Pacific Northwest
NATIONAL LABORATORY

PNNL-33773

Utilizing CellBox to Describe Varied Dynamical Systems

December 2022

Liam S. Mackey
Song Feng
Margaret S. Cheung

U.S. DEPARTMENT of ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-33773

Utilizing CellBox to Describe Varied Dynamical Systems

December 2022

Liam S. Mackey
Song Feng
Margaret S. Cheung

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Utilizing CellBox to describe varied dynamical systems

Liam S. Mackey [1,2,3], Song Feng [2], Margaret S. Cheung [1,2,4]

Environmental Molecular Sciences Laboratory [(1)], Pacific Northwest National Laboratory [(2)], Rensselaer Polytechnic Institute [(3)], University of Washington [(4)]

CellBox is a hybrid modeling framework that integrates machine-learning approaches with explicit mathematical modeling to determine the behavior of dynamical systems, intended for determining the complex interactions of molecule and phenotype within a cellular environment using perturbation/response-based data. The original work utilized analysis of a melanoma cell line, SK-Mel-133, which we sought to expand to and evaluate effectiveness in describing other nonlinear systems both within and outside of biology. We evaluated with two models. One model is based upon the free-fall of an object under a velocity-dependent drag force; this produced inherent perturbation/response data. The other model is based upon a form of mass-action molecular kinetics; for this, we ran the simulation to steady state in each species for each perturbation and select the steady state as input for CellBox. We determined that CellBox, outside of its original use case, can still strongly predict even non-biological dynamical systems in the small-perturbation regime, with $r = 0.99977$ to 0.85123, $p < 0.05$ for freefall. However, CellBox cannot accurately predict behavior within large-perturbation regimes, $p >> 0.05$ for freefall. In addition, we performed a naïve comparison of CellBox network structure to known network structure in the mass-action model to assess accuracy.

I. INTRODUCTION

The study of dynamical systems is key to our understanding of the natural world. In most general terms, a dynamical system is a description of evolutions of quantities by their relations, with mathematical form as a set of differential equations that take an evolution function $\Phi(t,x)$ and use it to map an initial state x to a unique image in the phase space X dependent on the evolution parameter t (typically time, in real-world applications). In biology, dynamical systems can be used to describe the evolution and behavior of systems as widespread as anti-clotting countermeasures in the bloodstream or the replication of cells under low-light conditions. Thus, in terms of biofuel production, predator/prey dynamics in biosphere self-regulation, homeostasis in the human body, targeted pharmaceutical effectiveness, and a thousand more disciplines besides, understanding dynamical systems is key to the U.S. Department of Energy's (DOE) Biological and Environmental Research (BER) mission.

However, dynamical systems at small scales tend to evade understanding. Due to the inherent and immense complexity of biological systems, building a complete understanding of them is impossible in practical terms; thus, we looked instead to machine learning models in order to create mathematical descriptions of biological systems, allowing us to predict future behavior of systems and successfully design experiments to probe new frontiers in these systems. Recent research at Pacific Northwest National Laboratory (PNNL)'s Environmental Molecular Science Laboratory (EMSL), a DOE User Facility, has paid significant attention to this interdisciplinary problem straddling the boundaries between computational science and the biological sciences, of which this paper is a part.

New innovations in the development of accurate models for dynamical systems have been made outside PNNL in recent years, including the development of the hybrid machine learning

model CellBox by Yuan *et al.* in 2021.[1] Originally developed with the SK-Mel-133 melanoma cell line and trained relative to understanding how cancer treatment drugs affected the melanoma system, CellBox can also be utilized for the purpose of evaluating other dynamical systems, which this research represents some of the first steps in examining. However, before applying CellBox to systems we do not fully understand (i.e., real-world systems), we must first assess its efficacy, precision, and accuracy using simulated model data in order to both prove that it is useful in the regimes and systems under consideration and evaluate how to best curate datasets in order to train CellBox models.

