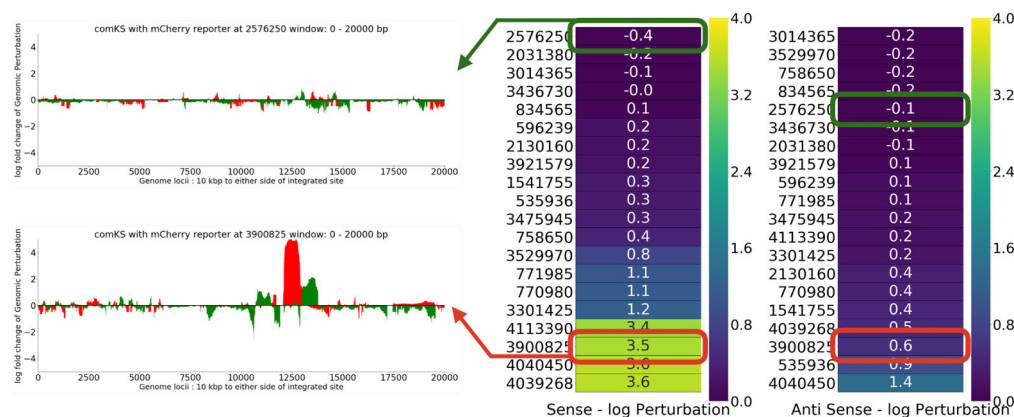


Figure 16: The genomic perturbation is calculated by calculating the 2-norm of the difference between the strain's transcriptomic profile and the wild-type across the entire genome. This represents the amount of change in the activity of the host genome upon insertion of the cassette. Ideal landing sites would impart very little genomic perturbation.

The implications of these findings are significant: the very act of genomic integration should be viewed as a perturbative force on the internal dynamics of a cell. Factors to consider in evaluating the quality of a novel genomic landing site are reporter activity in various culture conditions, the extent of the growth defect and the extent of perturbation to expression in neighboring (cis) and far-away (trans) genes. The work we have done here shows that some landing sites can be used to activate dormant gene expression in the genome, while others appear to have no effect. A key determinant function in predicting these outcomes is the modeled basal supercoiling density prior to and post-genomic integration. Transcriptional activity creates transient torsional stress in most situations, but in various loci across the genome, delicate balance of torsional stress propagation and sequestration can be disrupted clumsily by a genomic integration event, in the absence of landing site recommender models. The landing sites characterized here provide a standard resource for the synthetic biology community aiming to work in the *B. subtilis* type strain. More generally, our integrated experimental-computational approach provides a framework for onboarding and determining new, functional landing sites for genomic integration in novel host organisms.

5. Conclusions

In summary, our goal was to merge operator-theoretic methods with deep learning to automate synthetic biological circuit analysis and design. Our key insight was that deep neural networks acted as extraordinarily efficient representations of arbitrary function spaces, which could be used for describing the flow of dynamical systems. At the time, our preliminary results from Koopman operator theory, functional analysis, and dynamical systems suggested that many physical systems, including biological networks, could be described as a time-evolution of nonlinear functions on a defined manifold of physical variables.

In the first year of the program, we discovered that deep neural networks could be trained in real time on physical system data, to learn generalized representations of Koopman operators. We created an open-source code base on GitHub, using packages and functions developed in TensorFlow. We explored many example systems, including identification and analysis of the largest network system to date (over 100 states) with Koopman operator methods. Prior numerical methods for Koopman operator analysis were limited to planar or scalar systems. By casting Koopman operator learning problems as a deep learning problem, we immediately resolved the curse of dictionary explosion. We called this algorithm deep dynamic mode decomposition (deepDMD). Our first paper documenting deepDMD was posted on ArXiv in 2017, published in 2019 and has been cited 155 times since publication. This method of data-driven model discovery would form the foundation for closing the loop in design, build, test, learn cycles in the SD2 program.

