From Synthetic Data to Equation Recovery: A Computational Framework for Stochastic Biological Networks

Nolan Brown & Katherina Martinez

PHYS230 Final Project

Dynamical behavior in biological systems

- Biological systems - inherently dynamic in that they can be affected by external and/or internal noise
- These systems are governed by complex reaction networks that evolve stochastically overtime
- Experimental observation at a great resolution is challenging - gap in our ability to observe continuous dynamics of interacting components!
- How about modeling? Current approaches often trade off between accuracy and interpretability
- Can we have an approach that will minimize this trade off?

What has been done vs what we are trying to do

- Stochastic simulations capture biological noise but produce “black-box” trajectories that can obscure governing equations
- Symbolic regression excels at discovering deterministic equations can we use this for uncovering reaction dynamics?
- Can we bridge these two together?
    - Can put in tools that make symbolic regression viable here?

How can automated stochastic reaction networks and symbolic regression assist equation discovery from noisy biological data?

Project aims

1. Create reaction networks in an automated fashion.
2. Generate synthetic time-series data from the created reaction network using a stochastic differential equation.
3. Utilize the time series data to infer the original reaction network and equations.

Methods: Generating Reaction Networks

- Reaction networks can be generated randomly with some controllable parameters, e.g.
  - Number of species
  - Number of reactions
  - Reaction complexity
- System is restricted to first order differential equations

$$\frac{dx_0}{dt} = -3.352x_0^2 + 0.947x_1$$
$$\frac{dx_1}{dt} = 0.655x_1x_2^2$$
$$\frac{dx_2}{dt} = 3.352x_0^2 - 1.31x_1x_2^2$$