II. METHOD

In order to examine the within-model accuracy and multi-model applicability of CellBox, two models were constructed for the purposes of generating simulated data for use in CellBox. Model 1 was developed as a simple mathematical model off of known physics and calculated using SciPy's[2] ODE module, while Model 2 was developed based off of a mass-action kinetic model example in the documentation of the Python package SBbadger.[3] Once data was generated, CellBox was trained on each model through the use of the EMSL High-Performance Computing program's Tahoma computer.

The CellBox data input format is composed of three separate comma-separated value (csv) files, with one configuration JavaScript Object Notation (json) file acting as a container for the settings required to run a CellBox simulation. The data files were organized as expression (expr), perturbation (pert), and node-index files, while the json file contains various parameters such as the names of each of the three data files, the partitioning for training and validation sets

for the machine learning process, the numbers of nodes in each of three categories, and similar. In our case, the json file was reproduced from the provided configuration file CellBox is packaged with, with minor changes in order to suit the shape of a particular experiment's data and the names of its files.

Beginning with the simplest of these files, a CellBox node-index file is a single-column csv containing the names of all nodes in the network, one per row. For ease of use in pert and expr files, values for CellBox training were expressed as changes off of the initial values of the model before perturbation – in the case of Model 1, this was by simple subtraction, and in Model 2, the log2 fold change (FC). The following table showcases the format of CellBox expr and pert files.

Table 1: CellBox format example.

	Active node	Active node	Perturbation node	Perturbation node
Sample 1	x	x	x	x
Sample 2	x	x	x	x
Sample 3	x	x	x	x
Sample 4	x	x	x	x

For expr files, the csv file represents the observed, "expressed" response data in the perturbation/response set. For pert files, we took the start of the experiment to be the baseline – that is, we normalized to 0 such that all values for active nodes in the pert file are 0, and the perturbations result in expressed changes relative to 0.

The categories of nodes in CellBox's json file are protein nodes, active nodes (protein + phenotype), and n_x, or total nodes (protein + phenotype + perturbation). No phenotype nodes were used in either of the two models under consideration in this paper.

A. Model 1

Model 1 represented free-fall through a resistive medium, a classical example of a nonlinear dynamical system. The model was written in Python such that it took as input a series of three values – initial height (m), initial velocity (m/s), and drag coefficient (unitless) – and returned as output the time to impact (s) of the object in free-fall. For each of these initial conditions, three values were provided, thus producing 3^3 or 27 samples. In CellBox format, our three initial conditions were our perturbation nodes, while our active node was time to impact.

Initially, our perturbations were on the order of 200m for each step in height, 5 m/s in velocity, and 0.01 in drag coefficient. As will be discussed in our results section, this produced poor results, and thus perturbations were lowered greatly in scale to 1m-1 m/s-0.001, hereafter referred to as the "small-perturbation regime". After achieving significantly better results there, perturbation scale was slowly raised over the course of three more trials to 5m-4 m/s-0.005.

Prediction data was recovered from CellBox in initial condition/prediction form, and the initial conditions were then fed back into the original Model 1 in order to determine the actual output value for the given initial condition. Pearson's R test was then run between the set of actual output values and CellBox's predicted values using the SciPy stats module to

produce a quantifiable value for correlation, and the SciPy orthogonal distance regression (ODR) module was used to fit the data as shown in the results section.

B. Model 2

This model represented standard mass-action kinetics in a biological system, and was generated based in part off of examples available through SBbadger's documentation.[4] Models were repeatedly generated using SBbadger until ten models that reached steady state over a simulation timescale (using the tellurium Python package) of 1000 were generated. All models were recorded as Antimony strings for future loading and human-readable understanding of reaction rules. One model was then selected from this set, whose network is shown in the following figure.