A little while later in the program, we encountered an interesting challenge — many genetic circuits were being constructed with a reductionist's perspective, first realizing individual biological parts followed by assembly into larger, more complex circuits. Much of the data generated from intermediate, less complex pairwise or three-way part combinations was being discarded. This was no surprise, as this is also common practice, in the field of synthetic biology. Such control experiments are simply used to confirm that there are no issues with intermediate parts, but the dynamics of these parts does not inform models of the final, complex circuit realization.

In response to this challenge, we created structured dynamic mode decomposition, an algorithm that used Koopman operator theory and exploited the structure of reductionists experiments to quantify and discover emergent phenomena in genetic circuits as they scaled to increasing complexity. While resulting in a well-cited, peer reviewed publication, the greatest impact was to debug a series of NAND gates proposed by CELLO that exhibited strong toxicity effects. We found that one gene in the circuit, previously characterized as non-toxic, actually exhibited a unilateral pressure on the dynamics of up to four hundred and twenty genes, depending on the growth condition. Surprisingly, the strain exhibited no growth defect, but only in combination with other components did we observe emergent toxicity. This finding resulted in a new round of NAND2.0 gates, which avoided the use of this toxic protein, IcaR.

Over the course of the program, we realized that the scope of automated design for synthetic biological circuits was limited to digital logic design. We wondered if it would be possible for a researcher to simply draw on a whiteboard with the dynamics of a circuit they would like, while a

machine learning algorithm ingested the graphics of the drawing and made a suggestion for design. Such a design challenge emerged in the goal to build a hysteretic switch, proposed by SD2 collaborators from Duke University, Montana State University, and Rutgers University. We were given a hysteresis curve and asked to generate a circuit that would realize this network. We found two technical capability gaps in this process: A) the ability to identify appropriate circuit components, parts, and parametric realizations to meet a steady-state hysteresis design specification, B) the ability to iterate on a given design from data, to refine the parts and balance of gain in the circuit to converge to the target design specification.

Thus, we created the Koopman circuit optimization tool. The Koopman circuit optimization tool takes reference curves of dynamics or steady-state behavior to generate a reference circuit. The data from the actual, realized circuit may not look anything like the desired dynamics and this discrepancy in dynamics is fed to estimate a discrepancy Koopman operator, that pinpoints design revisions to the level of biological parts. We found in the design of a two-node switch, for example, that a double activator design was less optimal than an activator repressor design, when parametrically tuned away from oscillator behavior (the Hasty Stricker oscillator). This tool represents a new generation of synthetic biology tools, powered by machine learning, to automate the ideation process for novel biological circuits with unexplored behaviors. We can search over tens of thousands to hundreds of thousands of candidate-designs for a single behavioral constraint, drawn up by a user, in the course of several minutes.

These tools, combined with our Sensor Scraper and our Landing Site Recommender, enable rapid, data-driven iteration of circuit design to detect relevant analytes (the Sensor Scraper) and to engineer circuits that can function reliably on the genome and evade evolutionary escape. The Sensor Scraper has been able to automatically evaluate thousands of potential sensor promoters, in a matter of minutes, to find strong, responsive promoters to a target organophosphate compound. Likewise, the Landing Site Recommender searches over millions of potential landing sites on the genome, to find optimal landing sites which will result in high levels of circuit expression and minimal interference to nearby gene expression. These tools are now being adapted in a Department of Energy (DOE) program, led by Dr. Rob Egbert at Pacific Northwest National Laboratory, as well as by two emerging projects for the Army Research Office and the Institute of Collaborative Biotechnologies. They enable discovery of new biosensors for previously unexplored root exudate compounds, biochemical threats in soil and air, and rapid development of new field isolate microbes.

One major area of research, going forward, is how sequence maps to function. In regular conversations with Dr. Susan Buckout-White, we identified small datasets that would allow us to explore this problem, but the scope of the problem, at large, truly escaped the grasp of our SD2 program efforts. We feel a dedicated program centered on information innovation and sequence-to-function studies could be transformative for synthetic biology, systems biology, clinical biology, oncology, and many, many other fields. “Sequence-to-function” modeling is a catch all term, but even now, after SD2, it is unclear how to take a given sequence from the genome and predict which proteins bind to it, how strongly, and the dynamics of that promoter over the course of the cell cycle. This is a fundamental scientific knowledge gap that needs to be addressed, if we are to fully realize the vision of automated, biological design.

