PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Data-driven model discovery and model selection for noisy biological systems

Xiaojun Wu, MeiLu McDermott, Adam L. MacLean*

Department of Quantitative and Computational Biology, University of Southern California, Los Angeles, California, United States of America

* macleana@usc.edu

OPEN ACCESS

Citation: Wu X, McDermott M, MacLean AL (2025) Data-driven model discovery and model selection for noisy biological systems. PLoS Comput Biol 21(1): e1012762. https://doi.org/10.1371/journal.pcbi.1012762

Editor: Mark Alber, University of California Riverside, UNITED STATES OF AMERICA

Received: June 24, 2024

Accepted: December 31, 2024

Published: January 21, 2025

Data availability statement: All code associated with this study are available on GitHub at: github.com/maclean-lab/model-discovery. All raw data used in this study are publicly available on the Gene Expression Omnibus (GSE147405).

Funding: A.L.M. acknowledges support from the National Institutes of Health (R35GM143019) and from the National Science Foundation (DMS 2045327). Funders played no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Biological systems exhibit complex dynamics that differential equations can often adeptly represent. Ordinary differential equation models are widespread; until recently their construction has required extensive prior knowledge of the system. Machine learning methods offer alternative means of model construction: differential equation models can be learnt from data via model discovery using sparse identification of nonlinear dynamics (SINDy). However, SINDy struggles with realistic levels of biological noise and is limited in its ability to incorporate prior knowledge of the system. We propose a data-driven framework for model discovery and model selection using hybrid dynamical systems: partial models containing missing terms. Neural networks are used to approximate the unknown dynamics of a system, enabling the denoising of the data while simultaneously learning the latent dynamics. Simulations from the fitted neural network are then used to infer models using sparse regression. We show, via model selection, that model discovery using hybrid dynamical systems outperforms alternative approaches. We find it possible to infer models correctly up to high levels of biological noise of different types. We demonstrate the potential to learn models from sparse, noisy data in application to a canonical cell state transition using data derived from single-cell transcriptomics. Overall, this approach provides a practical framework for model discovery in biology in cases where data are noisy and sparse, of particular utility when the underlying biological mechanisms are partially but incompletely known.

Introduction

Mathematical models wielded skillfully can offer great insight into biological systems. The process of constructing models, however, is typically manual and labor-intensive. Data-driven model discovery, i.e. methods by which models can be learnt directly from data, offer a promising alternative to manual model-building. To perform such model discovery however, one must overcome the idiosyncrasies that biological systems present, including appropriate consideration of the extent/type of noise present in the data, and the need to evaluate results in an unbiased way.

Noise is pervasive in biological systems. Technical and biological (intrinsic and extrinsic) sources of noise must be taken into account for the accurate quantification and simulation of

Competing interests: The authors have declared that no competing interests exist.

biological dynamics. To do so, one can model a stochastic dynamical system starting from a discrete Markov process (stochastic reaction network) or a Chemical Langevin equation [1]. While treatment of the stochastic dynamics is at times necessary, models described by ordinary differential equations (ODEs), capturing the mean behavior of the underlying dynamics can be (unreasonably) effective at describing the dynamics of biological systems across many scales. A wide suite of tools readily available for the analysis of ODE models extends their utility. Here we focus on ODE models for biological systems, taking the general form $x' \equiv dx/dt = f(x(t))$, where $x = x(t)$ is the state vector of species at time t. We will use model discovery to learn ODEs for biological models, i.e. we will treat the biological noise as external to the system dynamics.

The indefatigable growth of biological datasets in size and scope offers fertile ground for the integration of machine learning and dynamical systems approaches to learn new biology [2]. Data-driven model discovery methods offer means by which model equations can be inferred directly from data, such as via sparse identification of nonlinear dynamics (SINDy) [3]. SINDy assumes that the right-hand-side of a model (e.g. $f(x(t))$ of an ODE model) can be expressed as a product of basis functions and a coefficient matrix. The goal of SINDy is then to learn the (sparse) coefficient matrix that best fits the data via regularized regression. A basis that defines the set of possible terms in the model must be constructed prior to regression. Often such bases are comprised of polynomial functions, but SINDy can also discover models containing non-polynomial terms, either by assuming that the entire right-hand side is a fraction of two polynomials [4], or simply by including bases like Hill equations [5], which are commonly found in biological networks. SINDy, and its extensions [6,7], enable model discovery from a large space of possible models; previous methods able to explore topological model spaces or model properties typically operated over smaller domains [8,9]. SINDy has been successful in a wide variety of contexts from modeling CAR-T cell therapy [10] to epidemiology [11]. Model discovery approaches extend beyond differential equation-based models as recent works combining model discovery with discrete/agent-based modeling have shown [12,13]. Recent works have also demonstrated the potential for model discovery on noisy data [14–19], although inferring models in the presence of noise remains challenging. As an alternative to sparse regression, there are also model discovery methods that use symbolic regression, providing a bottom-up approach for building symbolic mathematical expressions that become the equations of an ODE model [20,21].

In some cases, partial knowledge of a biological system is available a priori. For example: the birth or death terms of a biological species might be known, but the interaction terms for that species might be unknown. Partial knowledge can be incorporated into the model discovery framework by splitting $f(x(t))$ into a known and an unknown part, the latter of which is approximated by a neural network (NN) [22]. We refer to such model as hybrid dynamical systems. Building upon recent advances that combine differential equation- and NN-based approaches for machine learning [23–25], several recent works have demonstrated the potential for model discovery with neural ODEs [22,26–29]. Rackauckas et al. [22] developed "universal differential equations," a model discovery approach using hybrid dynamical systems with sparse regression to infer the unknown terms of a partially known model. Here, an NN within the hybrid dynamical system is trained on data via an appropriate objective function, and subsequently an ODE model is inferred using sparse regression with the derivatives estimated by the fitted hybrid dynamical model. This approach can successfully infer correct models using limited input data—short trajectories, albeit richly sampled in time. Inferring correct models however becomes difficult as the noise grows. Here, we seek means with which to evaluate models inferred using hybrid dynamical systems in an unbiased way, given the typical unavailability of ground truth for biological systems of interest.

In this paper, following methods introduced in [22], we develop methods for data-driven model discovery and model selection of complex, noisy biological dynamics. We show that we are able to learn models from noisy biological data using only short time spans for training, i.e. when species have not reached steady state. Moreover, we show how incorporating partial knowledge into the inference framework facilitates model discovery. To do so we apply a two-step model discovery framework. First, we fit the unknown dynamics using a neural network for smoothing and interpolation. Second, we use the trained neural network as input to SINDy-like sparse regression to learn symbolic model terms. At both steps we perform model selection to search over hyperparameter space with unbiased evaluation criteria. With application to synthetic noisy data from two canonical models (Lotka-Volterra [30,31] and the repressilator [32]) we demonstrate the advantages of hybrid dynamical systems for model discovery with complex data. We also apply these methods to a single-cell RNA-sequencing dataset describing a epithelial-mesenchymal transition (EMT) [33], and demonstrate that we can learn models from real data.

Materials and methods

Overview of inference framework

We seek to learn ordinary differential equation (ODE) models that govern the behavior of a dynamical system using data-driven model discovery. We define a hybrid dynamical system below as one for which the model is partially but not completely known. We take a two-step approach to model discovery. In the first step, we use neural network (NN) based approaches to learn the unknown derivatives of a hybrid dynamical system. In the second, we use sparse regression to infer the missing terms in a model using the derivatives inferred in the NN step. At both steps we employ model selection in the hyperparameter space; enabling the unbiased assessment of predictions when the ground truth is unknown. Finally, we describe the methods involved in the processing and analysis of single-cell RNA sequencing data with which we will infer models of cell state transition dynamics using our two-step approach.

Below, we use lowercase notation for variables (e.g. x) and uppercase for data (e.g. X). Subscripts denote the indices of multidimensional variables, data, or functions, e.g. x_i (for variable x) or $g_i(x)$ (for function $g(x)$); with the exception of x_0, which denotes the initial conditions. Superscripts in parentheses denote labels for data. E.g. $X^{(i)}$ denotes the ith sample of a dataset; $X^{(\text{true})}$ denotes ground truth data.

Formulation of a hybrid dynamical system

A dynamical system is quantified by n noisy observations $X = \left\{X^{(1)}, \ldots, X^{(n)}\right\}$ sampled on a time span $[0, T]$ (Fig 1A). A hybrid dynamical system is defined as an ODE system of the form:

$$x' = f(x) = g(x) + \mathrm{NN}(x) \tag{1}$$

where $g(x)$ is a closed-form function and $\mathrm{NN}(x)$ is a neural network. Note that other implementations of dynamical systems inference also use "hybrid" to describe their formulation [11,34,35], but these uses are distinct to that which we present here.

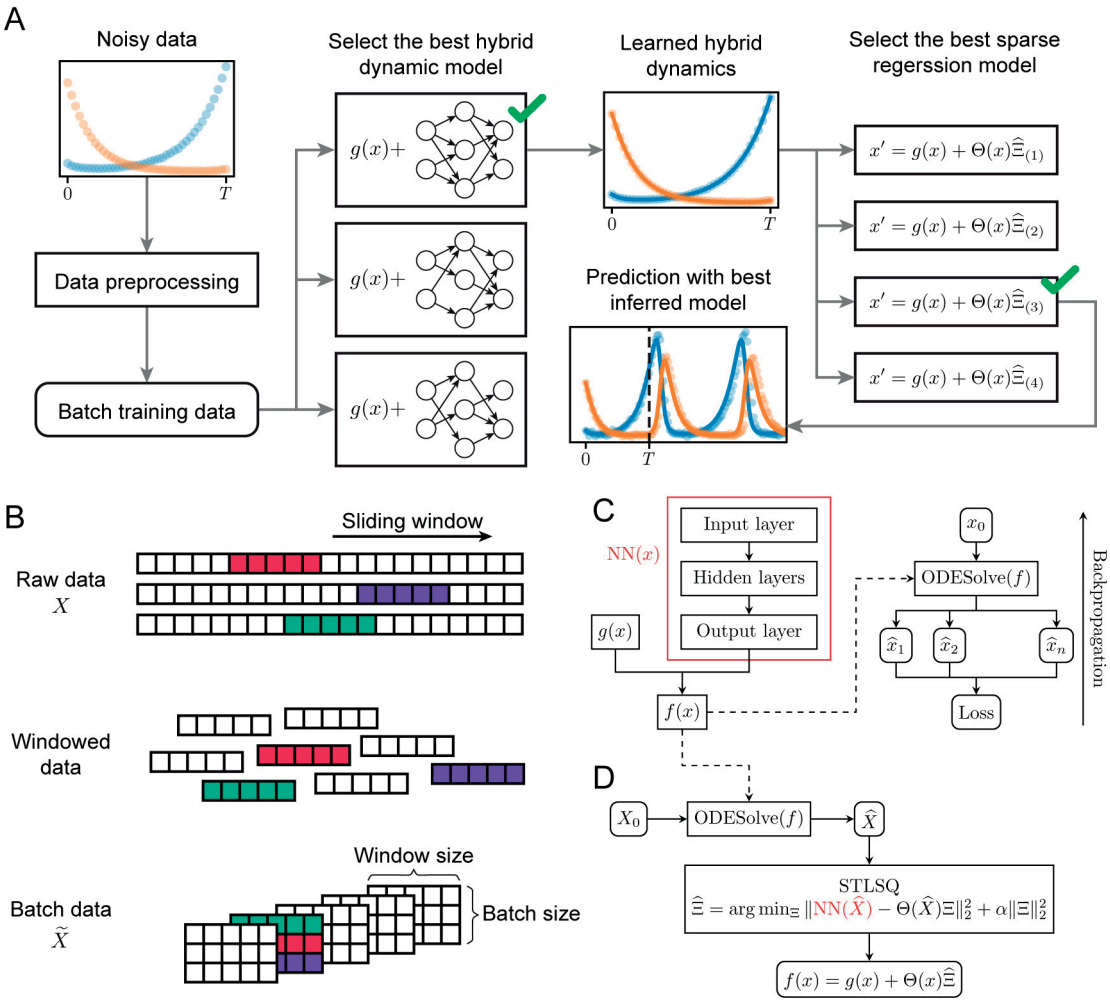

Fig 1. Pipeline for data-driven model discovery for dynamical systems with model selection. A. Overview of the pipeline. Raw data is converted to batch training data for learning hybrid dynamical models using different hyperparameters. Dynamics approximated by the best hybrid dynamical model are used for inferring ODE models with sparse regression. Inferred ODE models are evaluated on model fit and extrapolation. **B.** Each sample in the raw training data is split using a sliding window to obtain samples on shorter time spans. These short samples are assembled into batches for training. **C.** A hybrid dynamical model is trained using $\mathrm{NN}(x)$ by simulating over the training time span and backpropagating, using the loss between simulated data and training data. **D.** The trained hybrid dynamical model is input to sparse regression for equation-learning via the STLSQ sparse regression algorithm, and the final inferred model is output.

https://doi.org/10.1371/journal.pcbi.1012762.g001

Data generation

For a model with known ODE system $x' = f(x)$, we generate synthetic datasets comprised of samples for training, validation, and testing. We first simulate the model from its initial conditions, x_0, (using the LSODA method from SciPy) and obtain a noise-free time series $X^{(\mathrm{true})}$ on $[0, T]$ with step size Δt [36]. Noisy observations $X^{(i)}$ are then generated according to the specified type of noise (additive or multiplicative) and the specified noise level ϵ. We note that we are considering sources of extrinsic (measurement) noise here, and not the intrinsic sources of noise that are also at play. In the case of additive noise, we generate observations for state variable k at time t_j from: $X^{(i)}_{j,k} \sim \mathcal{N}\left(X^{(\mathrm{true})}_{j,k}, \epsilon \overline{X^{(\mathrm{true})}_{:,k}}\right)$, where $\overline{X^{(\mathrm{true})}_{:,k}}$ is the mean of state k for

$X^{(\text{true})}$. That is, variation in the magnitude of the noise is constant across time points for the state k. In the case of multiplicative noise, we generate observations for state variable k from: $X^{(i)}_{j,k} \sim \mathcal{N}\left(X^{(\text{true})}_{j,k}, \epsilon X^{(\text{true})}_{j,k}\right)$. The magnitude of the noise in the observation of $X^{(i)}_{j,k}$ depends on the value of $X^{(\text{true})}_{j,k}$ at time t_j only, i.e. we do not consider time-correlated noise measurements.