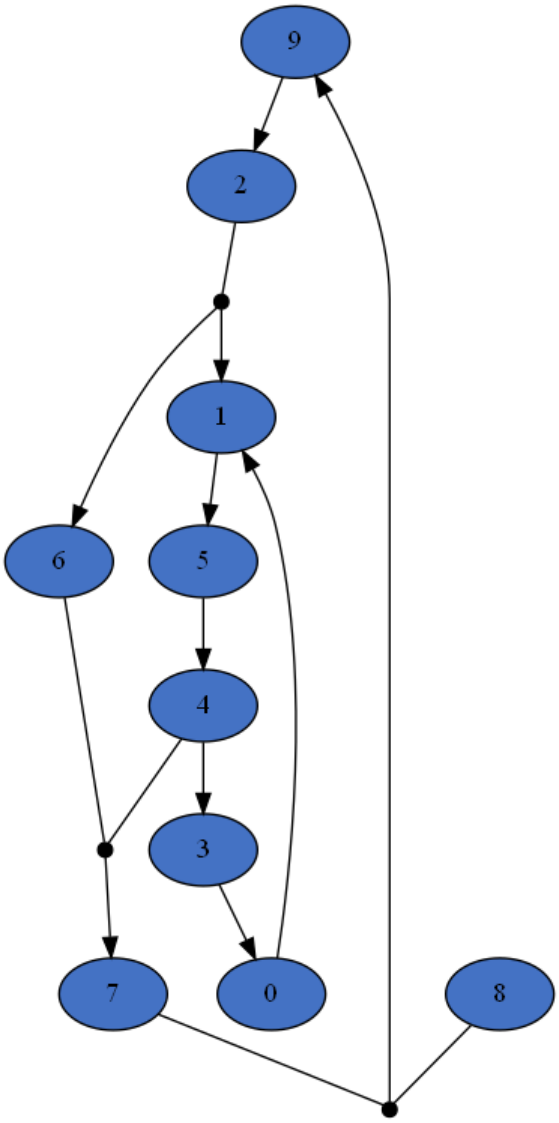

Fig. 1: The network/node architecture of Model 2; all numbers X here represent species SX (i.e., species S0 is represented as 0).

The reaction rules for this network are supplied below. Here, our species are denoted by S, and our reaction rate and equilibrium constants by kc/kr/kf (equilibrium constant/reverse reaction rate constant/forward reaction rate constant respectively).

J0: S9 -> S2; kc0*S9

J1: S8 + S7 -> S9; kc1*S8*S7

J2: S2 -> S1 + S6; kc2*S2

J3: S0 -> S1; kf3*S0 - kr3*S1

J4: S6 + S4 -> S7; kc4*S6*S4

J5: S4 -> S3; kc5*S4

J6: S5 -> S4; kc6*S5

J7: S1 -> S5; kf7*S1 - kr7*S5

J8: S3 -> S0; kf8*S3 - kr8*S0

SBbadger returns time-series data as opposed to the perturbation/response data required by CellBox; thus, running the model to steady state was vital, as it would be the steady state concentrations of the nine relevant species (S0, S1, S2… S7, S9) taken as samples for CellBox's data format. We note that as S8 was fixed and did not change in concentration, it would not be included in the data files for training. Data was collected by perturbing the equilibrium constants kc0, kc1, kc2, kc4, kc5, and kc6 from 5% to 25% increase in steps of 5%, individually and in groups, for a dataset comprising 15,625 samples. As this model could not directly be compared to CellBox's predictions for correlational purposes, a confusion matrix was instead constructed with true positive/true negative/false positive/false negative values based off of CellBox's estimations of its W parameter, which governs the strength of interactions between species and thus could directly be compared to the known interactions from the network model in Fig. 1.

Later, as this model produced poor results, we produced two more datasets, one that perturbed kc0, kc1, kc2, and kc4 from a -4-fold change to a +4-fold change without including "no perturbation" as an option, and one which perturbed kc0, kc1, and kc2

from a -4-fold change to a +4-fold change with “no perturbation” included as an option. These were composed of 1296 and 343 samples respectively, and evaluation of their efficacy is still undergoing as of this writing.