6. References

- [1] Guy Aidelberg, Benjamin D Towbin, Daphna Rothschild, Erez Dekel, Anat Bren, and Uri Alon. Hierarchy of non-glucose sugars in escherichia coli. *BMC systems biology*, 8(1):133, 2014.
- [2] WN Aldridge, JW Miles, DL Mount, and RD Verschoyle. The toxicological properties of impurities in malathion. *Archives of toxicology*, 42(2):95–106, 1978.
- [3] Uri Alon. *An introduction to systems biology: design principles of biological circuits*. CRCpress, 2019.
- [4] Orly Alter, Patrick O Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.
- [5] BrianDOAndersonandJohnBMoore. *Optimalfiltering*. CourierCorporation, 2012.
- [6] J Christopher Anderson, Elizabeth J Clarke, Adam P Arkin, and Christopher A Voigt. Environmentally controlled invasion of cancer cells by engineered bacteria. *Journal of molecular biology*, 355(4):619–627, 2006.
- [7] Milena Anguelova. *Observability and identifiability of nonlinear systems with applications in biology*. Chalmers Tekniska Hogskola (Sweden), 2007.
- [8] Shara Balakrishnan, Aqib Hasnain, Nibodh Boddupalli, Dennis M Joshy, Robert G Egbert, and Enoch Yeung. Prediction of fitness in bacteria with causal jump dynamic mode decomposition. In *2020 American Control Conference (ACC)*, pages 3749–3756. IEEE, 2020.
- [9] Yoshua Bengio and Olivier Delalleau. On the expressive power of deep architectures. In *International conference on algorithmic learning theory*, pages 18–36. Springer, 2011.
- [10] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *International conference on machine learning*, pages 552–560. PMLR, 2013.
- [11] Olivier Borkowski, Francesca Ceroni, Guy-Bart Stan, and Tom Ellis. Overloaded and stressed whole-cell considerations for bacterial synthetic biology. *Current opinion in microbiology*, 33:123–130, 2016.
- [12] Luc Bousse. Whole cell biosensors. *Sensors and Actuators B: Chemical*, 34(1-3):270–275, 1996.
- [13] Jennifer A N Brophy and Christopher A Voigt. Principles of genetic circuit design. *Nature methods*, 11(5):508–520, 2014.
- [14] Bing Na W Brunton, Lise A Johnson, Jeffrey G Ojemann, and J Nathan Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of neuroscience methods*, 258:1–15, 2016.

- [15] Steven L Brunton , Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- [16] Pablo Carbonell, Adrian J Jervis, Christopher J Robinson, Cun yu Yan, Mark Dunstan, Neil Swainston, Maria Vinaixa, Katherine A Hollywood, Andrew Currin, Nicholas JW Rattray, et al. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Communications biology*, 1(1):1–10, 2018.
- [17] Max Carbonell-Ballester, Eva Garcia-Ramallo, Raul Montanez, Carlos Rodriguez-Caso, and Javier Mac ía. Dealing with the genetic load in bacterial synthetic biology circuits: convergences with the ohm’s law. *Nucleic acids research*, 44(1):496–507, 2016.
- [18] S. Cardinale and A. Arkin. Contextualizing context for synthetic biology—identifying causes of failure of synthetic biological systems. *Biotechnology Journal*, 2012.
- [19] Francesca Ceroni, Rhys Algar, Guy-Bart Stan, and Tom Ellis. Quantifying cellular capacity identifies gene expression designs with reduced burden. *Nature methods*, 12(5):415–418, 2015.
- [20] Ye Chen, Shuyi Zhang, Eric M Young, Timothy S Jones, Douglas Densmore, and Christopher A Voigt. Genetic circuit design automation for yeast. *Nature Microbiology*, 5(11):1349–1360, 2020.
- [21] Donald J Chmielewski, Tasha Palmer, and Vasilios Manousiouthakis. On the theory of optimal sensor placement. *AIChE journal*, 48(5):1001–1012, 2002.
- [22] Huiqing Chong and Chi Bun Ching. Development of colorimetric-based whole-cell biosensor for organophosphorus compounds by engineering transcription regulator dmpr. *ACS synthetic biology*, 5(11):1290–1298, 2016.
- [23] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcze’sniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):1–19, 2016.
- [24] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004.
- [25] Natalie A Cookson, William H Mather, Tal Danino, Octavio Mondrag’ on-Palomino, Ruth J Williams, Lev S Tsimring, and Jeff Hasty. Queueing up for enzymatic processing: correlated signaling through coupled degradation. *Molecular systems biology*, 7(1):561, 2011.
- [26] Alan Costello and Ahmed H Badran. Synthetic biological circuits within an orthogonal central dogma. *Trends in biotechnology*, 39(1):59–71, 2021.
- [27] Bryan C Daniels and Ilya Nemenman. Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PloS one*, 10(3):e0119821, 2015.

- [28] Tal Danino, Arthur Prindle, Gabriel A Kwong, Matthew Skalak, Howard Li, Kaitlin Allen, Jeff Hasty, and Sangeeta N Bhatia. Programmable probiotics for detection of cancer in urine. *Science translational medicine*, 7(289):289ra84–289ra84, 2015.
- [29] I Desi, Gy Dura, L Gonczi, Zs Kneffel, A Strohmayer, and Z Szabo. Toxicity of malathion to mammals, aquatic organisms and tissue culture cells. *Archives of environmental contamination and toxicology*, 3(4):410–425, 1975.
- [30] Peter Flessel, PJE Quintana, and Kim Hooper. Genetic toxicity of malathion: a review. *Environmental and molecular mutagenesis*, 22(1):7–17, 1993.
- [31] Paul Francois and Vincent Hakim. Design of genetic networks with specified functions by evolution in silico. *Proceedings of the National Academy of Sciences*, 101(2):580–585, 2004.
- [32] Timothy S Gardner, Charles R Cantor, and James J Collins. Construction of a genetic toggle switch in *escherichia coli*. *Nature*, 403(6767):339–342, 2000.
- [33] Jorge Goncalves and Sean Warnick. Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks. *IEEE Transactions on Automatic Control*, 53(7):1670–1674, 2008.
- [34] Thomas E Gorochofski, Irem Avcilar-Kucukgoze, Roel AL Bovenberg, Johannes A Roubos, and Zoya Ignatova. *ACS synthetic biology*, 5(7):710–720, 2016.
- [35] Andras Gyorgy, Jose I Jimenez, John Yazbek, Hsin-Ho Huang, Hattie Chung, Ron Weiss, and Domitilla Del Vecchio. Isocost lines describe the cellular economy of genetic circuits. *Biophysical journal*, 109(3):639–646, 2015.
- [36] Aqib Hasnain, Nibodh Boddupalli, and Enoch Yeung. Optimal reporter placement in sparsely measured genetic networks using the koopman operator. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 19–24. IEEE, 2019.
- [37] Aqib Hasnain, Subhrajit Sinha, Yuval Dorfan, Amin Espah Borujeni, Yongjin Park, Paul Maschhoff, Uma Saxena, Joshua Urrutia, Niall Gaffney, Diveena Becker, et al. A data-driven method for quantifying the impact of a genetic circuit on its host. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2019.
- [38] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.
- [39] Ayaan Hossain, Eriberto Lopez, Sean M Halper, Daniel P Cetnar, Alexander C Reis, Devin Strickland, Eric Klavins, and Howard M Salis. Automated design of thousands of nonrepetitive parts for engineering stable genetic systems. *Nature biotechnology*, 38(12):1466–1475, 2020.
- [40] Robert W Jackson, Gail M Preston, and Paul B Rainey. Genetic characterization of *pseudomonas fluorescens sbw25* *rsp* gene expression in the phytosphere and in vitro. *Journal of bacteriology*, 187(24):8477–8488, 2005.

- [41] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315, 1931.
- [42] Krzysztof Kowalski and W-H Steeb. *Nonlinear dynamical systems and Carleman linearization*. World Scientific, 1991.
- [43] E Kreindler and P Sarachik. On the concepts of controllability and observability of linear systems. *IEEE Transactions on Automatic Control*, 9(2):129–136, 1964.
- [44] Brigitta Kurenbach, Delphine Marjoshi, Carlos F Am´abile-Cuevas, Gayle C Ferguson, William Godsoe, Paddy Gibson, and Jack A Heinemann. Sublethal exposure to commercial formulations of the herbicides dicamba, 2, 4-dichlorophenoxyacetic acid, and glyphosate cause changes in antibiotic susceptibility in *escherichia coli* and *salmonella enterica* serovar typhimurium. *MBio*, 6(2):e00009–15, 2015.
- [45] Travis L La Fleur, Ayaan Hossain, and HowardMSalis. Automated model-predictive design of synthetic promoters to control transcriptional profiles in bacteria. *bioRxiv*, 2021.
- [46] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with *deseq2*. *Genome biology*, 15(12):1–21, 2014.
- [47] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.
- [48] Mario A Marchisio and Jorg Stelling. Automatic design of digital synthetic gene circuits. *PLoS computational biology*, 7(2):e1001083, 2011.
- [49] Adam J Meyer, Thomas H Segall-Shapiro, Emerson Glassey, Jing Zhang, and Christopher A Voigt. *Escherichia coli* “marionette” strains with 12 highly optimized small-molecule sensors. *Nature chemical biology*, 15(2):196–204, 2019.
- [50] Igor Mezic. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.
- [51] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. Bionumbers: the database of key numbers in molecular and cell biology. *Nucleic acids research*, 38(suppl 1):D750–D753, 2010.
- [52] Goksel Mısırlı, Jennifer Hallinan, Matthew Pocock, Phillip Lord, James Alastair McLaughlin, Herbert Sauro, and Anil Wipat. Data integration and mining for synthetic biology design. *ACS synthetic biology*, 5(10):1086–1097, 2016.
- [53] Michael Moraskie, Harun Roshid, Gregory O’Connor, Emre Dikici, Jean-Marc Zingg, Sapna Deo, and Sylvia Daunert. Microbial whole-cell biosensors: Current applications, challenges, and future perspectives. *Biosensors and Bioelectronics*, pg. 113359, 2021.
- [54] Kay Lynn Newhart. Environmental fate of malathion. California Environmental Protection Agency, 11:1–20, 2006.

- [55] Alec AK Nielsen, Bryan S Der, Jonghyeon Shin, Prashant Vaidyanathan, Vanya Paralanov, Elizabeth A Strychalski, David Ross, Douglas Densmore, and Christopher A Voigt. Genetic circuit design automation. *Science*, 352(6281), 2016.
- [56] Axel Nystrom, Antonis Papachristodoulou, and Andrew Angel. A dynamic model of resource allocation in response to the presence of a synthetic construct. *ACS synthetic biology*, 7(5):1201–1210, 2018.
- [57] Paul Opgenorth, Zak Costello, Takuya Okada, Garima Goyal, Yan Chen, Jennifer Gin, Veronica Benites, Markus de Raad, Trent R Northen, Kai Deng, et al. Lessons from two design–build–test–learn cycles of dodecanol production in *Escherichia coli* aided by machine learning. *ACS synthetic biology*, 8(6):1337–1351, 2019.
- [58] Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.
- [59] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.
- [60] Yili Qian, Hsin-Ho Huang, Jose I Jimenez, and Domitilla Del Vecchio. Resource competition shapes the response of genetic circuits. *ACS synthetic biology*, 6(7):1263–1272, 2017.
- [61] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *international conference on machine learning*, pages 2847–2854. PMLR, 2017.
- [62] Jonathan M Raser and Erin K O’shea. Noise in gene expression: origins, consequences, and control. *Science*, 309(5743):2010–2013, 2005.
- [63] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.
- [64] Yannick Rondelez. Competition for catalytic resources alters biological network dynamics. *Physical review letters*, 108(1):018102, 2012.
- [65] Clarence W. Rowley, Igor Mezic, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson. *Journal of Fluid Mechanics*, 641:115, 2009.
- [66] Clarence W Rowley, Igor Mezic, Shervin Bagheri, Philipp Schlatter, Dans Henningson, et al. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641(1):115–127, 2009.
- [67] Howard Salis, Alvin Tamsir, and Christopher Voigt. Engineering bacterial signals and sensors. *Bacterial Sensing and Signaling*, 16:194–225, 2009.
- [68] Tobias Schladt, Nicolai Engelmann, Erik Kubaczka, Christian Hochberger, and Heinz Koeppl. Automated design of robust genetic circuits: Structural variants and parameter uncertainty. *ACS Synthetic Biology*, 2021.

- [69] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- [70] Nicholas J Schurch, Piet’a Schofield, Marek Gierliński, Christian Cole, Alexander Sherstnev, Vijender Singh, Nicola Wrobel, Karim Gharbi, Gordon G Simpson, Tom Owen-Hughes, et al. How many biological replicates are needed in an rna-seq experiment and which differential expression tool should you use? *Rna*, 22(6):839–851, 2016.
- [71] Rafael Silva-Rocha and V’ictor de Lorenzo. Mining logic gates in prokaryotic transcriptional regulation networks. *FEBS letters*, 582(8):1237–1244, 2008.
- [72] Subhrajit Sinha, Umesh Vaidya, and Enoch Yeung. On computation of koopman operator from sparse data. In *2019 American Control Conference (ACC)*, pages 5519–5524. IEEE, 2019.
- [73] Lawrence Sirovich. A novel analysis of gene array data: yeast cell cycle. *Biology Methods and Protocols*, 5(1):bpaa018, 2020.
- [74] Yizhi Song, Cordelia PN Rampley, Xiaoyu Chen, Fawen Du, Ian P Thompson, and Wei E Huang. Application of bacterial whole-cell biosensors in health. *Handbook of Cell Biosensors*, pages 945–961, 2022.
- [75] Brynne C Stanton, Alec AK Nielsen, Alvin Tamsir, Kevin Clancy, Todd Peterson, and Christopher A Voigt. Genomic mining of prokaryotic repressors for orthogonal logic gates. *Nature Chemical Biology*, 10(2):99–105, 2014.
- [76] Muhammad Syafrudin, Risky Ayu Kristanti, Adhi Yuniarto, Tony Hadibarata, Jongtae Rhee, Wedad A Al-Onazi, Tahani Saad Algarni, Abdulhadi H Almarri, and Amal M Al-Mohaimed. Pesticides in drinking water—a review. *International Journal of Environmental Research and Public Health*, 18(2):468, 2021.
- [77] Mao Taketani, Jianbo Zhang, Shuyi Zhang, Alexander J Triassi, Yu-Ja Huang, Linda G Griffith, and Christopher A Voigt. Genetic circuit design automation for the gut resident species *bacteroides thetaiotaomicron*. *Nature biotechnology*, 38(8):962–969, 2020.
- [78] Justin Tan, Anand V Sastry, Karoline S Fremming, Sara P Bjørn, Alexandra Hoffmeyer, Sangwoo Seo, Bjørn G Voldborg, and Bernhard O Palsson. Independent component analysis of *e. coli*’s transcriptome reveals the cellular processes that respond to heterologous gene expression. *Metabolic Engineering*, 61:360–368, 2020.
- [79] Jake P Taylor-King, Asbjørn N Riseth, Will Macnair, and Manfred Claassen. Dynamic distribution decomposition for single cell snapshot time series identifies subpopulations and trajectories during ipsc reprogramming. *PLoS computational biology*, 16(1):e1007491, 2020.
- [80] Dennis Tu, Jiwon Lee, Taylan Ozdere, Tae Jun Lee, and Lingchong You. *Engineering gene circuits: foundations and applications*. 2007.
- [81] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

- [82] Sipko Van Dam, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pedro de Magalhaes. Gene co-expression analysis for functional classification and gene–disease predictions. *Briefings in bioinformatics*, 19(4):575–592, 2018.
- [83] Eric Van Arsdale, Chen-yu Tsao, Yi Liu, Chen-yu Chen, Gregory F Payne, and William E Bentley. Redox-based synthetic biology enables electrochemical detection of the herbicides dicamba and roundup via rewired escherichia coli. *ACS sensors*, 4(5):1180–1184, 2019.
- [84] Christopher A Voigt. Genetic parts to program bacteria. *Current opinion in biotechnology*, 17(5):548–557, 2006.
- [85] XinyiWan, Behide Saltepe, Luyang Yu, and Baojun Wang. Programming living sensors for environment, health and biomanufacturing. *Microbial biotechnology*, 2021.
- [86] Baojun Wang, Mauricio Barahona, and Martin Buck. A modular cell-based biosensor using engineered genetic logic circuits to detect and integrate multiple environmental signals. *Biosensors and Bioelectronics*, 40(1):368–376, 2013.
- [87] Christopher M Whitford, Pablo Cruz-Morales, Jay D Keasling, and Tilmann Weber. The design-build-test-learn cycle for metabolic engineering of streptomyces. *Essays in Biochemistry*, 2021.
- [88] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.
- [89] Jewell D Wilson. Toxicological profile for malathion. Agency for Toxic Substances and Disease Registry, 2003.
- [90] Enoch Yeung, Jongmin Kim, Ye Yuan, Jorge Goncalves, and Richard Murray. Quantifying crosstalk in biochemical systems. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pages 5528–5535. IEEE, 2012.
- [91] Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In 2019 American Control Conference (ACC), pages 4832–4839. IEEE, 2019.
- [92] Yang-Chun Yong and Jian-Jiang Zhong. A genetically engineered whole-cell pigment-based bacterial biosensing system for quantification of n-butyl homoserine lactone quorum sensing signal. *Biosensors and Bioelectronics*, 25(1):41–47, 2009.

List of Symbols, Abbreviations, and Acronyms

ATC	anhydrotetracycline
cAMP	cyclic adenosine monophosphate receptor
cDeepDMD	Deep Dynamic Mode Decomposition with control
cDeepKCOT	Deep learning driven Koopman Circuit Optimization Tool
CELLO	Cellular Logic
cRELU	Complementary Rectified Linear Unit
CRP	catabolite receptor protein
CVXPY	convex python optimization package
DARPA	Defense Advanced Research Projects Agency
DBTL	Design, Build, Test, and Learn Cycle
DeepDMD	Deep Dynamic Mode Decomposition
DMD	Dynamic Mode Decomposition
DNA	Deoxyribo-Nucleic Acid
DOE	Department of Energy
dsGRN	Database of Signatures of Gene Regulatory Networks
EDMD	Extended Dynamic Mode Decomposition
ELU	Exponential Linear Unit
FPKM	Fragments per Kilobase of Exon per Million Base Pair Reads
GFP	Green Fluorescent Protein
GO	Gene Ontology
GPU	Graphical Processing Unit
GUI	Graphical User Interface
IPTG	Isopropyl-beta-Thio-Galactoside
LSTM	Long-Short Term Memory Module
NAND	not an acronym
OD	optical density
ODE	Ordinary Differential Equation
POLS	Partially Observed Linear System
RBS	Ribosome Binding Site
RFP	Red Fluorescent Protein
RELU	Rectified Linear Unit
RNA	Ribonucleic acid
RNAseq	Next Generation RNA Sequencing
SD2	Synergistic Discovery & Design Program
TetR	Tet Regulator
TPM	Transcripts per Millions of Base Pair Reads
UCF	User Constraint File