Data preprocessing

We apply gradient-based optimization approaches to batches of noisy data to facilitate learning. Raw data from the training set $X = \left\{X^{(1)}, \ldots, X^{(n)}\right\}$ are converted into shorter time series by taking a sliding window through each $X^{(i)}$ with a window of size w and a step size of 1 (Fig 1B). These short time series data are then shuffled and arranged into batches of size b for training (Fig 1B). We will denote the batch data as $\widetilde{X}$.

NN-based approaches to fit the unknown dynamics of a hybrid dynamical system

Given a hybrid dynamical system (Eq 1), following the approach taken by Rackauckas et al. in [22], we learn a neural network (NN) approximator $\text{NN}(x)$ for the unknown dynamics of the system using the preprocessed training samples $\widetilde{X}$ (Fig 1C). For each sample $\widetilde{X}^{(i)}$ on time points $\left\{t_j, \ldots, t_{j+w-1}\right\}$ in a batch, we simulate $f(x)$ from initial condition $\widetilde{X}^{(i)}_{j,\cdot}$ to obtain the predicted dynamics $\widehat{X}^{(i)}$ over this time range. The mean squared error (MSE) between $\widetilde{X}^{(i)}$ and $\widehat{X}^{(i)}$ is defined as the sample loss, and the batch loss, $\mathcal{L}$, is defined as the mean of all the sample losses in one batch. The gradient of $\mathcal{L}$ with respect to the weights of $\text{NN}(x)$ is then computed by backpropagation.

The NN approximator is implemented in PyTorch [37]. For numerical integration we use the `Dopri5` integrator as it is implemented in the torchode package [38]. For gradient descent we use the Adam algorithm to update the weights of $\text{NN}(x)$ from the gradient of $\mathcal{L}$ [39].

Sparse regression for model discovery from a hybrid dynamical system

We use a framework modified from SINDy [3] to infer the dynamics $f(x)$ of an ODE system that is partially known (Fig 1D). First, given $\text{NN}(x)$ trained above, we simulate the dynamics $f(x) = g(x) + \text{NN}(x)$ with initial conditions taken from the training samples. By simulating over the same time span as for training, $[0, T]$, with step size Δt (which can provide finer temporal resolution than in training), we obtain trajectories $\widehat{X}$ for input to sparse regression. We then select a library of basis functions, $\Theta(x)$. The basis can contain polynomial as well as non-polynomial terms, as we will show in the examples below. We find a sparse matrix $\widehat{\Xi}$ that best fits the regression problem specified by: $\text{NN}(\widehat{X}) = \Theta(\widehat{X})\Xi$. We use an implementation of the SINDy optimization algorithm Sequentially Thresholded Least Squares (STLSQ) in PySINDy that permits multi-sample input to determine $\widehat{\Xi}$. In brief, STLSQ iteratively performs ridge regression on the basis functions in $\Theta(x)$ with nonzero coefficients, setting coefficients to 0 if they drop below a threshold, as defined by the threshold parameter λ [3,6,7]. The full ODE model is then given by

$$f(x) = g(x) + \Theta(x)\widehat{\Xi} \tag{2}$$

Model selection part I: NN-based learning

We perform two model selections steps, as we fit $\mathrm{NN}(x)$ and then $\Theta(x)\widehat{\Xi}$ by sparse regression, to allow for flexibility in the choice of hyperparameters at each step. We see in practice that one choice of hyperparameters cannot necessarily perform well on all problem types, motivating this approach. To choose the $\mathrm{NN}(x)$ that best fits the training data X, we allow three hyperparameters to vary: the window size (w), the batch size (b) for $\widetilde{X}$, and the learning rate of the Adam algorithm (lr). For a predetermined set of possible values for each hyperparameter, we fit $\mathrm{NN}(x)$ on every possible combination of hyperparameter values (w, b, lr). For each parameter set, we train the model for 10 epochs; where in each epoch all batches in $\widetilde{X}$ are fit. At the end of each epoch, we evaluate the current trained model using validation data. To do so, for each validation sample we simulate the trained hybrid dynamical model $f(x)$ over the full time span to obtain the predicted dynamics; we then take observations of the predicted dynamics at the same time points as used for training. We then compute the validation loss of the sample as the MSE between the data and the predicted dynamics. The hybrid dynamical model with lowest mean validation loss is selected as the best model and retained for model discovery via sparse regression at the next step.

Model selection part II: sparse regression

To find the best model discovered by sparse regression, we allow for three parameters to vary: the step size Δt for $\widehat{X}$, which defines the resolution of the data simulated from $\mathrm{NN}(x)$ and input to sparse regression, the choice of basis function library, and the regularization parameter α in STLSQ. For a given model (set of ODEs recovered by sparse regression) that has k parameters, we compute the Akaike information criterion with correction (AICc) on n validation samples [40]:

$$\mathrm{AICc} = n\ln(\mathrm{RSS}/n) + 2k + \frac{2(k+1)(k+2)}{(n-k-2)}, \tag{3}$$

where $\mathrm{RSS} = \sum_{i=1}^{n} \left\| X^{(i)} - X^{(i)}_{\mathrm{pred}} \right\|_2^2$ is the residual sum of squares between validation data $X^{(i)}$ and model-predicted dynamics $X^{(i)}_{\mathrm{pred}}$. We refer to $n\ln(\mathrm{RSS}/n)$ as the AIC term and the remainder of Eq 3 as the correction term. The model with the lowest AICc is chosen as the best model for that dataset.

Analysis of single-cell RNA-sequencing data to infer models of the epithelial-to-mesenchymal transition

Publicly available single-cell sequencing data (read count matrices) is obtained from three studies [41–43] (see [33] for further details). Lung adenocarcinoma (A549) cell line stimulated by TGF-β from Cook and Vanderhyden [41] (NCBI GEO accession GSE147405) are used for model discovery; other datasets are processed similarly and used for out-of-sample prediction. Analyzed in scanpy [44], cells with fewer than 200 genes or more than 8% mitochondrial read counts, and genes present in fewer than 3 cells are filtered out. Read counts are normalized to 10,000 and log-plus-one transformed. Batch correction is performed with ComBat [45]. Total counts, percent mitochondrial counts, and cell cycle effects are regressed out in scanpy, subsequently the data are scaled to uniform variance and highly variable genes are identified. Clustering is performed using the Leiden algorithm [46] at a resolution of 0.4. Three clusters are found at this resolution, which are identified as epithelial, intermediate, and mesenchymal cell states based on canonical gene expression.

The cell state trajectory data is calculated as follows. Trajectory inference is performed with diffusion pseudotime (DPT) [47], with root nodes chosen as those cells with values near the extrema of the first two diffusion map components. To estimate the uncertainty in pseudotemporal ordering, five cells are chosen as root nodes and the pseudotemporal ordering is calculated for each. The median values across the five runs are used as the pseudotime values for each cell, and the standard deviation is estimated around each median value. The pseudotime is then divided into twelve equal bins; the cell state proportions are calculated as the number of cells in each EMT state at each time interval (bin), normalized by the total number of cells at that time point.

We augment the cell state trajectory data to generate training and validation datasets containing multiple trajectories. Cell state proportions are sampled at each time point from a truncated Gaussian distribution with the following constraints: 1) each sampled proportion is between 0 and 1; and 2) all proportions at the same time point add up to 1. More specifically, for each observation $X^{(i)}$ in a dataset, its value for cell state k at pseudotime bin j, denoted as $X^{(i)}_{j,k}$, is drawn from a truncated distribution based on $\mathcal{N}(\mu_{j,k}, \sigma_{j,k})$, where $\mu_{j,k}$ is the median proportion of cells in state k and $\sigma_{j,k}$ the standard deviation. Each $X^{(i)}_{j,k}$ is constrained so that it is within one standard deviation of the base distribution and also within $[0, 1]$. Once values for two cell states are generated, the proportion of the remaining state is computed by subtracting the generated values from 1.

Results

Here we apply our model selection framework to three biological systems: two test-cases and one real dataset. The first system is the Lotka-Volterra model [30,31]. We illustrate the model selection framework and demonstrate that it can successfully learn Lotka-Volterra models from noisy data. We also use this system to show that the hybrid dynamical system approach outperforms alternatives for model discovery. The second system studied is the repressilator [32], which exhibits more complex dynamics and a greater challenge for model discovery than the Lotka-Volterra model. Finally, we study cell state transition dynamics during epithelial-mesenchymal transition using single-cell RNA-sequencing data as an example of the application of our model discovery approach to real data.

A model selection framework for robust model discovery with hybrid dynamical systems

We will perform model discovery on dynamical models of the form $x' = f(x)$ with $x \in \mathbb{R}^d$, where a partial but incomplete closed-form expression of $f(x)$ is given. This can be represented as a hybrid dynamical system (Eq 1):

$$x' = f(x) = g(x) + \mathrm{NN}(x).$$

Here, $g : \mathbb{R}^d \to \mathbb{R}^d$ is a function known in closed-form, and $\mathrm{NN} : \mathbb{R}^d \to \mathbb{R}^d$ is a neural network that models the unknown latent dynamics, which will be learnt from data and then input to sparse regression (Fig 1A). In the case that we have no prior knowledge about the model (no terms in $f(x)$ are known), we set $g(x) = 0$ and learn the entire right-hand side via $\mathrm{NN}(x)$. From $\mathrm{NN}(x)$, we then perform regression on $\mathrm{NN}(x) = \Theta(x)\Xi$ to find a sparse coefficient matrix $\widehat{\Xi}$ over the basis $\Theta(x)$ that characterizes the learnt model (Fig 1D). The full ODE model is then given by $x' = g(x) + \Theta(x)\widehat{\Xi}$.

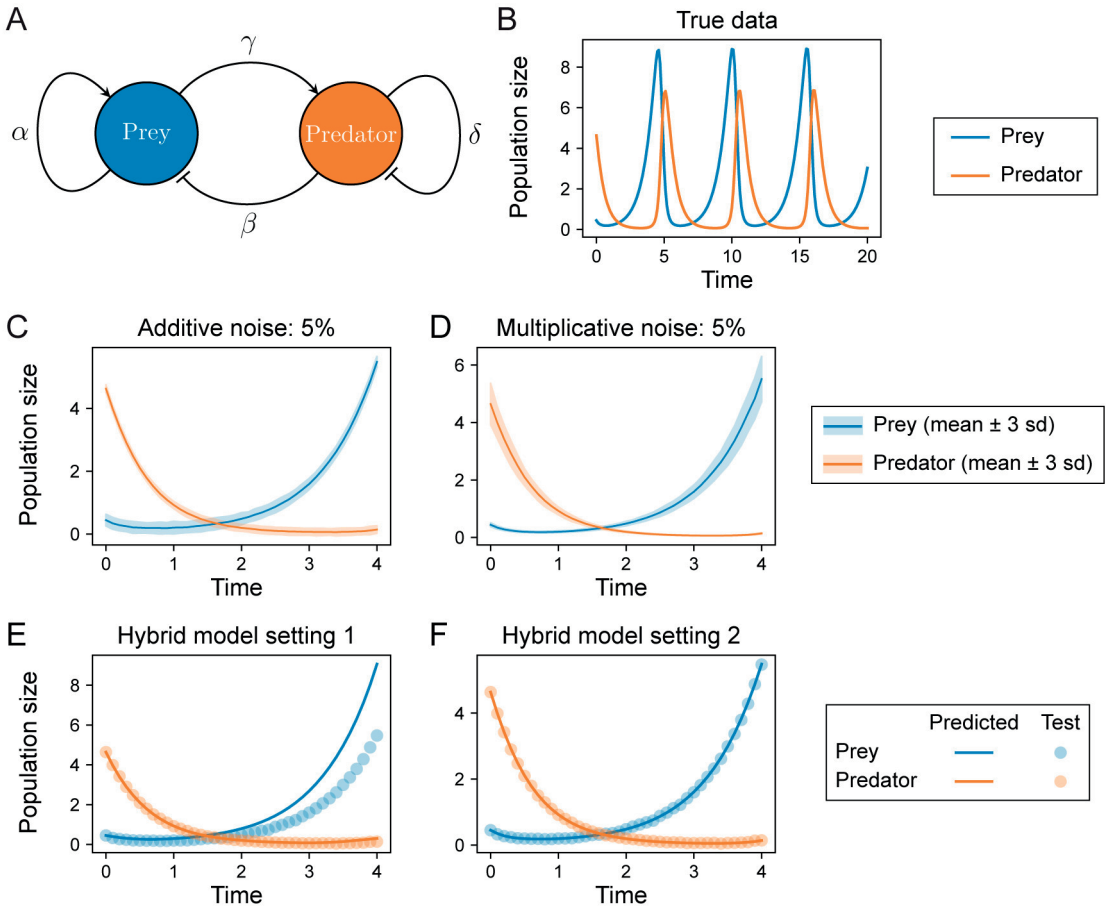

Fig 2. Evaluation of hybrid dynamical model fits on the Lotka-Volterra model. A. The Lotka-Volterra system describes predator-prey relationships in an ecosystem, here parameterized by $(\alpha, \beta, \gamma, \delta)$. **B.** Example simulation with parameters (1.3, 0.9, 0.8, 1.8). The population dynamics oscillate at a stable limit cycle. **C and D.** To evaluate model discovery methods, additive or multiplicative noise is added to the underlying deterministic dynamics at different noise levels. The mean trajectories of 200 samples for each noise model are shown; ribbons represent ±3 s.d. **E and F.** Comparison of fits using hybrid dynamical models. In setting 1, good fits to the data were not obtained (high validation loss). Training parameters: learning rate 0.001; window size 10; batch size 10. In setting 2, a good fit to the data was obtained. Training parameters: learning rate 0.01; window size 5; batch size 5.