III. Results

A. Model 1

Beginning with Model 1, we observed that in the initial large-perturbation regime, results were very poor correlationally between CellBox’s predictions and the model’s actual output for CellBox’s generated initial conditions, with $p >> 0.05$ in Pearson’s R test implying no statistical relevance. Later models, however, showed both very high positive correlation (r near 1) and $p < 0.05$, confirming statistical relevance. Figures 2, 3, 4, and 5 show the results of these tests, listed in order of smallest to largest perturbation of initial conditions.

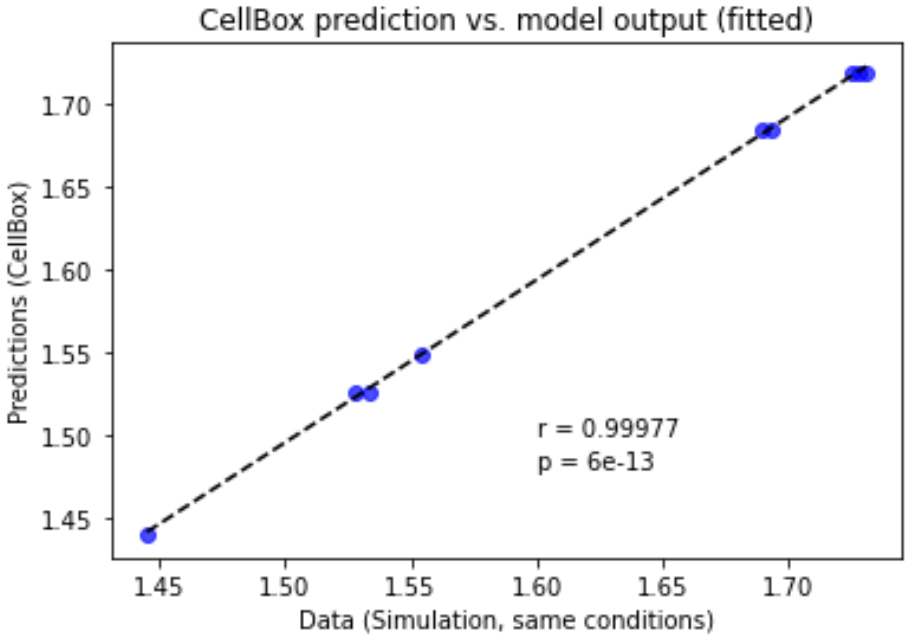

Fig. 2: Small-perturbation regime correlation graph for Model 1.