SymPy

Methods: Solving Stochastic Differential Equations

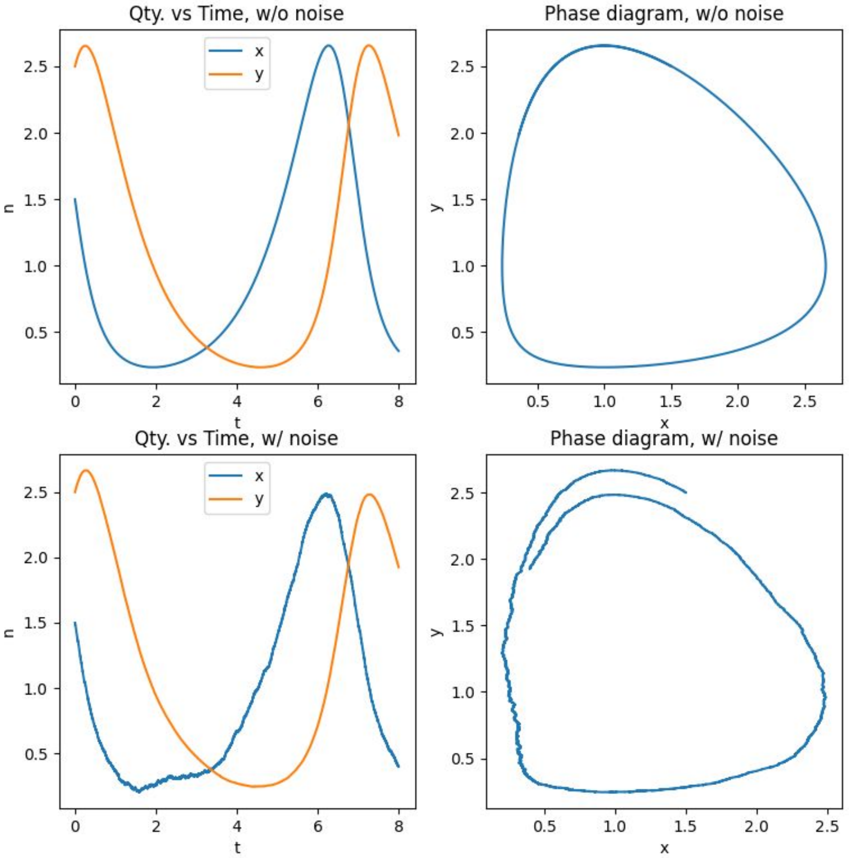
- Integrate using Euler's method

$$X(t+dt) = X(t) + \frac{dX}{dt} * dt + \eta$$

- Example: Lotka-Volterra

$$\frac{dx}{dt} = \alpha x - \beta xy$$

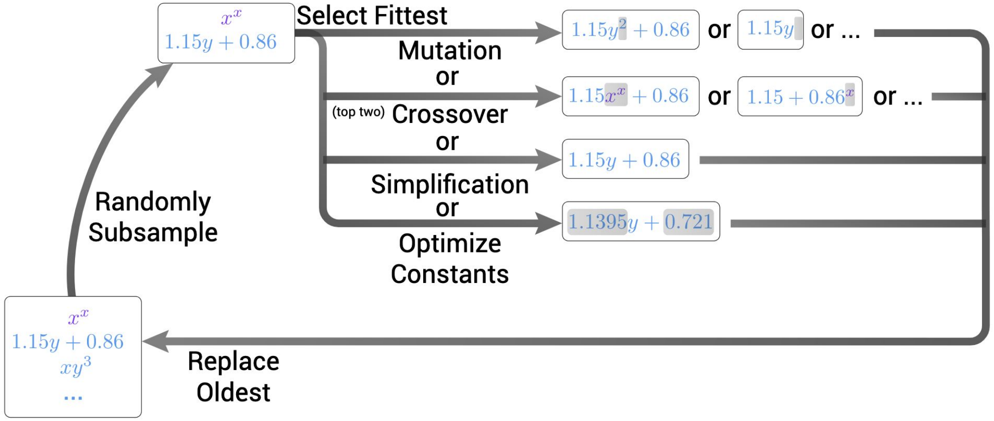
$$\frac{dy}{dt} = -\gamma y + \delta xy$$

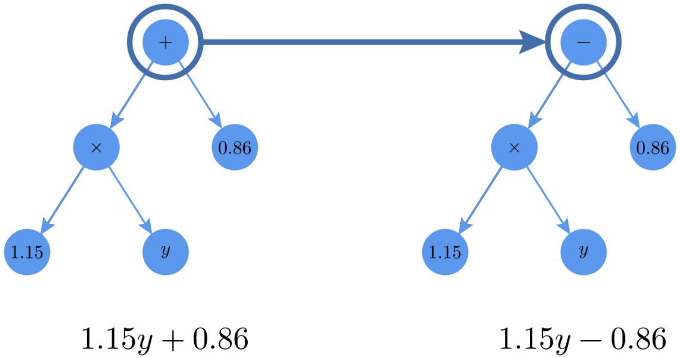


Methods: Symbolic Regression - Overview

- "...a type of machine learning which aims to discover human-interpretable symbolic models" - Prof. Miles Cranmer
- PySR