https://doi.org/10.1371/journal.pcbi.1012762.g002

At both the NN and the sparse regression steps, we perform a search over the hyperparameter space and evaluate inferred models on validation data (see Model selection parts I & II in Methods). There are two advantages of this approach. First, evaluating inferred models on validation data ensures that the models are assessed in terms of their ability to reliably predict unseen data and are not biased towards the training data. Second, model selection using a grid search increases the robustness of our method, allowing it to be adaptable to datasets with vastly different characteristics. To demonstrate the pipeline, we will use the Lotka-Volterra model as an example (Fig 2A), which is given by [30,31]:

$$\begin{aligned} x_1' &= \alpha x_1 - \beta x_1 x_2 \\ x_2' &= \gamma x_1 x_2 - \delta x_2 \end{aligned} \tag{4}$$

In Eqs 4, x_1 is the population size of the prey and x_2 is the population size of the predators.

Table 1. Model selection configuration for Lotka-Volterra.

Learning step	Hyperparameter	Values/options
Hybrid dynamical model	Window size (data preprocessing)	5, 10
	Batch size (data preprocessing)	5, 10, 20
	Learning rate (Adam optimization)	0.001, 0.01, 0.1
Sparse regression	Step size (for generating $\widehat{X}$)	0.05, 0.1
	Basis library	`poly_max_2`: $1, x_i, x_i^2, x_i x_j$ `poly_2_3`: $x_i^2, x_i x_j, x_i^3, x_i^2 x_j, x_i x_j^2$
	α (STLSQ regularization parameter)	0.05, 0.1, 0.5, 1, 5, 10
	λ (STLSQ coefficient threshold)	0.1

https://doi.org/10.1371/journal.pcbi.1012762.t001

Our model selection strategy consists of two stages. In the first, we determine the best $\mathrm{NN}(x)$ in Eq 1 to model the latent dynamics. For the Lotka-Volterra model, we assumed that the growth/death terms were known, and sought to learn the interaction terms via $\mathrm{NN} : \mathbb{R}^2 \to \mathbb{R}^2$ with:

$$\begin{aligned} x_1' &= \alpha x_1 + \mathrm{NN}_1(x) \\ x_2' &= -\delta x_2 + \mathrm{NN}_2(x) \end{aligned} \tag{5}$$

The neural network $\mathrm{NN}(x)$ in Eqs 5 had an input layer of length 2 and an output layer of length 2. Between those two layers, we added one fully-connected hidden linear layer of 8 neurons, omitting greater depth to avoid overfitting due to model complexity. For each dataset, we trained $\mathrm{NN}(x)$ with different learning rates on samples processed by various window and batch sizes (see Materials and methods); for the Lotka-Volterra model the hyperparameter values used in model selection are given in Table 1. The window sizes were set to 5 or 10 so that each window covered a small segment of the full time span but had sufficient data points to train from. Batch sizes ranged from 5 to 20 to allow smooth updating of the NN parameters while learning unique features of each batch. Learning rates were chosen at different orders of magnitude to enable the most efficient training of the NN.

In the second stage of our model selection approach, we sought to discovery accurate ODE models by sparse regression. We evaluated models by the Akaike information criterion with correction (AICc, Eq 3), which estimates the model prediction error with a penalty term to account for the number of model parameters [40]. For the Lotka-Volterra model, we considered three hyperparameters for model selection. The first hyperparameter was the step size (Δt) used in the generation of training data ($\widehat{X}$) for sparse regression from the fitted hybrid dynamical model. The values we used were $\Delta t = 0.1$, i.e. the same as the training data, and $\Delta t = 0.05$ to augment the trajectory with additional data points. We note that generating trajectories with an arbitrary step sizes (interpolation) is only possible because of the fitted hybrid dynamical system; methods such as that of finite differences only compute derivatives on the time points provided. The second hyperparameter was the library of basis functions. Here, we assumed that the possible interaction terms were all polynomial, but with unknown order. We constructed two basis libraries consisting of polynomial terms of different orders: the first included all polynomial terms up to order 2 (denoted `poly_max_2`); the second included only polynomial terms of orders 2 and 3 only (denoted `poly_2_3`), which was motivated by the observation that we sought to infer interaction terms, i.e. the first-order terms in the model may

be already completely known. The third hyperparameter was the regularization parameter α for STLSQ. Here a wide range of α values were tested so that we could obtain models with either few or many terms. Table 1 gives the full details of the hyperparameter space explored for model selection. We kept λ (the STLSQ coefficient threshold) constant at $\lambda = 0.1$, i.e. only basis terms with coefficients greater than 0.1 are retained in the final model.

Lotka-Volterra models can be inferred from noisy data

To test whether we could correctly infer the Lotka-Volterra model (Fig 2A) from noisy data, we generated datasets from a single set of true parameters, $\{\alpha, \beta, \gamma, \delta\} = \{1.3, 0.9, 0.8, 1.8\}$ and initial conditions $x_0 = [0.4425, 4.6281]$. Under these conditions the system oscillates, reaching a stable limit cycle with a period of approximately five time steps (Fig 2B). A total of 12 datasets were generated by considering either additive or multiplicative noise at six different noise levels: 0.1%, 0.5%, 1%, 5%, 10%, and 20% (examples for the 5% noise level are shown in Fig 2C and D). Each dataset comprised 200 training samples and 50 validation samples on $t = [0, 4]$, and 50 test samples on $t = [0, 20]$, all generated with step size $\Delta t = 0.1$. This Lotka-Volterra example model was studied by Rackauckas et al. [22] to demonstrate how a hybrid formulation could be used to approximate derivatives for use with sparse regression. Here we sought to learn from noisy biological datasets that contain multiple measurements of each species by learning the dynamics from all the data together, rather than fitting a model to a single trajectory. We also implemented a model selection strategy in order to find models that generalize on both seen and unseen data.

We found wide variation in the results of training the hybrid model (Eqs. 5) based on the choice of hyperparameters. For example, at 1% additive noise, a learning rate of 0.001 produced underfit models with high validation loss (Fig 2E), whereas higher learning rates quickly attained low validation losses and fit the data well (Fig 2F). We examined hyperparameter values used to attain lowest validation loss for all datasets and found that the best hyperparameter combination varied by datasets (S1 Table), highlighting the utility of a flexible approach. Overall, we found our model selection scheme to improve the fits of hybrid dynamical models in an automated manner; avoiding the need for ad hoc approaches in hyperparameter choice for $\mathrm{NN}(x)$.

Given the best fit NN model, we simulated training data and performed sparse regression for model selection for all configurations specified in Table 1, to produce a total of 24 discovered models. We considered a model correct if it had the correct topology, i.e. correct terms in Eqs 4 with no extra terms; the parameter values did not have to be accurately inferred, although we saw in practice that the inferred parameters of most models with correct terms were close to the true values. For datasets of both additive and multiplicative noises up to 5% noise level, the ODE models with lowest AICc were all correct (Table 2). Those models could be used to accurately predict future states for test data on $t = [0, 20]$ (Fig 3A, B, D, E). For 10% and 20% noise levels, extra terms were recovered in models with lowest AICc (Table 2); those models failed to extrapolate well beyond $t = 4$ (Fig 3C, F), exhibiting damped oscillations. Nonetheless, correct models could still be successfully recovered at 10% noise level (S2 Table). Those models produced stable oscillations beyond $t = 4$, albeit with small inaccuracies in the period and amplitude (Fig 3G and H). In several cases, we found that the best-fit models were inferred from a hybrid dynamical model simulated at finer temporal resolution (step size $\Delta t = 0.05$) than the data, highlighting that the hybrid dynamical model approach allowed us to obtain better results as it permits data interpolation with arbitrary step size (S3 Table).

Table 2. Best ODE model inferred from Lotka-Volterra data at each noise level.

Noise type	Noise level	Inferred model
Additive	0.1%	$x_1' = 1.3x_1 - 0.907x_1x_2$ $x_2' = -1.8x_2 + 0.804x_1x_2$
Additive	0.5%	$x_1' = 1.3x_1 - 0.911x_1x_2$ $x_2' = -1.8x_2 + 0.780x_1x_2$
Additive	1%	$x_1' = 1.3x_1 - 0.906x_1x_2$ $x_2' = -1.8x_2 + 0.781x_1x_2$
Additive	5%	$x_1' = 1.3x_1 - 0.893x_1x_2$ $x_2' = -1.8x_2 + 0.814x_1x_2$
Additive	10%	$x_1' = 1.3x_1 + 0.206x_2^2 - 0.842x_1x_2 - 1.044x_1x_2^2$ $x_2' = -1.8x_2 + 0.924x_1x_2$
Additive	20%	$x_1' = 1.3x_1 + 0.199x_2^2 - 0.525x_1^2x_2 - 1.707x_1x_2^2$ $x_2' = -1.8x_2 + 0.798x_1x_2 + 0.237x_1x_2^2$
Multiplicative	0.1%	$x_1' = 1.3x_1 - 0.889x_1x_2$ $x_2' = -1.8x_2 + 0.826x_1x_2$
Multiplicative	0.5%	$x_1' = 1.3x_1 - 0.910x_1x_2$ $x_2' = -1.8x_2 + 0.761x_1x_2$
Multiplicative	1%	$x_1' = 1.3x_1 - 0.935x_1x_2$ $x_2' = -1.8x_2 + 0.814x_1x_2$
Multiplicative	5%	$x_1' = 1.3x_1 - 0.836x_1x_2$ $x_2' = -1.8x_2 + 0.846x_1x_2$
Multiplicative	10%	$x_1' = 1.3x_1 - 0.884x_1x_2 - 0.333x_1^2x_2$ $x_2' = -1.8x_2 + 0.883x_1x_2$
Multiplicative	20%	$x_1' = 1.3x_1 + 0.127x_1 - 1.194x_1x_2$ $x_2' = -1.8x_2 + 0.502x_2 + 0.247x_1x_2 - 0.198x_2^2$

https://doi.org/10.1371/journal.pcbi.1012762.t002

To test the generalizability of the approach, we performed model discovery on an alternative parameterization of the Lotka-Volterra model. We used the parameter values ($\{\alpha, \beta, \gamma, \delta\} = \{1.3, 0.9, 0.8, 1.8\}$) and initial conditions ($x_0 = [2.5, 0.7]$). This model still exhibits a stable limit cycle but the population sizes of prey and predator are closer in magnitude and smaller overall than for the previous parameterization example (S1 Fig). The model discovery results obtained were similar to those above. In fact, for this parameterization, the hybrid approach for model discovery performed better than in the previous example: the best models ranked by AICc were correct up to 10% noise level for both additive and multiplicative noise (S1 Fig).