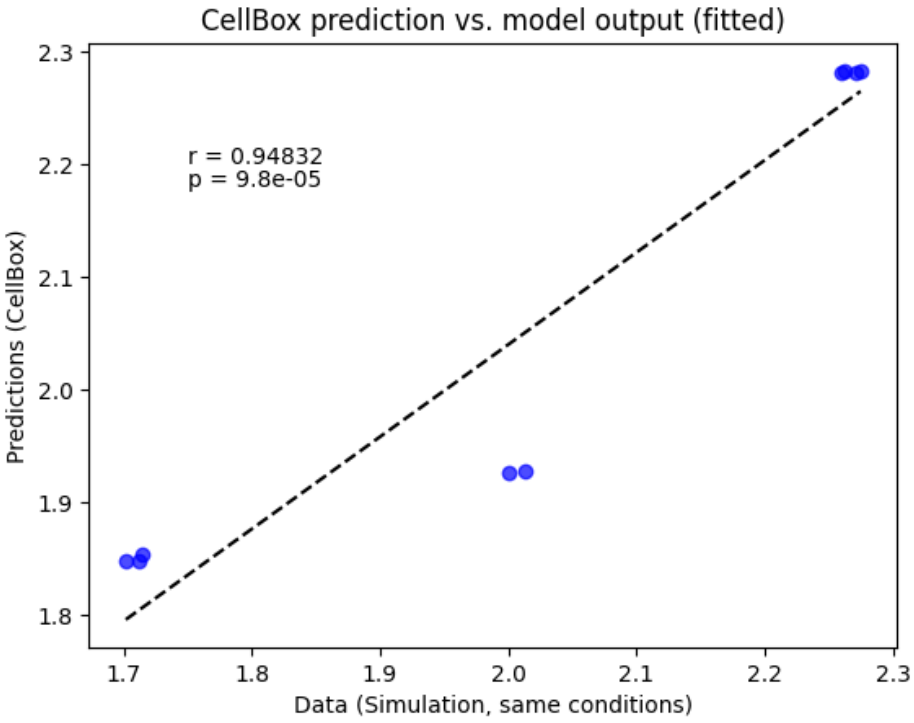

Fig. 3: First larger-perturbation correlation graph for Model 1.

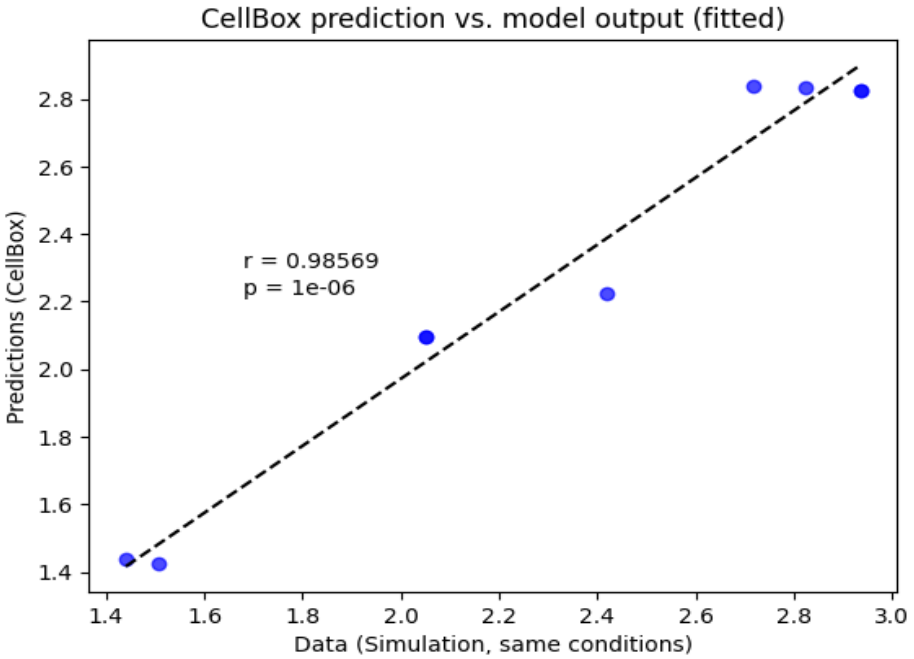

Fig. 4: Second larger-perturbation correlation graph for Model 1.