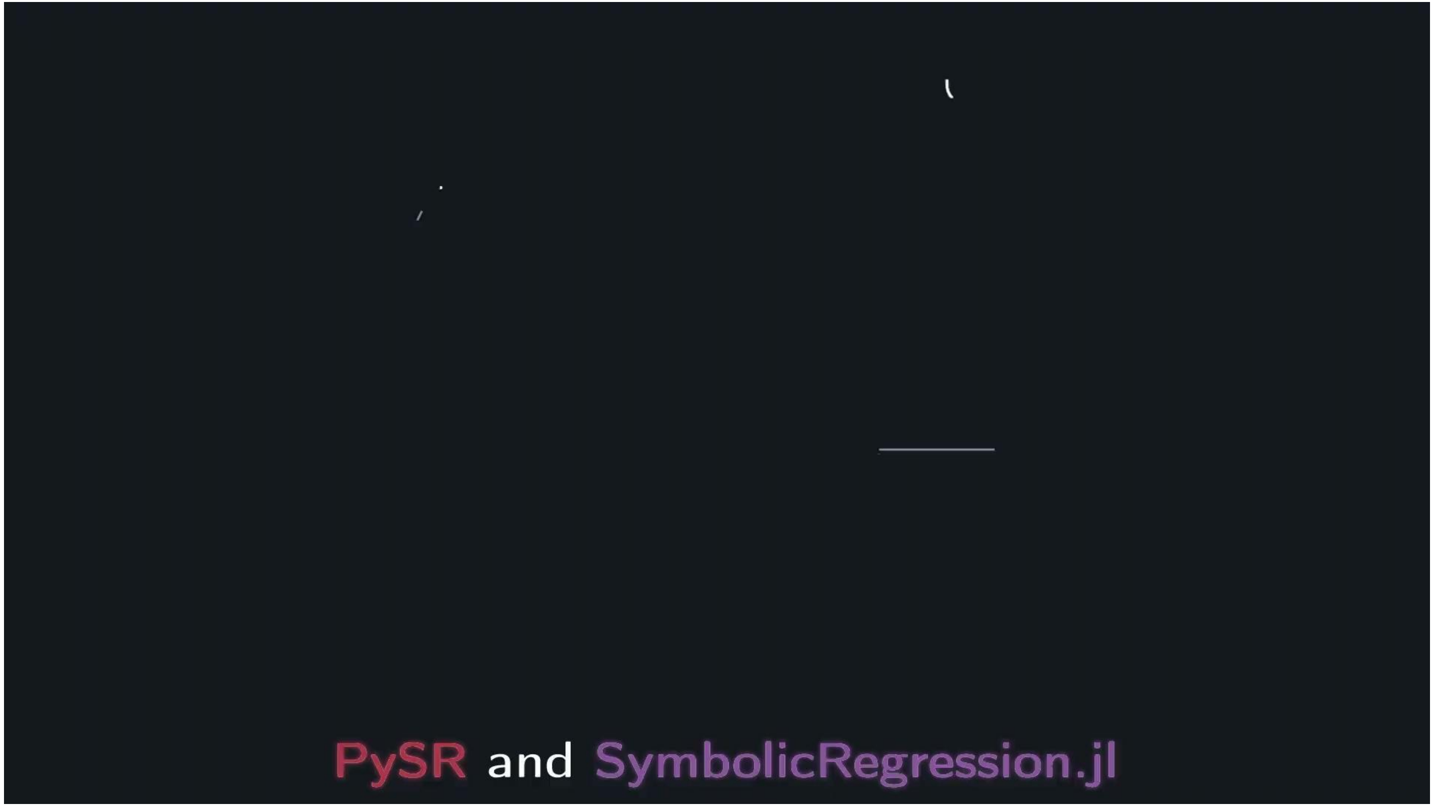
$$\frac{dX}{dt} = f(X, t)$$





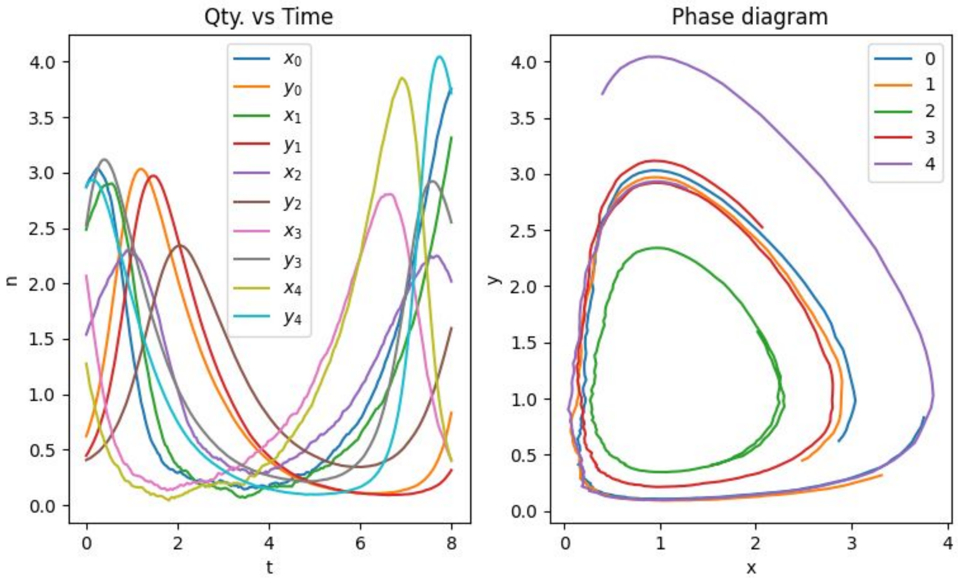
(Cranmer, ArXiv, 2023)

Methods: Symbolic Regression - Video



https://github.com/MilesCranmer/PySR

Methods: Symbolic Regression - Searching



$$\frac{dx}{dt} = \alpha x - \beta x y$$

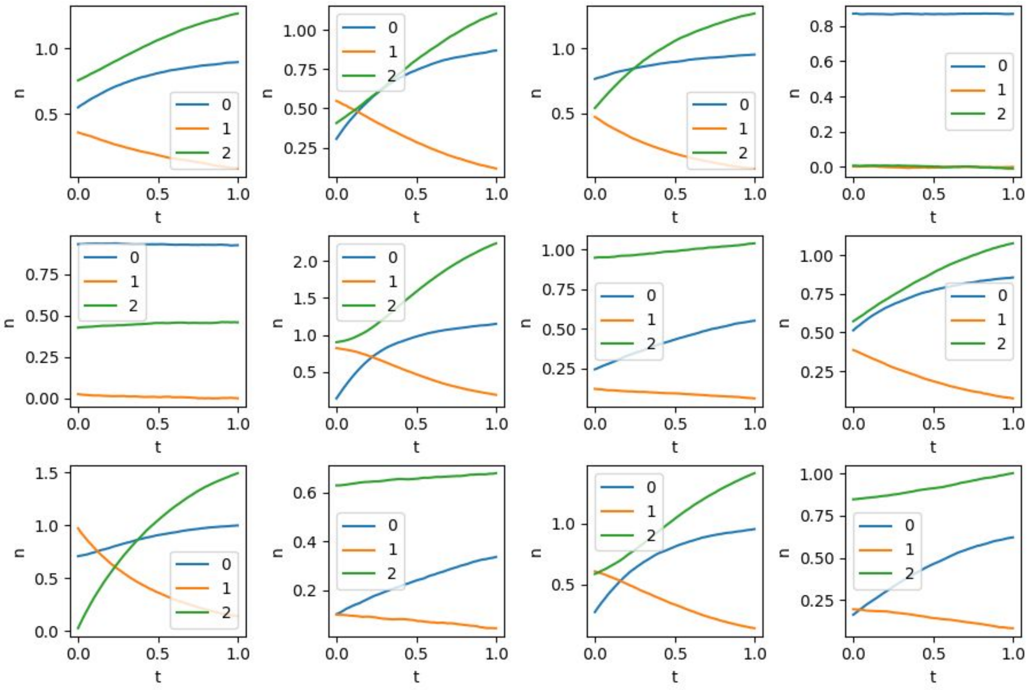
$$\frac{dy}{dt} = -\gamma y + \delta x y$$

	complexity	loss	equation	score
0	1	1.260550	0.05352724	0.000000
1	3	0.463843	x0 + -1.008298	0.499878
2	5	0.000647	(x0 + -0.9996065) * x1	3.287549
3	7	0.000645	((x0 + -0.9985899) * x1) + -0.001954513	0.001402
4	9	0.000644	((x1 * 0.0014742549) + (x0 + -1.0032189)) * x1	0.000887
5	11	0.000644	(((x1 * 0.001478863) + (x0 + -1.0032443)) * 1...	0.000099

	complexity	loss	equation	score
0	1	1.499422	-0.008472832	0.000000e+00
1	3	1.025602	x1 * -0.49425724	1.898999e-01
2	5	0.518070	(x1 * -1.03251) + 1.0343899	3.414621e-01
3	7	0.060325	x0 * ((x1 * -0.9981239) + 0.9923181)	1.075185e+00
4	9	0.060324	((x1 * -0.99739236) + 0.99132454) * (x0 + 0.00...	8.868761e-06
5	11	[illegible]	[illegible]	[illegible]

Results

$$\frac{dx_0}{dt} = -4.106x_0^2x_1 + 1.802x_1x_2 + 2.284x_1$$
$$\frac{dx_1}{dt} = -2.053x_0^2x_1 + 0.901x_1x_2 - 1.142x_1$$
$$\frac{dx_2}{dt} = 4.106x_0^2x_1 - 0.901x_1x_2 + 1.142x_1$$



Results

Original Equations:

$$\frac{dx_0}{dt} = -4.106x_0^2x_1 + 1.802x_1x_2 + 2.284x_1$$

$$\frac{dx_1}{dt} = -2.053x_0^2x_1 + 0.901x_1x_2 - 1.142x_1$$

$$\frac{dx_2}{dt} = 4.106x_0^2x_1 - 0.901x_1x_2 + 1.142x_1$$

Found Equations:

$$x_1\left(x_2 \cdot 1.5136646 + \left(x_0 - 0.7900634\right)\left(x_0\left(-3.9374795\right) - 2.886241\right)\right)$$

$$x_1\left(x_0x_0\left(-1.9554864\right) + x_2 \cdot 0.8329185 - 1.1354636\right)$$

$$x_0x_1x_0 \cdot 4.4217706 + x_1\left(x_0x_2\left(-0.80320853\right) + 0.7310002\right)$$

Future Ideas

- Attempt to fit larger networks
- Try to handle larger amounts of noise, possibly introduce pre-SR filtering
- Experiment with different SR configurations such as weighting or fine-tuned complexity limits
- Improve aggregation of final results to improve interpretability
- Incorporate more advance reaction dynamics