Incorporating prior knowledge improves models inferred from sparse regression

To assess the performance of the hybrid dynamical system formulation for model discovery (which incorporates partial prior knowledge of the system), we compared it to four alternative methods:

- Base SINDy: sparse regression in SINDy run in a single step on all the data. This takes as input the training data X and estimates the derivatives X' by finite differences [6,7]. No prior knowledge is incorporated;
- Base-known: similar to Base SINDy, but with known terms (i.e. αx_1 and $-\delta x_2$ in Eqs 5) subtracted from X', estimated by finite differences before regression. The right-hand

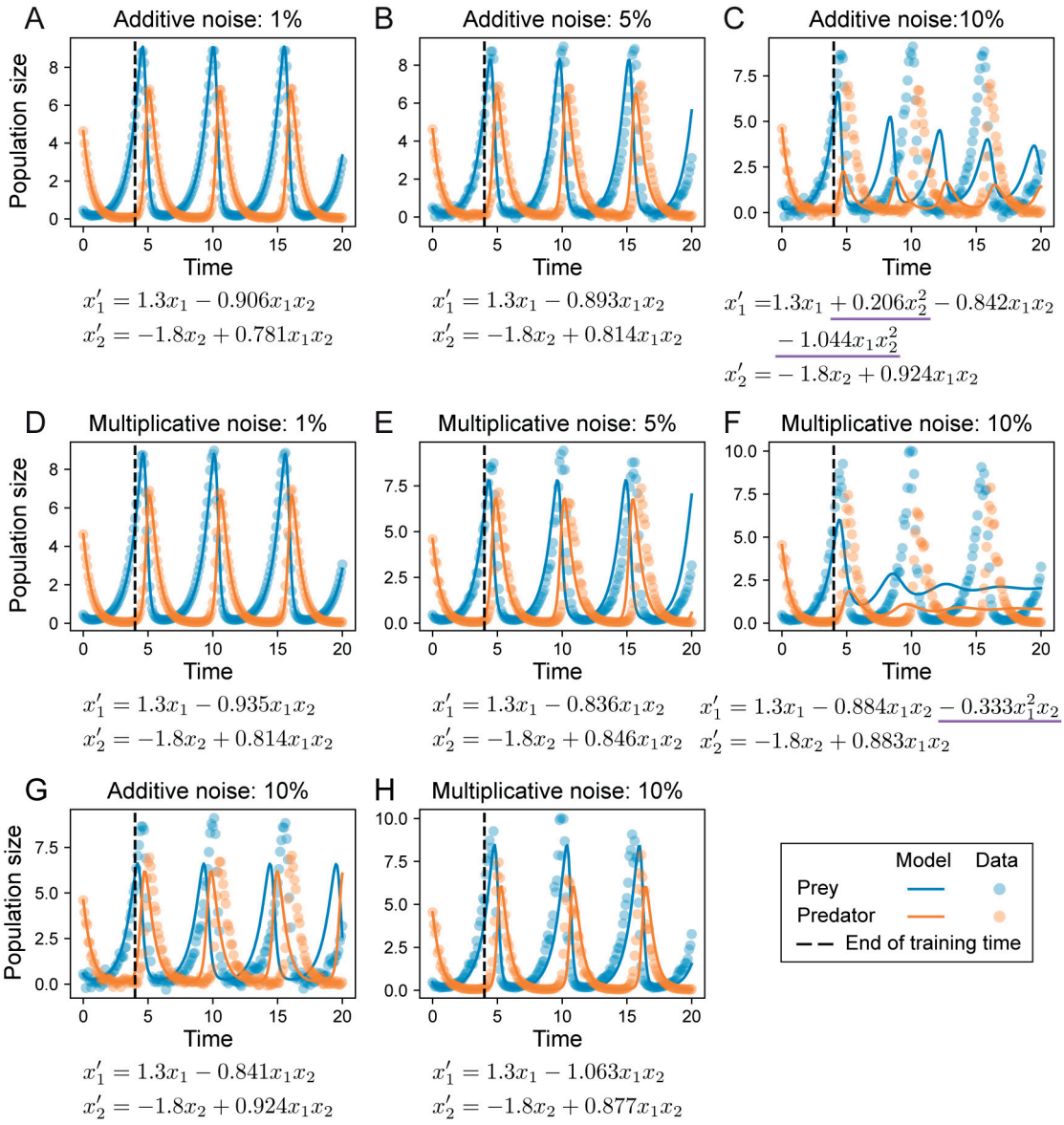

Fig 3. Inferred models from Lotka-Volterra datasets. For each model, the first term in x_1' and x_2' is known and the remaining terms are inferred. Underline denotes terms that are incorrect relative to the true model (Eqs 4). **A–C.** Inferred models with lowest AICc for datasets with additive noise, at 1, 5 and 10%. **D–F.** Inferred models with lowest AICc for datasets with multiplicative noise at at 1, 5 and 10%. **G.** The model with correct terms could be inferred at 10% additive noise, but was not ranked highest by AICc. **H.** The model with correct terms could be inferred at 10% multiplicative noise, but was not ranked highest by AICc.

https://doi.org/10.1371/journal.pcbi.1012762.g003

side of the inferred model is the sum of the known terms and terms inferred by sparse regression;

- Weak SINDy: SINDy performed on weak formulation of the underlying ODE system [17–19], as implemented in PySINDy [6,7]. No prior knowledge is incorporated;
- Pure NN: the entire derivative was approximated by a NN, which is then input to sparse regression, i.e. $x' = f(x) = \mathrm{NN}(x)$. No prior knowledge is incorporated.

Table 3. Lowest AICc values of models inferred from Lotka-Volterra datasets using various methods.

Noise type	Noise level	Base	Base-known	Weak	Pure NN	Hybrid
Additive	0.1%	-309	-309	n/a	**-433**	-369
Additive	0.5%	-299	-297	n/a	n/a	**-365**
Additive	1%	-283	-280	n/a	-300	**-368**
Additive	5%	-178	-175	-195	-39	**-216**
Additive	10%	-105	-104	-113	**-147**	-146
Additive	20%	**-26**	-28	n/a	n/a	n/a
Multiplicative	0.1%	-298	-296	n/a	-332	**-372**
Multiplicative	0.5%	-254	-249	n/a	-318	**-393**
Multiplicative	1%	-215	-209	**-334**	-81	-208
Multiplicative	5%	n/a	n/a	-179	n/a	**-137**
Multiplicative	10%	n/a	n/a	n/a	n/a	**-59**
Multiplicative	20%	n/a	n/a	n/a	n/a	n/a

https://doi.org/10.1371/journal.pcbi.1012762.t003

For fair comparison, we also performed model selection on each of the methods above; see Table 1 for hyperparameters where applicable to the method. For the first three methods above, model selection was performed for sparse regression without the step size hyperparameter for generating $\widehat{X}$, as those methods do not interpolate.

We compared the inferred models by each method via AICc. Base SINDy and base-known inferred correct models at all levels of additive noise tested, but did not infer correct models for multiplicative noise > 1% (Table 3, S4 Table, S5 Table). Of note, only these two methods could infer correct models for high levels (20%) of additive noise. Weak SINDy was able to infer correct models at moderate noise levels (around 5%), but could not infer correct models at the lowest or the highest noise levels (Table 3, S6 Table). The pure NN approach inferred correct models for most levels of additive noise (0.1%, 1%, 5%, 10%) and for low levels of multiplicative (up to 1%) (Table 3, S7 Table). The hybrid formulation achieved the best performance overall, inferring correct models for 10/12 datasets (not successful at 20% noise of either type; Table 3). Unlike most methods, it was able to handle higher levels of multiplicative noise, successfully discovering correct models at 5-10%. In terms of model quality, the hybrid formulation outperformed all other methods for 7/12 datasets and was within a small margin of error for one other dataset (10% additive noise), confirming that overall the hybrid approach offers a powerful framework for model discovery with noisy data (Table 3). An important addendum: the number of hyperparameters varies between methods, potentially biasing the comparison against those methods with fewer. e.g. non NN-based methods.

To analyze the performance of each method in more depth, we studied the derivatives inferred by each, as these are crucial to the sparse regression. (Note that weak SINDy was not included in this analysis as it performs sparse regression on integrals instead of derivatives.) We found that the derivatives for all methods tested deviated from the true values, especially for multiplicative noise (Fig 4), but that the derivatives from the hybrid method deviated the least overall, with small deviations from the true value appearing only for multiplicative noise near the end of the time window (Fig 4D). This goes to explain in part how the hybrid approach is able to infer more accurate models overall (Table 3).

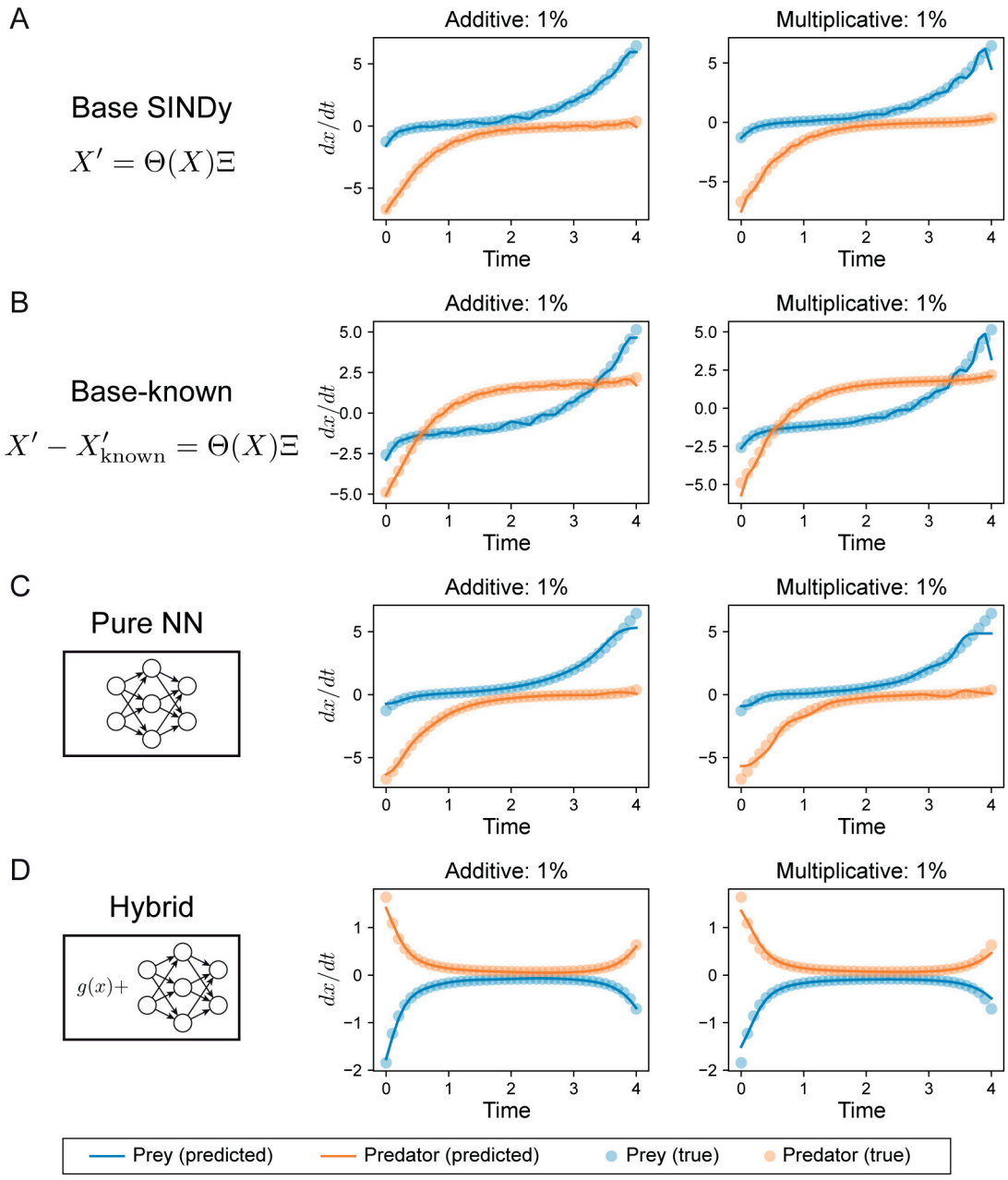

Fig 4. Comparison of derivatives approximated from Lotka-Volterra datasets using various methods. A. Derivatives approximated by base SINDy, which uses the method of finite differences. **B.** Derivatives approximated by finite differences, with known terms (i.e. αx_1 and $-\delta x_2$ in Eqs 5) subtracted, which are used by base-known. **C.** Derivatives approximated by a pure neural network formulation, in which the full right-hand side of an ODE system is approximated by a neural network. **D.** Derivatives approximated by a neural network in the hybrid model; where the neural network is fitted to partial dynamics (Eqs 5).

https://doi.org/10.1371/journal.pcbi.1012762.g004

Model discovery from noisy data for biological reaction networks with complex dynamics

We next turn our attention to a larger model with complex dynamics, as a more challenging test of model discovery with hybrid dynamical systems. The repressilator [32] describes a synthetic transcriptional network that can produce stable oscillations. Originally constructed in

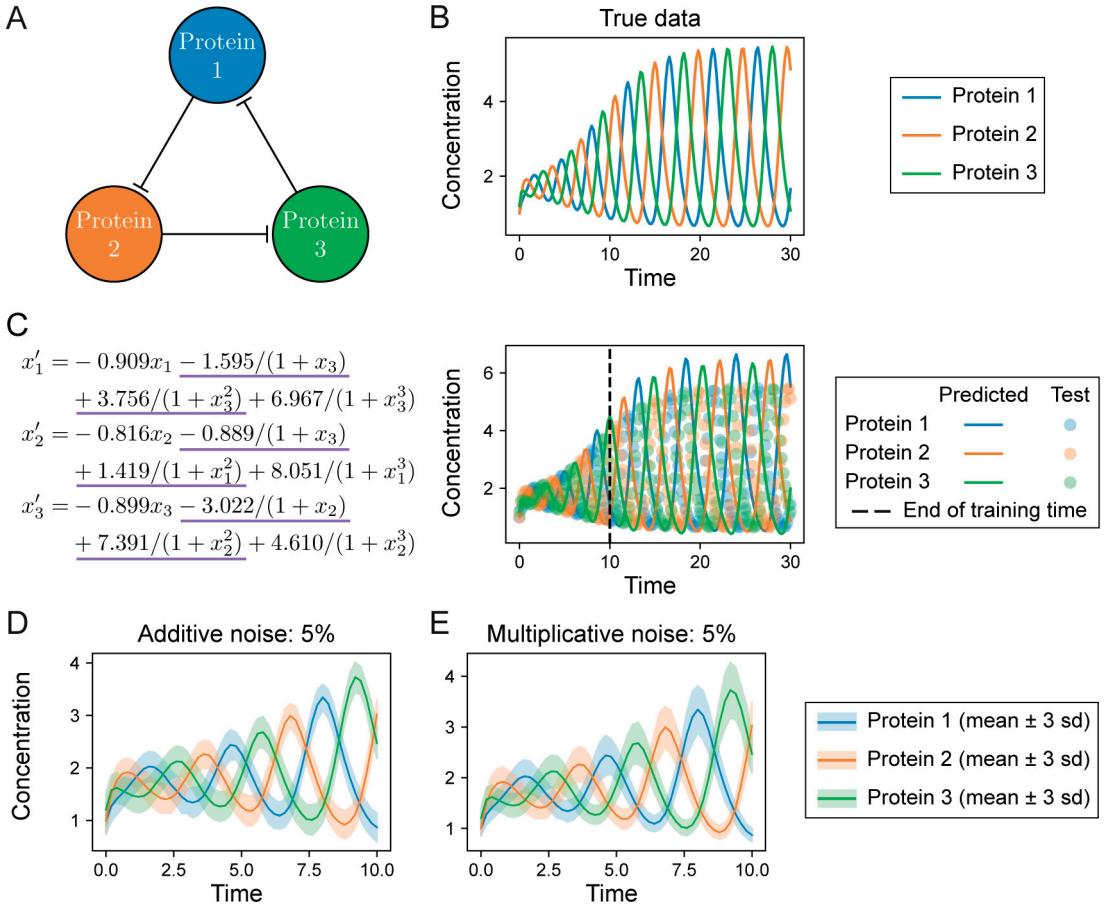

Fig 5. Evaluation of base SINDy fits and data generation for the repressilator model. A. The repressilator models describes a coupled negative feedback loop between three interacting proteins. **B.** Example simulation of the repressilator model without noise. The concentrations of each of the three proteins oscillate and the system eventually reaches a stable limit cycle. **C.** Evaluation of base SINDy to infer repressilator models from noise-free data on $t = [0, 10]$. Inferred equations (left) and simulation of the inferred model on $t = [0, 30]$ (right). Underlined terms are incorrect in comparison to the true repressilator model (Eqs 6). **D–E.** To evaluate model discovery methods, additive or multiplicative noise is added to the underlying deterministic dynamics at different noise levels. The mean trajectories of 200 samples for each noise model are shown; ribbons represent ±3 s.d.

https://doi.org/10.1371/journal.pcbi.1012762.g005

E. coli, it consists of three proteins that each represses its neighbor, thus coupling all three in a negative feedback loop (Fig 5A). It can be modeled with six species: the mRNA and protein products of each gene, or one can simplify the model to consider one variable per species. Taking the latter approach, the system can be described by the following equations:

$$\begin{aligned} x_1' &= \frac{\beta}{1+x_3^n} - \gamma x_1 \\ x_2' &= \frac{\beta}{1+x_1^n} - \gamma x_2 \\ x_3' &= \frac{\beta}{1+x_2^n} - \gamma x_3, \end{aligned} \tag{6}$$

where β and γ represent the basal production and degradation rates, respectively, and where we have assumed symmetry between the species. Below, we will set the degradation rate for

each protein as $\gamma = 1$ and assume these are known for each species. The remaining terms in Eqs 6 represent the inhibitions of each protein by its neighbor in the network. The inhibition terms are represented by Hill functions of order n, where we have assumed a half maximal effective concentration of each protein $k = 1$. We studied a specific parameterization of the repressilator system with $n = 3$, $\beta = 10$ and initial conditions $x_0 = [1, 1, 1.2]$. At this point, the system displays oscillations that increase in amplitude for the first 15 time steps (units are dimensionless) and then reaches a stable limit cycle thereafter (Fig 5B). We tested whether we would discover correct models for the repressilator (Eqs 6) from data on $t = [0, 10]$, a time interval that does not let the system reach its steady state.

We first tested whether we could discover correct models for the repressilator using a direct SINDy approach (base SINDy). To construct a basis for regression in SINDy, we made the assumptions that: the degradation term for each protein is linear; proteins interact only via inhibition; and that the strength of each inhibitory interaction can be characterized by a Hill function $1/(1 + u^n)$ of order n, with $n \in \{1, 2, 3\}$. I.e. we seek to discover the interaction structure of the network for the pairs of proteins: who interacts with whom, and in what direction. Given these assumptions, the basis library for SINDy consisted of the Hill functions $\{1/(1 + x_i), 1/(1 + x_i^2), 1/(1 + x_i^3)\}$ and the linear functions $\{x_i\}$ for $i = 1, 2, 3$; a total of 12 basis functions and 36 coefficients to be determined. In testing model discovery with SINDy on this basis, we realized that it was difficult to set λ, the basis coefficient threshold for inclusion of that term. For λ close to or greater than 1, the (correct) linear terms would either be eliminated or have coefficients much larger than the true values. For $\lambda \ll 1$, all of the discovered models would admit extra Hill terms with wrong orders and small coefficients. For example, with $\lambda = 0.5$ and true data generated noise-free and sampled every 0.2 seconds in $t = [0, 10]$, the discovered model is shown in Fig 5C. This model has Hill terms of orders $n = 1$ and $n = 2$ and cannot accurately predict future states (Fig 5C). This highlights the challenge in estimating derivatives for sparse regression, in this case even before any noise is added to the data. In contrast, SINDy could easily recover the correct Lotka-Volterra model from noise-free data with a simple degree 2 polynomial basis library and STLSQ threshold 0.1 (S2 Fig).

The hybrid formulation overcomes the challenge of threshold choice in STLSQ. We separate the linear (known) degradation terms from the unknown interaction terms in the hybrid dynamical system with $\mathrm{NN} : \mathbb{R}^3 \to \mathbb{R}^3$ to give:

$$x_i' = \mathrm{NN}_i(x) - x_i, \text{ for } i \in (1, 2, 3), \tag{7}$$

where $\mathrm{NN}(x)$ is a neural network with two fully-connected hidden linear layers of 8 neurons each. In the hybrid system, the choice of λ no longer affects the linear terms since these are known; and need to learn only the interaction terms from the latent dynamics approximated by $\mathrm{NN}(x)$. To do so we used a basis consisting of inhibitory terms, i.e. Hill functions between pairs of proteins with different Hill coefficients: $\{1/(1 + x_i), 1/(1 + x_i^2), 1/(1 + x_i^3)\}$. In addition, our model selection scheme for sparse regression would recover models using basis libraries of Hill functions of the same order. We will use `hill_n` for a basis library of Hill functions of order n only and `hill_max_n` for a basis library of all Hill functions up to order n. Similar to the Lotka-Volterra examples, we will experiment with two step sizes for estimating derivatives from trained $\mathrm{NN}(x)$, and six different values of the regularization parameter α. The full configuration for model selection on repressilator models is given in Table 4.

To test whether we can successfully discover the repressilator model from noisy data, we again generated datasets with additive or multiplicative noise at six different noise levels from

Table 4. Model selection configuration for the repressilator model.

Learning step	Hyperparameter	Values/options
Hybrid dynamical model	Window size (data preprocessing)	5, 10
	Batch size (data preprocessing)	5, 10, 20
	Learning rate (Adam optimization)	0.001, 0.01, 0.1
Sparse regression	Step size (for generating $\widehat{X}$)	0.1, 0.2
	Basis library	`hill_1`: $1/(1+x_i)$ `hill_2`: $1/(1+x_i^2)$ `hill_3`: $1/(1+x_i^3)$ `hill_max_3`: $1/(1+x_i)$, $1/(1+x_i^2)$, $1/(1+x_i^3)$
	α (STLSQ regularization parameter)	0.05, 0.1, 0.5, 1.0, 5.0, 10.0
	λ (STLSQ coefficient threshold)	1.0

https://doi.org/10.1371/journal.pcbi.1012762.t004

0.1 – 20% (example for 5% noise shown in Fig 5D and E). Each dataset had 200 training samples and 50 validation samples on $t = [0, 10]$, as well as 50 test samples on $t = [0, 30]$, all sampled with step size $\Delta t = 0.2$. As for the example with the Lotka-Volterra model, we considered an inferred ODE model correct if it identified all terms of Eqs 6 with no extra terms (i.e. correct topology). We did require that "correct models" had highly accurate parameter values, although we found in practice that they often were very close to the true values.

We found that correct models were inferred for all noise types and levels up to 1%, and for multiplicative noise up to 5% (Table 5). The inferred models could accurately predict future states (Fig 6A, B, D, E). For higher noise values, the inferred models with lowest AICc did not correctly recover Eqs 6 and could not predict future dynamics beyond the training time span (Fig 6C, F). However, analysis of the top 10 inferred models for each noise type/level at higher levels of noise (S6–S10 Tables) revealed that in many cases correct models could be found. For additive noise at 5% and 10% and for multiplicative noise at 10%, correct ODE models were discovered in the top 10 predicted models, e.g. Models 4 and 5 in S8 Table, Models 6 and 7 in S9 Table, Models 7 and 8 in S10 Table. Simulations of these models showed that—despite larger parameter variation from the true values—they produced stable limit cycles with oscillations close to the true data well beyond the time range used for training (Fig 6G and H). These correct models tended to have higher AICc than a few incorrect ones since their prediction errors accumulated over successive cycles due to mismatched periodicity. However, we would argue that in practice they should be the preferred models. As an example, the model in Fig 6G should be a better model than the model in Fig 6C, because the former was showing signs of stable oscillations on $t \in [0, 10]$ similar to the training data, whereas all proteins in the latter quickly approached a fixed point.

At the 20% noise level, there were no correct models in the top 10 predicted for either additive or multiplicative noise. However, even at this high noise level, inferred models with low AICc scores contained aspects of the true system. Some inferred models contained Hill terms of the correct order ($n = 3$) but with an extra term (e.g. Models 4 and 5 in S11 Table, or Models 7 and 8 in S12 Table). In these models the correct terms had coefficients close to true values ($\beta = 10$) and the incorrect terms had coefficients small in magnitude. Additionally, models were predicted that contained Hill terms of the wrong order but which recapitulated the correct repressilator network. As an example, Model 1 (S11 Table) inferred the correct network topology:

$$\text{protein 1} \dashv \text{protein 2} \dashv \text{protein 3} \dashv \text{protein 1} \tag{8}$$

Table 5. Best ODE model inferred from repressilator data at each noise level.

Noise type	Noise level	Inferred model
Additive	0.1%	$x_1' = 10.017/(1+x_3^3) - x_1$ $x_2' = 9.988/(1+x_1^3) - x_2$ $x_3' = 10.223/(1+x_2^3) - x_3$
Additive	0.5%	$x_1' = 9.981/(1+x_3^3) - x_1$ $x_2' = 9.948/(1+x_1^3) - x_2$ $x_3' = 9.821/(1+x_2^3) - x_3$
Additive	1%	$x_1' = 10.358/(1+x_3^3) - x_1$ $x_2' = 10.012/(1+x_1^3) - x_2$ $x_3' = 10.158/(1+x_2^3) - x_3$
Additive	5%	$x_1' = -2.152/(1+x_2^2) + 9.303/(1+x_3^2) - x_1$ $x_2' = 7.755/(1+x_1^2) - x_2$ $x_3' = -1.491/(1+x_1^2) + 10.563/(1+x_2^2) - 1.897/(1+x_3^2) - x_3$
Additive	10%	$x_1' = -2.129/(1+x_2^2) + 9.345/(1+x_3^2) - x_1$ $x_2' = 9.502/(1+x_1^2) - 2.427/(1+x_3^2) - x_2$ $x_3' = -2.177/(1+x_1^2) + 10.042/(1+x_2^2) - x_3$
Additive	20%	$x_1' = 7.800/(1+x_3^2) - x_1$ $x_2' = 7.514/(1+x_1^2) - x_2$ $x_3' = 7.401/(1+x_2^2) - x_3$
Multiplicative	0.1%	$x_1' = 10.150/(1+x_3^3) - x_1$ $x_2' = 9.986/(1+x_1^3) - x_2$ $x_3' = 10.133/(1+x_2^3) - x_3$
Multiplicative	0.5%	$x_1' = 9.931/(1+x_3^3) - x_1$ $x_2' = 9.944/(1+x_1^3) - x_2$ $x_3' = 9.876/(1+x_2^3) - x_3$
Multiplicative	1%	$x_1' = 9.841/(1+x_3^3) - x_1$ $x_2' = 9.464/(1+x_1^3) - x_2$ $x_3' = 9.705/(1+x_2^3) - x_3$
Multiplicative	5%	$x_1' = 10.560/(1+x_3^3) - x_1$ $x_2' = 9.824/(1+x_1^3) - x_2$ $x_3' = 10.386/(1+x_2^3) - x_3$
Multiplicative	10%	$x_1' = -1.873/(1+x_2^2) + 9.551/(1+x_3^2) - x_1$ $x_2' = 10.506/(1+x_1^2) - 1.660/(1+x_2^2) - 1.610/(1+x_3^2) - x_2$ $x_3' = -1.821/(1+x_1^2) + 10.355/(1+x_2^2) - 1.087/(1+x_3^2) - x_3$
Multiplicative	20%	$x_1' = -1.887/(1+x_2^2) + 9.666/(1+x_2^2) - x_1$ $x_2' = 8.569/(1+x_1^2) - x_2$ $x_3' = -3.325/(1+x_1^2) + 11.144/(1+x_2^2) - x_3$

Terms colored in green are known terms in Eqs 7.

https://doi.org/10.1371/journal.pcbi.1012762.t005

although with lower cooperativity (smaller Hill term: $n = 2$) than the true system. We saw again in the case of models with incorrect Hill terms, that if they contained additional (incorrect) network edges (e.g. Model 1 in S12 Table), these would likely have much smaller coefficients than the terms that represented correct edges in the network (Eq 8). This suggests that the network topology shown in Eq 8 be the most likely one out of many other possible network topologies.

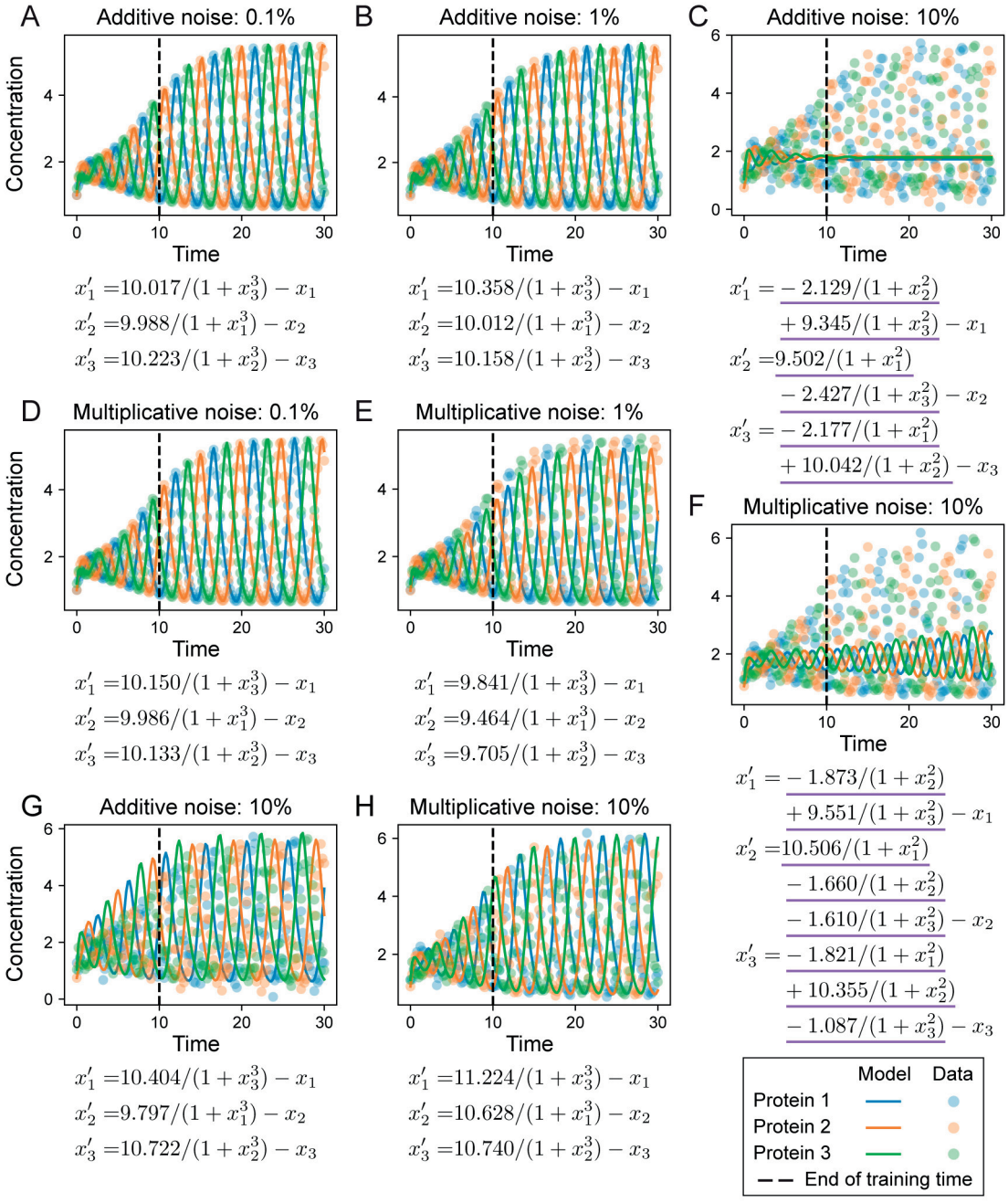

Fig 6. Inferred models inferred from repressilator datasets. For each model, the linear decay terms are known and the remaining terms are inferred. Underline denotes terms that are incorrect relative to the true model (Eqs 6). **A–C.** Inferred models with lowest AICc for datasets with additive noise, at 0.1, 1 and 10%. **D–F.** Inferred models with lowest AICc for datasets with multiplicative noise at at 0.1, 1 and 10%. **G.** The model with correct terms could be inferred at 10% additive noise, but was not ranked highest by AICc. **H.** The model with correct terms could be inferred at 10% multiplicative noise, but was not ranked highest by AICc.

https://doi.org/10.1371/journal.pcbi.1012762.g006

This analysis highlights useful applications and limitations of sparse regression to recover complex models from noisy data. At high noise levels it might not be possible to infer correct models with confidence. Nonetheless, through careful analysis of the predicted models in

light of: model topology, parameter values (coefficients), and the consistency of ranked model predictions (e.g. if models show the same network but with different Hill coefficients) we can gain predictions regarding model structure and interactions for further testing.

We also ran model discovery on a repressilator model with non-symmetric Hill responses. In this new model, the order n of Hill functions are 2, and their coefficients are $\beta = (7, 6, 8)$. Instead of stable oscillations with the same magnitude across all states in the previous system after $t = 10$, the oscillations in the new system are damped and approach a fixed point (S3 Fig). The results were similar to those of the previous model for low and moderate noise levels. For both additive and multiplicative noises up to 1%, the inferred models with lowest AICc were correct (S3 Fig). No correct models were found for noise level 5% or higher. However, in many models with low AICc, coefficients for correct Hill terms were still much larger than coefficients for incorrect terms (S3 Fig).

Model discovery of cell state transition dynamics from single-cell transcriptomics

To demonstrate the potential of model discovery to infer the structure of biological systems from real data, we applied the hybrid dynamical system approach to a dataset characterizing the epithelial-to-mesenchymal transition (EMT), a canonical cell state transition of importance in development, wound healing, and cancer [48–50]. We studied EMT induced by TGF-β in lung carcinoma cells (A549). Theoretical and experimental evidence has indicated that intermediate states exist along the EMT spectrum [51,52]; via cell clustering we found evidence for one stable intermediate cell state in this dataset (see Methods). We thus sought to infer models describing the dynamics of three cell populations over pseudotime: epithelial (E), intermediate (I), and mesenchymal (M) (Fig 7A). A direct transition from $E \rightarrow M$ might also occur, as well as reverse transitions. By projecting the cell states over pseudotime (see Methods) we obtain the cell population pseudo-dynamics, from which we can infer models of EMT (Fig 7B). (The dynamics over real time do not provide adequate temporal resolution to infer a state transition; S4 Fig.)

To train models, 200 trajectories of the cell population dynamics were sampled over pseudotime values $t \in [1, 9]$ with step size $\Delta t = 1$; for validation of fitted models, 50 additional trajectories were sampled over the same time points. The data was sparse on the time axis compared to previous examples but sufficient for model discovery, since our methods are not sensitive to step sizes as long as the fluctuations in each species are well-represented in the data. To test inferred models' ability to extrapolate, 50 trajectories were sampled on $t \in [1, 13]$ with step size $\Delta t = 1$. We first tested base SINDy for model discovery on EMT data with sparse regression hyperparameters specified in S13 Table (except for step size, which is not used in base SINDy). We found that it could not produce models with dynamics that fit the data (Fig 7C). This highlights the importance of accurate derivatives for sparse regression, and the challenges in obtaining these from noisy biological data.

We next used a hybrid dynamical system for model discovery, assuming that each cell state had a known loss term with a rate parameter of 1, i.e. we fit the model:

$$x_i' = \mathrm{NN}_i(x) - x_i, \text{ for } i \in (1, 2, 3), \tag{9}$$

where (x_1, x_2, x_3) denote the epithelial, intermediate, and mesenchymal states, respectively. We inferred models using the same two-step approach as above; the model selection configuration is given in S13 Table. The basis libraries consisted of either only polynomial terms or a mixture of polynomial and Hill terms. We compared the models inferred by our

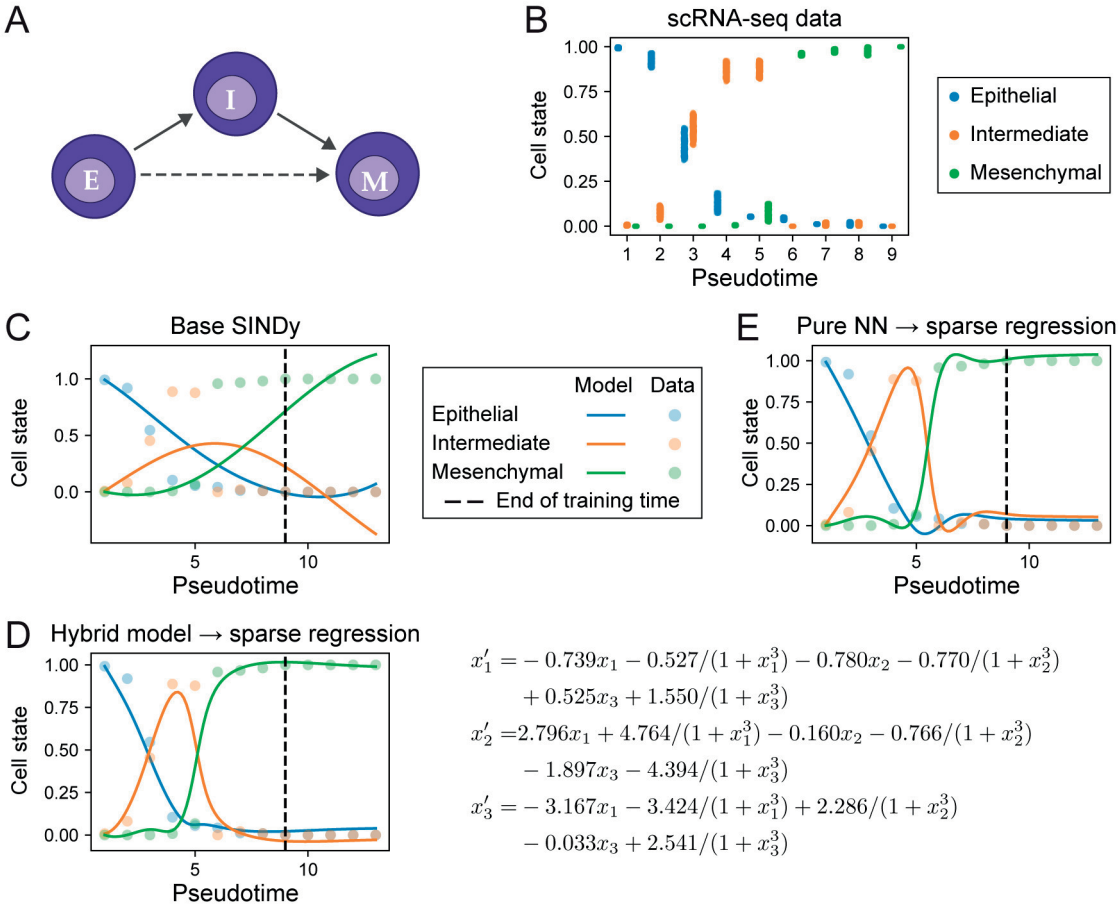

Fig 7. Model discovery of cell state transition dynamics from scRNA-seq data. A. Diagram of possible cell state transitions during epithelial-mesenchymal transition (EMT). **B.** EMT data generated with uncertainty estimates for model training from scRNA-seq of cell state transitions over pseudotime. **C.** Simulations from ODE model inferred by base SINDy. **D.** Simulations from ODE model inferred from the hybrid model (Eq 9); model equations of the inferred model (right). x_1: epithelial cell state; x_2 intermediate cell state; x_3: mesenchymal cell state. **E.** Simulations from ODE model inferred from the pure NN model. For **C**–**E**, the vertical dashed lines indicate the end of time span on which the inferred models were trained.

https://doi.org/10.1371/journal.pcbi.1012762.g007

hybrid approach with a pure NN approach in which we do not consider any known terms, i.e. fit the entire derivative by $\mathrm{NN}(x)$.

Both the hybrid model (Fig 7D) and the pure NN model (Fig 7E) could fit the dynamics of EMT. The pure NN model fits were slightly better than those for the hybrid model evaluated by AICc, but the former showed signs of overfitting given the temporal resolution of the data, as evidenced by the biologically implausible dip in epithelial and intermediate cell states for $t \in [5, 7]$ (Fig 7E). Thus while the pure NN approach provides the better fit quantitatively, due to the overfitting observed, the fit obtained by the hybrid model is preferable. Analysis of the derivatives inferred by the hybrid and pure NN models (S4 Fig) highlighted the gradual cell state transition $E \to I$ occurring in $t \in [0, 5]$ (second derivatives small) and the faster transition $I \to M$ in $t \in [4, 8]$ (second derivatives of I and M large and reach maxima).

Comparison of the models inferred from either hybrid (S14 Table) or pure NN (S15 Table) approaches found that the top ranked models of each were consistent, sharing both the same topological terms and in many cases similar parameter values. The basis of the top inferred

model was `hill_3_poly_1`: a mixture of polynomial terms with Hill terms of order 3. Notably, all polynomial terms in the model are linear, indicating a paucity of evidence for higher order polynomial interactions. The inferred model contains terms characterizing simple cell state transitions, e.g. $x_1 \rightarrow x_2$, albeit with a nonlinear modification to the transition rate for x_1 to x_2 (first two terms of x_1' in Fig 7D). The inferred model suggests that the transition from E to I is stimulated by the presence of intermediate state cells. There is also some evidence of a direct reverse transition from *M* back to *E*. Considering the loss terms in Eq 9, it is notable that the total inferred loss of x_3 is close to zero (the sparse regression model-predicted term ($+0.967x_3$) cancels out the loss term ($-x_3$) from the prior). This highlights the ability of the hybrid model framework to "correct" model terms when the data do not support the prior. Biologically, a loss rate for x_3 of zero is sensible since the all cells in the population are in a mesenchymal state at the final time point. In terms of the loss terms for x_1 and x_2, both have nonzero rates within plausible ranges; the rates vary over time due to nonlinear modifications (similar to those described above). Thus the loss rate of x_1 is in the range $[0.5, 0.9]$ and the loss rate of x_2 is in the range $[0.2, 0.7]$.

Finally, we tested the ability of the inferred model to perform out-of-sample predictions. For various different cell lines and stimuli, we determine the cell state transition dynamics over pseudotime [33]. We then simulate the inferred model (from A549 cells stimulated by TGF-β) using the initial conditions of every other cell type–stimulus dataset. Ordered by the fit accuracy (best to worst evaluated by MSE; S16 Table), we see that for many of the samples, a good fit is obtained. 4/11 samples have a close fit with mean MSE < 0.04. Within these good fits, we find multiple cell types (HMLE, OVCA) stimulated by the same stimulus (TGF-β), as well as other cell types (OVCA) stimulated by multiple stimuli (S5 Fig). Among the less well-fit samples, we find other stimuli (Zeb1, TNF) or experimental conditions (day 8 of the HMLE samples) that display different dynamics than those of the inferred model.

Discussion

In this study, we developed a framework for data-driven discovery of ODE models. We presented methods to infer models from noisy data via a two-step model selection framework. In the first, we learnt the latent (unknown) model dynamics with a NN; in the second we used sparse regression to infer equations to model the system. We showed how the use of hybrid dynamical systems outperformed purely data-driven approaches. This highlights that—while data-driven machine learning approaches for modeling complex systems hold great promise—best performance may be achieved with supervision when one has some prior knowledge about the system.

Analysis of model discovery on two canonical models (Lotka-Volterra and the repressilator) showed that our approach performed well on both additive and multiplicative noise at magnitudes up to 5%. At noise of levels of 10% and above, correct or partially correct models could still be found although not always ranked highest. This highlights that it can be important to consider a suite of possible models, rather than a single model, and evaluate them comparatively via specific features or interactions. Analysis of the levels of noise observed for biological systems that exhibit Lotka-Volterra-like oscillations (e.g. p53 [53,54], Hes1 [55,56] or NF-κB [57]) or repressilator dynamics [32,58] finds average noise levels to be around 1–5%, ranging up to 20–40% in some cases. Thus, it appears that many biological systems exhibit noise within the range possible to infer correct models from with our hybrid dynamical system approach.

The primary goal of this study was to identify models with low generalization error; this may come at the expense of underfitting the data, especially in regimes where noise is high.

The general question of identifying the underlying dynamics of biological systems with fidelity in the presence of noise remains challenging. In our work, we tested our methods using synthetic datasets that reflect common noise levels in biological systems. Future work might incorporate, for example, inference of stochastic models from the data. There is also room for improvement in inferring models within the current framework, especially for noise levels of 10% and above. One possible way to improve may be to denoise the data using conventional or deep learning-based smoothing techniques for time series [59,60] before learning a hybrid dynamical model. We also saw that at higher noise levels, incorrect models could be ranked above correct models by AICc. In the repressilator example, we saw that some correctly inferred models were lower ranked due to larger MSE caused by mismatched oscillations; and we suggest that useful predictions regarding the topology of the model can be made by considering a subset of highly-ranked models rather than only one. In addition, the AIC term in AICc grows with the noise, but the correction term does not; future work could explore alternative correction terms to the AIC that explicitly take the noise level into account.

The computational cost of model selection currently grows exponentially with the number of hyperparameters. To complete our numerical experiments in a reasonable amount of time, we carefully chose hyperparameters and their values so that the total number of candidate models was limited but the models were distinct enough to cover a large model space. The computational cost could also be reduced by alternative methods for hyperparameter optimization, thus enabling wider explorations of the model space [61,62]. These algorithms work not only for continuous hyperparameters, but also for discrete or categorical ones, and could be suitable both for NN-based learning and for sparse regression. Alternative NN architectures and optimizers for sparse regression may also improve future performance, including the SR3 algorithm [63,64], and could be incorporated as categorical variables into model selection.

The model discovery methods presented in this work belong to a larger class of network inference problems: a network of biological species is represented by the inferred equations. Network inference methods include means by which to use gene expression time series data [65] to infer gene-gene interaction networks [66,67]. However, the problem addressed by model discovery differs from network inference in key ways. Equation-learning offers much more than network topology: we learn the form of the interactions (linear vs nonlinear terms, etc) as well as the coefficients (rate constants) that govern them.

Sources of biological noise are myriad and complex. Yet the most popular algorithms for sparse regression have only been tested on zero-mean Gaussian noise [3,63,64]. An advantage of this choice is that a simple L_2 norm can be used for the error term in the objective function. However, such an objective function may be unsuitable for data with asymmetric or colored noise. We used the STLSQ optimizer here, considering only datasets with zero-mean Gaussian noise, although it would also be possible to learn the noise empirically. Kaheman et al. [68] proposed a SINDy-based method that fits the noise as a parameter on a per-time point basis, from which it could recover ODE models from measurements with asymmetric noise. However, the number of noise parameters that one must fit is then equal to the number of time points and thus grows with the number of time points in training data. An alternative approach to model the noise that could overcome this issue would be to use variational inference approaches [69].

Data-driven model discovery combines many fields, from numerical methods and statistical learning to dynamical systems. Its applications are equally broad, spanning engineering and the physical sciences. But applications of model discovery in biology are among the most exciting, precipitated by the astounding quantities of biological data and the extent to which we do not know the structure of most biological networks; there is a lot to learn. Here we have

tested the utility of model discovery methods on some challenging problems consisting of data sampled noisily from systems with complex dynamics. We have shown that hybrid methods can successfully learn incomplete models with limited noisy input data, and demonstrated the importance of incorporating prior knowledge into the inference framework. We were also able to predict candidate models describing regulations occurring during EMT from single-cell RNA-sequencing data. We anticipate further development of data-driven model discovery methods to improve our ability to infer models robustly, from limited noisy data, with prior knowledge integration where appropriate. This will accelerate our ability to produce new hypotheses in the form of differential equation models and, eventually, discover new biology.

Supporting information

S1 Fig. Lotka-Volterra model with alternative parameters. A. Example simulation with parameters $\{\alpha, \beta, \gamma, \delta\} = \{2.1, 1.2, 0.7, 1.1\}$. **B.** Inferred model with lowest AICc for datasets with 10% additive noise. **C.** Inferred model with lowest AICc for datasets with 10% multiplicative noise.
(TIF)

S2 Fig. Model inferred by SINDy from noise-free Lotka-Volterra data. Evaluation of base SINDy to infer Lotka-Volterra models from noise-free data on $t = [0, 4]$. True model parameters are $\{\alpha, \beta, \gamma, \delta\} = \{1.3, 0.9, 0.8, 1.8\}$; initial conditions are $x_0 = [0.4425, 4.6281]$. Inferred equations (left) and simulation of the inferred model on $t = [0, 20]$ (right).
(TIF)

S3 Fig. Repressilator model with alternative parameters. A. Example simulation with parameters $\beta = (7, 6, 8)$ and $n = 2$. **B.** Inferred model with lowest AICc for 1% additive noise dataset. **C.** Inferred model with lowest AICc for 1% multiplicative noise dataset. **D.** Inferred model with lowest AICc for 5% additive noise dataset. **E.** Inferred model with lowest AICc for 5% multiplicative noise dataset.
(TIF)

S4 Fig. Latent dynamics of cell state transition estimated by trained neural networks. A. Cell state proportions in epithelial-mesenchymal transition in experimental time. **B.** Derivatives estimated by the trained neural network $\mathrm{NN}(x)$ in hybrid formulation (Eqs 9). **C.** Derivatives estimated by the trained neural network $\mathrm{NN}(x)$ in pure neural network formulation (i.e. $x' = \mathrm{NN}(x)$). **D.** Derivatives of latent dynamics in A. **E.** Derivatives of latent dynamics in B.
(TIF)

S5 Fig. Cell state transition predicted by ODE model inferred from hybrid formulation vs data from other samples. Samples are sorted from smallest mean squared error (**A**) to largest (**K**). Each sample is named by "cell line – stimuli", with label for replicate appended if applicable. The cell lines are: HMLE – human mammary epithelial cell, OVCA – human ovarian cancer cell, MCF10A – human mammary epithelial cell, and DU145 – human prostate cancer cell. For replicates, "d8" and "d10" mean 8 days and 10 days after stimulation, respectively; "r1" means replicate 1 and "r2" replicate 2.
(TIF)

S1 Table. Hyperparameters used to attain lowest mean squared losses for hybrid dynamical models from Lotka-Volterra datasets.
(XLSX)

S2 Table. Best correct ODE models across noise levels inferred from Lotka-Volterra datasets using hybrid formulation. Terms colored in green are known terms in Eqs 5. Models given in Table 2 are not repeated here.
(XLSX)

S3 Table. Hyperparameters used to attain lowest AICcs for sparse regression on Lotka-Volterra datasets.
(XLSX)

S4 Table. Best correct ODE models across noise levels inferred from Lotka-Volterra datasets using base SINDy, according to AICc.
(XLSX)

S5 Table. Best correct ODE models across noise levels inferred from Lotka-Volterra datasets using base-known, according to AICc.

S6 Table. Best correct ODE models across noise levels inferred from Lotka-Volterra datasets using weak SINDy, according to AICc.
(XLSX)

S7 Table. Best correct ODE models across noise levels inferred from Lotka-Volterra datasets using pure neural network formulation, according to AICc.
(XLSX)

S8 Table. Top 10 models inferred from repressilator data with 5% additive noise (according to AICc).
(XLSX)

S9 Table. Top 10 models inferred from repressilator data with 10% additive noise (according to AICc).
(XLSX)

S10 Table. Top 10 models inferred from repressilator data with 10% multiplicative noise (according to AICc).
(XLSX)

S11 Table. Top 10 models inferred from repressilator data with 20% additive noise (according to AICc).
(XLSX)

S12 Table. Top 10 models inferred from repressilator data with 20% multiplicative noise (according to AICc).
(XLSX)

S13 Table. Model selection configuration for EMT data. For model selection at the sparse regression step, we considered basis libraries that include polynomial terms and/or Hill terms — `poly_max_2`: polynomial terms of degree up to 2 (bias term included); `poly_1_2`: polynomial terms of degrees 1 and 2 (no bias term); `hill_n_poly_1`: Hill terms of order n plus linear terms; `hill_n_poly_xy`: Hill terms of order n plus linear terms and interaction terms of the form xy; `hill_n_poly_1_2`: Hill terms of order n plus polynomial terms of degrees 1 and 2; and `hill_max_3_poly_1`: Hill terms of order up to 3 plus linear terms.
(XLSX)

S14 Table. Top 10 models inferred from EMT data using hybrid formulation (according to AICc).
(XLSX)

S15 Table. Top 10 models inferred from EMT data using pure neural network formulation (according to AICc).
(XLSX)

S16 Table. Prediction errors by inferred EMT model for original EMT data and out-of-sample data. E, I, M indicate mean squared errors for epithelial, intermediate, mesenchymal. Average is the average MSE.
(XLSX)

Author contributions

Conceptualization: Xiaojun Wu, Adam L. MacLean.

Investigation: Xiaojun Wu, MeiLu McDermott, Adam L. MacLean.

Methodology: Xiaojun Wu, MeiLu McDermott, Adam L. MacLean.

Project administration: Adam L. MacLean.

Software: Xiaojun Wu.

Supervision: Adam L. MacLean.

Writing – original draft: Xiaojun Wu, Adam L. MacLean.

Writing – review & editing: Xiaojun Wu, MeiLu McDermott, Adam L. MacLean.

References

1. Gillespie DT. The chemical Langevin equation. J Chem Phys. 2000;113(1):297–306. https://doi.org/10.1063/1.481811
2. Alber M, Buganza Tepole A, Cannon WR, De S, Dura-Bernal S, Garikipati K, et al. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. npj Digit Med. 2019;2(1):115. https://doi.org/10.1038/s41746-019-0193-y PMID: 31799423
3. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci U S A. 2016;113(15):3932–7. https://doi.org/10.1073/pnas.1517384113 PMID: 27035946
4. Mangan NM, Brunton SL, Proctor JL, Kutz JN. Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Trans Mol Biol Multi-Scale Commun. 2016;2(1):52–63. https://doi.org/10.1109/tmbmc.2016.2633265
5. Weiss JN. The Hill equation revisited: uses and misuses. FASEB J. 1997;11(11):835–41. https://doi.org/10.1096/fasebj.11.11.9285481
6. de Silva BM, Champion K, Quade M, Loiseau JC, Kutz JN, Brunton SL. PySINDy: a Python package for the sparse identification of nonlinear dynamics from data. 2020.
7. Kaptanoglu A, de Silva B, Fasel U, Kaheman K, Goldschmidt A, Callaham J, et al. PySINDy: a comprehensive Python package for robust sparse system identification. JOSS. 2022;7(69):3994. https://doi.org/10.21105/joss.03994
8. Liepe J, Filippi S, Komorowski M, Stumpf MPH. Maximizing the information content of experiments in systems biology. PLoS Comput Biol. 2013;9(1):e1002888. https://doi.org/10.1371/journal.pcbi.1002888 PMID: 23382663
9. Babtie AC, Kirk P, Stumpf MPH. Topological sensitivity analysis for systems biology. Proc Natl Acad Sci. 2014;111(52):18507–12. https://doi.org/10.1073/pnas.1414026112 PMID: 25512544
10. Brummer AB, Xella A, Woodall R, Adhikarla V, Cho H, Gutova M, et al. Data driven model discovery and interpretation for CAR T-cell killing using sparse identification and latent variables. Fronti Immunol. 2023;14:1115536. https://doi.org/10.3389/fimmu.2023.1115536 PMID: 37256133

11. Mangan NM, Askham T, Brunton SL, Kutz JN, Proctor JL. Model selection for hybrid dynamical systems via sparse regression. Proc R Soc A: Math Phys Eng Sci. 2019;475(2223):20180534. https://doi.org/10.1098/rspa.2018.0534
12. Nardini JT, Baker RE, Simpson MJ, Flores KB. Learning differential equation models from stochastic agent-based model simulations. J R Soc Interface. 2021;18(176):20200987. https://doi.org/10.1098/rsif.2020.0987 PMID: 33726540
13. Nieves E, Dandekar R, Rackauckas C. Uncertainty quantified discovery of chemical reaction systems via Bayesian scientific machine learning. Fron Syst Biol. 2024;4:1338518. https://doi.org/10.3389/fsysb.2024.1338518
14. Woodall RT, Esparza CC, Gutova M, Wang M, Cunningham JJ, Brummer AB, et al. Model discovery approach enables noninvasive measurement of intra-tumoral fluid transport in dynamic MRI. APL Bioeng. 2024;8(2):026106. https://doi.org/10.1063/5.0190561 PMID: 38715647
15. Lagergren JH, Nardini JT, Michael Lavigne G, Rutter EM, Flores KB. Learning partial differential equations for biological transport models from noisy spatio-temporal data. Proc R Soc A: Math Phys Eng Sci. 2020;476(2234):20190800. https://doi.org/10.1098/rspa.2019.0800 PMID: 32201481
16. Nardini JT, Lagergren JH, Hawkins-Daarud A, Curtin L, Morris B, Rutter EM, et al. Learning equations from biological data with limited time samples. Bull Math Biol. 2020;82(9):119. https://doi.org/10.1007/s11538-020-00794-z PMID: 32909137
17. Reinbold PAK, Gurevich DR, Grigoriev RO. Using noisy or incomplete data to discover models of spatiotemporal dynamics. Phys Rev E. 2020;101(1):010203. https://doi.org/10.1103/PhysRevE.101.010203 PMID: 32069592
18. Messenger DA, Bortz DM. Weak SINDy: Galerkin-based data-driven model selection. Multiscale Model Simul. 2021;19(3):1474–97. https://doi.org/10.1137/20m1343166 PMID: 38239761
19. Messenger DA, Bortz DM. Weak SINDy for partial differential equations. J Comput Phys. 2021;443:110525. https://doi.org/10.1016/j.jcp.2021.110525 PMID: 34744183
20. De Florio M, Kevrekidis I, Karniadakis G. AI-Lorenz: a physics-data-driven framework for black-box and gray-box identification of chaotic systems with symbolic regression. 2023.
21. Ahmadi Daryakenari N, De Florio M, Shukla K, Karniadakis GE. AI-Aristotle: a physics-informed framework for systems biology gray-box identification. PLoS Comput Biol. 2024;20(3):e1011916. https://doi.org/10.1371/journal.pcbi.1011916 PMID: 38470870
22. Rackauckas C, Ma Y, Martensen J, Warner C, Zubov K, Supekar R, et al. Universal differential equations for scientific machine learning. 2021. https://doi.org/10.21203/rs.3.rs-55125/v1
23. Chen R, Rubanova Y, Bettencourt J, Duvenaud D. Neural ordinary differential equations. Adv Neural Inf Process Syst. 2018.
24. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045
25. Lagergren JH, Nardini JT, Baker RE, Simpson MJ, Flores KB. Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. PLoS Comput Biol. 2020;16(12):1–29. https://doi.org/10.1371/journal.pcbi.1008462 PMID: 33259472
26. Roesch E, Rackauckas C, Stumpf MPH. Collocation based training of neural ordinary differential equations. Stat Appl Genet Mol Biol. 2021;20(2):37–49. https://doi.org/10.1515/sagmb-2020-0025 PMID: 34237805
27. Sha Y, Qiu Y, Nie Q. NeuralGene: Inferring gene regulation and cell-fate dynamics from neural ODEs. J Mach Learn Model Comput. 2023;4(3):1–15. https://doi.org/10.1615/jmachlearnmodelcomput.2023047369
28. Qin T, Wu K, Xiu D. Data driven governing equations approximation using deep neural networks. J Comput Phys. 2019;395:620–35. https://doi.org/10.1016/j.jcp.2019.06.042
29. Chen Z, Churchill V, Wu K, Xiu D. Deep neural network modeling of unknown partial differential equations in nodal space. J Comput Phys. 2022;449:110782. https://doi.org/10.1016/j.jcp.2021.110782
30. Lotka AJ. Analytical note on certain rhythmic relations in organic systems. Proc Natl Acad Sci. 1920;6(7):410–5. https://doi.org/10.1073/pnas.6.7.410 PMID: 16576509
31. Volterra V. Fluctuations in the abundance of a species considered Mathematically. Nature. 1926;118(2972):558–60. https://doi.org/10.1038/118558a0
32. Elowitz MB, Leibler S. A synthetic oscillatory network of transcriptional regulators. Nature. 2000;403(6767):335–8. https://doi.org/10.1038/35002125 PMID: 10659856
33. McDermott M, Mehta R, Roussos Torres ET, MacLean AL. Modeling the dynamics of EMT reveals genes associated with pan-cancer intermediate states and plasticity. 2024. https://doi.org/10.1101/2024.10.03.616309

34. Lee D, Jayaraman A, Kwon JS. Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling. PLoS Comput Biol. 2020;16(12):e1008472. https://doi.org/10.1371/journal.pcbi.1008472 PMID: 33315899
35. Hamilton F, Lloyd AL, Flores KB. Hybrid modeling and prediction of dynamical systems. PLoS Comput Biol. 2017;13(7):e1005655. https://doi.org/10.1371/journal.pcbi.1005655 PMID: 28692642
36. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17(3):261–72. https://doi.org/10.1038/s41592-019-0686-2 PMID: 32015543
37. Ansel J, Yang E, He H, Gimelshein N, Jain A, Voznesensky M, et al. PyTorch 2: faster machine learning through dynamic python bytecode transformation and graph compilation. In: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. La Jolla CA USA: ACM; 2024. p. 929–47.
38. Lienen M, Günnemann S. Torchode: a parallel ODE solver for PyTorch. 2023.
39. Kingma DP, Ba J. Adam: a method for stochastic optimization. 2017.
40. Mangan NM, Kutz JN, Brunton SL, Proctor JL. Model selection for dynamical systems via sparse regression and information criteria. Proc Math Phys Eng Sci. 2017;473(2204):20170009. https://doi.org/10.1098/rspa.2017.0009 PMID: 28878554
41. Cook DP, Vanderhyden BC. Context specificity of the EMT transcriptional response. Nat Commun. 2020;11(1):2142. https://doi.org/10.1038/s41467-020-16066-2
42. Panchy N, Watanabe K, Takahashi M, Willems A, Hong T. Comparative single-cell transcriptomes of dose and time dependent epithelial-mesenchymal spectrums. NAR Genom Bioinform. 2022;4(3):lqac072. https://doi.org/10.1093/nargab/lqac072 PMID: 36159174
43. van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, et al. Recovering gene interactions from single-cell data using data diffusion. Cell. 2018;174(3):716-729.e27. https://doi.org/10.1016/j.cell.2018.05.061 PMID: 29961576
44. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19(1):15. https://doi.org/10.1186/s13059-017-1382-0 PMID: 29409532
45. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics. 2007;8(1):118–27. https://doi.org/10.1093/biostatistics/kxj037 PMID: 16632515
46. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep. 2019;9(1):5233. https://doi.org/10.1038/s41598-019-41695-z PMID: 30914743
47. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ. Diffusion pseudotime robustly reconstructs lineage branching. Nat Methods. 2016;13(10):845. https://doi.org/10.1038/nmeth.3971 PMID: 27571553
48. Lambert AW, Weinberg RA. Linking EMT programmes to normal and neoplastic epithelial stem cells. Nat Rev Cancer. 2021;21(5):325–38. https://doi.org/10.1038/s41568-021-00332-6 PMID: 33547455
49. Nieto MA, Huang RY-J, Jackson RA, Thiery JP. EMT: 2016. Cell. 2016;166(1):21–45. https://doi.org/10.1016/j.cell.2016.06.028 PMID: 27368099
50. Malagoli Tagliazucchi G, Wiecek AJ, Withnell E, Secrier M. Genomic and microenvironmental heterogeneity shaping epithelial-to-mesenchymal trajectories in cancer. Nat Commun. 2023;14(1):789. https://doi.org/10.1038/s41467-023-36439-7 PMID: 36774358
51. MacLean AL, Hong T, Nie Q. Exploring intermediate cell states through the lens of single cells. Curr Opin Syst Biol. 2018;9:32–41. https://doi.org/10.1016/j.coisb.2018.02.009 PMID: 30450444
52. Sha Y, Haensel D, Gutierrez G, Du H, Dai X, Nie Q. Intermediate cell states in epithelial-to-mesenchymal transition. Phys Biol. 2019;16(2):021001. https://doi.org/10.1088/1478-3975/aaf928 PMID: 30560804
53. Lahav G, Rosenfeld N, Sigal A, Geva-Zatorsky N, Levine AJ, Elowitz MB, et al. Dynamics of the p53-Mdm2 feedback loop in individual cells. Nat Genet. 2004;36(2):147–50. https://doi.org/10.1038/ng1293 PMID: 14730303
54. Wilkinson DJ. Stochastic modelling for quantitative description of heterogeneous biological systems. Nat Rev Genet. 2009;10(2):122–33. https://doi.org/10.1038/nrg2509 PMID: 19139763
55. Hirata H, Yoshiura S, Ohtsuka T, Bessho Y, Harada T, Yoshikawa K, et al. Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop. Science. 2002;298(5594):840–3. https://doi.org/10.1126/science.1074560 PMID: 12399594
56. Ochi S, Imaizumi Y, Shimojo H, Miyachi H, Kageyama R. Oscillatory expression of Hes1 regulates cell proliferation and neuronal differentiation in the embryonic brain. Development. 2020;147(4):dev182204. https://doi.org/10.1242/dev.182204 PMID: 32094111

57. Hoffmann A, Levchenko A, Scott ML, Baltimore D. The IκB-NF-κB signaling module: temporal control and selective gene activation. Science. 2002;298(5596):1241–45. https://doi.org/10.1126/science.1071914 PMID: 12424381

58. Pokhilko A, Fernández AP, Edwards KD, Southern MM, Halliday KJ, Millar AJ. The clock gene circuit in Arabidopsis includes a repressilator with additional feedback loops. Mol Syst Biol. 2012;8574. https://doi.org/10.1038/msb.2012.6 PMID: 22395476

59. Brockwell PJ, Davis RA. Time series: Theory and methods. 2nd ed. Springer Series in Statistics. New York Berlin Heidelberg: Springer; 1991.

60. Frusque G, Fink O. Robust time series denoising with learnable wavelet packet transform. 2022.

61. Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. NIPS'11. Red Hook, NY, USA: Curran Associates Inc.; 2011. p. 2546–54.

62. Feurer M, Hutter F. Hyperparameter optimization. In: Hutter F, Kotthoff L, Vanschoren J, editors. Automated Machine Learning. Cham: Springer International Publishing; 2019. p. 3–33.

63. Zheng P, Askham T, Brunton SL, Kutz JN, Aravkin AY. A Unified Framework for Sparse Relaxed Regularized Regression: SR3. IEEE Access. 2019;7:1404–23. https://doi.org/10.1109/access.2018.2886528

64. Champion K, Zheng P, Aravkin AY, Brunton SL, Kutz JN. A unified sparse optimization framework to learn parsimonious physics-informed models from data. IEEE Access. 2020;8:169259–71. https://doi.org/10.1109/access.2020.3023625

65. Mitra R, MacLean AL. RVAgene: generative modeling of gene expression time series data. Bioinformatics. 2021;37(19):3252–62. https://doi.org/10.1093/bioinformatics/btab260 PMID: 33974008

66. Marku M, Pancaldi V. From time-series transcriptomics to gene regulatory networks: a review on inference methods. PLoS Comput Biol. 2023;19(8):e1011254. https://doi.org/10.1371/journal.pcbi.1011254 PMID: 37561790

67. Ding J, Bar-Joseph Z. Analysis of time-series regulatory networks. Curr Opin Syst Biol. 2020;21:16–24. https://doi.org/10.1016/j.coisb.2020.07.005

68. Kaheman K, Brunton SL, Nathan Kutz J. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. Mach Learn: Sci Technol. 2022;3(1):015031. https://doi.org/10.1088/2632-2153/ac567a

69. Course K, Nair PB. State estimation of a physical system with unknown governing equations. Nature. 2023;622(7982):261–7. https://doi.org/10.1038/s41586-023-06574-8 PMID: 37821594