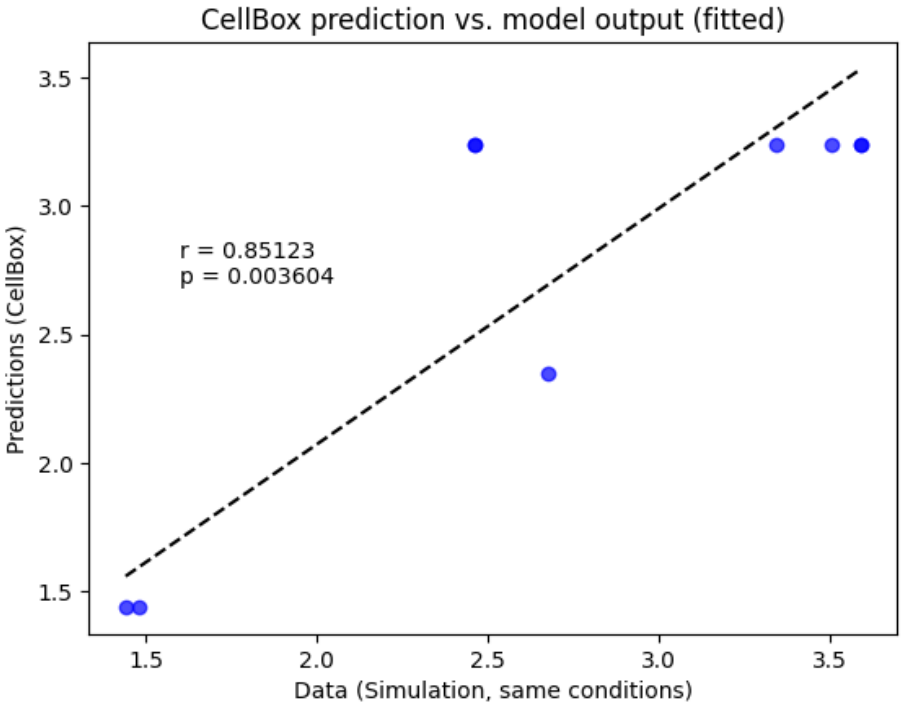

Fig. 5: Last larger-perturbation correlation graph for Model 1.

These results showcase that not only is CellBox capable of operating effectively with dynamical systems models outside of its original application – and outside of biology itself – it also decreases in accuracy notably with the size of perturbations in its input data, beginning to break down entirely in terms of predictive efficacy of its model when perturbations grow too large. Further research will explore whether this is a function of the absolute size of a perturbation, or the size of a perturbation as relative to its initial baseline value for the model parameter. Additionally, we find that a naïve analysis of network architecture through examining the W-parameter in CellBox's trained model confirms that CellBox correctly

identifies that there is no relationship between the three initial conditions, but that all three are related to time-to-impact.

B. Model 2

With respect to Model 2, this naïve analysis through the W-parameter is all that is available. As this is a far more complex system, we produce the following form of confusion matrix, shown as Figure 6.

	S0	S1	S2	S3	S4	S5	S6	S7	S9
S0	0	6.49E-07	2.17E-06	1.93E-07	-2.32E-06	-1.17E-06	-1.17E-06	1.65E-06	-9.53E-08
S1	-4.50E-07	0	-3.38E-06	-1.25E-06	-3.08E-06	-1.54E-06	5.34E-07	1.27E-06	-6.74E-07
S2	-2.55E-06	-4.63E-07	0	2.05E-06	-0.03727	-1.14E-07	-1.77E-06	-0.03577	-0.02514
S3	1.07E-06	-1.11E-06	2.95E-06	0	-1.53E-06	-2.00E-07	2.40E-07	2.98E-06	-1.27E-06
S4	-9.90E-07	1.01E-07	0.004013	-1.23E-06	0	-5.12E-07	-2.01E-06	0.003737	0.005007
S5	2.15E-06	-2.10E-06	-1.97E-06	1.20E-06	-3.38E-06	0	-1.71E-06	-9.16E-07	-1.87E-07
S6	7.60E-07	-1.25E-06	-8.35E-07	4.22E-06	-2.54E-06	1.25E-06	0	-1.20E-06	-2.63E-06
S7	-8.89E-07	2.73E-07	-0.04114	-3.34E-07	0.00062	-3.49E-06	8.12E-07	0	-0.04902
S9	-3.51E-06	-2.21E-07	-0.01171	-7.90E-07	0.000641	-8.45E-07	-7.33E-07	-0.04793	0

Fig. 6: A modified confusion matrix of the interaction between species in CellBox, with the value in each cell being the value of the W-parameter quantifying strength of interaction. Thus, if a cell has a high (absolute) value compared to others in its row, it is likely CellBox believes that cell to be an actual connection between species. Here, true positive results are highlighted in green – for example, S1 and S2 interacting – while false positive results are highlighted in red. False negative results – CellBox failing to see an interaction where it exists – are colored red. True negative results have no color.

As can be seen in Fig. 6, the false positive rate is exceptionally high at a glance. To confirm this, we perform some simple quantitative statistics on the model using our matrix;

we calculate the true positive rate, false positive rate, and precision of the model, obtaining a TPR of 0.591, a FPR of 0.551, and a precision of 0.325 or 32.5%. Our true positive rate is indeed higher than our false positive rate, if barely, but the model's precision is very low compared to our threshold for a successful or accurate model, which lies at approximately 80% precision. As such, we can safely discard this model.

IV. DISCUSSION

Notably, the result in Section IIIB reveals the other problem of CellBox and perturbation scale; where, in the Model 1 case, CellBox broke down at large perturbation sizes but operated well in the small-perturbation regime, the Model 2 case showcases that CellBox can also break down when perturbations are too small for the optimization process of the machine learning model to sift meaningful interactions from the supplied data. As well, we theorized that the size of the dataset supplied to CellBox also has influence; if the dataset is too large, the optimization process of CellBox will also be overwhelmed and fail to correctly train the model, compounded by the small perturbation size into an effect similar to training the model on extremely noisy data.

This is also related to an externality of applying CellBox to real biological data, not just the simulated models used for the purposes of this paper; a 5% perturbation in real-world data is on the same scale as the absolute minimum error for data produced in the laboratory, and in terms of assessing differential expression of genes, a log2 fold change of > 1 is typically a sign of perturbation meaningfully affecting a species. In this first dataset, no log2 FC was greater than 1.

Thus, datasets 2 and 3 were created for Model 2 using these lessons, as discussed in Section IIB of this paper. The same analysis as described above (construction of modified confusion matrix and application of TPR/FPR/precision heuristic) is being performed on dataset 2 as of this writing, while CellBox is set to be trained on dataset 3 via EMSL HPC assets and the same analysis performed in the near future.

V. CONCLUSION

We have developed a deeper understanding of the limitations and possibilities of the CellBox hybrid machine learning model as applied to a variety of simulated dynamical systems, types of perturbation, and perturbation scales in preparation for applying CellBox to real-world biological data. We establish that CellBox functions accurately in a small-perturbation regime even outside of biological systems, though care should be taken in the curation of datasets to avoid too-small perturbations and feeding too much data into CellBox in order to increase the precision of the model and avoid false positives. We determine that CellBox's accuracy of prediction scales inversely with the scale of perturbations in the training dataset. Additionally, we propose further directions for research into the capabilities and limitations of CellBox before applying it directly to real-world biological data in service of the DOE BER mission, ranging from further examination of network architecture prediction precision to better assessing the space of allowable perturbations for curating datasets suitable to produce high-accuracy CellBox models.

VI. ACKNOWLEDGEMENTS

With thanks for the invaluable aid of my mentor Margaret S. Cheung and my co-mentor Song Feng; the U.S. Department of Energy, for funding my research; PNNL, for providing the opportunities and facilities to make my research possible; EMSL's HPC resources, for their storage and data processing capabilities; and my family, without whom none of this would have been possible.

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

VII. REFERENCES

1. Yuan, B., Shen, C., Luna, A., Korkut, A., Marks, D. S., Ingraham, J., & Sander, C. (2021). Cellbox: Interpretable machine learning for perturbation biology with application to the design of cancer combination therapy. Cell Systems, 12(2). https://doi.org/10.1016/j.cels.2020.11.013

2. Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272.

3. Michael A Kochen, H Steven Wiley, Song Feng, Herbert M Sauro, SBbadger: biochemical reaction networks with definable degree distributions, Bioinformatics, Volume 38, Issue 22, 15 November 2022, Pages 5064–5072, https://doi.org/10.1093/bioinformatics/btac630

4. https://sbbadger.readthedocs.io/en/latest/examples.html

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov