Unpacking the role of complexity in multi-class models of the tumor microenvironment

Jason Y. Cain

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Neda Bagheri, Chair

David Beck

Cole DeForest

Elizabeth Nance

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Unpacking the role of complexity in multi-class models of the tumor microenvironment

Jason Y. Cain

Chair of the Supervisory Committee:
Associate Professor Neda Bagheri
Biology and Chemical Engineering

Cancer is an emergent phenotype generally resulting from the disregulation of cell decision processes as well as the disregulation of cell interactions with their microenvironment. However, the tumor microenvironment (TME) is difficult to probe and experimentally interrogate. This relative inaccessibility motivates the construction of computational models to generate hypothesies and design high impact experiments. Efforts to study this system holistically neccessitate considerations of the resulting complexity in the compuational models describing it. Leveraging diverse modeling paradigms can generate in silico approaches towards biological parity, but the compounding complexity resulting from this integration must be properly managed.

Agent-based models (ABM) are a popular approach to study emergence in multi-scale systems with complex interactions. ABM modularity provides a means of incorporating mutiple classes of models that can be regulated at different scales in an intuitive manner. However, robust analysis methods remain an open challenge. The multi-class nature of

many ABMs leads to limited application of traditional parameter assessment methods such as sensitivity analysis or optimization, which can present challenges to their validation. Thus, many techniques to analyze ABMs are analagous to those found in high-resolution experimental methods, where ABMs are treated as *in silico* test beds.

I discuss many of the challenges and approaches to studying biological phenomena with complex ABMs, especially in cases with the dynamic coupling between agents and their environment. I highlight a review article discussing both the growing popularity and the resulting challenges of combining modeling paradigms. I then present work leveraging machine learning techniques to emulate the simulation outputs from an ABM towards prioritizing data acquisition and resolution in the TME. We identify a gap between the between spatio-temporal emergent phenomena and information used to build robust emulation models. Counter-intuitively, spatial information confers far less benefit than leveraging temporal information to predict tumor aggression metrics. I also present results towards leveraging ABMs as an platform for translating *in vitro* derived models to *in situ* contexts. I integrate a mechanistic model of hypoxia-induced factors (HIFs), an ubiquitous feature in cancer progression, into a TME ABM. This platform enables predictions of additional regulation requirements for key growth factors.

I conclude with a perspective on the current state of code commonly found in academic journals and methods to improve code and reproducibility, through analogies from experimental biology. I then highlight many possible directions of continuing and extending the

research outlined in this dissertation specifically in the context of emulating agent-based models and characterizing angiognesis in the TME.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ABM: Agent-based model

ARCADE: Agent-based Representation of Cells And Dynamic Environments

HIF: Hypoxia-inducible factor

HRE: Hypoxia-responsive elements

ML: Machine learning

ODE: Ordinary-differential equation

PDE: Partial-differential equation

PHD: Prolyl hydroxylases

SL: Statistical learning

TME: Tumor microenvironment

VEGF: Vascular endothelial growth factor

DEDICATION

for Billy, truly the best of us. Rest in peace.

ACKNOWLEDGMENTS

It certainly took a village.

Neda, thank you for all your endless support and flexibility, no matter the situation (maybe even a little too supportive of my bad ideas). You are, and will continue to be, truly invaluable in my growth as a scientist, teacher, mentor, and communicator. The lab environment that you cultivated and the energy that you brought made everyday fun, even when the science was challenging. You made the difficult conversations easier and gave me the space to be honest—whether the topic was emotional or sociological.

Thank you all the members of the Bagheri lab, you all inspired me to be a better scientist. Talking, learning, laughing, and celebrating with all of you has truly been a pleasure. I would also like to highlight a couple that the work in thesis would not have been possible without: Jess, thank you for setting such a great foundation and helping me think through all my problems. Jacob, thank you for all the hard work you put in through all of my crazy ideas, even when the story felt tenuous.

Erika, thank you for making me take some time off to get chicken nuggies and fixing all my comma splices. You inspire me everyday to take work a little less seriously and replace my stress with your infectious joy. You celebrate all my successes that I was never capable

of acknowledging. You kept me sane and provided me all the motivation and support that I needed to finish. But most of all, you brought Cin into my life.

Billy, I miss you and I cannot believe you're gone. Thank you for believing in me more than any other person had before. I would not have had the confidence to tackle graduate school without your confidence in me. Writing this thesis while grieving you was one of the hardest challenges of my life, but I would not trade my experiences of your love of life and people.

To all the wonderful friends I have made along the way: Trivia with the Jardos, Katie, Justin, and Kirsten; Elsa with her pocket beers; Kaylyn with our cider nights and nacho afternoons; Sophia with her schemes; Kyra and the way too long lunch breaks; Christine and our coffee walks; Hannah and the sad punk shows; Melissa and her love for Cin; Elizabeth with our musical movie nights; Jake, Hannah, Sarah, and Katie who were so welcoming despite my keyboard; and finally the PSSBL Rocky Hops for giving me a desperately needed competitive outlet.

I would also like to thank Peppercorns, Tempesta Market, Sauce and Bread Kitchen, Sun Wah BBQ, Pho Bac Sup, Pho Than Brothers, City Teriyaki, College Inn. Most importantly, I would like to thank Backlot Coffee, Reprise Coffee Roasters, Dark Matter, Analog Coffee, Ghost Note, Onda Origins, Empire Roasters and Records, Overcast Coffee, and Cloud City Coffee. Special thanks to Phoebe Bridgers, Patti Smith, Carly Rae Jepsen, Open Mike Eagle, and Car Seat Headrest.

Chapter 1

INTRODUCTION

1.1 Overview

Computational models allow us to *streamline* science. Need a distribution? Run simulations. Interested in a perturbation? Change parameters. Want to generate hypotheses? Modify behaviors. The philosophical utility of computational models is clear: they save time by shifting the laborious component of running experiments to model development. What happens when models become as complex as the systems they are meant to study?

Currently, computational models offer unrealized potential to predict how the whole of a biological system is different than the sum of its individual parts, but the rising level of complexity—here, interactions, species, parameters, etc.—introduces new challenges. Unpacking biological complexity requires computational models capable of linking emergent behavior to underlying "rules of life". However, as we include more and more interactions there are universal considerations that need to be addressed regardless of the class of model that is being used: (1) Assumptions—comprimises between biological parity and computational abstraction—accumulate throughout the development process; (2) Extracting insight from these models can require layering even more complexity by requiring more sophisticated analysis techniques; (3) Capturing more information and calculations incurs computational

cost by increasing memory and processor loads, respectively. At minimum, these limitations must be acknowledged to truly unlock the potential of computational approaches. In this dissertation, I explore these considerations through agent-based modeling, a framework uniquely suited to modeling emergence.

In **Chapter 2**, I discuss the recent trend in systems computational biology to combine modeling classes or design paradigms together in order to build models that can support novel biological insights. Much of managing complexity in computational models has traditionally been through the deliberate use of specific design paradigms, here presented as `data-driven`, `mechanism-driven`, and `agent-driven`. However, many recent models have been leveraging composite models, combining the utility and benefits of each individual component. While these models demonstrate incredibly promising results, especially those fostering collaborations among computational researchers and with experimental researchers, there are still many open challenges.

In **Chapter 3**, I present research on the composite modeling strategy of emulation. Emulation is a potential avenue to reducing the computational burden of running increasingly complex high-resolution computational models. We find that the predictive power of many machine learning models are fairly limited unless the temporal evolution of the environment is accounted for. The limited performance of these models underscore the importance of building models to learn patterns in spatio-temporal emergence with sufficiently complex computational models.

In **Chapter 4**, I present research about how leveraging ABMs as a platform to extend

the generalizability of *in vitro* experiments associated mechnanistic models. Experimental results, especially those *in vitro*, are difficult to translate to *in vivo* or *in situ* contexts. Using agent-based models, I translate a noteworthy dynamic model of a key oncogene into an ABM and compare the results in different *in silico* contexts. The results help us highlight knowledge gaps in our original models, supporting further hypothesis generation.

In **Chapter 5**, I discuss the links between code quality and reproducibility and strategies to improve code quality. Much of the current literature surrounds the *how* of improving code, but very limited discussion of the *why*. We hope that providing the additional context surrounding better coding strategies would motivate computational scientists to prioritize code quality as part of their scientific process.

In **Chapter 6**, I propose future directions for the research themes in this dissertation.

1.2 Background

1.2.1 The tumor microenvironment

Cancers are complex diseases in which DNA mutations can give rise to unintuitive emergent phenomena, such as uncontrolled tissue growth[1]. The permissive environment that not only facilitates uncontrolled growth but also encourages genetic instability is known as the tumor microenvironment[2,3]—defined here as both the physiological (nutrient gradients, pO_2 , pH) and cellular (healthy, cancer, vasculature) microenvironments. A general pattern of cancer is usually the disruption of feedback loops required to maintain homeostasis and the reinforcement or activation of robust development processes[1,2]. Continuous disruption from

homeostasis can cause malignant cells to escape from their environment with an invasive phenotype, otherwise known as metastasis. Metastasis, not primary tumors, is usually the cause of death in cancer patients[4]. Therefore, metastatic potential is an indicator of the aggressiveness or the severity of disease[3].

Metastasis is complicated, but there are common features of cells and the tumor microenvironment that are associated with an increased aggressiveness[4]. The complex interplay between individual cells and the permissive tumor microenvironment propagates across multiple spatial and temporal scales[5,6]. For example, differential nutrient utilization resulting from phenotypic heterogeneity can alter local tissue environmental contexts[7]. These environmental contexts can promote specific cell phenotypes as a result of nutrient limitations or selective pressure. One such observation was the Warburg effect[8], where cancer cells have a higher affinity towards glycolysis over oxidative phosphorylation, unlike most cells in the human body. This phenomena was originally hypothesized to be a result of mitochondria dysfunction, but recent understanding has indicated that this is a result of hypoxia, or non-physiological levels of oxygen concentration[9,10].

Hypoxia, a key feature of the tumor microenvironment, is exemplary of complex tissue-level phenomena that significantly affects subcellular processeses[11]. As tumors grow, the demand for nutrients increases as higher cell crowding leads to higher competition and potential damage to existing blood vessels. Subsequent oxygen tension manifests in significant intra- and inter-tumor heterogeneity arising out of diffusion-limited oxygen delivery and heterogeneous vasculature[12,13]. This condition, unique to (normal tissue pO_2 are usually

between 4.5–9.4%[11] compared to hypoxia at $< 2\%$) and ubiquitous in pathologenic tumors, drives cells to promote the activity of HIF (hypoxia-inducible factor) and hypoxia associated pathways[14]. In addition to the aforementioned metabolic shift, poor oxygenation and elevated expression of the HIF pathway have correlated to resistance to traditional therapies[15–18], metastasis[4,19], and overall poor patient outcomes[20]. HIF is implicated in many important processes such as angiogenesis, the formation of new blood vessels[21,22]. This developmental process is required to sustain tumor growth, acting as a tissue-level physiological feedback loop by supplying nutrients to nutrient limited cells and promoting cell proliferation[23]. Unfortunately, complexity across spatio-temporal scales has led to an incomplete understanding of this physiological feedback loop. Our abilities to observe and modulate this property *in vivo* or *in vitro* with the resolution and throughput required for robust data-driven approaches remains a grand challenge[24].

1.2.2 Agent-based models

Capturing the dynamics of both inter- and intra-cellular properties and nutrient diffusion is unwieldy; accounting for cellular heterogeneity adds additional, though necessary, complexity. Tissue-level models are often abstracted to continuum models to capture some of the interactions of interest, forgoing resolution for computational efficiency[25]. Agent-based models (ABMs) address these challenges by capturing the interplay between cells and their environment at a single cell resolution[26].

To tackle hypoxia and its consequences, we must consider heterogeneity and the environ-

ment across multiple scales. Cells that share a specific genotype do not necessarily share the same phenotype. Cellular characteristics, like protein expression and responses to stimuli, exhibit significant heterogeneity and are impacted by prior cellular experiences. Many computational methodologies are unable to tractably capture these details[27]. Many continuum models that average population level behavior are unable to capture distinct heterogeneous characteristics like bimodality or switch-like behavior[28]. Capturing the environment by itself is feasible using partial differential equation models, but interactions within a cellular population make abstraction difficult[29]. In order to leverage computational models to study hypoxia and tumor development, we need a methodology that is capable capturing intra- and inter-cellular heterogeneity, switch-like responses to nutrients, phenotypic states, and environmental factors.

ABMs are discrete rule-based models where agent interactions and decisions are determined by discrete rule sets derived from experimental observations[28,30]. ABMs are relatively expensive computationally as a result of the high resolution to capture multiple spatio-temporal scales. Computational cost has historically been a barrier to leveraging ABMs, but powerful tools like cloud computing have redefined the landscape of what is possible with computational models. Hybrid ABMs have emerged as a popular modeling paradigm within the field of cancer modeling[25,31]. In hybrid ABMs, decisions and rules remain discrete, but criteria are frequently represented as continuum models. These ABMs leverage validated continuum models for components like nutrient diffusion and signaling networks[28]. This type of model is uniquely suited to studying oncogenesis as biological development is het-

erogeneous within the dynamic microenvironment and is affected by the introduction of new cells and new cell phenotypes. As an alternative to data driven approaches, ABMs leverage a "bottom-up" modeling approach that is particularly interpretable by not only implementing biologically relevant rules and decisions, but also generating outputs that are intuitive from an experimental perspective[26]. This characteristic facilitates easier collaboration and iteration with experimental collaborators.

ABMs have been used to characterize cancers by studying signaling networks[30,32], metabolism[33], immune interactions[34,35], and metastasis[36]. The increasing popularity of ABMs has resulted in the development of several open-source platforms and libraries to assist in model development[37–39]. As the cancer modeling field progresses, the agent definitions become more and more complex, leading to increased computational cost. In exchange for an increased computational cost, we learn more about the specific processes and patterns in our agent definitions that give rise to emergent phenomena.

Chapter 2

FIGHTING FIRE WITH FIRE: DEPLOYING COMPLEXITY IN COMPUTATIONAL MODELING TO EFFECTIVELY CHARACTERIZE COMPLEX BIOLOGICAL SYSTEMS

Alexis N. Prybutok[1,2†], Jason Y. Cain[3†], Joshua N. Leonard[1,2,4,5,6*], Neda Bagheri[1,2,3,7*]

1 Department of Chemical and Biological Engineering, Northwestern University, Evanston, IL 60208, USA
2 Center for Synthetic Biology, Northwestern University, Evanston, IL 60208, USA
3 Department of Chemical Engineering, University of Washington, Seattle, WA 98195, USA
4 Interdisciplinary Biological Sciences Program, Northwestern University, Evanston, IL 60208, USA
5 Chemistry of Life Processes Institute, Northwestern University, Evanston, IL 60208, USA
6 Robert H. Lurie Comprehensive Cancer Center, Northwestern University, Evanston, IL 60208, USA
7 Department of Biology, University of Washington, Seattle, WA 98195, USA

† These authors contributed equally to this work.
* Co-corresponding authors: j-leonard@northwestern.edu & nbagheri@uw.edu

Author's note: *This work was published in Current Opinion in Biotechnology in June 2022.*[40] *This opinion piece was co-authored by Alexis Prybutok and myself, and we contributed equally to the conceptualization and writing of the article. This work highlights the synergy that exists between different types of modeling paradigms, but also acknowledges the challenges associated with including additional complexity in the model design process. These benefits and tradeoffs are significantly considered and discussed in more detail with the statistical emulation work in chapter 3.*

ABSTRACT

Computational modeling empowers systems biologists to interrogate and understand increasingly complex biological phenomena, and the growing suite of computational approach presents both opportunities and challenges. Choosing the right computational approaches to address a given question requires managing a model's complexity, balancing goals and limitations including interpretability, data resolution, and computational cost. Excess model complexity can diminish the utility for building understanding, while excess simplicity can render the model insufficient for addressing the questions of interest. Using systems immunology as a case study, we review how different model design strategies uniquely manage complexity, ending with a consideration of composite models, which combine the benefits of individual paradigms but present additional challenges arising from added layers of complexity. We anticipate that considering general model design challenges and potential solutions through the lens of complexity will foster enhanced collaboration among computational and experimental researchers.

Highlights:

- The complexity of a model is intimately tied to its performance and utility.
- Each modeling approach has a unique relationship to complexity.
- Excess detail can result in computationally intractable models.
- Composite modeling approaches can help to manage complexity.
- Available data and empirical observations should guide deliberate model design.

2.1 Introduction

How does one choose the "right" computational model? With the increasing availability of high-throughput and high-resolution characterization technologies, the fields of systems biology and immunology rely on computational advances to guide and interpret experiments.[41,42] A growing library of methods and resources provide more accessibility to computational models than ever before.[43–46] However, the choice of which modeling approach to pair with a question of interest is consequential and requires careful consideration. Here, we use systems immunology as a case study to discuss the relationship between complexity in computational models and their respective biological questions. Simpler models can provide more insight than their more complex counterparts—considering the available empirical observations and data is a key step towards a more deliberate model design.

In this Opinion, we present a perspective for managing model complexity (a term we employ to describe the number, or layers, of interacting components: e.g. equations, species, or rules), which is critical to maximizing model utility. The design space of a model includes size (number of parameters), scale (temporal/spatial/physiological), and level of biological detail (species/interactions). We classify relevant modeling approaches into data-driven, mechanism-driven, and agent-driven design paradigms, each of which entails a unique relationship to complexity (Figure 2.1). Whereas many approaches fall into distinct design paradigms, more recent composite approaches blend and leverage the complementary nature of independent design frameworks, considering their costs and benefits. Approaching

model design through the lens of managing complexity should enhance the integration of experimental and computational approaches and inform future considerations of experimental design.

2.2 Data-driven model design

Data-driven modeling approaches extract information from biological datasets by identifying statistical patterns. These approaches are generally applied to infer significant system interactions[47] or to predict important properties[48] on datasets like single cell sequencing[49] or histopathology images[50]. Data-driven modeling encompasses both machine and statistical learning. Learning approaches involve model training: the use of computational algorithms to identify parameter values (fitting) that minimize an error (cost) function in order to facilitate prediction or inference. These algorithms are designed to learn general patterns that extend beyond the training data. However, the characteristics of the training data will be embedded into the model during the fitting process. Thus, data-driven model performance will be better when new data resembles the training data (interpolation). When new inputs fall outside the scope of the training data (extrapolation), data-driven algorithms often perform poorly, particularly for more complex/nonlinear/non-monotonic systems. Therefore, researchers should seek to obtain data that are of high quality (i.e. focusing on precision and accuracy of the measurements) and are relevant to the phenomena of interest, rather than compiling a high quantity of tangentially related observations. Higher quality measurements reduce parameter uncertainty and facilitate more interpretable analysis.

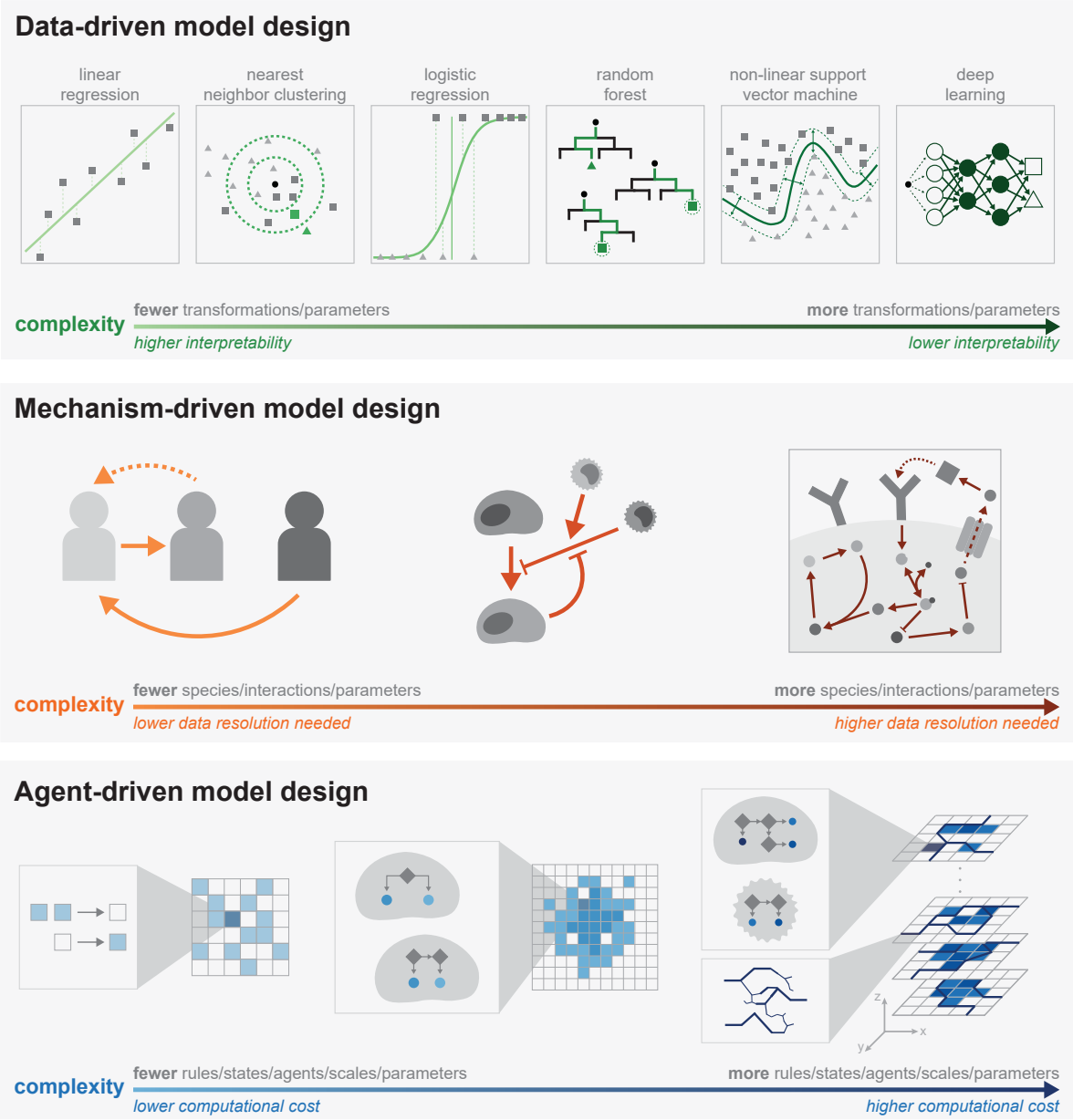

Figure 2.1: **Each modeling paradigm distinctly relates to complexity**: Increasing complexity in data-driven models can decrease interpretability, but enables models to capture non-linear and unintuitive behavior (top panel). Increasing the number of interactions in mechanism-driven models can require higher data resolution, but enables researchers to capture more intricate phenomena (middle panel). Increasing complexity—in the form of rules, cell types, and environmental scales—can incur higher computational costs in agent-driven modeling, but enables increased biological detail and incorporation of multiple scales to address complex, multi-scale phenomena (bottom panel).

Excess complexity during model selection can lead to overfitting, which impacts performance and predictive capacity. Overfitting occurs when the parameters characterize the noise, rather than the underlying signal, in the training data.[48] Performance is often quantified by the correlation between prediction and ground truth or by classification accuracy, the latter of which is evaluated in terms of both sensitivity and selectivity. Thus, overfitting can yield a spuriously promising performance metric on training data while also leading to poor predictive or inferential capacity on new data.

Data-driven models must also consider parameter interpretability in conjunction with overfitting when evaluating the appropriate complexity of a model. Popular learning methods span multiple levels of complexity based on algorithm design. Among the least complex is multiple linear regression, which assumes an interpretable linear combination of independent variables. A key step in using the least complex algorithms is to manually identify key input variables that provide a sufficiently predictive model. Conversely, deep learning extracts predictive and unobservable variables by adding complexity through parameters in hidden signal processing layers. The complexity in deep learning models necessitates external methods for interpreting the meaning of abstracted variables.[51,52] Interpretation methods are rapidly evolving.[52] Ensuring transparency of the learner's decision-making process is still an open challenge for the reproducibility of deep-learning models.[52]

The parameters and structures of less complex and more interpretable models support identifying generalizable patterns. Simple models can identify important correlates and detangle collinearities in data, for example to identify the relationship between patient baselines

and serologic responses to vaccination.[49] Another study developed a generalizable method using simple classifiers to identify key signaling genes in spatial single-cell sequencing.[53] Similarly, interpretable clustering methods, such as nearest-neighbors-based approaches, have helped to identify cell subtypes important for COVID-19 progression[54] or immune response dynamics in cancer treatment[55]. Focusing on interpretability during model selection supports hypothesis generation.

Models focusing on performance over interpretability learn indirect relationships between inputs and outputs to extract features from unstructured data. Many such models link sequence to function through neural networks to capture the nonlinear combinations of variables. For example, deep learning models predicting peptide-ligand binding, like netMHCpan[56], have helped identify important candidate antigens in the vast design space of enhanced T-cell binding[57]. Another common use case of deep learning models is to associate images with independent quantitative measurements. One such study leverages neural network models to integrate histopathology images (input) and gene expression data (output) to characterize image and associated biomarker heterogeneity.[50] Although deep learning models lack the interpretability of simpler approaches, they provide an alternative to manually extracting predictive variables.

2.3 Mechanism-driven model design

Mechanism-driven approaches incorporate domain knowledge to determine which interactions and species give rise to observed behaviors through testing candidate models. These

models incorporate known or hypothesized interactions using first-principles from kinetics and thermodynamics. Parameters are fit to experimental data similar to how they are fit in data-driven models. Likewise, the data quality impacts parameter interpretation and confidence. Commonly used mathematical frameworks include ordinary differential equations (ODEs), partial differential equations (PDEs), and equilibrium expressions (i.e. from thermodynamics). Comparing competing hypotheses, posed as distinct models, enables one to evaluate which hypothesis is most consistent with the available data, knowledge, and observations. A well-developed model can also be useful for predicting how a system will respond to perturbations not included in the training data, making mechanistic models better at extrapolating than are data-driven models.

Mechanism-driven model complexity—the number of defined species, relationships, and resulting parameters—should be constrained by data resolution. Data resolution dictates parameter identifiability, which is the ability to determine a parameter value with adequate precision given available data.[58] Developing mechanistic models that describe biological relationships at multiple scales is challenging without sufficiently detailed data connecting these scales.[59] High-resolution observations (e.g. protein or mRNA concentration profiling and temporal data[60,61]) support complex model development. Some phenomena can be observed at low resolution (e.g. population-level rates of infection[62,63]) and support abstracted models. Model reduction can be used to simplify a model to match data resolution and to ensure that all parameters are well-constrained and meaningful towards building insightful models.[58]

Some modeling goals are well-suited to abstracted biological descriptions and lower res-

olution data. Classic examples include population-level analyses and epidemiological phenomena.[62–64] Simple ODE models framed at the level of cell populations provided unique insights into human immunodeficiency virus (HIV) dynamics and treatment, revealing a viral production rate of ~ 1010 virions/day in contrast to the previously expected low replication rate for this virus.[65–68] Given limited data resolution, reducing model complexity facilitated parameter identifiability[65], and these models proved highly informative. More recently, cell population-level models have facilitated the design of chemotherapy preconditioning before chimeric antigen receptor (CAR) T-cell therapy.[64] Similarly, simple susceptible-infected-removed models describe infectious disease case counts over time. During the COVID-19 pandemic, such models have informed vaccine distribution[62] and provided insights into viral spread within counties as a function of age, race, and movement/interaction[63]. In each of these examples, useful simple models resulted from matching complexity to data availability.

High-resolution data can facilitate building complex models that enable interrogating explicit interactions and mechanisms. These datasets are commonly associated with subcellular measurements, such as protein or mRNA concentrations over time. For example, high-resolution models of NK-cell activation, signaling, and regulation have fostered greater insight into and design of anti-cancer and immunotherapy treatments by forecasting the impact of diverse cellular engineering strategies[69] and identifying key factors regulating hypoxia response[61]. Complex models describing native and engineered T-cell activation and signaling pathways elucidated how T-cell receptors (TCRs) discriminate between self and foreign antigens[70] and guided CAR-T cell engineering to enhance therapeutic efficacy[71–73]. A model

that incorporated intercellular regulation and intracellular signaling generated unique hypotheses describing how macrophages coordinate their activation in a manner that depends on local population density[60]. Notably, each of these complex models enabled fine-grained insight into subcellular biological phenomena.

2.4 Agent-driven model design

Agent-driven models—simulations designed around the behavior of autonomous agents—facilitate studying emergent functions. Rules guiding agent behavior are abstracted from observed or hypothesized behaviors of individuals within populations. These models can serve as intuitive testbeds to understand how different rules impact emergent behavior. This paradigm is synonymous with agent-based models (ABMs), which include the common subtypes of cellular automata and cellular Potts. Depending on context and scale, agents typically represent individual cells (e.g. in immunological applications) or individual organisms/people (e.g. in epidemiology or sociology applications).[26,74] In contrast to the other paradigms, agent-driven design does not rely on training, but it does rely on accurate characterization of higher-level dynamics. These models enable researchers to evaluate how hypothesized rules underlying biological phenomena give rise to observed emergent behavior.

Increasing model complexity increases computational cost. ABM design can be as simple as the two agent states and four rules included in Conway's Game of Life (a common, low-cost case study of the phenomenon of emergence).[75] Other agent-based model designs are more complex, computationally costly (run time and memory usage), and difficult to

parameterize.[74] Computational cost results from biological detail, scaling with the number of agents and agent properties in the simulation. Model reduction or abstraction can help avoid development of intractable models.

In systems immunology, ABMs with abstracted, rule-guided agents link subcellular or cellular-level changes to population-level emergent phenomena. ABMs with rule-guided agents that take on limited states and actions—such as proliferation, migration, death, genetic mutation, and/or environmental interaction—have revealed how tumor microenvironment conditions[76] and cell migration rate[77] affect tumor morphology, growth, and genetic diversity. ABMs can also be used to compare hypotheses. One modeling framework supported two different rule sets to compare how competing hypotheses on genetic mutation can drive the evolution of aggressive phenotypes in cancer progression.[76] Both rule sets resulted in a few aggressive phenotypes dominating tumor genetic makeup. Relatively simple agent descriptions can provide profound insight when recapitulating complex biological behavior.

Designing an ABM to describe intricate processes—such as intercellular interactions, sensing, signaling, environmental features, and trafficking—may necessitate reducing complexity to manage computational cost. One strategy is to lump functionally related features (e.g. intracellular signaling mediators) into aggregate signals, an approach applied to generate a simple representation of tumor-macrophage interactions.[57] This reduction facilitated a multiparametric sensitivity analysis that identified parameters influencing tumor survival and helped propose cell therapy strategies. Another study employed a simplified model of vasculature structure and dynamics (omitting details used in other ABMs[78,79]) to simulate

checkpoint inhibitor therapy and identify biomarkers that may be useful for guiding treatment.[80] A useful approach for integrating phenomena involving fine-grained detail is a Potts model, which explicitly describes cell boundaries and surface interactions. Potts models have been used to incorporate pMHC-TCR interactions and build understanding of anti-tumor CD8+ T-cell responses in heterogenous contexts.[81] Designing ABM complexity to fit the specific questions of interest helps manage computational cost and generate relevant insights.

2.5 Composite model design

Multi-paradigm approaches integrate the capabilities and considerations of each individual approach. Combining modeling approaches can increase computational efficiency, enable direct incorporation of data, expand scale representation, and facilitate sophisticated analyses.[59] While the challenge of achieving repeatability and reproducibility is not unique to composite models, there exist fewer tools to standardize novel model types compared to traditional/individual paradigms due to the added layers of complexity. Thus, specific safeguards designed to promote reproducibility of individual paradigms[82–85] should be integrated into these implementations. When wielded carefully, combining all three paradigms can synergistically provide in silico testbeds for complementing experimentally intractable systems.[86]

ABMs that incorporate mechanistic-driven or data-driven methods to guide agent state can effectively capture multi-scale phenomena and incorporate experimental data, which is difficult when using agent-driven models in isolation. The Agent-based Representation of

Cells And Dynamic Environments (ARCADE) is a framework that simulates tissues and cells in various environmental contexts, layering ODE signaling models and stoichiometric metabolic expressions within cell agents to influence rule-based state changes.[46] This study explicitly evaluated how varying the complexity of these subcellular models (which influence computational cost) impacted emergent outcomes. To avoid computational cost associated with solving ODEs within an ABM, a recent study used a neural network trained on a mechanistic model guiding differentiation as a drop-in replacement to enable tractable exploration of macrophage-based immunotherapies.[87] ABMs can predict outcomes or elucidate fundamental phenomena by incorporating mechanism-driven models. To reduce computational costs, data-driven models can be used in place of mechanistic descriptions.

External machine learning approaches comprise a strategy that can facilitate low-cost data-driven analysis, iterative adaptation, or emulation of mechanistic-driven or agent-driven models. An ODE model of CAR T-cell antigen-induced activation and signaling utilized multiple mechanistic frameworks and machine learning approaches to identify important signaling mediators and highlight counterintuitive relationships between receptor expression levels and signaling.[73] Many ABMs have integrated learning methodologies to improve patient treatment plans[88] or to adaptively choose statistically impactful simulations to run[89,90], automating the non-trivial simulation selection process. Emulating ABMs using data-driven approaches enables more robust sensitivity analyses at low computational cost by replacing the ABM with a predictive machine learning model.[91] Data-driven models can reduce the computational demand by autonomously probing models, minimizing unnecessary manual

curation of simulation conditions, or emulating an established model's behavior towards more rapid iterations.

Recent data-driven models have applied domain knowledge, drawing principles from mechanism-driven design, to improve the interpretability and performance of algorithms. Towards the goal of guiding cancer combination therapies, deep learning optimization strategies were used to fit parameters in a network—where the interactions of the network are constrained by biologically intuitive ODE models—to observations of cellular responses to various perturbations.[92] In this case, the biologically realistic mechanistic models constrained the deep learning model, thus providing interpretable parameters. Another approach employed a parameter penalty, derived from input from expert immunologists, to prioritize "important" signal transduction pathways to improve predictions of clinical outcomes in longitudinal term pregnancy and chronic periodontitis based on immune profiling.[93] Embedding mechanisms into data-driven algorithms promotes both interpretability and performance of resulting models.

2.6 Concluding thoughts

Complexity is a double-edged sword. Enhanced complexity can make models more predictive or biologically descriptive, but it can also obscure interpretability, necessitate higher data resolution, or increase computational cost. Deliberate design reduces undue complexity when pairing computational models to questions in systems immunology. There are still several open challenges surrounding complexity in modeling, especially in composite mod-

eling approaches. For example, layered complexity introduces different considerations for our data integration capabilities, as adding complexity reduces our ability to identify bias in our modeling designs. Additionally, model training can be more complicated with these hybrid frameworks, and model validation is further complicated. Standards of reproducibility should encourage models with transparency throughout each additional layer of complexity. Deliberate model design is the key in leveraging and layering complexity effectively, such that researchers are only limited by their creativity.

Chapter 3

INCORPORATING TEMPORAL INFORMATION DURING FEATURE ENGINEERING BOLSTERS EMULATION OF SPATIO-TEMPORAL EMERGENCE

Jason Y. Cain[1†], Jacob I. Evarts[2†], Jessica S. Yu[2], Neda Bagheri[1,2*]

1 Chemical Engineering, University of Washington, Seattle, WA 98195
2 Biology, University of Washington, Seattle, WA 98195

† These authors contributed equally to this work.
* nbagheri@uw.edu

Author's note: *This work was submitted to Bioinformatics in August 2023.*[94] *This article was co-authored by Jacob Evarts and myself. I was lead on conceptualization and methodology; we contributed equally to all other aspects of the project. This work combines data-driven and agent-based modeling paradigms in order to build emulation models.*

ABSTRACT

Motivation: Emergent biological dynamics derive from the evolution of lower-level spatial and temporal processes. A long-standing challenge for scientists and engineers is identifying simple low-level rules that give rise to complex higher-level dynamics. High-resolution biological data acquisition enables this identification and has evolved at a rapid pace for both experimental and computational approaches. Simultaneously harnessing the resolution and managing the expense of emerging technologies—e.g. live cell imaging, scRNAseq, agent-based models—requires a deeper understanding of how spatial and temporal axes impact biological systems. Effective emulation is a promising solution to manage the expense of increasingly complex high-resolution computational models. In this research, we focus on the emulation of a tumor microenvironment agent-based model to examine the relationship between spatial and temporal environment features, and emergent tumor properties.

Results: Despite significant feature engineering, we find limited predictive capacity of tumor properties from initial system representations. However, incorporating temporal information derived from intermediate simulation states dramatically improves the predictive performance of machine learning models. We train a deep-learning emulator on intermediate simulation states and observe promising enhancements over emulators trained solely on initial conditions. Our results underscore the importance of incorporating temporal information in the evaluation of spatio-temporal emergent behavior. Nevertheless, the emulators exhibit inconsistent performance, suggesting that the underlying model characterizes unique

cell populations dynamics that are not easily replaced.

Availability: All source codes for the agent-based model, emulation, and analyses are publicly available at `github.com/bagherilab/ARCADE`, `github.com/bagherilab/emulation`, and `github.com/bagherilab/emulation_analysis`, respectively.

Contact: nbagheri@uw.edu

Keywords: emulation, surrogate modeling, spatio-temporal emergence, deep learning, agent-based models

---

3.1 Introduction

Grand challenges in biology have required tools with increasingly higher resolution and throughput while also sampling across spatial and temporal axes[95–98]. In particular, dynamic data from live cell imaging is positioned to become the next "omics"[99,100] with lower-level resolution data, like scRNAseq, contributing a more nuanced understanding of individual system components across space[101,102]. Our ability to utilize high resolution data has often lagged behind our ability to generate said data[95,96,103]. Parity between computational and experimental approaches will allow researchers to synergistically utilize computational models to explain nonintuitive observations, identify "rules of life", and design model-driven experiments that test new hypotheses. This parity will identify both temporal and spatial components that are fundamental to specific biological systems *a priori*[104,105]. In order to address this knowledge gap, we interrogate the use of statistical emulation to estimate

emergent behavior in a comprehensive analysis of a high-resolution agent-based model.

Computational models capable of characterizing cellular dynamics over time and space across multiple scales are fundamental to scientific progress. Agent-based models (ABMs) have gained popularity due to their ability to simulate populations of heterogenous agents dynamically over time and in space in order to predict system-level emergent properties. ABMs are designed using the behavior and interactions of individual agents (usually cells) in an evolving spatial and temporal context[106]. This rule-based approach is well suited to modeling emergent behaviors, such as the development of biological tissues[107–109] or the spread of infectious diseases[110]. ABMs can capture the heterogeneity and the stochasticity of biological systems, as well as the bilateral relationship between the local microenvironment and agents' behaviors[79]. Furthermore, ABMs can be used to investigate the effects of different parameters and conditions on emergent behaviors, providing insights into complex biological processes that may be difficult or impossible to observe experimentally[26,79,111,112]. As such, ABMs are an increasingly important—albeit computationally expensive—tool for understanding complex biological systems and generating testable hypotheses that can inform experimental design.

Statistical emulation (SE)—the mapping of independent variables (inputs) to dependent variables (outputs) via statistical inference and machine learning (ML)—generates simplified statistical models that are computationally cheaper to analyze, and therefore interrogate, than the original simulation model. The flexible pattern recognition of ML provides two key benefits: (1) the emulation model building process codifies selection of requisite inputs to

identify dominant statistical patterns; and (2) the mathematical frameworks involved in pattern recognition—once trained—are easily calculated via sequences of simple mathematical operations. On the contrary, the feature selection and design process required for SE is laborious, and the algorithms demand significant data for adequate performance. Balancing these trade-offs supports effective integration into multi-class models[87,113], sensitivity analyses for complex models[91], and parameter sweeps of mechanistic models[114,115]. Even with these successes, there have been few comprehensive analyses of the application of emulation models to rule-based models designed to simulate tissue dynamics that emerge from temporal and spatial interactions[91,116,117]. SE of ABMs is largely under-explored despite knowing that the computational demands of large multi-scale ABMs can outpace their utility to generate hypotheses between continuous input variables and heterogeneous outputs[118,119]. Synergistic development of SE frameworks for ABMs would facilitate expedited analyses and provide systematic methods towards hypothesis generation. Understanding the characteristics (e.g. spatial, temporal, etc.) of data required to emulate key ABM dynamics provides a more thorough understanding of the drivers of emergent outcomes.

3.1.1 The tumor microenvironment as a model system

ABMs of the tumor microenvironment have leveraged *in silico* networks to represent the vascular environment[79]. Physical and structural characteristics (e.g. pressure, shear, radius) are often encoded into network elements like edges and nodes[120,121]. Networks are flexible data structures that can be modified and interrogated, enabling researchers to study interplay

among agents and between agents and environments. Functional coupling between the agents and the environment is required to capture experimentally observable divergent emergence like vascular collapse and necrotic core formation[79].

Network analysis provides a means to abstract high-resolution, spatial information into summary statistics[122,123]. Network topology and morphology have been used to understand ecological systems[124,125], interrogate biological pathways[126], analyze neurological structure[127–129], and identify novel treatment[130]. Specifically in tumor development, network analyses are a promising approach to study healthy and pathogenic vascular mimicry and angiogenesis[131–133]. Thus, we hypothesize that vascular network analysis enables the encoding of complex network architecture as interpretable inputs for SE.

In this study, we utilize network analysis to construct SE models of ARCADE, an existing agent-based model of a tumor microenvironment with heterogeneous and realistic vascular networks[79]. Our investigation reveals a limited relationship between spatial network topology characteristics and emergent tumor properties. We demonstrate the efficacy of incorporating temporal evolution of network metrics to predict emergent properties of the system. Leveraging this temporal information, we develop deep-learning models that improve the predictive power of emulators. Our results highlight the role of temporal dynamics in understanding and predicting emergent properties of diverse cell populations that evolve from lower-level spatial and temporal processes.

3.2 Materials and methods

3.2.1 Data and code availability

All source code for the ARCADE ABM is publicly available on GitHub at `github.com/bagherilab/ARCADE`. The scripts used to perform analyses and generate figures reported in this paper are also publicly available on GitHub at `github.com/bagherilab/emulation_permutation`.

3.2.2 ARCADE simulations and workflow

All ABM simulations and model analyses were performed similar to those described in a previous publication[79] using ARCADE v2.4[134]. Two agent populations, healthy and cancer cells, exist in the simulations. *In silico* cancer cell populations exhibit hallmarks of cancer: increased crowding tolerance, increased preference for metabolic glycolysis than oxidative phosphorylation, and increased migratory invasiveness versus healthy cell populations. The `colony` and `tissue` contexts describe simulations comprised of a solely cancer cell population or a combination of cancer and healthy cell populations, respectively. Prior work highlighted differences in emergent behavior between colony and tissue contexts[26,79,135].

We focus on emulating three emergent tumor properties: activity (instantaneous state of system), growth rate (cumulative temporal behavior), and symmetry (instantaneous spatial state). Activity is the ratio of active to dead cells, describing the degree of necrosis in the tumor. Growth rate is a temporal emergent property that generally describes tumor

aggression. Symmetry is a spatial emergent property describing the density and implying the aggressiveness of the tumor. Specific calculations for these properties are detailed in Supp. Section 3.5.1.

The initial vascular structure is the only differing variable between simulations of the same context. Vasculatures are stochastically generated using starting root geometries, detailed in a previous article[79]. 100 seeds generated a unique vasculature for each starting root geometry and seed combination.

Vascular structure and function change over time based on their coupling with cell agent populations[79]. Functional stresses from tissue cells lead to vascular remodeling, changing vessel radius and wall thickness as a function of hemodynamic properties (shear stress, circumferential stress, and flow rate) and metabolic demand. Cancer cells degrade the vasculature by removing components; this removal can lead to larger scale disruption of vascular structure, as functional vasculature requires perfusion.

We quantify properties of the *in silico* vasculature and use these features to predict spatial and temporal emergent dynamics. Aggregate hemodynamics, such as flow and wall thickness, are calculated for each vessel segment and then averaged. Topological features are calculated using the `igraph` package[136] in Python to create a network representation of the vasculature. Hemodynamic edge features are the same network metrics used to define topological features with one modification: edges in the graph are weighted by hemodynamic properties. Finally, spatial features use distance from the center of the simulation (the tumor seeding location) as edge weights, such that vessels within and closer to the tumor core are

weighted more heavily.

3.2.3 Network analysis and feature extraction

To represent the intricate characteristics of vasculatures within ABM simulations, we employed network analysis using the `igraph` Python package. By using a graphical representation of the vasculature, we were able to utilize graph theory metrics as structural and functional features for our ML models. Graph theory provides a number of benefits: it comprises diverse metrics that account for the topological structure of the vasculature; it has mechanisms for specifying vessel importance; and it can quantify overall vessel connectivity. Vascular structure is represented as a network where blood vessels are represented as edges, and where junctions and end points are represented as nodes. The supplementary materials (Supp. Table 3.4, 3.5, and 3.6) offer detailed information about specific graph metrics and how they were obtained.

3.2.4 Statistical emulation

Machine learning models

Python package `sklearn` was used to build the ML models[137]. The specific modules used are listed in Table 3.1, with accompanying abbreviations used throughout this article. All hyperparameters specified are referred to as their respective arguments for each model.

Model	Python module
MLR	`sklearn.linear_model.ElasticNet`
SVR	`sklearn.svm.SVR`
RF	`sklearn.ensemble.RandomForestRegressor`
MLP	`sklearn.neural_network.MLPRegressor`

Table 3.1: `sklearn` package implementation used for each ML model.

Hyperparameter selection and cross-validation

A Sobol random search[138] was used to select the tested hyperparameters during cross-validation using the `scipy`[139] package in Python. The ranges used for the Sobol random search are detailed in Table 3.2. Parameter types categorized as "linear" used Sobol indices in the linear space, whereas those types categorized as "logarithmic" used the log of the bounds as the search range. Every discrete parameter value was exhaustively tested in combination with parameter values selected using a Sobol random search on continuous parameter spaces. Each set of Sobol and discrete hyperparameters was used to generate an independent ML model. The model with the best average performance metric (we use the coefficient of determination R^2) compared across all hyperparameters during cross-validation was used for training and testing. 10-fold cross-validation was used in all cases. The coefficient of determination is defined as $R^2 = 1 - \frac{SS}{SS_{tot}}$, where SS is the sum of squares of residuals ML model and SS_{tot} is the total sum of squares from the mean. R^2_{val} was calculated from ML models evaluated on the validation data set. The reported R^2_{train} and R^2_{test} were calculated on the training and withheld testing sets after cross-validation.

		Bounds		
	Hyperparameter	Lower	Upper	Type
MLR	alpha	0.001	1.000	Log
	l1_ratio	0.1	1.0	Linear
SVR	C	0.0001	1.000	Log
	epsilon	0.0	1.0	Linear
RF	n_estimators	1	100	Linear
	max_features	0.01	1.0	Log
	max_depth	1	100	Linear
	min_samples_split	0.01	1.0	Log
	max_samples_leaf	0.01	1.0	Log
MLP	alpha	0.0001	1.000	Log

Table 3.2: Continuous hyperparameters used in cross-validation Sobol search.

	Hyperparameter	Values
SVR	kernel	linear, poly, rbf, sigmoid
RF	bootstrap	True, False
MLP	activation	identity, logistic, tanh, relu
	hidden_layer_sizes	(5,), (5,5), (5,10), (25,), (25, 25), (25,50), (50,), (50,25), (50,50)

Table 3.3: Discrete hyperparameters used in cross-validation.

3.2.5 Recurrent neural network architecture and training

In order to account for dynamic network evolution, we trained deep neural networks (NNs) to use time course information to improve predictive performance using the `keras`[140] API of the `tensorflow`[141] package in Python. The purpose of the trained NNs was to predict network evolution from the initial vascular structure in order to increase prediction performance of the emulation models. The neural network utilized a long short-term memory (LSTM) layer, a type of recurrent NN layer capable of capturing sequential patterns[142]. The LSTM layer was followed by 3-4 fully connected layers. Full network topology details are described in Supp. Figure 3.12.

Network topology features from each simulation day were collected and stacked into multivariate time series to facilitate transfer learning of NN parameters. The recurrent NN (RNN) was pre-trained on the full-length time series, encompassing the entire temporal evolution of the network in order to constrain the RNN parameters. To further fine tune the RNN, bootstrapped samples from a subset of the time series (10 days, 5 days, 3 days) were used to sequentially retrain the model (using the previous pretrained deep-learning models as a starting point) and retrain on the initial conditions to provide the ultimate emulator. We applied the best performing model to predict the two week network structures for a reserved test set from only the initial network topology, and then passed the predicted features into the naive ML models (Figure 3.4A).

3.3 Results

The objective was to predict the emergent tumor output metrics by their respective simulation inputs using naive ML algorithms (Figure 3.1). Specifically, we used multiple linear regression (MLR), random forest (RF), support vector regression (SVR), and multi-layer perceptron (MLP) to predict emergent tumor output metrics (activity, growth, and symmetry) from the initial condition of the microenvironment vasculature. In general, we use "emulators" to describe those models accepting initial conditions as the sole inputs and we us "ML models" to describe those accepting any other timepoint. The microenvironment was represented by a network analysis of the vascular architecture to provide features for ML model.

Various network analyses generated a suite of feature sets that were used to train the ML models. The `topological` feature set characterized traditional structural and topological information of the vasculature system (e.g. eccentricity, betweenness, average degrees), as well as mean hemodynamics across the vascular system (e.g. mean pressure, mean shear, mean radius, etc.). The `hemodynamic` feature set ascribed hemodynamics properties (e.g. flow, pressure, wall thickness) as edge weights to the topological features in order to capture vascular function. The `spatial` feature set integrated relative locations of edges and nodes from the center of the simulation—which represents the center of the tumor—and accounted for these properties as both additional edge weights and penalties in weighted average calculations. Each feature set is inclusive of the previous set:

`topological` ⊆ `hemodynamic` ⊆ `spatial`. A comprehensive feature set breakdown is described in Supp. Tables 3.4, 3.5, and 3.6.

3.3.1 Unweighted and hemodynamic-weighted network topologies do not predict emergent tumor output metrics

In order to represent vascular structures in a quantitative format, topological network data from initialization vasculatures were used to train the emulators. Aggregate network metrics (e.g. number of nodes and edges) characterized the size and density of the network. Additional metrics—e.g. average eccentricity, betweenness, and coreness of each node—were used to characterized the average behavior of nodes in the network. This `topological` feature set does not include spatial node embeddings as a factor in the analysis.

Vascular structures represented by initial `topological` features are not predictive of emergent behavior, resulting in models that exhibit both overfit and underfit characteristics. The coefficient of determination for test data in all `topological` emulation models is less than 0.3 (Figure 3.2A), suggesting that these models are underfit as a result of the variance in our features not explaining the variance in the emergence. The substantial performance gaps between training and testing results that derive from more complex algorithms (i.e. SVR, MLP) indicate overfitting, despite implementing regularization (Figure 3.2A).

Performance is variable as a result of context; activity predictions in `colony` contexts (Figure 3.2A, left) performed more accurately than comparable models in `tissue` contexts (Figure 3.2A, right). On the contrary, emulators that comprise healthy cell background

in the `tissue` context (Figure 3.2A, right) reflect more accurate symmetry and growth predictions. Notably, an adequate model—exhibiting a test coefficient of determination over 0.0—for symmetry in the `colony` context could not be trained.

We then included hemodynamic characteristics as edge weights in the network analysis for network-distance metrics in the distance-based analyses. These `hemodynamic` features describe higher-resolution physical characteristics of the environment that have clinical implications. For example, pressure-based metrics have been associated with system-level hypertension.[143] The resulting weighted network analysis metrics offered limited improvement in prediction accuracy; most results were statistically insignificant relative to the unweighted case, determined by a two-way ANOVA (Figure 3.2A). Training and testing performance of `hemodynamic` emulators experienced diminishing returns as the amount of training data increased (Supp. Figure 3.5).

3.3.2 Network topologies with hemodynamic and spatial information do not predict emergent tumor output metrics

We hypothesized that the spatial variance in vascular structure seeds would account for a significant amount of the variance found in emergent tumor behaviors. This hypothesis was motivated by the finding that vascular collapse, a large-scale dynamic event, substantially impacts the spatial structure of simulations. The consequence of collapse in concert with vessel degradation is required to observe the formation of a necrotic tumor core[79]. Thus, in order to capture spatial information in the network analysis, we applied a proportional

penalty for edges and nodes based on their respective euclidean distance from the center of the simulation (a proxy for the tumor core).

We investigated whether the distance of network nodes from the center of the simulation would explain tumor behavior (Figures 3.2A and 3.2B). Including `spatial` features led to negligible, if any, improvement on the emulators' testing performance. Regularization in the cross-validated emulation models led to predicted targets that spanned less variance in the response variable (Figures 3.2), which was indicated by the low coefficients of determination (Figure 3.2A). The RMSE of the predicted targets showed minimal improvement as we increased the number of training data (Figure 3.2C), suggesting that the poor performance was not a result of insufficient training data; instead, it likely derived from an incomplete representation of the drivers of emergence.

3.3.3 Network analysis of environmental snapshots at later timepoints predict emergent tumor output metrics

In order to interrogate the impact of network evolution on the performance of our ML models, we calculated hemodynamic network metrics for the vascular structure on each simulation day. Features were derived from snapshots of later vascular architectures. These temporally dynamic features were then used to predict emergent tumor dynamics.

We found that network metrics that were generated from later timepoints provided better predictive performance (Figure 3.3A). In the `colony` context activity was significantly more predictable when using features representing network properties at later timepoints (Fig-

ure 3.3B, left). Additionally, we observed non-monotonic improvements in growth prediction with a notable improvement in performance using snapshots of network properties from the middle of the time course (Figure 3.3B, middle, Supp. Figure 3.9). Symmetry showed minimal improvement (Figure 3.3B, right). Conversely, in the `tissue` context, predictions of activity did not improve, while predictions of growth improved monotonically (Figure 3.3B, Supp. Figure 3.10). The improvement in predicting symmetry from initial features relative to features from two simulation weeks was larger in `tissue` contexts than in `colony` contexts (Figure 3.3B, Supp. Figure 3.10).

The analysis of vascular structure included features post-vascular collapse, which likely accounted for some of the better performance. Alternatively, including differential timepoint analyses did not provide additional benefits to predictive power (Supp. Figure 3.6, Supp. Figure 3.7). Simply shortening the simulation time, and therefore the prediction horizon, to one week resulted in a sharp decline in emulator performance (Supp. Figure 3.8). Once again, there was limited improvement in test performance with additional training data, indicating there was sufficient data to train these ML models (Figure 3.3C).

3.3.4 Neural network structures trained on network dynamics support ABM emulators

We hypothesized that including temporal dynamics in the training of an emulator could improve performance based on the enhanced predictive potential of later timepoints. First we trained a recurrent NN (RNN) on the network evolution of the vascular architecture in order to forecast the network metrics of the final timepoint based on the initial condition

(Figure 3.4A). Then, we used statistical learning models to predict emergent outputs using the predicted final network metrics as features (Figure 3.4B).

The efficacy of using forecasted features varied widely by emergent outcome and context. Emulators trained on forecasted features never performed worse than those trained on initial-condition-derived features (Figure 3.4B, left). In the `colony` context the RNN model led to a small improvement in activity and growth, while symmetry remained largely unpredictable. In the `tissue` context, the model was improved across all emergent behaviors. Activity predictions improved in the `tissue` context with a test R^2 nearly three times greater than the naive emulation models (Figure 3.4B).

While growth exhibited the largest benefits from training on temporal dynamics, all predictions across all emergent outcomes presented at least minor improvements, suggesting that temporal information could benefit the prediction of diverse emergent behaviors. However, these improvement did not match the predictive accuracy of the final timepoints directly from the simulation. Based on a principal component analysis, the features derived from the forecasted network architectures were comparable to those from simulated architectures at the final timepoint, independent of the training-testing split of data (Supp. Figure 3.11). These results indicated that the propagation of minor errors resulting from ML regression techniques may account for the differences in predictive accuracy of the forecasted versus simulated timepoints.

3.4 Discussion

Parity between biological systems and computational models requires algorithms capable of considering higher resolutions of heterogeneous species and their interactions. This objective adds complexity to analysis of experimental data and the development of predictive models. Emulation is a powerful tool that computational scientists can use to reduce the computational expense of model simulations (accelerating hypothesis generation), and to identify and understand key drivers of simulation dynamics (elucidating biological insight). This work unpacks the challenges of parity by using emulation to predict the evolution of tumor development in a dynamic environment that accounts for multilateral regulation among diverse cell agents and between cell agents and their supporting vasculature. The vascular architecture (topology) and function (hemodynamics) change as a function of time and cell population, encompassing a relevant level of biological complexity. While other studies have focused on leveraging emulation to interrogate ABM parameters[91,117], we focused on emulators using initial heterogeneous environmental conditions to predict the evolution of the cell population holistically to maintain close analogs with current experimental methodologies (e.g., dynamic imaging).

We built ML models designed to predict emergent behaviors of *in silico* tumors after two simulation weeks. These behaviors include tumor activity, growth, and symmetry. ML model predictions were based on features that derived from a network analysis of the tumor environment's initial vascular architecture and hemodynamics. Counter-intuitively, we found

that spatial characteristics of the environment were largely *un*successful at improving predictions of emergent behaviors that were shown to be associated with spatial phenomena[79]. The resulting models were prone to both underfitting (evident from poor performance and limited improvement from additional data) and overfitting (indicated by substantial gaps between testing and training splits). Instead, we found that ML models that were trained on environmental snapshots of later simulation timepoints were more predictive of these emergent behaviors.

We then built an emulator using a combination of a RNN-forecaster model to predict the network evolution of the vasculature, and we used these as inputs into ML regression models to predict emergent behaviors. The final RNN-based emulator showed promising, albeit limited, improvement over using emulators that were strictly trained on initial environmental conditions highlighting the role of temporal dynamics in spatio-temporal cell population models. We were able to demonstrate the ability of an RNN model to make accurate forecasts of the network evolution. However, the limited predictive potential of those forecasted timepoints for predicting emergent behavior in turn emphasizes the importance of leveraging ABMs to make more accurate representations of our systems.

By leveraging network analysis and the evolution of network metrics, this study reinforces the importance of temporal information when predicting the behavior of cell population dynamics, even when the emergent behavior derives from spatial dynamics. The difference in performance between our two simulation contexts necessitate careful consideration when translating `colony` results to `tissue` contexts—analogous to translating `in vitro` to `in`

vivo results. A general conclusion of this study suggests that out-of-the-box ML approaches are limited in their ability to characterize spatially heterogeneous systems. Despite their limited performance, we believe that SE would become an invaluable tool to the scientific community once we overcome challenges underlying their predictive performance for general application. Until then, ABMs are a necessary and unmatched alternative to predicting spatio-temporal dynamics.

The emulation of ABMs is difficult due to the complexity (e.g., number of species and corresponding interactions) and emergent nature of the biological phenomena they are well suited to simulate. As ABMs become increasingly multi-scale and complex[40], challenges in emulation will persist and likely magnify. Mitigating these challenges is necessary to mediate the quantity of ABM simulations required to identify patterns, sample high-dimensional spaces, and generate testable hypotheses across emergent dynamics; such simulations can be cost-prohibitive. This cost is particularly relevant in context of personalized medicine where computational models must be used for real-time control (as is the case of insulin delivery)[144]. In order to develop similar interventions that can help mitigate or drive population dynamics, the cost of predicting the spatio-temporal dynamics of cell populations must be addressed head on.

Computational expense remains a significant consideration in supporting emulator development, training, and analysis of perturbations' impact on outcomes. Furthermore, the high-resolution spatio-temporal data required for emulation necessitates efficient storage, dissemination, and management protocols. The associated computational costs with emulation

also call for economic considerations when investigating relationships among state variables or between inputs and outputs, as well as rigorous sampling of an ML model's hyperparameter space. Additionally, emergent dynamics arising from the evolution of complex spatial and temporal processes poses challenges for representing said data in a ML framework such that the resulting models remain interpretable and useful. These challenges need to be addressed to maximize the potential of SE in advancing our understanding of biological systems through computational modeling.

Acknowledgments

This work was supported by the National Science Foundation CAREER award CBET-1653315 (N.B.) and the Washington Research Foundation (N.B.).

Declaration of interests

J.S.Y. is Scientist at the Allen Institute for Cell Science. N.B. is Adjunct Associate Professor of Chemical & Biological Engineering at Northwestern University and Sr. Advisor of Modeling, Dissemination, & Alliances at the Allen Institute for Cell Science.

3.5 Supplemental information

3.5.1 Emergent behavior metrics

Emergent behavior metrics are defined and calculated as presented in a previous study of heterogeneous vasculatures in ARCADE[79].

Growth rate

Growth rate quantifies the change in colony diameter over time. First, colony diameter is calculated at each time index. Growth rate is the slope of the simple linear regression between [0, 0.5, ..., t_i] and corresponding diameters [D_0, $D_{0.5}$, ..., D_i], where i indicates the timepoint. These calculations were performed using the Python function `polyfit` from package `numpy` with degree of 1.

Symmetry

Symmetry (S) quantifies colony shape at a given timepoint, ranging from 0 (not symmetric) to 1 (perfectly symmetric). For hexagonal coordinates, a colony is perfectly symmetric if for each location (u,v,w), the corresponding five locations (-w,-u,-v), (v,w,u), (-u,-v,-w), (w,u,v), and (-v,-w,-u) are all occupied. Symmetry is calculated as:

$$S = 1 - \frac{1}{N} \sum_{i}^{N} \frac{n_i}{5}$$

where N is the number of unique locations and n_i is the number of corresponding unoccupied locations for a unique location i.

Activity

Activity (A) quantifies the ratio of active (proliferative and migratory) to inactive (necrotic and apoptotic) cells at a given timepoint. This metric ranges between -1 (all non-quiescent cells are inactive) and +1 (all non-quiescent cells are active). An activity value of 0 indicates

an equal number of active and inactive cells. Activity is calculated as:

$$A = 2 * \frac{N_a}{N_a + N_i} - 1$$

where N_a is the number of active cells and N_i is the number of inactive cells.

Feature	Code	Equation	Description
Radius	RADIUS	$\frac{\sum_{e \in E} \texttt{radius of e}}{\lvert E \rvert}$	Average vessel radius
Length	LENGTH	$\frac{\sum_{e \in E} \texttt{length of e}}{\lvert E \rvert}$	Average vessel length
Wall	WALL	$\frac{\sum_{e \in E} \texttt{wall thickness of e}}{\lvert E \rvert}$	Average vessel wall thickness
Shear	SHEAR	$\frac{\sum_{e \in E} \texttt{shear force on e}}{\lvert E \rvert}$	Average vessel shear force
Circumference	CIRCUM	$\frac{\sum_{e \in E} \texttt{circumference of e}}{\lvert E \rvert}$	Average vessel circumference
Flow	FLOW	$\frac{\sum_{e \in E} \texttt{flow through e}}{\lvert E \rvert}$	Average vessel flow
Nodes	NODES	$\lvert V \rvert$	Number of nodes in the graph
Edges	EDGES	$\lvert E \rvert$	Number of edges in the graph
Average eccentricity	AVG_ECCENTRICITY	$\frac{\sum_{v \in V} max(dist(v))}{\lvert V \rvert}$ where $dist(v)$ is the shortest distances between v and all other nodes	The average longest shortest path that each nodes is from another node
Graph radius	GRADIUS	$min(eccentricity)$	The minimum eccentricity value
Graph diameter	GDIAMETER	$max(eccentricity)$	The maximum eccentricity value
Average shortest path	AVG_SHORTEST_PATH	$\frac{\sum_{v \in V} \frac{(dist(v)}{\lvert V \rvert}}{\lvert V \rvert} * \frac{1}{\lvert V \rvert * (\lvert V \rvert - 1)}$ where $dist(v)$ is the shortest distances between v and all other nodes	The average shortest path between each nodes and every other node
Average in degree	AVG_IN_DEGREES	$\frac{\sum_{v \in V} indegree(v)}{\lvert V \rvert}$	The average in degree in a directed graph
Average out degree	AVG_OUT_DEGREES	$\frac{\sum_{v \in V} outdegree(v)}{\lvert V \rvert}$	The average out degree in a directed graph
Average degree	AVG_DEGREE	$\frac{\sum_{v \in V} degree(v)}{\lvert V \rvert}$	The average degree in an undirected graph
Average clustering	AVG_CLUSTERING	$\frac{\sum_{v \in V} Cl(v)}{\lvert V \rvert}$ where $Cl(v)$ is the ratio of the ratio between the number of existing connections between neighbors of v and the maximum possible connections between them	The average clustering coefficient of a graph; the tendency for nodes to form tightly interconnected clusters
Average closeness	AVG_CLOSENESS	$\frac{(\lvert V \rvert - 1) * (\lvert V \rvert - 1)}{\frac{\sum_{v \in V} dist(v)}{\lvert V \rvert}}$ where $dist(v)$ is the shortest distances between v and all other nodes	The average closeness between nodes using the Wasserman and Faust imrpoved formula
Average betweenness	AVG_BETWEENNESS	$\frac{\sum_{v \in V} B(v)}{\lvert V \rvert} * \frac{1}{(\lvert V \rvert - 1) * (\lvert V \rvert - 2)}$ where $B(v)$ is the sum of the fraction of shortest paths passing through v over all pairs of nodes in the graph	The average betweenness of a graph; the average importance of nodes connectors between other nodes in the network
Average coreness	AVG_CORENESS	$\frac{\sum_{v \in V} C(v)}{\lvert V \rvert}$ where $C(v)$ is the coreness of node v	The average coreness of a graph; the level of connectedness of nodes within a graph

Table 3.4: Supplement: Topological feature list

Feature	Code
Graph radius weighted by flow	GRADIUS:FLOW
Graph diameter weighted by flow	GDIAMETER:FLOW
Average eccentricity weighted by flow	AVG_ECCENTRICITY:FLOW
Average shortest path weighted by flow	AVG_SHORTEST_PATH:FLOW
Average closeness weighted by flow	AVG_CLOSENESS:FLOW
Average betweenness weighted by flow	AVG_BETWEENNESS:FLOW
Graph radius weighted by wall	GRADIUS:WALL
Graph diameter weighted by wall	GDIAMETER:WALL
Average eccentricity weighted by wall	AVG_ECCENTRICITY:WALL
Average shortest path weighted by wall	AVG_SHORTEST_PATH:WALL
Average closeness weighted by wall	AVG_CLOSENESS:WALL
Average betweenness weighted by wall	AVG_BETWEENNESS:WALL
Graph radius weighted by shear	GRADIUS:SHEAR
Graph diameter weighted by shear	GDIAMETER:SHEAR
Average eccentricity weighted by shear	AVG_ECCENTRICITY:SHEAR
Average shortest path weighted by shear	AVG_SHORTEST_PATH:SHEAR
Average closeness weighted by shear	AVG_CLOSENESS:SHEAR
Average betweenness weighted by shear	AVG_BETWEENNESS:SHEAR
Graph radius weighted by radius	GRADIUS:RADIUS
Graph diameter weighted by radius	GDIAMETER:RADIUS
Average eccentricity weighted by radius	AVG_ECCENTRICITY:RADIUS
Average shortest path weighted by radius	AVG_SHORTEST_PATH:RADIUS
Average closeness weighted by radius	AVG_CLOSENESS:RADIUS
Average betweenness weighted by radius	AVG_BETWEENNESS:RADIUS
Graph radius weighted by average pressure	GRADIUS:PRESSURE_AVG
Graph diameter weighted by average pressure	GDIAMETER:PRESSURE_AVG
Average eccentricity weighted by average pressure	AVG_ECCENTRICITY:PRESSURE_AVG
Average shortest path weighted by average pressure	AVG_SHORTEST_PATH:PRESSURE_AVG
Average closeness weighted by average pressure	AVG_CLOSENESS:PRESSURE_AVG
Average betweenness weighted by average pressure	AVG_BETWEENNESS:PRESSURE_AVG
Graph radius weighted by pressure delta	GRADIUS:PRESSURE_DELTA
Graph diameter weighted by pressure delta	GDIAMETER:PRESSURE_DELTA
Average eccentricity weighted by pressure delta	AVG_ECCENTRICITY:PRESSURE_DELTA
Average shortest path weighted by pressure delta	AVG_SHORTEST_PATH:PRESSURE_DELTA
Average closeness weighted by pressure delta	AVG_CLOSENESS:PRESSURE_DELTA
Average betweenness weighted by pressure delta	AVG_BETWEENNESS:PRESSURE_DELTA
Graph radius weighted by average oxygen	GRADIUS:OXYGEN_AVG
Graph diameter weighted by average oxygen	GDIAMETER:OXYGEN_AVG
Average eccentricity weighted by average oxygen	AVG_ECCENTRICITY:OXYGEN_AVG
Average shortest path weighted by average oxygen	AVG_SHORTEST_PATH:OXYGEN_AVG
Average closeness weighted by average oxygen	AVG_CLOSENESS:OXYGEN_AVG
Average betweenness weighted by average oxygen	AVG_BETWEENNESS:OXYGEN_AVG
Graph radius weighted by oxygen delta	GRADIUS:OXYGEN_DELTA
Graph diameter weighted by oxygen delta	GDIAMETER:OXYGEN_DELTA
Average eccentricity weighted by oxygen delta	AVG_ECCENTRICITY:OXYGEN_DELTA
Average shortest path weighted by oxygen delta	AVG_SHORTEST_PATH:OXYGEN_DELTA
Average closeness weighted by oxygen delta	AVG_CLOSENESS:OXYGEN_DELTA
Average betweenness weighted by oxygen delta	AVG_BETWEENNESS:OXYGEN_DELTA

Table 3.5: Supplement: Hemodynamic feature list

Feature	Code
Graph radius weighted by inverse distance	GRADIUS:INVERSE_DISTANCE
Graph diameter weighted by inverse distance	GDIAMETER:INVERSE_DISTANCE
Average eccentricity weighted by inverse distance	AVG_ECCENTRICITY:INVERSE_DISTANCE
Average shortest path weighted by inverse distance	AVG_SHORTEST_PATH:INVERSE_DISTANCE
Average closeness weighted by inverse distance	AVG_CLOSENESS:INVERSE_DISTANCE
Average betweenness weighted by inverse distance	AVG_BETWEENNESS:INVERSE_DISTANCE
Average eccentricity weighted by distance	AVG_ECCENTRICITY_WEIGHTED
Average closeness weighted by distance	AVG_CLOSENESS_WEIGHTED
Average coreness weighted by distance	AVG_CORENESS_WEIGHTED
Average betweenness weighted by distance	AVG_BETWEENNESS_WEIGHTED
Average in degree weighted by distance	AVG_IN_DEGREES_WEIGHTED
Average out degree weighted by distance	AVG_OUT_DEGREES_WEIGHTED
Average degree weighted by distance	AVG_DEGREE_WEIGHTED

Table 3.6: Supplement: Spatial feature list

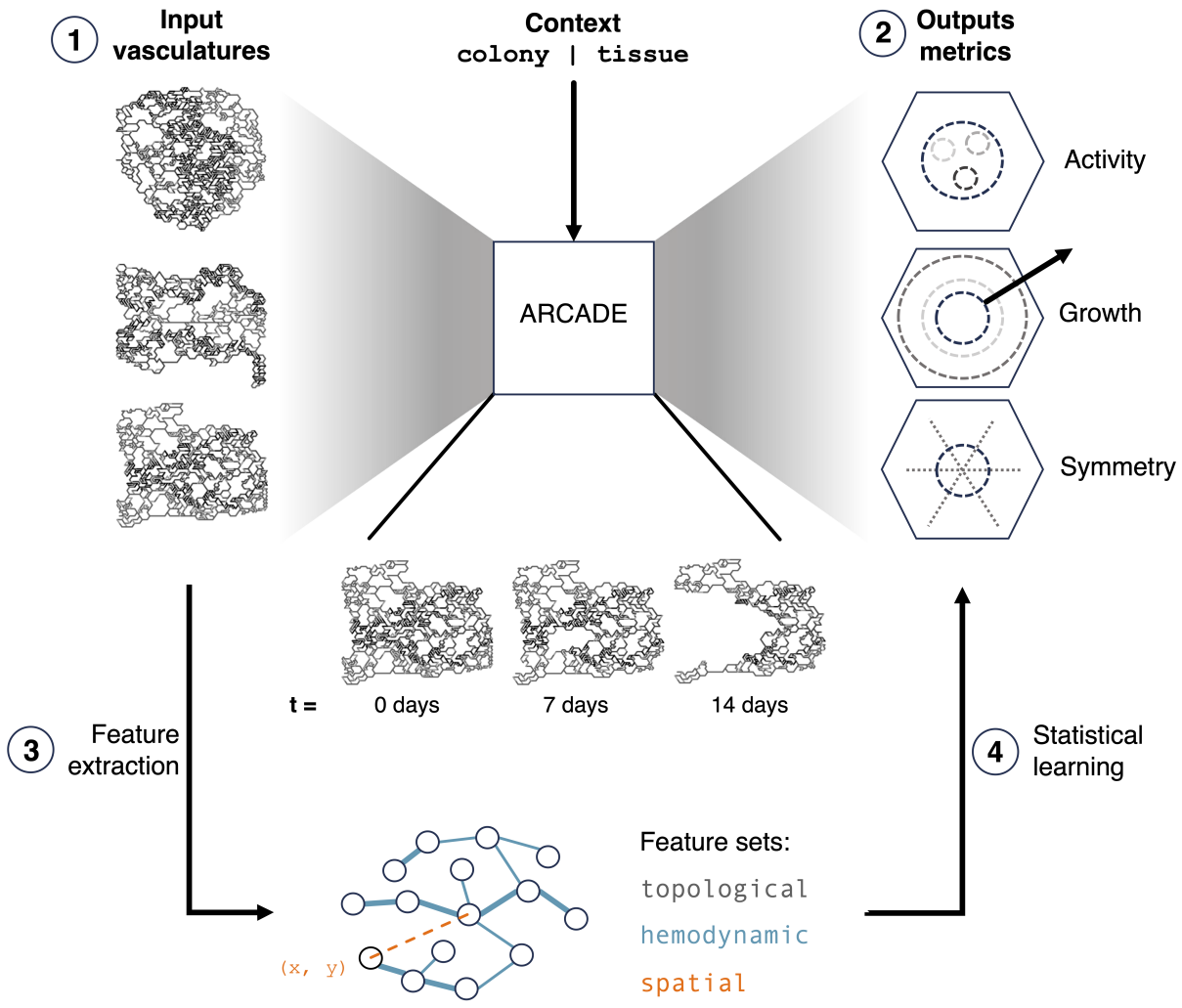

Figure 3.1: **Emulation workflow** — A summary of the overall emulation workflow. (1) ARCADE, an ABM of the tumor microenvironment, receives *in silico* vasculature networks and initial cell population colonies as inputs. ARCADE simulates interactions among diverse agents to predict the evolution of vascular architecture and function, as well as cell populations, over space and time. (2) Spatio-temporal dynamics are summarized with output metrics that evaluate emergent tumor properties at the end of the simulations: activity, growth, and symmetry. (3) Network metric-based feature sets are extracted from vascular architectures. Nodes represent junctions in the vasculature; edges represent sources of nutrients in the simulation. Topological features are extracted from the unweighted structure of the network. Hemodynamic features are extracted from attributes of network topologies including hemodynamic characteristics as edge weights. Spatial features account for distance between the information in the network from the center of the simulation. (4) Statistical learning models use network metric-based feature sets to predict emergent tumor output metrics.

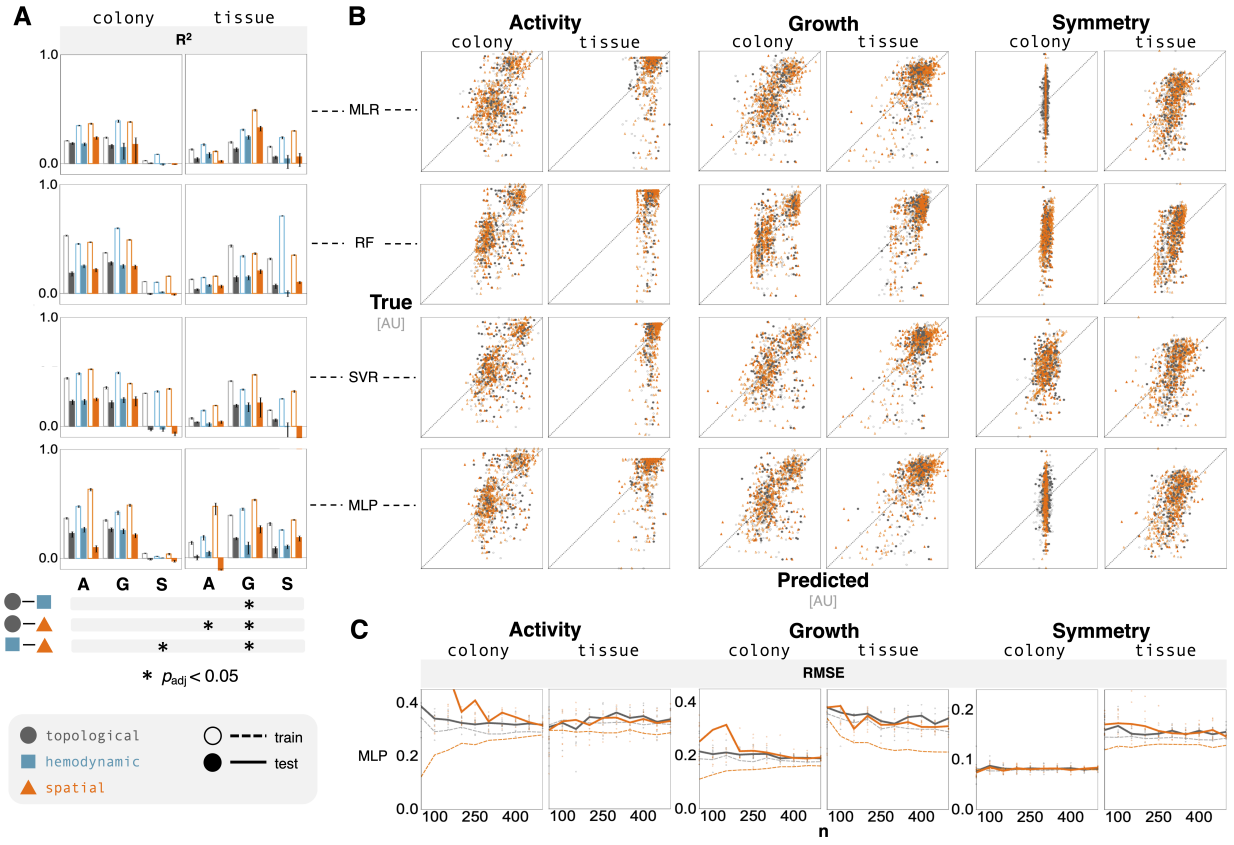

Figure 3.2: **Spatial information does not support emulation** — (**A**) Bar plots show predictive performance of emergent outputs (A: activity, G: growth, S: symmetry) across feature sets for different models (MLR, RF, SVR, and MLP). Feature engineering offered limited improvement. Bar chart values range from -0.1 to 1.0; the horizontal axis is at 0.0. The Bonferroni corrected p-values from a two-way ANOVA highlight significant results (noted with asterisks) that have an adjusted p-value less that 0.05. (**B**) Parity plots show differences between the variance in the predicted response and the true response, comparing the topological and spatial feature sets. (**C**) Additional training data offered diminishing returns on predictive performance of MLP models that were trained on both spatial and topological features. These subplots show the average RMSE as a function of the size of training data. The individual points represent the RMSE from randomized test sets.

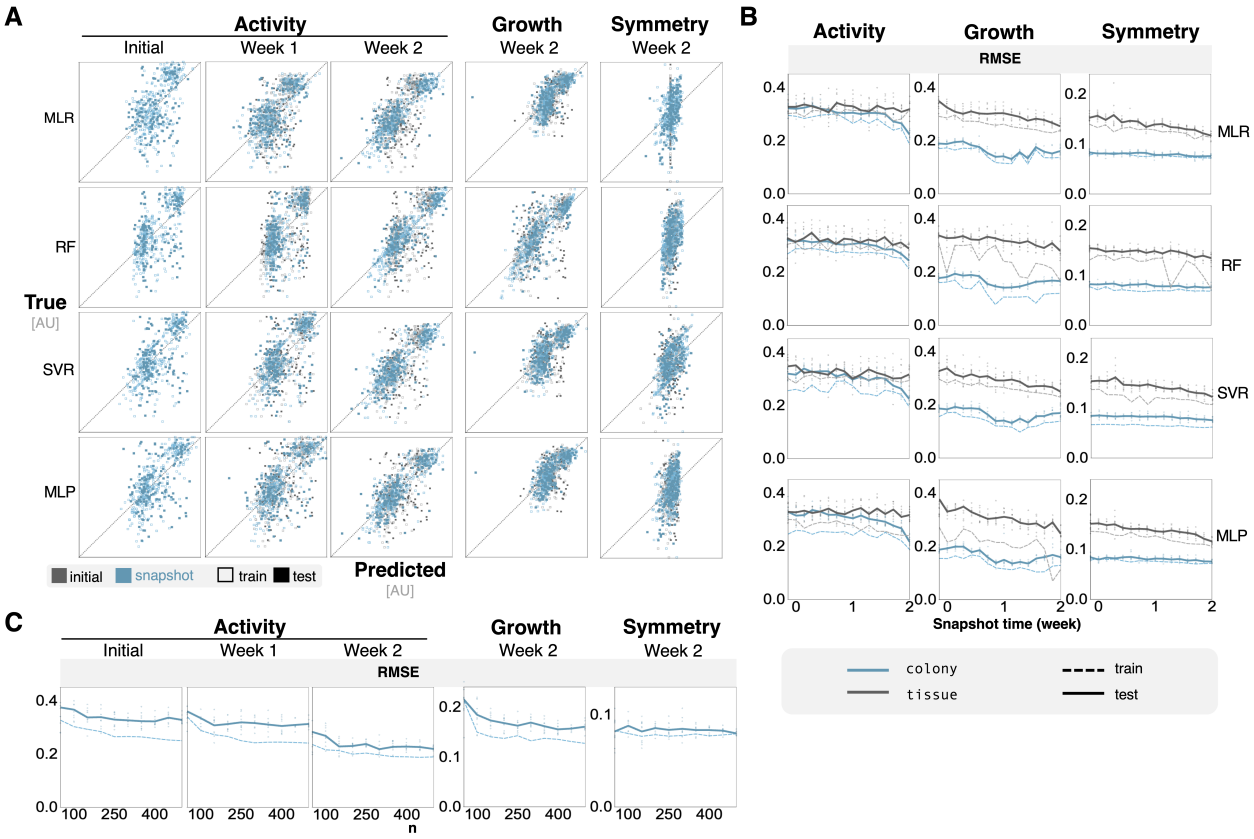

Figure 3.3: **Temporal information improves accuracy of ML models** — (**A**) Parity plots show the predictive performance of `colony` context ML models that were trained on features from later timepoints. The trends are consistent for all three predicted outputs. The week 1 parity plots for growth and symmetry, and the corresponding results for `tissue` context, are included in Supp. Fig. 3.9, and 3.10. (**B**) Line plots show improvement of ML models in both `colony` and `tissue` contexts when they are trained on features from timepoints later in the simulation. (**C**) Predictive performance as a function of training data for MLP models at later timepoints. These subplots show the average RMSE as a function of the size of training data. The individual points represent the RMSE from randomized test sets.

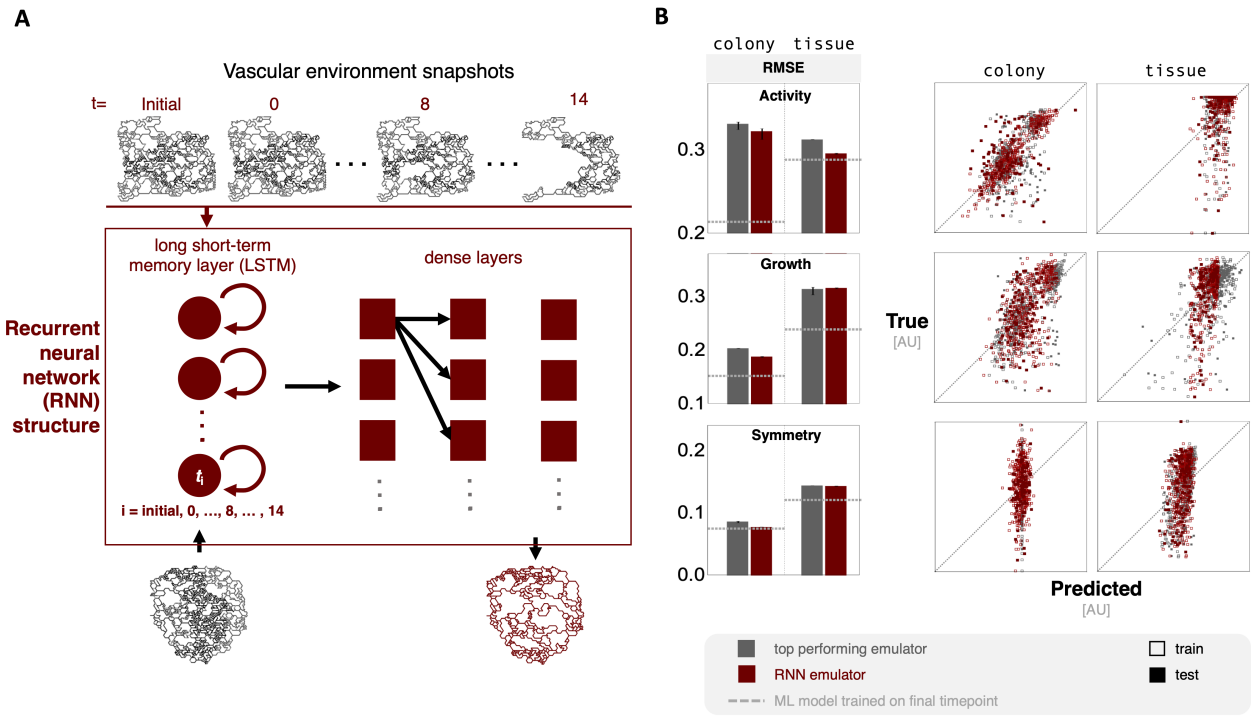

Figure 3.4: **Incorporating temporal information can improve emulation performance** — (**A**) A summary of the vascular network structure and ML model workflow. Network metrics from vascular structures from consecutive simulation days are used as a sequence to train the RNN. Initial vasculatures (represented by the greyscale network) are then used to predict network properties of the vasculatures after two simulation weeks (red network); the two-week predictions are then used to predict emergent behaviors with ML models. (**B**) Bar plots show performance between emulators trained on forecasted network metrics and the top performing naive emulation models. Associated parity plots show the prediction performance of the RNN models across contexts for each emergent output.

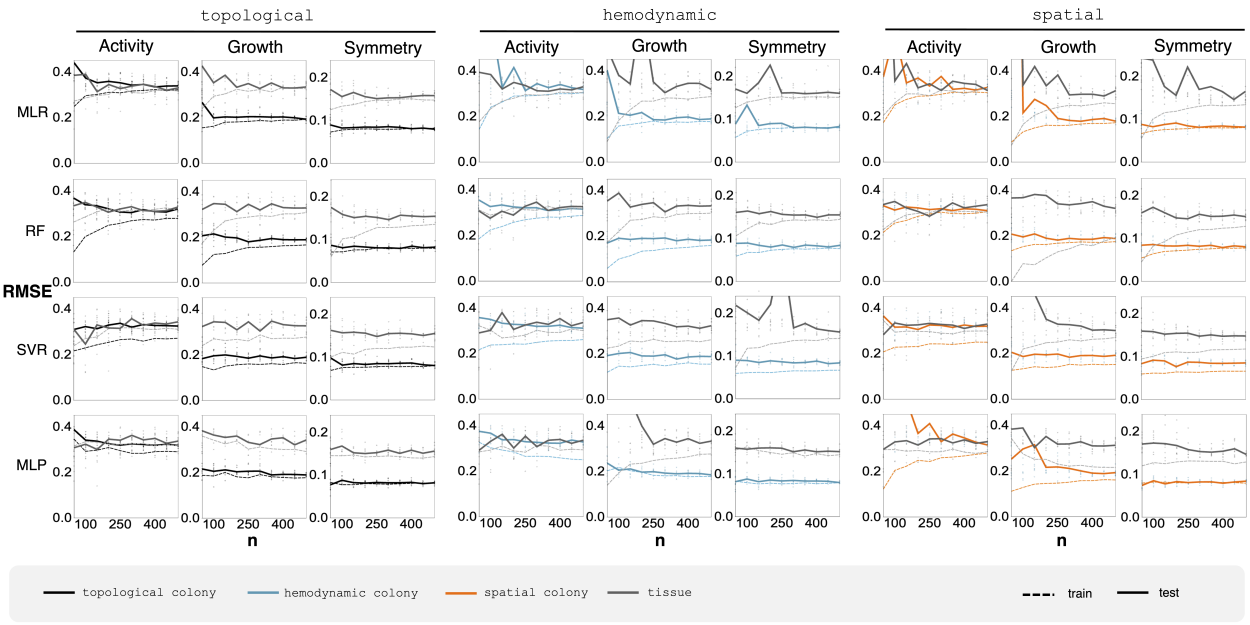

Figure 3.5: **Training data points to diminishing returns** — Line plots indicate the predictive performance of models trained on increasingly large training data sets. In most cases, the RMSE shows diminishing returns of model performance across all feature sets (topolgical, hemodynamic, and spatial), emergent targets (activity, growth, and symmetry), and algorithm (MRL, RF, SVR, MLP).

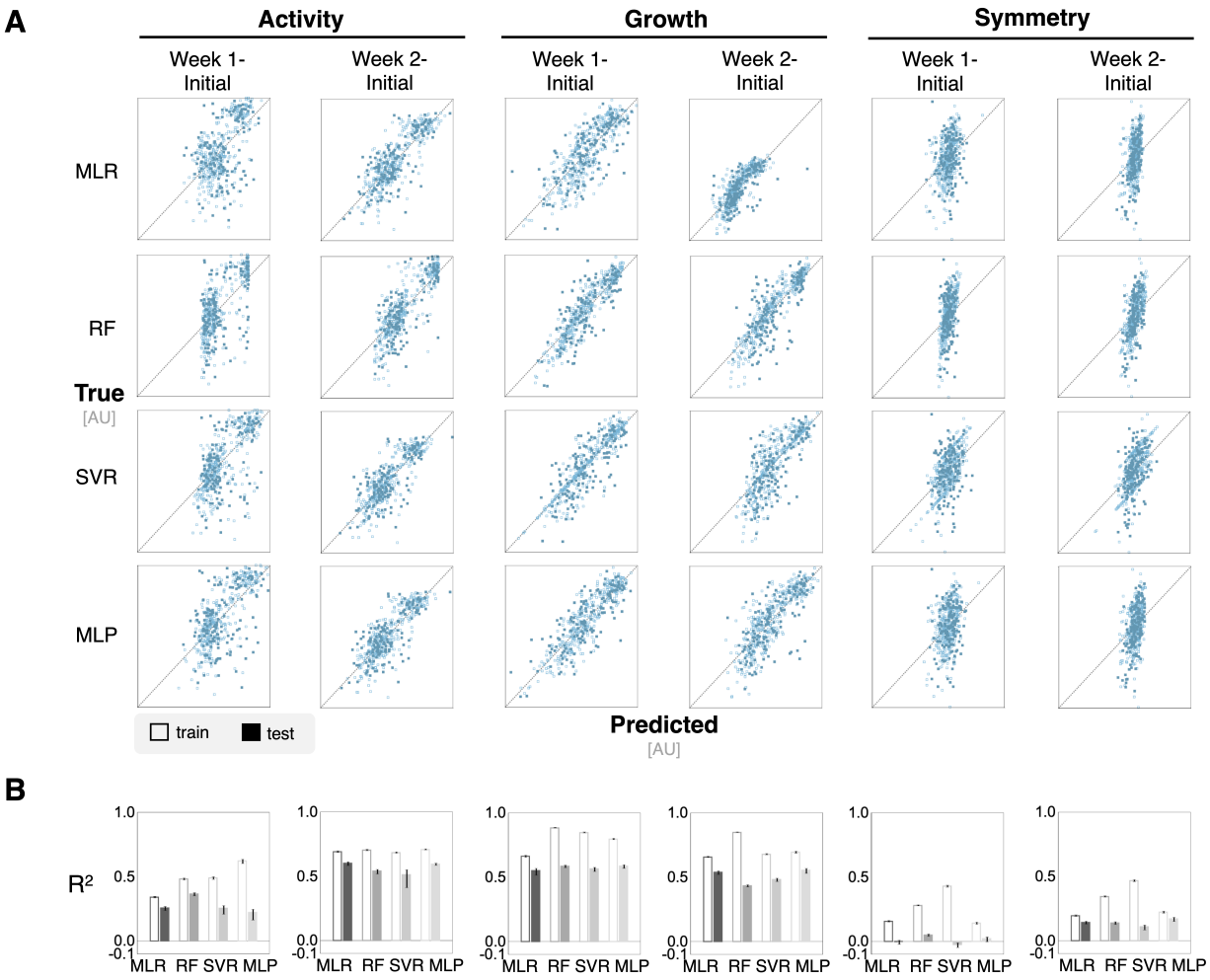

Figure 3.6: **Differential timepoint analysis shows minimal improvement in performance in a `colony` context** — Differential features were calculated by subtracting the features at either one or two weeks from initial features in order to capture the evolution of the features over time. (**A**) Parity plots show the predicted values of emergent targets against the real value to demonstrate fine-grained predictive performance. (**B**) Bar plots show R^2 values for ML models trained on differential features. Growth predictions in the `colony` context improve with simulations using week one and initial timepoint differential features.

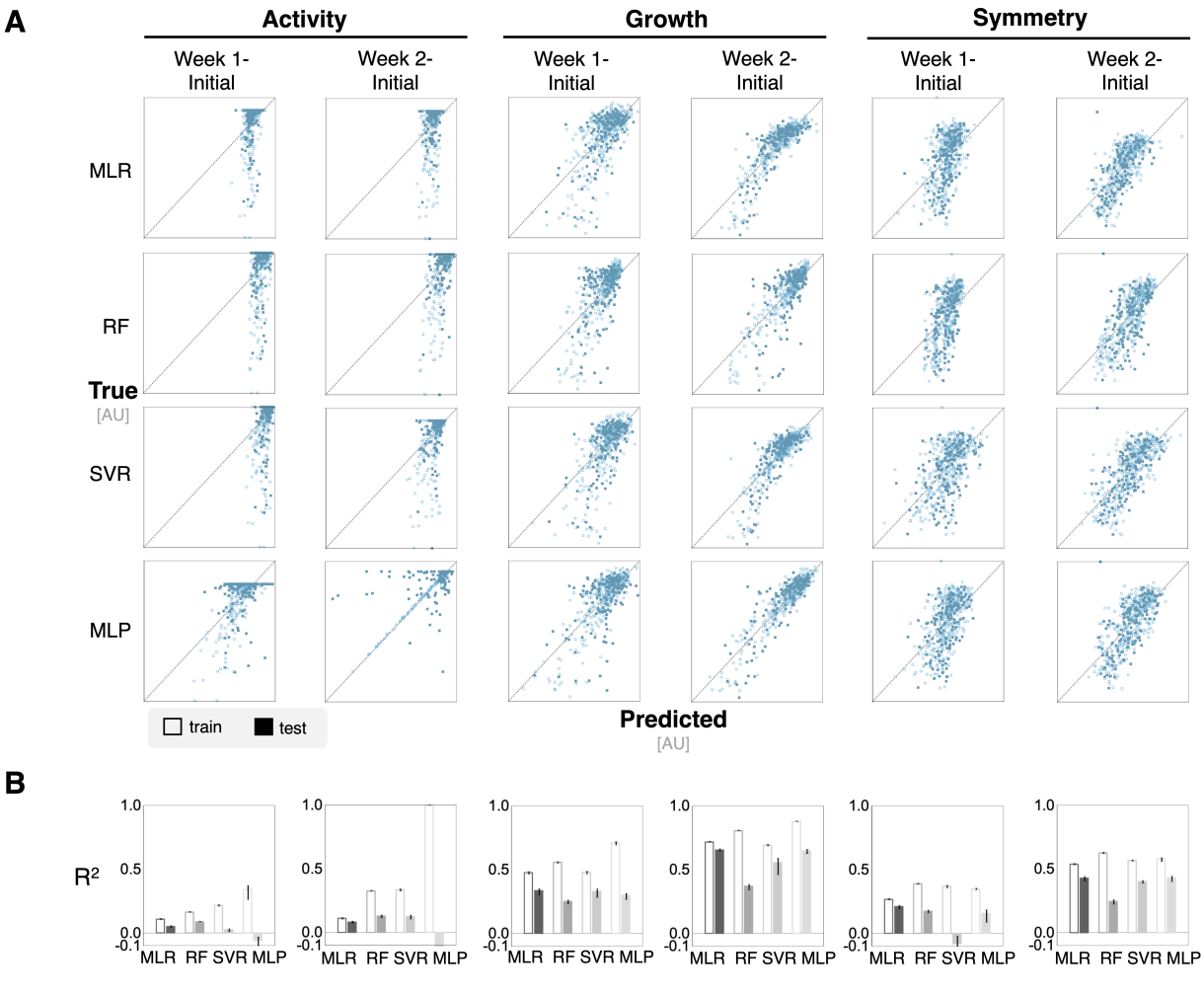

Figure 3.7: **Differential timepoint analysis does not improve performance in a `tissue` context** — Differential features were calculated by subtracting the features at either one or two weeks from initial features in order to capture the evolution of the features over time. (**A**) Parity plots show the predicted values of emergent targets against the real value to demonstrate fine-grained predictive performance. (**B**) Bar plots show R^2 values for ML models trained on differential features.

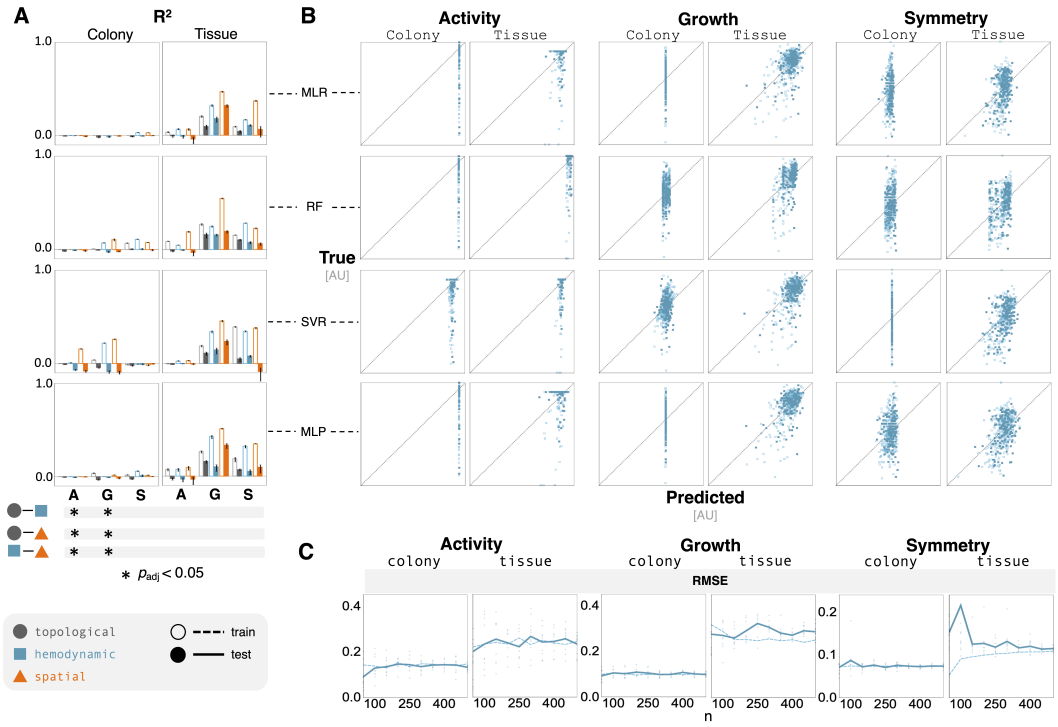

Figure 3.8: **Shortened prediction horizon has minimal effect on emulator performance** — Models predicting spatio-temporal dynamics from after one simulation week. (**A**) Bar plots indicate very poor predictive performance in all cases. Bar chart values range from -0.1 to 1.0; the horizontal axis is at 0.0. The Bonferroni corrected p-values from a two-way ANOVA highlight significant results (noted with black circles) that have an adjusted p-value less that 0.05. (**B**) Parity plots reveal substantial discrepancies in the variance between the predicted and true responses. (**C**) Line plots show predictive performance of the MLP models (the average RMSE) as a function of the size of training data. Performance improvements are limited from additional data. The data points highlight the RMSE from randomized test sets.

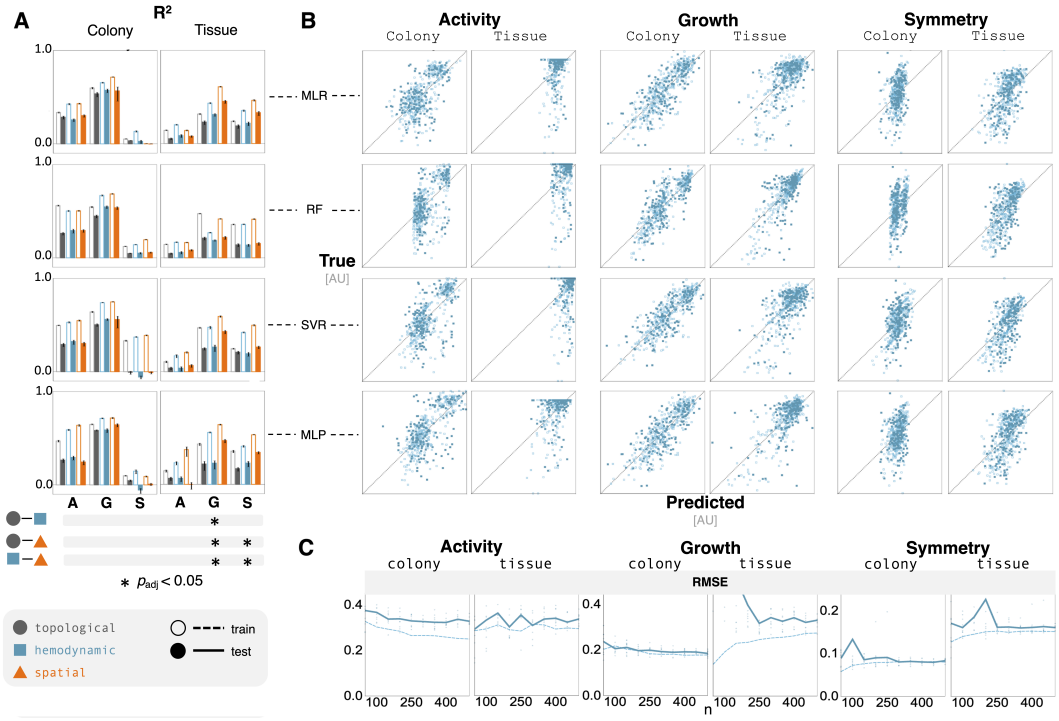

Figure 3.9: **Mid-simulation features show improvement in performance in both contexts** — Models trained on vascular features characterizing network structure at one simulation week result. (**A**) Bar plots show limited improvement from training on mid-simulation features. Bar chart values range from -0.1 to 1.0; the horizontal axis is at 0.0. The Bonferroni corrected p-values from a two-way ANOVA highlight significant results (noted with black circles) that have an adjusted p-value less that 0.05. (**B**) Parity plots reveal large amounts of variance in predicted values with some improvement in growth predictions in the `colony` context. (**C**) Line plots show predictive performance of the MLP models (the average RMSE) as a function of the size of training data. Performance improvements are limited from additional data. The data points highlight the RMSE from randomized test sets.

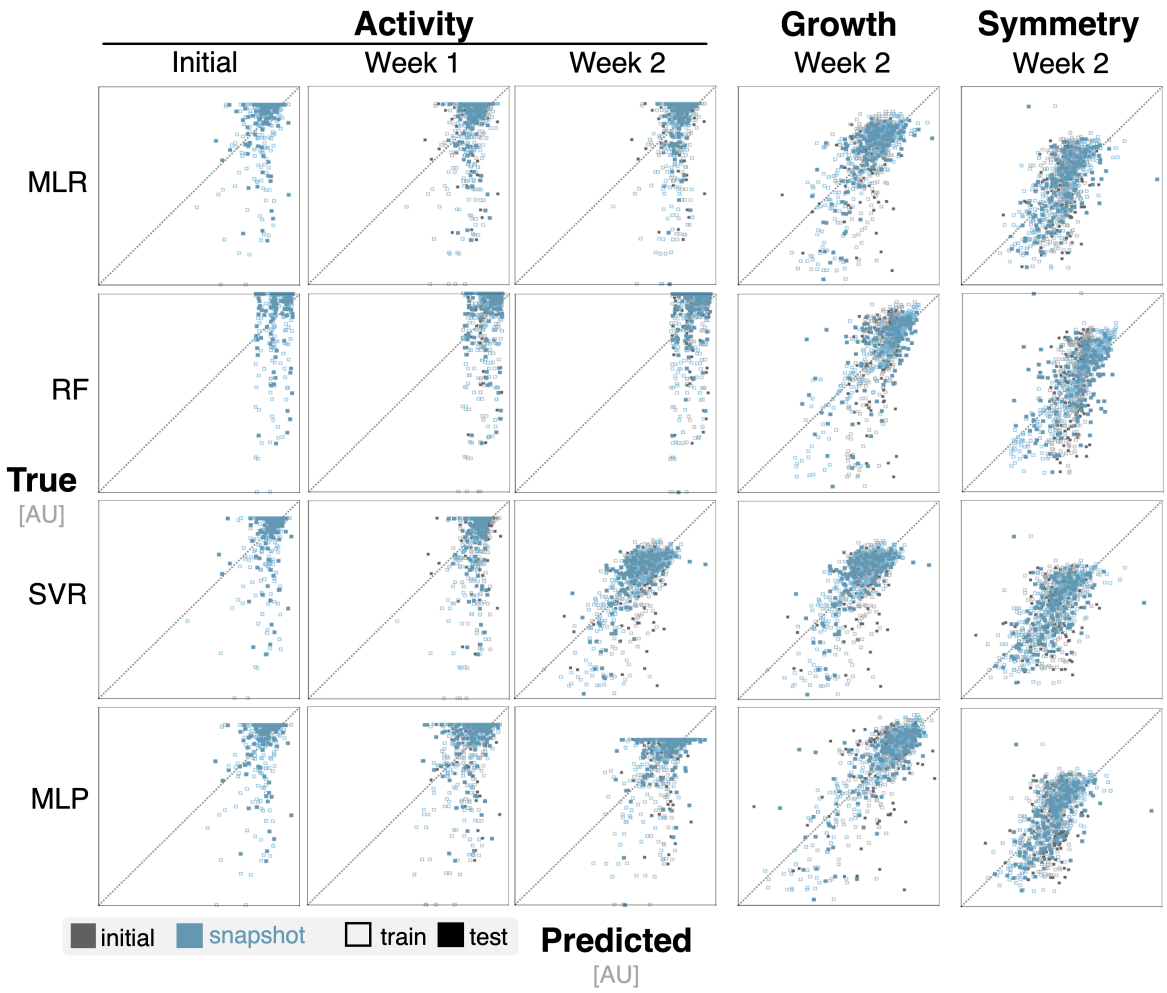

Figure 3.10: **Temporal information improves ML model predictions in `tissue` context** — Parity plots show the predictive performance of ML models trained on features from later timepoints against emulators trained on features from the initial timepoint. Improved prediction of activity is limited; growth and symmetry reflect minimal improvements.

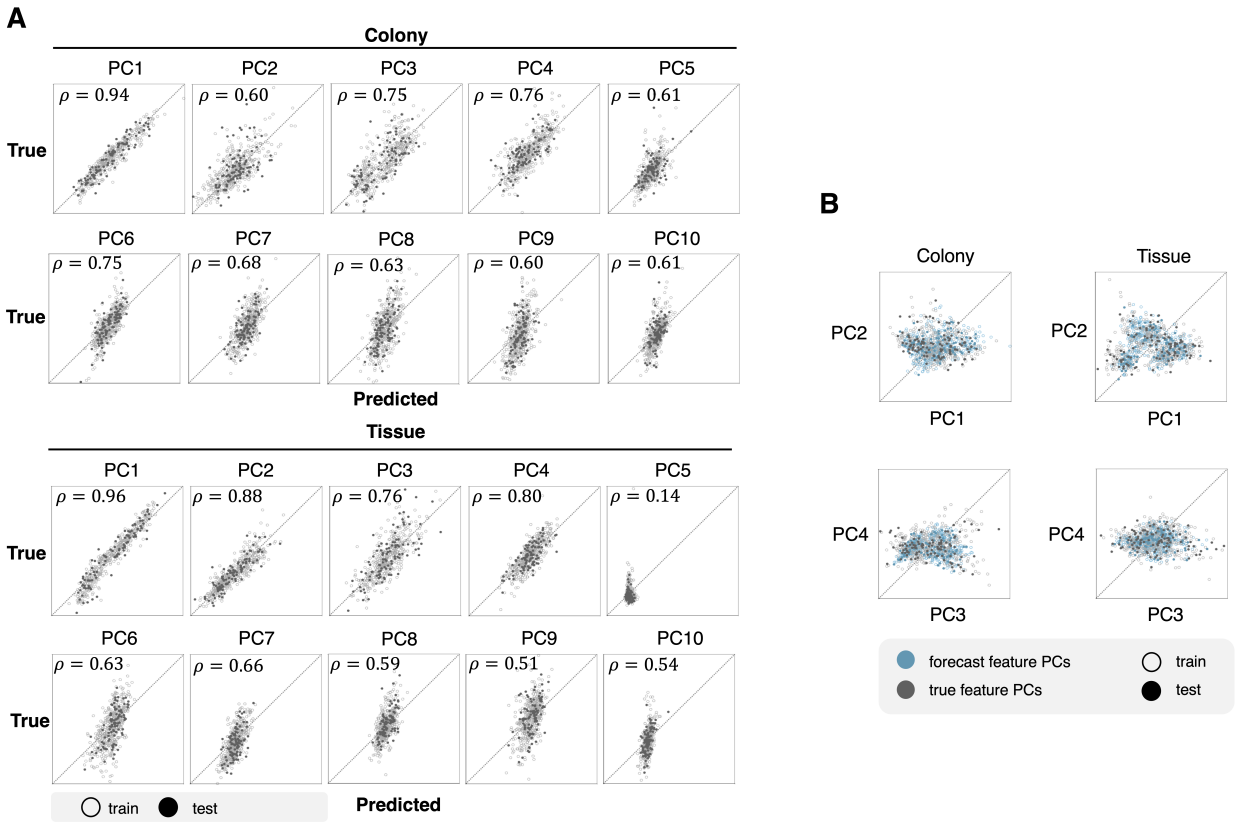

Figure 3.11: **RNN model captures vascular feature variance** — (**A**) Parity plots highlight the performance of a RNN model at predicting network metric features. Both the simulated features and forecasted features were combined to perform PCA. The dimensionality of the feature set are reduced to the first 10 principal components, which represent 95% of the feature variance. The Pearson correlation coefficient (ρ) is reported for each parity plot. (**B**) Scatter plots illustrate the overlap between true and forecasted features using the first four principal components.

Model	Target	Feature set	Hyperparameters
MLR	Activity	Topological	alpha:0.001;l1_ratio:0.1;max_iter:10000
		Hemodynamic	alpha:0.0023; l1_ratio:0.6625; max_iter:10000
		Spatial	alpha:0.0015; l1_ratio:0.9437; max_iter:10000
	Growth	Topological	alpha:0.001; l1_ratio:0.1 ; max_iter:10000
		Hemodynamic	alpha:0.0036; l1_ratio:0.3812; max_iter:10000
		Spatial	alpha:0.0036; l1_ratio:0.3812; max_iter:10000
	Symmetry	Topological	alpha:0.0056; l1_ratio:0.775; max_iter:10000
		Hemodynamic	alpha:0.0036; l1_ratio:0.3812; max_iter:10000
		Spatial	alpha:0.0316; l1_ratio:0.55; max_iter:10000
RF	Activity	Topological	n_estimators:13; max_features:0.1778; max_depth:38; min_samples_split:0.0177; min_samples_leaf:0.0177; bootstrap:True
		Hemodynamic	n_estimators:81; max_features:0.2371; max_depth:81; min_samples_split:0.0133; min_samples_leaf:0.0749; bootstrap:False
		Spatial	n_estimators:81; max_features:0.2371; max_depth:81; min_samples_split:0.0133; min_samples_leaf:0.0749; bootstrap:False
	Growth	Topological	n_estimators:26; max_features:0.3162; max_depth:75; min_samples_split:0.3162; min_samples_leaf:0.0316; bootstrap:False
		Hemodynamic	n_estimators:81; max_features:0.2371; max_depth:81; min_samples_split:0.0133; min_samples_leaf:0.0749; bootstrap:False
		Spatial	n_estimators:81; max_features:0.2371; max_depth:81; min_samples_split:0.0133; min_samples_leaf:0.0749; bootstrap:False
	Symmetry	Topological	n_estimators:57; max_features:0.0749; max_depth:7; min_samples_split:0.4216; min_samples_leaf:0.0237; bootstrap:False
		Hemodynamic	n_estimators:20; max_features:0.0421; max_depth:94; min_samples_split:0.0749; min_samples_leaf:0.1333; bootstrap:False
		Spatial	n_estimators:94; max_features:0.0133; max_depth:69; min_samples_split:0.0237; min_samples_leaf:0.04216; bootstrap:True
SVR	Activity	Topological	C:0.5623; epsilon:0.1778; kernel:rbf
		Hemodynamic	C:0.5623; epsilon:0.1778; kernel:rbf
		Spatial	C:0.5623; epsilon:0.1778; kernel:rbf
	Growth	Topological	C:0.1; epsilon:0.001; kernel:rbf
		Hemodynamic	C:0.2371; epsilon:0.0749; kernel:rbf
		Spatial	C:0.2371; epsilon:0.0749; kernel:linear
	Symmetry	Topological	C:0.2371; epsilon:0.0749; kernel:rbf
		Hemodynamic	C:0.2371; epsilon:0.07498; kernel:rbf
		Spatial	C:0.2371; epsilon:0.0749; kernel:rbf
MLP	Activity	Topological	alpha:0.3162; activation:logistic; hidden_layer_sizes:[50, 25]; solver:lbfgs; max_iter:1000
		Hemodynamic	alpha:0.5623; activation:logistic; hidden_layer_sizes:[25, 50]; solver:lbfgs; max_iter:1000
		Spatial	alpha:0.5623; activation:logistic; hidden_layer_sizes:[25, 50]; solver:lbfgs; max_iter:1000
	Growth	Topological	alpha:0.1778; activation:logistic; hidden_layer_sizes:[25, 25]; solver:lbfgs; max_iter:1000
		Hemodynamic	alpha:0.5623; activation:logistic; hidden_layer_sizes:[5]; solver:lbfgs; max_iter:1000
		Spatial	alpha:0.3162; activation:logistic; hidden_layer_sizes:[25, 25]; solver:lbfgs; max_iter:1000
	Symmetry	Topological	alpha:0.3162; activation:logistic; hidden_layer_sizes:[50]; solver:lbfgs; max_iter:1000
		Hemodynamic	alpha:0.0005; activation:identity; hidden_layer_sizes:[5, 10]; solver:lbfgs; max_iter:1000
		Spatial	alpha:0.1778; activation:logistic; hidden_layer_sizes:[50, 50]; solver:lbfgs; max_iter:1000

Table 3.7: Supplement: Top performing hyperparameters for emulation models in a colony context

Model	Target	Feature set	Hyperparameters
MLR	Activity	Topological	alpha:0.0086; l1_ratio:0.2687; max_iter:10000
		Hemodynamic	alpha:0.0056; l1_ratio:0.775; max_iter:10000
		Spatial	alpha:0.0749; l1_ratio:0.2125; max_iter:10000
	Growth	Topological	alpha:0.0056; l1_ratio:0.775; max_iter:10000
		Hemodynamic	alpha:0.0133; l1_ratio:0.4375; max_iter:10000
		Spatial	alpha:0.0023; l1_ratio:0.6625; max_iter:10000
	Symmetry	Topological	alpha:0.0015; l1_ratio:0.9437; max_iter:10000
		Hemodynamic	alpha:0.0015; l1_ratio:0.9437; max_iter:10000
		Spatial	alpha:0.0015; l1_ratio:0.9437; max_iter:10000
RF	Activity	Topological	n_estimators:38; max_features:0.0562; max_depth:63; min_samples_split:0.5623; min_samples_leaf:0.0562; bootstrap:False
		Hemodynamic	n_estimators:38; max_features:0.0562; max_depth:63; min_samples_split:0.5623; min_samples_leaf:0.0562; bootstrap:False
		Spatial	n_estimators:20; max_features:0.0421; max_depth:94; min_samples_split:0.0749; min_samples_leaf:0.1333; bootstrap:False
	Growth	Topological	n_estimators:13; max_features:0.1778; max_depth:38; min_samples_split:0.0177; min_samples_leaf:0.0177; bootstrap:True
		Hemodynamic	n_estimators:57; max_features:0.0749; max_depth:7; min_samples_split:0.4216; min_samples_leaf:0.0237; bootstrap:True
		Spatial	n_estimators:81; max_features:0.2371; max_depth:81; min_samples_split:0.0133; min_samples_leaf:0.0749; bootstrap:False
	Symmetry	Topological	n_estimators:94; max_features:0.0133; max_depth:69; min_samples_split:0.0237; min_samples_leaf:0.0421; bootstrap:False
		Hemodynamic	n_estimators:94; max_features:0.0133; max_depth:69; min_samples_split:0.0237; min_samples_leaf:0.0421; bootstrap:False
		Spatial	n_estimators:81; max_features:0.2371; max_depth:81; min_samples_split:0.0133; min_samples_leaf:0.0749; bootstrap:False
SVR	Activity	Topological	C:0.5623; epsilon:0.1778; kernel:linear
		Hemodynamic	C:0.2371; epsilon:0.0749; kernel:linear
		Spatial	C:0.2371; epsilon:0.0749; kernel:rbf
	Growth	Topological	C:0.5623; epsilon:0.1778; kernel:rbf
		Hemodynamic	C:1.333; epsilon:0.0133; kernel:linear
		Spatial	C:4.8696; epsilon:0.0205; kernel:linear
	Symmetry	Topological	C:0.2371; epsilon:0.0749; kernel:linear
		Hemodynamic	C:0.2371; epsilon:0.0749; kernel:linear
		Spatial	C:0.1; epsilon:0.001; kernel:linear
MLP	Activity	Topological	alpha:0.3162; activation:logistic; hidden_layer_sizes:[50, 25]; solver:lbfgs; max_iter:1000
		Hemodynamic	alpha:0.5623; activation:logistic; hidden_layer_sizes:[50, 25]; solver:lbfgs; max_iter:1000
		Spatial	alpha:0.5623; activation:logistic; hidden_layer_sizes:[50, 25]; solver:lbfgs; max_iter:1000
	Growth	Topological	alpha:0.5623; activation:logistic; hidden_layer_sizes:[50]; solver:lbfgs; max_iter:1000
		Hemodynamic	alpha:0.5623; activation:identity; hidden_layer_sizes:[25]; solver:lbfgs; max_iter:1000
		Spatial	alpha:0.5623; activation:identity; hidden_layer_sizes:[5]; solver:lbfgs; max_iter:1000
	Symmetry	Topological	alpha:0.5623; activation:tanh; hidden_layer_sizes:[50, 50]; solver:lbfgs; max_iter:1000
		Hemodynamic	alpha:0.5623; activation:identity; hidden_layer_sizes:[5, 5]; solver:lbfgs; max_iter:1000
		Spatial	alpha:0.5623; activation:identity; hidden_layer_sizes:[5, 10]; solver:lbfgs; max_iter:1000

Table 3.8: Supplement: Top performing hyperparameters for emulation models in a `tissue` context

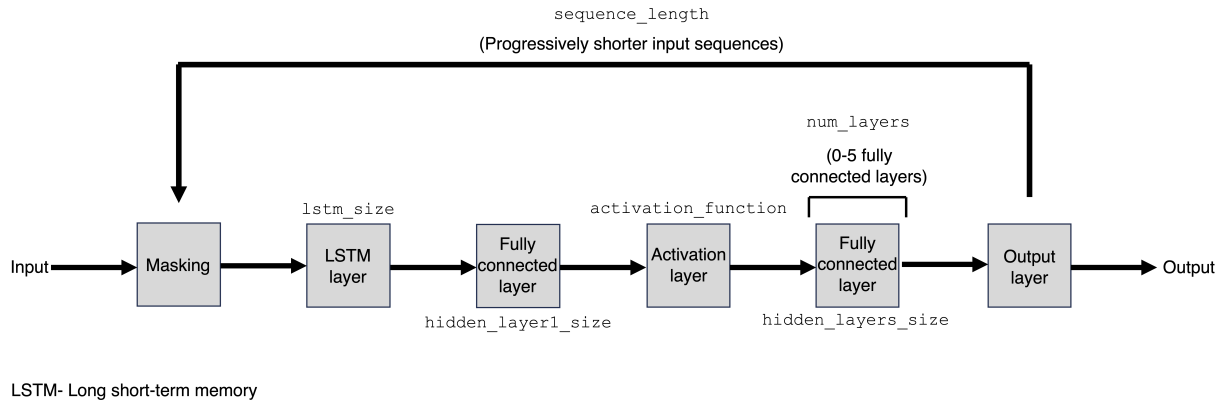

Figure 3.12: **Architecture and structure of the RNN used for feature prediction** — This flowchart describes the layers used while training the feature prediction RNN. All parameter values are defined in Supp. Table 3.9 and optimal values are provided in Supp. Table 3.10.

Hyperparameter	Code	Potential value
Training sequence lengths	sequence_length	[15, 10, 5, 3, 1]] [15, 10, 5, 1] [15, 5, 3, 1]
LSTM layer size	lstm_size	64 128 256 512 1024
Size of first hidden layer	hidden_layer1_size	64 128 256 512 1024
Size of the remaining hidden layers	hidden_layers_size	64 128 256 512 1024
Number of hidden layers after the first	num_layers	0 1 2 3 4 5
Activation function after first hidden layer	activation_function	`relu` `tanh`

Table 3.9: Supplement: Hyperparameters used in RNN architecture grid search

Context	Parameter	Value
colony	sequence_length	[15, 10, 5, 3, 1]
	lstm_size	512
	hidden_layer1_size	1024
	hidden_layers_size	1024
	num_layers	2
	activation_function	relu
tissue	sequence_length	[15, 10, 5, 3, 1]
	lstm_size	512
	hidden_layer1_size	512
	hidden_layers_size	1024
	num_layers	1
	activation_function	relu

Table 3.10: Supplement: Top performing architecture parameters for RNN

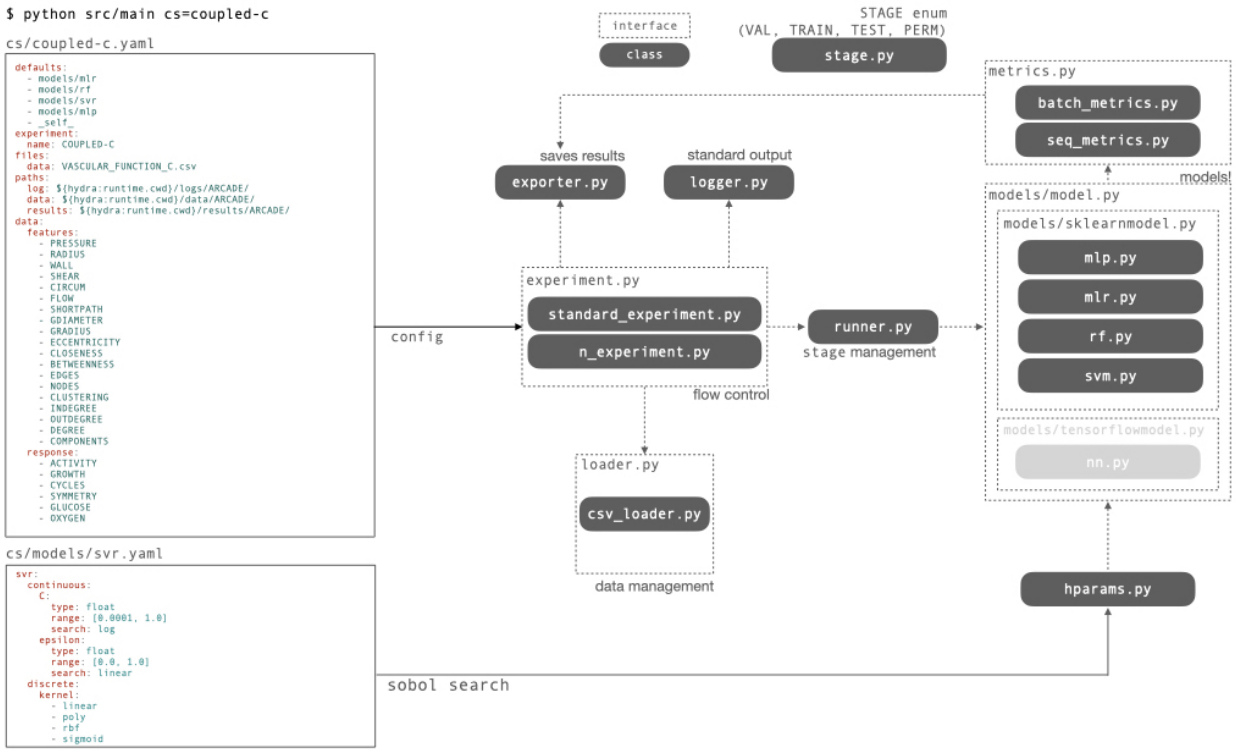

Figure 3.13: **Architecture and structure of the `emulation` codebase** — This flowchart details the design of `emulation` which can be extended to use alternative ML models with the same inputs.

Chapter 4

AGENT-BASED MODELS AS AN *IN SILICO* PLATFORM FOR TRANSLATING *IN VITRO* DERIVED MODELS TO *IN SITU* CONTEXTS

Jason Y. Cain[1], Neda Bagheri[1,2]*

1 Chemical Engineering, University of Washington, Seattle, WA 98195
2 Biology, University of Washington, Seattle, WA 98195

* nbagheri@uw.edu

Author's note: *This work is in preparation, and may be submitted and published in a different format.*

4.1 Introduction

Cancer often co-opts physiological processes that eventually manifest as pathological phenotypes. One such process, angiogenesis, is a key driver in development to support cell growth by sprouting blood vessels to supply nutrients at the tissue-scale.[145] However, there is a clear distinction between pathogenic and nonpathogenic vasculatures.[12,146] Angiogensis is an intuitive target for therapy as avascular tumors cannot normally grow past 1–2mm[147]. However, the application of anti-angiogenic therapies has been largely unsuccessful, indicating gaps in our understanding of the overall regulation of this process.

Angiogenesis is mediated by two common biological cellular regulators: oxygen tension sensing and the secretion of a growth factor.[12,148] Hypoxia—severe oxygen tension—is a result of resource competition and vascular degradation in solid tumors.[149,150] Hypoxia inducible factors (HIFs) are transcriptional regulators that provide sensing machinery for cells. One such process regulated by HIF, specifically HIF-1α, is the secretion of vascular endothelial growth factor (VEGF).[148] VEGF then promote migration and proliferation in the vascular endothelial cells, supporting angiogensis.

There is a distinct need and gap for computational models that can capture this complexity. A computational model designed to include mechanistic insights can help elucidate potential therapeutic targets in a multi-scale control system. We hypothesize that agent-based models (ABMs) that accurately describes the system *in situ* can translate *in vitro* derived models into *in vivo* contexts.

We integrate a validated dynamic model of HIF-1α into ARCADE[46,79]. We investigate the challenges of translating the *in vitro* context of the validated model towards *in vivo* and *in situ* representations of the system. We first validate parameters within the most analagous environment to the original *in vitro* context. We show that this validation does not extend to more complex contexts, resulting from a lack of endogenous regulation in VEGF production from healthy cells. We then discuss the next steps required to complete this work.

4.2 Background

4.2.1 Hypoxia, tumor development, and the tumor microenvironment

Biological systems are rife with sensors. These sensors have been shown to enable responses and adaptation to environmental conditions, to coordinate development, and to be (re)engineered by synthetic biologists to perform computations. Cellular reactions to activation of these sensors can also impact the environment. This bilateral dynamic results in complex control systems that have evolved to manage stress conditions and physiological processes. However, these same sensors are also commonly found to be associated with disregulation implicated in pathogenic phenotypes like cancer.

Effectively treating cancer requires understanding and controlling the disregulation of these sensors[1]. Many solid tumors display an emergent spatial phenomenon in which proliferating cells localize to the outer shell of the tumor[151]. Within the proliferative rim, nutrient limitation leads to a necrotic core surrounded by a layer of quiescent cells[152]. Rapid proliferation of cancer cells and an increased crowding tolerance lead to hypoxia ($<\sim$ 2% pO_2)

within the tumor. In addition to the spatial heterogeneity, tumors also experience temporal heterogeneity comprising three main types of hypoxia: chronic, acute, and cycling[11]. The characterization of these terms and the temporal patterns they characterize are relatively inconsistent in literature due to limited experimental tools to impose hypoxia[24], but there are attempts to capture consistent emergent phenomena[153].

Chronic hypoxia is generally considered as hypoxia that persists for more than 24 hours; it is believed to be the first type of hypoxia encountered during tumor development. Chronic hypoxia can lead to loss of vital cell functions, apoptosis, and cell cycle arrest at a cellular level; it is believed to be associated with therapy resistance, immune suppression, and poor patient outcomes[17]. Chronic hypoxia can arise from diffusion-limited hypoxia as a result of enlarged diffusion distances, enhanced competition, or dysfunctional vasculature. Acute hypoxia is considered shorter term hypoxia (on the order of hours) with greater oxygen tension than chronic hypoxia; it is associated with spontaneous metastasis[149]. This temporal pattern is largely attributed to a temporary flow stoppage or inconsistent red blood cell flux, which is commonly a result of dysfunctional blood vessel development during pathologic angiogenesis[154]. Perpetual exposure of cells to cycles of acute hypoxia results in chronic hypoxia, and the causes and consequences of cyclic hypoxia are less well studied[11].

These different types of hypoxia are associated with dissimilar patterns of differential activation of hypoxia inducible factor (HIF) proteins. HIFs are transcription factors that are important for early embryonic development in the oxygen-limited environment and act as defense mechanisms for cells to avoid necrosis as a result of oxygen limitation[14]. Three iso-

forms of HIFα subunit have been identified in humans HIF-1α, HIF-2α, HIF-3α[155]. HIF-1α and HIF-2α are the two that are most active in cancers and will be the focus of discussion[155], and HIF-1α is the subject of this research. HIF-1α is constitutively expressed in all tissues, but HIF-2α is differentially expressed in certain tissues[11,21]. These two isoforms have unique targets and regulators, with some overlap[156]. They also have similar mechanisms for activation through stabilization.

4.3 Results

Our objective was to demonstrate the utility of agent-based modeling for translation of *in vitro* models towards replicating *in situ* conditions. We add a dynamic subcellular model[157] of HIF to ARCADE[79] as an agent-specific sensing module for hypoxia, the model is detailed in Section 4.5.2, Table 4.5.2 and initial conditions described in Table 4.5.2. Similar to previous work[79], we include the functional coupling of agents to their environment through degradation and remodeling of the vasculature.

Different cellular contexts and populations were used to initialize the system. The `colony` simulations were run with only cancer cells. The `tissue` simulations were run with both healthy cells and cancer cells. The `healthy` simulations were run with only healthy cells. The parameter governing VEGF secretion derived from HIFd-HRE complex formation was manually tuned according to the `colony` context as the most analogous system to the original experimental system.[157]

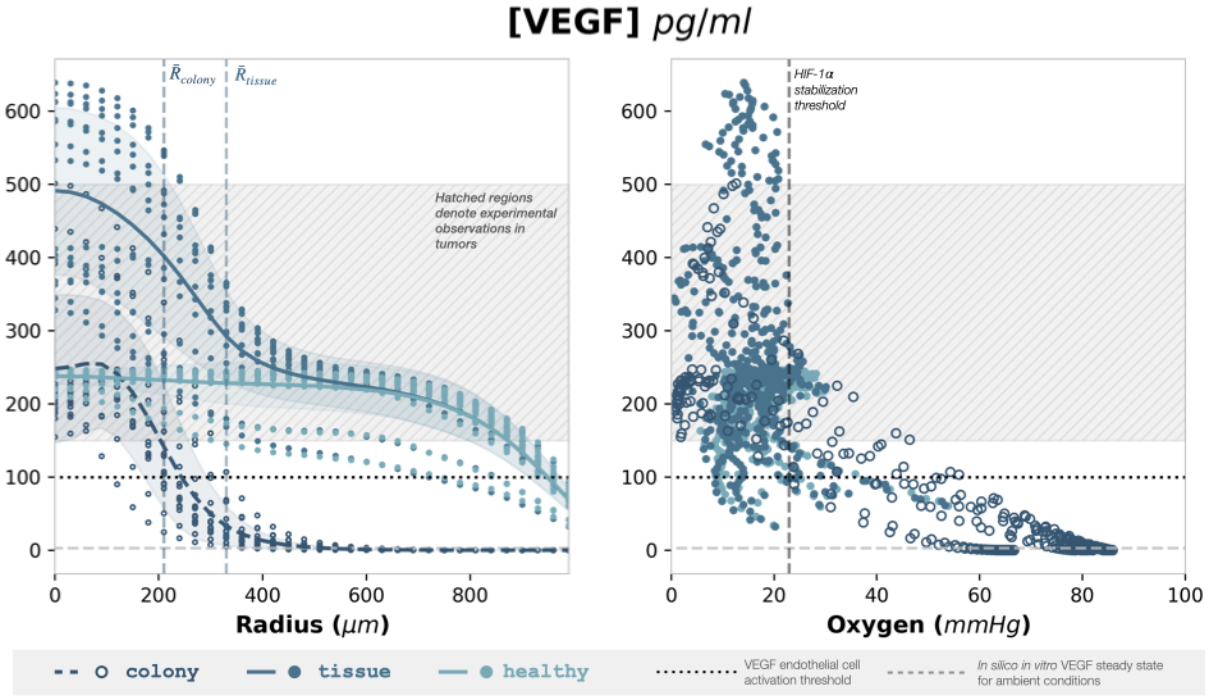

Figure 4.1: **ABM is consistent with emergent observations** — (**LEFT**) Line plot of the VEGF concentration profile versus the distance from the center of the simulation (n=3). Vertical lines indcate the average tumor radius for the `colony` and `tissue` contexts. Horizontal lines indicate key thresholds for VEGF concentrations. Hatched regions show pathological VEGF levels in tumors. Black dotted line indicates angiogenic VEGF concentration threshold.[158] Grey dashed line shows the steady state behavior of analogous `in vitro` simualtions at ambient temperatures. (**RIGHT**) Scatter plot of average VEGF concentrations versus the average oxygen partial pressure at along the radius of the simulation. The vertical line indicates threshold for HIF-1α stabilization.[157] The horizontal lines show the same thresholds as left plot.

4.3.1 `Colony` VEGF profile with dynamic vasculature is consistent to key cancer observations

In order to determine effective integration into the ABM, the VEGF concentration profiles were compared to those obtained experimentally.[159,160] Specifically, the kinetic mass-action

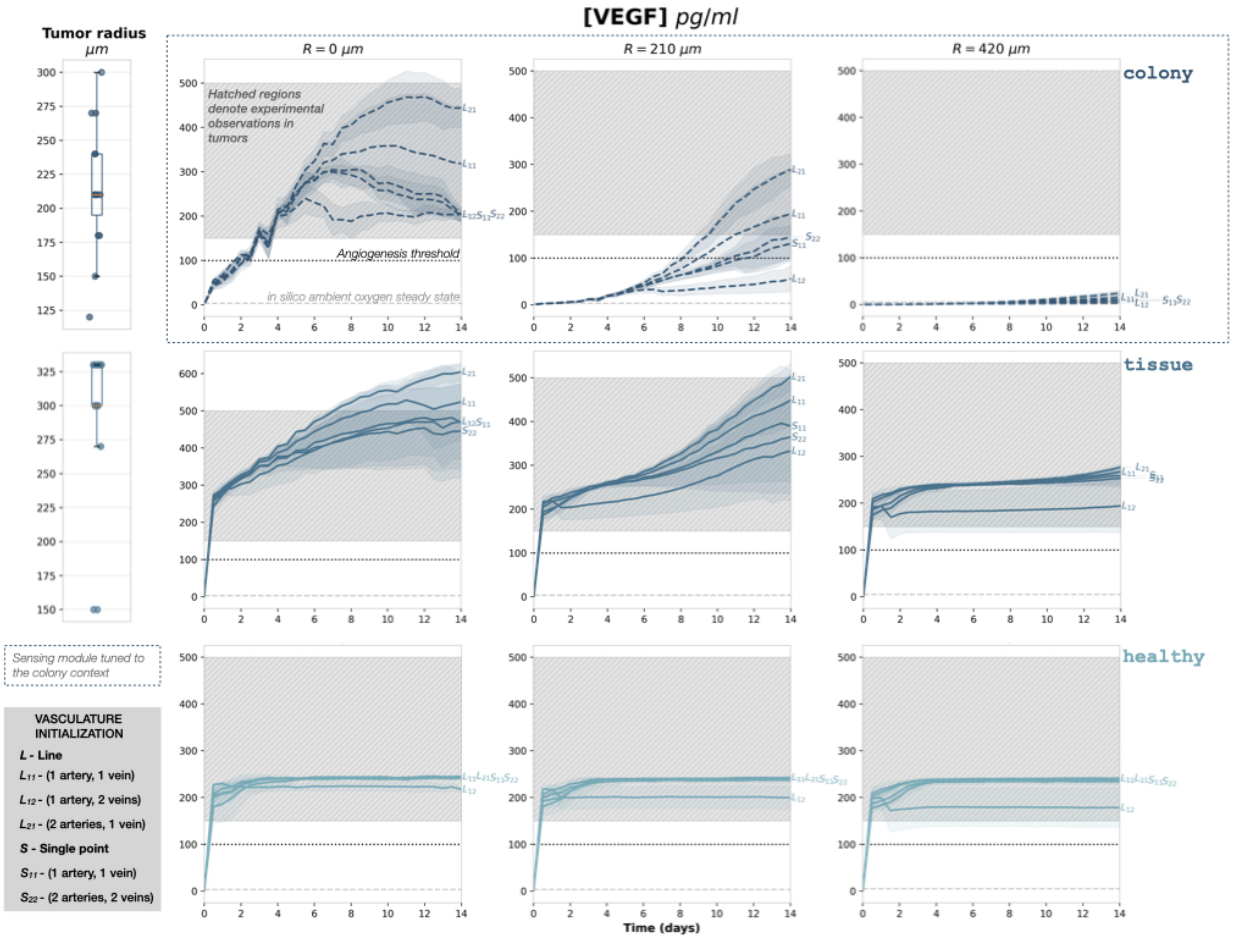

Figure 4.2: **VEGF profiles vary by context** — (**TOP**) ABM was tuned to mimic VEGF behavior found in tumors[159,160], ensuring that the `colony` context passes an activation threshold[158] and reaches a steady state within the experimentally derived range. The dynamic VEGF concentrations is plotted over time, based on the average concentration at a radius from the center of the tumor, for each type of root vasculature seed. The associated radii are plotted in boxplots. (**MIDDLE**) `Tissue` and (**BOTTOM**) `healthy` simulation results are also shown for the same radii as the `colony` simulations. `Tissue` results show non-additive behavior between colony and healthy values, and `healthy` simulations show high VEGF indicative of tumor formation.

parameter translating HRE transcription to VEGF secretion was tuned in the model to match experimental phenotypes. There are two key emergent conditions to which we compare

our model, the VEGF concentration profile and the correlative relationship with hypoxia. Previous computational studies have suggested that the spatial VEGF profile within tumors is fairly consistent since diffusion is fast and VEGF profiles are diffusion limited outside the tumor as the quiescent population dominates VEGF production in tumors.[147] The switch-like behavior of HIF-1α should only support an excess of VEGF for conditions when HIF-1α is stabilized, at oxygen concentrations lower than 3%.

We found that within the tumor radius, VEGF concentration levels are consistent with those experimentally observed for `colony` contexts (Figure 4.3, LEFT; Figure 4.3, TOP-LEFT). Past the radius of the tumor, VEGF is diffusion limited at the rim, still exceeding the angiogenic threshold at later timepoints (Figure 4.3, LEFT, Figure 4.3, TOP-MIDDLE). At large distances from the center of the tumor, the VEGF conditions are minimal and match those conditions. We also observed that the large majority of VEGF exceeding an angiogenic threshold is under hypoxic conditions (Figure 4.3, RIGHT).

4.3.2 VEGF profile is inconsistent with experimental hypoxic distributions

We applied the same HIF-1α dynamic model to two alternate contexts, `tissue`—a combination of tumor population similar to those found in `colony`, and healthy cells which are seeded throughout the radius of the simulation—and `healthy`—healthy cells throughout the radius of the simulation. The same analysis was performed for both contexts, with the exception of calculating tumor radius as is not applicable for `healthy` simulations. Notably, we find that the `tissue` VEGF concentration is not simply additive between the `colony`

and healthy contexts (Figure 4.3).

We find that there are significant inconsistencies with previous observations. The VEGF concentration were much higher in the tissue context than in experimental observations. Furthermore, the inclusion of healthy cells show that there is a significant concentration of VEGF at steady state conditions for tissues. VEGF is not accumulating but is being secreted at a constant rate, despite degradation of VEGF, at in situ environmental conditions. We find that the HIF leakiness in this context indicates that the dynamic model used is an incomplete description of the regulation of HIF-1α in situ, despite considerable validation in a previous publication.[157]

4.4 Discussion

Translating results from in vitro conditions to in situ environments is a long-standing grand challenge in biology. In order to begin closing this gap we use a validated multi-class model that has been designed with a bottom-up approach to mimic the TME. We focus on the regulation of VEGF, which is a key growth factor secreted in hypoxic conditions, a driver of angiogenesis, and a hallmark of solid tumors. We find that when the emergent VEGF concentration is tuned to be consistent with literature, adding healthy cells to the environment leads to significant discrepancies between theory and experiment. The VEGF concentration profile that is experimentally derived from healthy cells is several fold lower than those found in this study.

This translation is especially difficult as hypoxia is uniquely difficult to probe in vitro

as the experimental tools to create hypoxic conditions are limited.[23,161] The main discrepancy between healthy cells *in vitro* and *in situ* is a large difference between ambient and physiological tissue oxygen pressures. These results suggest that there is an additional regulator preventing the leakiness of HIF behavior.

This work is incomplete in its current iteration. This framework provides researchers the means to hypothesize additional regulation that can be included in this model; I propose such model extensions below. The lack of *in silico* dynamic behavior of tissue-derived VEGF suggests that the model incorporate more switch-like behavior, hypothetically resulting from cooperativity. However, this model assumes that the sole transcription factor driving VEGF is HIF-1α. One could include additional hypoxia regulators that could limit VEGF production from healthy cells, like HIF-2α or secretion regulation. Another key multi-scale behavior that is not described in this model is the sequestering of VEGF in the vasculature. This factor motivates the need to include dynamic vascular growth (i.e. angiogenesis) into ARCADE.

We believe that ABMs are promising tools to translate results to multiple contexts, but certain challenges need to be addressed. Tuning an ABM is extremely laborious, thus optimization methods are important to continue developing for this class of model. Validation of ABMs remains challenging due to the multiplexed nature of their design. ABMs can incur long development times and high computational costs, motivating standardization—much like other classes of models—to support both thrusts. Once these limitations are addressed, there is incredible potential in the widespread adoption of ABMs.

4.5 Materials and Methods

4.5.1 Data and code availability

All source code for the adapted ARCADE ABM is publicly available on GitHub at `github.com/bagherilab/ARCADE`.

4.5.2 Sensing module

The sensing module has been adapted from a dynamic model of HIF stability dynamics.[157] Many of the equations are the same as those outlined in the previous publication. k_i refer to reaction constants, with i corresponding to a respective ν_i reaction rate. Km refer to Michaelis-menten kinetic equations. O_2 is taken from the local environment of the cell in the agent-based model. Consitent with the original model, oxygen levels are treated as percentage of normoxic conditions, assumed to be 160 mmHg. All parameters were fit or derived from existing literature to constrain the search space. However, the generation of VEGF has been added to this model and had not been fit previously, a key subject of this study. We list the reaction rates below.

Cells first poll the environment `lattice` to determine oxygen concentration. The reactions were calculated to determine the amount of VEGF produced per model step, defined in MASON[162]. VEGF concentration was then reduced to zero and all VEGF was placed in a VEGF environment `lattice`.

$$\nu_1 = k_1$$

$$\nu_2 = k_2[\text{HIF-1}\alpha]$$

$$\nu_3 = k_3[\text{PHD}]\frac{\text{O}_2}{Km_{3a} + \text{O}_2}\frac{[\text{HIF-1}\alpha]}{Km_{3b} + [\text{HIF-1}\alpha]}$$

$$\nu_4 = k_4[\text{VHL}]\frac{[\text{HIF-1}\alpha\text{-pOH}]}{Km_4 + [\text{HIF-1}\alpha\text{-pOH}]}$$

$$\nu_5 = k_5[\text{FIH}]\frac{\text{O}_2}{Km_{5a} + \text{O}_2}\frac{[\text{HIF-1}\alpha]}{Km_{5b} + [\text{HIF-1}\alpha]}$$

$$\nu_6 = k_6[\text{HIF-1}\alpha\text{-aOH}]$$

$$\nu_7 = k_7[\text{PHD}]\frac{\text{O}_2}{Km_{7a} + \text{O}_2}\frac{[\text{HIF-1}\alpha\text{-aOH}]}{Km_{7b} + [\text{HIF-1}\alpha\text{-aOH}]}$$

$$\nu_8 = k_8[\text{VHL}]\frac{[\text{HIF-1}\alpha\text{-aOHpOH}]}{Km_4 + [\text{HIF-1}\alpha\text{-aOHpOH}]}$$

$$\nu_9 = k_9[\text{HIF-1}\alpha]$$

$$\nu_{10} = k_{10}[\text{HIF-1}\alpha_n]$$

$$\nu_{11} = k_{11}[\text{PHD}]$$

$$\nu_{12} = k_{12}[\text{PHD}_n]$$

$$\nu_{13} = k_{13}[\text{HIF-1}\alpha\text{-aOH}]$$

$$\nu_{14} = k_{14}[\text{HIF-1}\alpha_n\text{-aOH}]$$

$$\nu_{15} = k_{15}[\text{PHD}_n]\frac{\text{O}_2}{Km_{15a} + \text{O}_2}\frac{[\text{HIF-1}\alpha_n]}{Km_{15b} + [\text{HIF-1}\alpha_n]}$$

$$\nu_{16} = k_{16}[\text{VHL}_n]\frac{[\text{HIF-1}\alpha_n\text{-pOH}]}{Km_{16} + [\text{HIF-1}\alpha_n\text{-pOH}]}$$

$$\nu_{17} = k_{17}[\text{FIH}_n]\frac{\text{O}_2}{Km_{17a} + \text{O}_2}\frac{[\text{HIF-1}\alpha_n]}{Km_{17b} + [\text{HIF-1}\alpha_n]}$$

$$\nu_{18} = k_{18}[\text{HIF-1}\alpha_n\text{-aOH}]$$

$$\nu_{19} = k_{19}[\text{PHD}_n]\frac{\text{O}_2}{Km_{19a} + \text{O}_2}\frac{[\text{HIF-1}\alpha_n\text{-aOH}]}{Km_{19b} + [\text{HIF-1}\alpha_n\text{-aOH}]}$$

$$\nu_{20} = k_{20}[\text{VHL}_n]\frac{[\text{HIF-1}\alpha_n\text{-aOHpOH}]}{Km_{20} + [\text{HIF-1}\alpha_n\text{-aOHpOH}]}$$

$$\nu_{21} = k_{21f}[\text{HIF-1}\alpha_n][\text{HIF}\beta] - k_{21r}[\text{HIF}_d]$$

$$\nu_{22} = k_{22f}[\text{HIF}_d][\text{HRE}] - k_{21r}[\text{HIF}_d\text{-HRE}]$$

$$\nu_{23} = k_{23}[\text{HIF}_d\text{-HRE}]$$

$$\nu_{24} = k_{24}[\text{PHD}]$$

$$\nu_{25} = k_{25}[\text{HIF}_d\text{-HRE}]$$

$$\frac{d[\text{HIF-1}\alpha]}{dt} = \nu_1 - \nu_2 - \nu_9 + \nu_{10} - \nu_3 - \nu_5 + \nu_6$$
$$\frac{d[\text{HIF-1}\alpha\text{-pOH}]}{dt} = \nu_3 - \nu_4$$
$$\frac{d[\text{HIF-1}\alpha\text{-aOH}]}{dt} = \nu_5 - \nu_6 - \nu_7 - \nu_{13} + \nu_{14}$$
$$\frac{d[\text{HIF-1}\alpha\text{-aOHpOH}]}{dt} = \nu_7 - \nu_8$$
$$\frac{d[\text{HIF-1}\alpha_n]\text{-pOH}}{dt} = \nu_{15} - \nu_{16}$$
$$\frac{d[\text{HIF-1}\alpha_n]}{dt} = \nu_9 - \nu_{10} - \nu_{17} + \nu_{18} - \nu_{15} - \nu_{21}$$
$$\frac{d[\text{HIF}_d]}{dt} = \nu_{21} - \nu_{22}$$
$$\frac{d[\text{HIF}_d\text{-HRE}]}{dt} = \nu_{22}$$
$$\frac{d[\text{HIF-1}\alpha_n\text{-aOH}]}{dt} = \nu_{17} - \nu_{18} - \nu_{19}$$
$$\frac{d[\text{HIF-1}\alpha_n\text{-aOHpOH}]}{dt} = \nu_{19} - \nu_{20}$$
$$\frac{d[\text{PHD}]}{dt} = \nu_{23} - \nu_{24} - \nu_{11} + \nu_{12}$$
$$\frac{d[\text{PHD}_n]}{dt} = \nu_{11} - \nu_{12}$$
$$\frac{d[\text{HIF}\beta]}{dt} = -\nu_{21}$$
$$\frac{d[\text{HRE}]}{dt} = -\nu_{22}$$
$$\frac{d[\text{VEGF}]}{dt} = \nu_{25}$$

4.5.3 VEGF `lattice` properties

The diffusion of VEGF was calcualted using a reaction diffusion equation:

$$\frac{\partial C}{\partial t} = D\nabla C + R_a - R_d$$

Reaction	Description
1	Constitutive generation of HIF-1α
2	HIF-1α degredation
3	Prolyl hydroxylation by PHD
4	VHL degredation of HIF-1α-pOH
5	Asparaginyl hydroxylation of HIF-1α by FIH
6	HIF-1α-aOH de-hydroxylation
7	Prolyl hydroxylation by PHD for HIF-1α-pOH
8	VHL degredation of HIF-1α-aOHpOH
9	HIF-1α translocation to nucleus
10	HIF-1α translocation to cytoplasm
11	PHD translocation to nucleus
12	PHD translocation to cytoplasm
13	HIF-1α-aOH translocation to nucleus
14	HIF-1α-aOH translocation to cytoplasm
15	Nuclear localization of reaction 3
16	Nuclear localization of reaction 4
17	Nuclear localization of reaction 5
18	Nuclear localization of reaction 6
19	Nuclear localization of reaction 7
20	Nuclear localization of reaction 8
21	HIF-1α nuclear dimerization with HIFβ
22	HIF_d binding to HRE
23	Production of PHD from HIF_d-HRE
24	Degredation of PHD
25	Production and secretion of VEGF from HIF_d-HRE

Table 4.1: Description of reactions in dynamic HIF signaling model.

Molecule	Initial concentration (nM)
[HIF-1α]	5
[HIF-1α-pOH]	0
[HIF-1α-aOH]	0
[HIF-1α-aOHpOH]	0
[HIF-1α_n]-pOH	0
[HIF-1α_n]	0
[HIF$_d$]	0
[HIF$_d$-HRE]	0
[HIF-1α_n-aOH]	0
[HIF-1α_n-aOHpOH]	0
[PHD]	100
[PHD$_n$]	0
[HIFβ]	170
[HRE]	50
[VEGF]	0

Table 4.2: Initial conditions for HIF signaling pathway.

Where C is the concentration of VEGF. D is the diffusivity constant, here 10 $\mu m^2/s$.[147] R_a is the amount added from the reaction detailed in previous section, and R_d is the degredation rate, here $0.0045s^{-1}$.[147]

4.5.4 ARCADE simulations and workflow

Input file tags

`*_invitro_ambient.xml`

```
<globals>
    <global id="CONCENTRATION_OXYGEN" value="160" units="mmHg" />
</globals>
<components>
    <component type="sites" class="source"/>
</components>
```

*_invitro_normoxia.xml

```
<globals>
    <global id="CONCENTRATION_OXYGEN" value="40" units="mmHg" />
</globals>
<components>
    <component type="sites" class="source"/>
</components>
```

*_invitro_hypoxia.xml

```
<globals>
    <global id="CONCENTRATION_OXYGEN" value="15" units="mmHg" />
</globals>
<components>
    <component type="sites" class="source"/>
</components>
```

*_invitro_anoxia.xml

```
<globals>
    <global id="CONCENTRATION_OXYGEN" value="0.02" units="mmHg" />
</globals>
<components>
    <component type="sites" class="source"/>
</components>
```

colony_*.xml

```
<agents initialization="0">
    <populations>
        <population type="C" fraction="0.0">
            <variables>
                <variable id="max_height" scale="1.5" />
                <variable id="meta_pref" scale="1.5" />
                <variable id="migra_threshold" scale="0.5" />
            </variables>
        </population>
    </populations>
    <helpers>
        <helper type="insert" delay="1440" populations="0" bounds="0.05"/>
    </helpers>
</agents>
```

tissue_*.xml

```
<agents initialization="0">
    <populations>
        <population type="C" fraction="0.0">
            <variables>
                <variable id="max_height" scale="1.5" />
                <variable id="meta_pref" scale="1.5" />
                <variable id="migra_threshold" scale="0.5" />
            </variables>
        </population>
        <population type="H" fraction="1.0" />
    </populations>
    <helpers>
        <helper type="insert" delay="1440" populations="0" bounds="0.05"/>
        <helper type="insert" delay="720" populations="1" bounds="1.0"/>
    </helpers>
</agents>
```

healthy_*.xml

```
<agents initialization="0">
    <populations>
        <population type="H" fraction="1.0" />
    </populations>
    <helpers>
        <helper type="insert" delay="720" populations="0" bounds="1.0"/>
    </helpers>
</agents>
```

Chapter 5

THE IN SILICO LAB: IMPROVING ACADEMIC CODE USING LESSONS FROM BIOLOGY

Jason Y. Cain[1], Jessica S. Yu[2], Neda Bagheri[1,2]*

1 Chemical Engineering, University of Washington, Seattle, WA 98195
2 Biology, University of Washington, Seattle, WA 98195

† These authors contributed equally to this work.
* nbagheri@uw.edu

Author's note: *This work was published in Cell Systems in January 2023.*[163] *This article highlights the importance of code quality as models become more sophisticated.*

ABSTRACT

"Good code" is often regarded as a nebulous, impractical ideal. Common best practices towards improving code quality can be inaccessible to those without a rigorous computer science or software engineering background, contributing to a gap between advancing scientific research and FAIR practices. We seek to equip researchers with the necessary background and context to tackle the challenge of improving code quality in computational biology research using analogies from biology to synthesize *why* certain best practices are critical for advancing computational research. Improving code quality requires active stewardship; we encourage researchers to deliberately adopt and share practices that ensure reusability, repeatability, and reproducibility.

KEYWORDS: code quality; reproducibility; reusability; computational biology

5.1 Introduction

Complex code bases often resemble fragile ecosystems on the verge of collapse, vulnerable to any perturbation. The study of larger and more intricate biological systems has motivated both the ubiquity and increasing complexity of computational models. However, inconsistent management of code complexity introduces gaps between best intentions and best practices. The computational biology community is not exempt from the ongoing reproducibility crisis[164,165]. Proactively improving the quality of code would significantly improve both reproducibility of, and trust in, code. Unfortunately, translating jargon around best

practices is difficult for those without a background in software development.

Researchers should prioritize building code that is consistent with FAIR practices[84]. There exist countless community resources that aim to standardize modeling tools[166–169] as well as accessible introductions to coding and relevant best practices[170–174]. Far fewer discussions focus on, and motivate practices for, improving code quality in academic research[175,176].

This perspective presents computer science and software engineering concepts through the lens of biological analogies to support academic researchers of varying programming proficiency. The guidelines introduced in this perspective are not intended to be static nor dogmatic. We simply aim to highlight some important considerations that reinforce the impact of computational research for both the individual researcher and the greater academic community. Just as research evolves and refines over time, the following principles will similarly evolve and refine.

5.1.1 Context: Understanding the environmental niche

Just as the environmental context guides the questions asked and approaches used when studying biological systems, the unique considerations specific to academic research motivate a nuanced selection of best practices rather than wholesale adoption. We acknowledge that most industrial implementations of software engineering practices are broader in scope than the requirements, use-cases, and especially audience of academic code. Universal recommendations do not account for the context of one's greater research community. A research community may maintain certain conventions (e.g. model-type specific markup languages

like SBML[177]) that support reproducible code. Different approaches are better when designing tools versus one-off analyses. Practices must be in place to maintain a publication legacy of an improving product. Understanding *why* certain coding practices exists equips researchers to identify the most impactful conventions.

A substantial amount of software engineering terminology has particular relevance to the dynamic needs of academic research. For example, reducing **coupling**—or avoiding code that needs to be edited simultaneously—provides flexibility to make adjustments or course corrections without disruptive systemic changes. Likewise, researchers will inherently need to adapt the design of the code and deliverable(s) to changing contexts. Improving modularity in code design supports the flexibility to account for the exploratory nature of research. Another example is **refactoring**—improving the readability without changing the functionality—which helps improve the transparency of methods. While Table 5.1 is not comprehensive, it is a useful starting point for contextualizing and accelerating the implementation of better coding practices.

To empower researchers with the necessary context to independently evaluate the impact of best practices, we highlight the following considerations:

- The choice of programming language has nuanced impacts on both computational and development time in addition to syntax considerations.
- A deliberate approach to version control supports improved transparency and organization.
- Tests provide far more utility than simply verifying code correctness.

- Different deployment strategies have a significant impact on the reach and accessibility of research results.
- Readability and design are equally if not more important than documentation for determining the quality of code.

By introducing common best practices and jargon through analogous biological principles, we hope to guide and facilitate nuanced discussion supporting how researchers write, distribute, and publish code.

5.2 Computational counterparts to biological principles

5.2.1 Instrumentation: Pick the right tool for the job

No single experimental instrument can meet all sensitivity and throughput demands for diverse research samples. Thus, researchers identify, evaluate, and select appropriate tools for their question and system. Similarly, all programming languages have strengths and weaknesses. One's preferred programming language may be adequate, but another language, or a combination thereof, may excel at the same task.

Different languages offer various levels of abstraction or features that can be more appropriate for certain applications. Programming languages can be roughly classified on a spectrum of abstraction—how far removed the software interface is from the hardware considerations. Higher-levels of abstraction often hide implementation details for the sake of ease-of-use. Generally, higher-level languages provide shorter development times by silently handling operations such as matrix multiplication. Some higher-level languages stream-

line development with native functionality for specific types of analyses, and therefore are designed to be specialized tools rather than general-purpose languages; popular examples include R and MATLAB. Higher-level languages are often designed with cross-platform functionality in mind. In contrast, lower-level languages provide more control, usually resulting in better performance, but the burden is placed on the developer to manage computational resources more intentionally. Lower-level languages allow for more flexibility of implementation and applications; a common example is C++.

Researchers need to consider both computational and development time when selecting a programming language, while also considering the conventions in their field. A common challenge is that readability does not always support efficient computing, and vice versa. Development time—the time required to design, write, test, and turnover code—can be shortened by using appropriate languages that emphasize readability. These languages tend to be **interpreted** languages, which are executed directly from the code syntax. Computational expense—the time and memory required to execute the code—is a bottleneck that can be managed with high degrees of parallel computing and additional resources. This expense can be managed through the use of **compiled** languages, which are optimized in another language prior to execution. Compiled languages often require more explicit **typing**, which allows minor bugs to be caught during execution.

Languages are hard to classify, but there are generalizations due to abstraction (Figure 5.1A). It is often easier to call lower-level languages from higher-level languages than the alternative. Higher-level languages are more likely to be interpreted, whereas lower-level

languages are more likely to be compiled, but it is worth noting that the two distinctions are unique. Python and Java, two languages that integrate both compiled and interpreted components into their functionality, are sometimes considered to have similar levels of abstraction as they hide similar implementation details. However, their use-cases diverge as a result of their typing and execution procedures. An important factor to consider is that some proprietary languages have open-source counterparts that fill a similar functionality niche while also providing the benefit of transparency (e.g. Julia as an open-source alternative to MATLAB). A universally superior language does not exist, but there is likely one designed to excel at any necessary task.

5.2.2 Records: Treat version control like a lab notebook

Version control software maintains the legacy of projects and decisions, much like the ubiquitous lab notebook. Managing a significant amount of source code can be difficult, but tools like Git have been developed as a solution to properly tracking changes. Git is not the only version control software, but it is the most common. Multiple tools have extended Git to help developers maintain and organize version control of both code and data. All interactions with version control are documented. If time is invested in keeping these systems organized, the result naturally resembles a well-maintained lab notebook.

Versioning code allows developers to maintain, and return to, the code base at significant points of development. Maintaining an organized history allows researchers to go back through code changes and find the reasoning behind specific changes. A version can be as-

signed to any commit—a collection of code changes—as an indicator for a useful checkpoint in development. Importantly, versions can be associated with a DOI and used to indicate the code that produced results in a publication.

Version control systems are designed with organization in mind, providing many technical structures to compartmentalize work. Repositories are abstracted file systems that support Git to track changes made to a collection of source code. Commit labels are comprised of a unique identifier, timestamp, and author. The repository is a great place to include additional user information: installation instructions, example usage, public API documentation. If commits are kept small, their annotations can be highly specific to the change made.

Many tools integrate with version control software, facilitating **continuous integration** and collaboration. Branches are collections of commits that can help developers organize work-in-progress code. Branches can be merged into the main project using a pull request, a common repository hosting feature for codifying significant changes and supporting collaborative code development and review. Pull requests are immensely useful not only as a commit history of adding a feature to the code base, but also as a mechanism for collaborating through **code reviews**. Online repository hosts include automated workflows (e.g. GitHub Actions, BitBucket Pipelines) that use continuous integration tools when new commits are added. These workflows can include automated linters—programs that check style—or test runners—programs that execute tests—that provide useful feedback.

5.2.3 Selection: Write tests to favor better code

Similar to how evolutionary pressures select for fitter phenotypes, intentionally developed tests can select for better code by driving code to follow best practices that improve modularity and avoid coupling while ensuring correctness. Early test design helps researchers write code proactively and have more utility than tests that are used reactively for debugging. An added benefit of writing tests is to break down wide-reaching design goals into a series of smaller, easily solvable test cases.

Tests provide a problem-solving framework to write and improve code. Simple tests, with streamlined inputs and outputs, run faster and are less disruptive to a workflow than complicated tests. Writing a trivial initial test—such as checking for an empty object when no input is given—is a useful foundation to continue adding to a suite of simple tests as projects grow and evolve. Tests provide additional rigorous documentation (simpler tests are clearer documentation) for intended function behavior and interactions for anticipated edge cases.

Tests should focus on assessing interfaces (inputs and outputs) rather than assessing implementation (underlying structure). Testing what code should do, rather than how it works, enables the developer to avoid rewriting tests even if the code implementation changes. **Unit tests** (or developer tests) generally evaluate one function or class of code. These tests should be explicit about the testing criteria so that it is clear to other developers what the point of a method, class, or function is. If a unit test fails, the code does not complete its intended purpose. **Functional tests** (or integration tests) evaluate a user's experience, and

should test multiple modules together. These tests are useful indicators for the anticipated use cases or user stories that were considered in the original design. If an integration or functional test fails, it could indicate that the modules may not correctly interact with one another or with an external service.

Code that cannot be tested is "bad code". It is more challenging to design tests that evaluate multi-purpose functions containing 100 lines of code than it is to design tests for focused functions with 10 lines of code. If it is difficult to conceive of a test that evaluates a function in isolation, a likely cause is that the code is highly coupled. The solution is usually to organize code with focused responsibilities, and therefore selecting for modular design. When refactoring, passing tests assures the developer that changes made toward more intuitive or readable implementations are still functionally correct.

5.2.4 Robustness: Design systems to work in diverse environments

Environments can be hostile to non-native species that have not evolved the requisite fitness. Similarly, code should be designed to navigate challenges of adapting to new environments. Reproducibility suffers when code only works on specific systems; fortunately, there are many emerging tools to support portability. In a world where it is increasingly common for people to use different operating systems and computational architectures, designing code that can be used by diverse audiences is critical. Instructions should support users trying to install code, and describe the necessary steps to run installation programs, or **build tools**. A common approach to ensure code fitness is virtualization, which mimics an operating system

within a program through **containerization**.

Instructions on how to install software is a necessary, but not always sufficient, requirement for sharing computational projects. Installation instructions can be sufficient when associated with standardized package management systems like PyPI (Python Package Index, often associated with `pip`) for Python. To support other contributors, installation instructions should be supplemented with a list of software **dependencies**, hardware requirements, and respective version numbers. In general, the reliability of instructions are limited by the extent of the combinations of software and hardware configurations that developers can reliably test.

Build tools automate the compilation and installation process to make the final software more accessible. Automated installation software can ensure smaller variance on the installation experience. These tools allow for the implementation of continuous integration steps, such as running tests, managing dependencies, and generating documentation. Languages often have unique build tools, meaning that any user will need to install the specified build tool. This approach is often desirable for sharing code with other developers to extend code functionality.

Containerization software (e.g. Docker) ensures that the installation and execution of code is consistent across machines. Containerizing code creates portable computational projects that work on different machines. In order to containerize code, the developer sets up a virtual file system and operating system for running the code, independent of the user system. Building containers is simply an exercise in determining the necessary dependencies

and steps required for a clean install, and can often be used in conjunction with build tools. However, the container still accesses the host machine's hardware, and therefore inherits any limitations from the original code that may result. For projects requiring a high degree of parallelization, containers can ensure code runs the same way on clusters and cloud computing as on a local machine.

5.2.5 Redundancy: Documentation and readability are complimentary, not interchangeable

Just as functional and genetic redundancy enables consistent development in biological systems, redundancy in code documentation practices and code readability provide a similar benefit in computational development. Redundant documentation provides accessibility of code to a wider range of users, while readable code that can be understood and interpreted like prose supports method transparency. Ensuring that the code is readable includes making sure the code is as intuitive as possible to other developers, an idea closely related to **cohesion**. Providing traditional documentation and improving readability through code reviews are tangible methods of providing clearer code.

In practice, documentation refers to multiple scales of potentially redundant information that are designed to help different types of users engage with different levels of the project. There are many ways to signify intent as a developer that go beyond pure syntax. Users rely on examples and instructions to understand and apply code; future developers, including the original author, rely on comments, readability, and tests. Comments alone are not sufficient documentation as they cannot be tested and are prone to mistakes. Code written to be as

readable as possible (a practice referred to as self-documenting code) is helpful, but should not be the sole form of documentation. Some of these considerations are detailed in Figure 5.1B, which highlights the organization and readability of consistently styled code.

Syntax alone can be misleading since the structure of code can lead to incorrect assumptions about implementation. For example, it is helpful to be explicit in the documentation when giving a function permission to change attributes of an object (pass by reference) rather than just passing a function its value (pass by value). Not doing so can lead to unintentional side-effects (manipulation of data). **Functional programming** is less prone to unintentional side-effects by definition, while using **object-oriented programming** provides a framework for intuitive data structures and methods for manipulating data. Leveraging appropriate design principles profoundly impacts how immediately understandable the code may be.

Code reviews and pair programming provide opportunities to assess the efficacy of documentation and readability. Reviewers do not need to be more fluent or knowledgeable than the author to be helpful; code review and pair programming are productive exercises that benefit both parties independent of their relative levels of expertise. Tools should be reviewed and tested by multiple users; these exercises are bolstered by, and help assess efficacy of, documentation. Collective ownership of code between collaborators and the subsequent knowledge sharing within a community offers the best safeguard against “bad code”.

5.3 Conclusion

Complexity is inherent to both biology and code. Functional code emerges from simple code interactions. No one actively sets out to write "bad code", but "good code" does not emerge passively. It derives from intentionally applying and maintaining coding practices that are appropriate for the context of the problem. We introduced biological analogies of coding practices to equip computational biologists with the perspective to address and proactively mitigate shortcomings in code quality. A collective effort to improve the quality of code in research will improve reproducibility and trust within and across scientific communities.

A: Common terms used to describe programming languages

Term	Description
Compiled	Languages that leverage an intermediate step of taking source code and rewriting, and sometimes optimizing, it into another programming language before execution.
Interpreted	Languages that directly execute code from the source code as written in a text editor, where the file is continuously accessed during runtime. Parts of interpreted languages can also be compiled.
Typing	How languages handle the underlying data structures of variables. Typing usually amounts to when, or if, type verification happens and how flexible that verification is.
Object-oriented design	The organization of code such that data are stored into objects that can interact and potentially change other objects. Object-oriented programming is typically regarded as the standard approach to well-designed modular code.
Functional design	The organization of code into functions with a focus on inputs and outputs, with no side effects. Functional programming is popular in data analysis or workflow designs due to the design principle of not changing underlying data.

B: Common terms used to discuss coding practices

Term	Description	Related terms
Style guides	Guidelines of community conventions for a specific language. These are often useful to enforce consistent style decisions and help catch mistakes.	*Reformatters; Linters*
Unit tests	A classification of light-weight tests designed to assess the execution of a single function or class. These tests are often written from the perspective of a developer to help write and refactor code. Ideally, these tests are isolated from other tests and are often designed to isolate the function/class being tested.	*Developer tests*
Functional tests	Tests that are designed to evaluate a collection of code (suite or package) working together and are generally developed based on users/use cases. There is not a significant distinction between functional and integration tests and are often used as interchangeable terms. These tests are often run in continuous integration configurations as they often are less isolated and require more time.	*Integration tests*
Continuous integration	The practice of using automated tools to build and test new code changes. Continuous integration tools are often associated with deploying code, but can also be easily adapted to encourage better day-to-day practices.	*Continuous delivery*
Coupling	A consideration of how much code might need to change as a result of adding or removing code in another function or class. Coupling is often associated the scalability or flexibility of source code.	*Modularity; Interfaces*
Cohesion	A descriptor of the intuitive structure of code. A cohesive design focuses on the grouping (and separation) of code to be organized into distinct purposes.	*Design pattern*

C: Common terms used to describe workflows

Term	Description	Related terms
Dependency	A package or module that a code project relies on to work. Dependencies may need to be isolated from one another in order to work. Portability issues arise from the mismanagement of inaccessible or incompatible dependencies.	*Virtual environment*
Version control	Tools that help developers maintain collections of source code as separate versions by tracking changes and annotations. The most common version control software is Git.	*Source code control*
Refactoring	Rewriting functioning code to be more readable, scalable, and maintainable.	*Code smells*
Code review	Collaborative evaluations of code readability, correctness, and maintainability.	*Pair programming*
Build tools	A tool that automates the installation of software. This process involves downloading dependencies, compiling code, and packaging as an executable.	*Build automation*
Containerization	Packaging code alongside the required operating system and dependencies. Software is then able to run containers by virtualizing the operating system on any machine.	*Virtualization*

Table 5.1: **Common terminology in coding practices literature** | Collection of common terms in software development. These terms are grouped based on their prevalence in code development practices. The "related terms" in B and C are not necessarily interchangeable terms, but are closely associated in discussions of the original term.

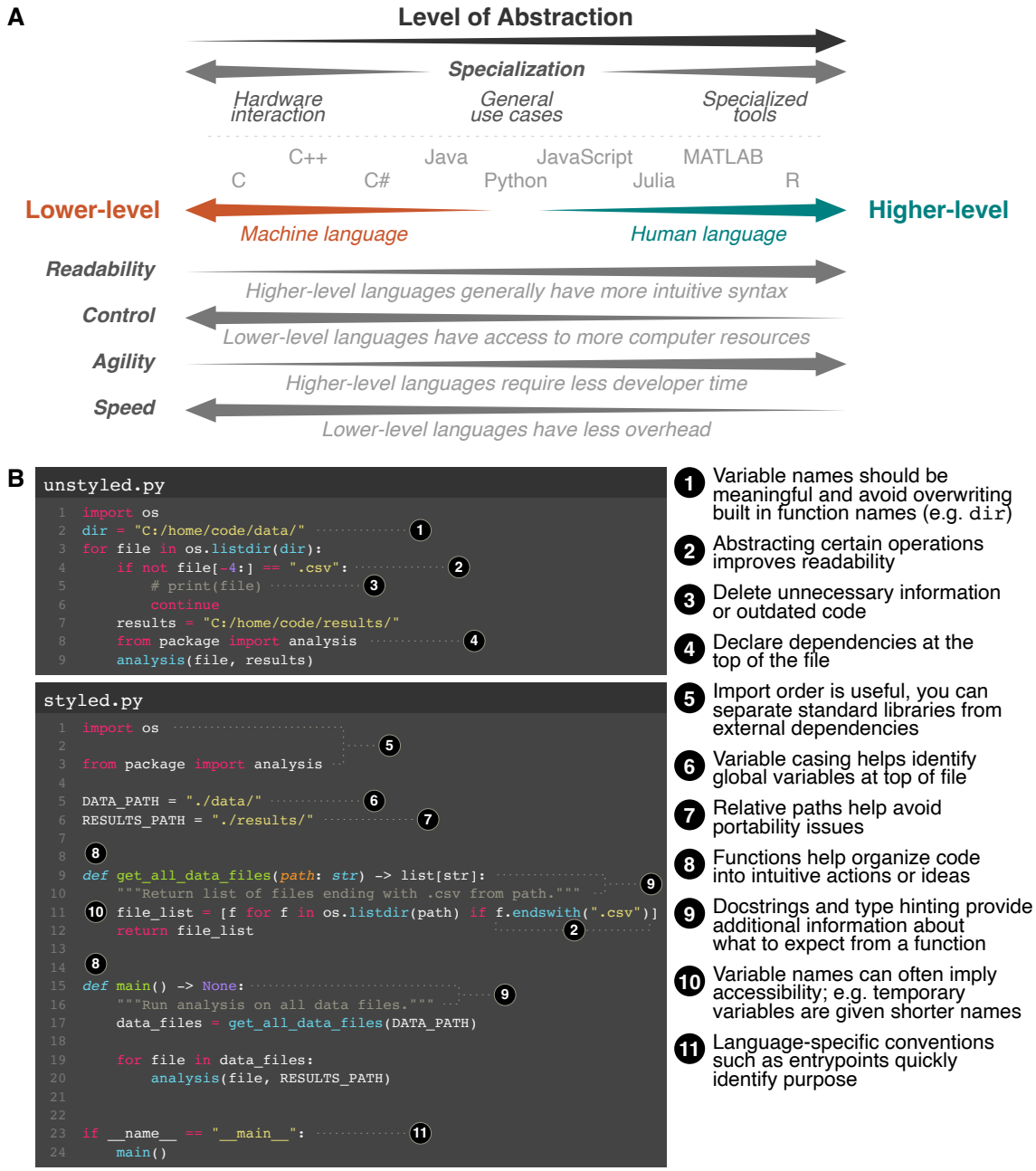

Figure 5.1: **Considering programming languages and style | (A) Patterns of abstraction -** Languages (e.g., C, C++, C#, etc.) are often designed with certain properties (e.g., readability, control, agility, and speed) consistent to a relative level of abstraction. The ordering of languages is approximate. The list of languages in this figure is not comprehensive; it represents a subset of the more popular languages in computational biology. **(B) Importance of style -** In addition to streamlining decisions about organization, style guides (e.g. PEP8 for python) impose meaning and structure on the organization of syntax. The two files perform the exact same task, yet `styled.py` provides far more clarity than the original, `unstyled.py`.

Chapter 6

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

6.1 Conclusion

Computational models, when wielded effectively, continue to be necessary to modern science in order to generate hypotheses, interrogate assumptions, and verify results. As our computational expertise evolves and models become more sophisticated, tools to interrogate these models and principles to integrate these models need to be established. Without said tools and principles, we are at risk of creating more problems than we are solving.

In **Chapter 1**, I discuss the nuances of model class decisions, their primary utilities, as well as the inherent complexity to their application and combination thereof. It is increasingly important to be deliberate with these decisions to extract the key research objectives from a computational model. In **Chapter 2**, I apply an emulation framework, integrating all three modeling paradigms discussed in **Chapter 1**, to interrogate the information axes available in high-resolution models. I believe the conclusion of finding temporal information to be significant in predicting, and therefore assocating, temporal evolution to spatio-temporal emergence. **Chapter 3** details work integrating a mechanistic model from an alternate context and highlighting a missing regulatory process in VEGF secretion. Careful translation of mechanistic models into agent-based models provides exciting opportunities to highlight

knowledge gaps, and therefore generate hypotheses of *in situ* dynamics. **Chapter 4** provides analogies from experimental biology to computational biology to bolster the reproducibility of computational models. Current computational models are increasing in complexity at a rapid pace, and without a solid foundation, a gap between model design and FAIR principles will emerge.

6.2 Future Directions

There are a number of directions that I would consider if I were to continue the work included in this dissertation. I have divided the proposed directions into two major themes: (1) *Data-driven models of agent-based models* and (2) *Agent-based modeling of angiogenesis*.

6.2.1 Data-driven emulation of agent-based models

As demonstrated in **Chapter 2**, there is much room for improvement in the development of emulation strategies of complex multi-scale models. The limited performance of those emulators support strategies that leverage more complex ML techniques to capture the emergent phenomena that we can simulate *in silico*. I propose a number of extensions of the work, specifically for including dynamics within the emulator, active learning approaches towards personalized medicine.

Dynamic emulation of environment

One of the key results from **Chapter 2** was the importance of temporal information, however the emulation models were not designed to predict the evolution of the environment.

While we demonstrated some utility in predicting the dynamics of the system, the small inconsistencies prevented robust performance. I propose two possible implementations to capture further environment information in the emulator: (1) Multi-objective learning could be used to predict characteristics of the environment and refine the optimization such that the parameters stayed within interpretable ranges. If the model parameters were more constrained by multi-objective optimization, the variance in the feature importance could be mitigated, thus providing opportunity for robust sensitivity analyses. (2) Graphical recurrent neural networks are an option to capture the environment in a higher resolution than we were able to with network metrics. The resulting models would be less interpretable than the method presented in this dissertation, but this could result in better predictive accuracy. (3) Another promising approach in recent literature is the integration of mechanistic or physical models into the cost function. As the emulation of ABMs is a fairly nascent field, further study from carrying out these exercises would also provide more insight into what types of tumors—considering that the models in **Chapter 2** were largely agnostic to cancer cell properties—or environments are easier to predict. Importantly, improving the predictive performance of these models and their dynamics would allow for the reduction of computational expense, supporting surrogate modeling.

Reinforcement learning towards intervention planning

Integrating reinforcement-learning models into ABMs is likely beneficial to reducing the computational burden of exhaustively running simulations. Reinforcement learning allows

researchers to narrow the decision space by optimizing for a specific goal. Narrowing the decision space—in this scenario—would help identify the most impactful interventions without sampling the entire possible parameter space. A number of challenges would need to be addressed: the representation of the *in silico* tumor in feature space is non trivial, and the implementation and validation of treatment modules would also need to be completed. There are likely patterns in treatment strategies could be more helpful to address different types of emergent properties of the tumor than others. Programming the ABM parameters to be more specific to observed tumor or environmental phenotypes could be a platform for studying the role of ABMs and personalized medicine.

6.2.2 Agent-based modeling of angiogenesis

In **Chapter 3** I focused on the integration of sensing modules into ARCADE in order to support the consideration of hypoxia as a key component of the TME. In doing so, by extending that work, we could include further discovery of *in situ* HIF and VEGF dynamics. These two biomolecules are widely implicated in many studies of hypoxia, angiogenesis, metastasis, and the associated pathways have been proposed as targets for potential therapies. I highlight two possible extensions—modeling angiogenesis, and its resulting effects on treatment strategies—that would greatly expand the utility of ARCADE while also supporting new research aims in the Bagheri lab.

Cellular sensing and dynamic angiogenesis

Another key aspect of hypoxia and the dissemination of VEGF from cellular sensing pathways is the co-opting of physiological development processes. Angiogenesis is not only interesting but is also a ubiquitous feature of solid tumors. Tumors can only grow 1–2mm without the external supply of nutrients derived from angiogenesis.[147] However, the application of anti-angiogenic therapies has been largely unsuccessful, indicating gaps in our understanding of the overall regulation of this process. Adding dynamic angiogenesis to ARCADE of tumor development would significantly help understand the implications of the multi-scale process. Modeling the dynamic interplay between cells and developing vasculature would allow us to probe the consequences of disregulation in this process. This objective would also provide insight into modeling decisions, where alternative representations of the environment could be used to model this dynamic process (e.g. cellular potts vs. network model)

Treatment in complex control systems

Development of multi-scale models allow us to study the control of complex systems. Unifying disparate models of different classes provide an opportunity to investigate decentralized control schema. Through the integration of the HIF signaling network detailed in **Chapter 3** with the intention of implementing angiogenesis, this would be the only agent-based model, to my knowledge, with a mechanistic model underlying the growth factor role in angiogenesis. This mechanistic model provides opportunities to use sensitivity analyses to evaluate therapy targets in this complex environment. An alternative direction is the addition of

alternative cell-based therapies to sample the design space in a less expensive manner. One such cell-based therapy is the opportunity to model microbe-based therapies driven by recent advances in synthetic biology. Applying principals from decentralized control would provide narrower design guidelines to build effective therapies in this novel engineering field. Layering these types of therapies with dynamic angiogenesis and previously published research on immune cell-based therapies[135] could help realize many of the most ambitious synthetic biology solutions to cancer.

6.3 Concluding thoughts

Validation has long been the sole gold standard by which computational models have been assesssed. Validated models can be incomplete, and unvalidated models can still be useful. While necessary, the widespread adoption of computational approaches in recent decades requires more nuanced communication and assessments when it comes to evaluating a model. I believe that it is becoming more and more important to reproducibility and extensibiliy to explicitly communicate assumptions and abstractions that are embedded into a model. These assumptions are made more clear via bettter code and better communication. While many journals and societies are requiring the inclusion of code, simple inclusion is not sufficient.

BIBLIOGRAPHY

[1] Douglas Hanahan and Robert A. Weinberg. Hallmarks of Cancer: The Next Generation. *Cell*, 144(5):646–674, March 2011. ISSN 0092-8674. doi: 10.1016/j.cell.2011.02.013. URL http://www.sciencedirect.com/science/article/pii/S0092867411001279.

[2] Bashar Emon, Jessica Bauer, Yasna Jain, Barbara Jung, and Taher Saif. Biophysics of Tumor Microenvironment and Cancer Metastasis - A Mini Review. *Computational and Structural Biotechnology Journal*, 16:279–287, January 2018. ISSN 2001-0370. doi: 10.1016/j.csbj.2018.07.003. URL http://www.sciencedirect.com/science/article/pii/S2001037018300291.

[3] Varvara Petrova, Margherita Annicchiarico-Petruzzelli, Gerry Melino, and Ivano Amelio. The hypoxic tumour microenvironment. *Oncogenesis*, 7(1):10, January 2018. ISSN 2157-9024. doi: 10.1038/s41389-017-0011-9. URL https://doi.org/10.1038/s41389-017-0011-9.

[4] Yasir Suhail, Margo P. Cain, Kiran Vanaja, Paul A. Kurywchak, Andre Levchenko, Raghu Kalluri, and Kshitiz. Systems Biology of Cancer Metastasis. *cels*, 9(2):109–127, August 2019. ISSN 2405-4712. doi: 10.1016/j.cels.2019.07.003. URL https://www.cell.com/cell-systems/abstract/S2405-4712(19)30234-0. Publisher: Elsevier.

[5] Mina J. Bissell and Derek Radisky. Putting tumours in context. *Nature Reviews Cancer*, 1(1):46–54, October 2001. ISSN 1474-1768. doi: 10.1038/35094059. URL https://doi.org/10.1038/35094059.

[6] Thomas S. Deisboeck, Zhihui Wang, Paul Macklin, and Vittorio Cristini. Multiscale Cancer Modeling. *Annu Rev Biomed Eng*, 13, August 2011. ISSN 1523-9829. doi: 10.1146/annurev-bioeng-071910-124729. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3883359/.

[7] Géraldine Gentric, Virginie Mieulet, and Fatima Mechta-Grigoriou. Heterogeneity in Cancer Metabolism: New Concepts in an Old Field. *Antioxid Redox Signal*, 26(9):462–485, March 2017. ISSN 1523-0864. doi: 10.1089/ars.2016.6750. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5359687/.

[8] Otto Warburg, Franz Wind, and Erwin Negelein. THE METABOLISM OF TUMORS IN THE BODY. *J Gen Physiol*, 8(6):519–530, March 1927. ISSN 0022-1295. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2140820/.

[9] Rupert Courtnay, Darleen C. Ngo, Neha Malik, Katherine Ververis, Stephanie M. Tortorella, and Tom C. Karagiannis. Cancer metabolism and the Warburg effect: the role of HIF-1 and PI3K. *Mol. Biol. Rep.*, 42(4):841–851, April 2015. ISSN 1573-4978. doi: 10.1007/s11033-015-3858-x.

[10] Wafaa Al Tameemi, Tina P. Dale, Rakad M. Kh Al-Jumaily, and Nicholas R. Forsyth. Hypoxia-Modified Cancer Cell Metabolism. *Front. Cell Dev. Biol.*, 7, 2019. ISSN 2296-634X. doi: 10.3389/fcell.2019.00004. URL https://www.frontiersin.org/articles/10.3389/fcell.2019.00004/full. Publisher: Frontiers.

[11] Kritika Saxena and Mohit Kumar Jolly. Acute vs. Chronic vs. Cyclic Hypoxia: Their Differential Dynamics, Molecular Mechanisms, and Effects on Tumor Progression. *Biomolecules*, 9(8), August 2019. ISSN 2218-273X. doi: 10.3390/biom9080339. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6722594/.

[12] Jake C Forster, Wendy M Harriss-Phillips, Michael JJ Douglass, and Eva Bezak. A review of the development of tumor vasculature and its effects on the tumor microenvironment. *Hypoxia (Auckl)*, 5:21–32, April 2017. ISSN 2324-1128. doi: 10.2147/HP.S133231. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5395278/.

[13] Avihai Ron, Xosé Luís Deán-Ben, Sven Gottschalk, and Daniel Razansky. Volumetric Optoacoustic Imaging Unveils High-Resolution Patterns of Acute and Cyclic Hypoxia in a Murine Model of Breast Cancer. *Cancer Res*, 79(18):4767–4775, September 2019. ISSN 0008-5472, 1538-7445. doi: 10.1158/0008-5472.CAN-18-3769. URL https://cancerres.aacrjournals.org/content/79/18/4767. Publisher: American Association for Cancer Research Section: Convergence and Technologies.

[14] Luana Schito and Gregg L. Semenza. Hypoxia-Inducible Factors: Master Regulators of Cancer Progression. *Trends in Cancer*, 2(12):758–770, December 2016. ISSN 2405-8033. doi: 10.1016/j.trecan.2016.10.016. URL http://www.sciencedirect.com/science/article/pii/S2405803316301595.

[15] Minsi Zhang, Qiong Qiu, Zhizhong Li, Mohit Sachdeva, Hooney Min, Diana M. Cardona, Thomas F. DeLaney, Tracy Han, Yan Ma, Lixia Luo, Olga R. Ilkayeva, Ki Lui, Amanda G. Nichols, Christopher B. Newgard, Michael B. Kastan, Jeffrey C. Rathmell, Mark W. Dewhirst, and David G. Kirsch. HIF-1α regulates the response of primary sarcomas to radiation therapy through a cell autonomous mechanism. *Radiat Res*, 183(6):594–609, June 2015. ISSN 0033-7587. doi: 10.1667/RR14016.1. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4800000/.

[16] Tomaz Makovec. Cisplatin and beyond: molecular mechanisms of action and drug resistance development in cancer chemotherapy. *Radiology and Oncology*, 53(2):148–158, March 2019. doi: 10.2478/raon-2019-0018. URL https://content.

sciendo.com/view/journals/raon/53/2/article-p148.xml. Publisher: Sciendo Section: Radiology and Oncology.

[17] Nadine Rohwer and Thorsten Cramer. Hypoxia-mediated drug resistance: novel insights on the functional interaction of HIFs and cell death pathways. *Drug Resist. Updat.*, 14(3):191–201, June 2011. ISSN 1532-2084. doi: 10.1016/j.drup.2011.03.001.

[18] Peipei Xu, Miao Wang, Ying Jiang, Jian Ouyang, and Bing Chen. The association between expression of hypoxia inducible factor-1α and multi-drug resistance of acute myeloid leukemia. *Translational Cancer Research*, 6(1):198–205–205, February 2017. ISSN 2219-6803. doi: 10.21037/11945. URL http://tcr.amegroups.com/article/view/11945. Number: 1.

[19] Carine Michiels, Céline Tellier, and Olivier Feron. Cycling hypoxia: A key feature of the tumor microenvironment. *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 1866(1):76–86, August 2016. ISSN 0304-419X. doi: 10.1016/j.bbcan.2016.06.004. URL http://www.sciencedirect.com/science/article/pii/S0304419X16300440.

[20] Johannes Schödel, Steffen Grampp, Eamonn R. Maher, Holger Moch, Peter J. Ratcliffe, Paul Russo, and David R. Mole. Hypoxia, Hypoxia-inducible Transcription Factors, and Renal Cancer. *European Urology*, 69(4):646–657, April 2016. ISSN 0302-2838. doi: 10.1016/j.eururo.2015.08.007. URL http://www.sciencedirect.com/science/article/pii/S0302283815007502.

[21] Veronica A. Carroll and Margaret Ashcroft. Role of Hypoxia-Inducible Factor (HIF)-1α versus HIF-2α in the Regulation of HIF Target Genes in Response to Hypoxia, Insulin-Like Growth Factor-I, or Loss of von Hippel-Lindau Function: Implications for Targeting the HIF Pathway. *Cancer Res*, 66(12):6264–6270, June 2006. ISSN 0008-5472, 1538-7445. doi: 10.1158/0008-5472.CAN-05-2519. URL https://cancerres.aacrjournals.org/content/66/12/6264. Publisher: American Association for Cancer Research Section: Cell, Tumor, and Stem Cell Biology.

[22] Keqiang Zhang, Ernest S. Han, Thanh H. Dellinger, Jianming Lu, Sangkil Nam, Richard A. Anderson, John H. Yim, and Wei Wen. Cinnamon extract reduces VEGF expression via suppressing HIF-1α gene expression and inhibits tumor growth in mice. *Molecular Carcinogenesis*, 56(2):436–446, 2017. ISSN 1098-2744. doi: 10.1002/mc.22506. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mc.22506. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mc.22506.

[23] Barbara Muz, Pilar de la Puente, Feda Azab, and Abdel Kareem Azab. The role of hypoxia in cancer progression, angiogenesis, metastasis, and resistance to therapy.

Hypoxia (Auckl), 3:83–92, December 2015. ISSN 2324-1128. doi: 10.2147/HP.S93413. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5045092/.

[24] Roland H Wenger, Vartan Kurtcuoglu, Carsten C Scholz, Hugo H Marti, and David Hoogewijs. Frequently asked questions in hypoxia research. *Hypoxia (Auckl)*, 3:35–43, September 2015. ISSN 2324-1128. doi: 10.2147/HP.S92198. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5045069/.

[25] Nicholas A. Cilfone, Cory R. Perry, Denise E. Kirschner, and Jennifer J. Linderman. Multi-Scale Modeling Predicts a Balance of Tumor Necrosis Factor-α and Interleukin-10 Controls the Granuloma Environment during Mycobacterium tuberculosis Infection. *PLoS One*, 8(7), July 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0068680. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3711807/.

[26] Jessica S Yu and Neda Bagheri. Multi-class and multi-scale models of complex biological phenomena. *Current Opinion in Biotechnology*, 39:167–173, June 2016. ISSN 0958-1669. doi: 10.1016/j.copbio.2016.04.002. URL http://www.sciencedirect.com/science/article/pii/S0958166916301100.

[27] Wenying Shou, Carl T Bergstrom, Arup K Chakraborty, and Frances K Skinner. Theory, models and biology. *eLife*, 4:e07158, July 2015. ISSN 2050-084X. doi: 10.7554/eLife.07158. URL https://doi.org/10.7554/eLife.07158. Publisher: eLife Sciences Publications, Ltd.

[28] Lucas B. Edelman, James A. Eddy, and Nathan D. Price. In silico models of cancer. *Wiley Interdiscip Rev Syst Biol Med*, 2(4):438–459, 2010. ISSN 1939-5094. doi: 10.1002/wsbm.75. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3157287/.

[29] Daniel G. Brown, Rick Riolo, Derek T. Robinson, Michael North, and William Rand. Spatial process and data models: Toward integration of agent-based models and GIS. *J Geograph Syst*, 7(1):25–47, May 2005. ISSN 1435-5949. doi: 10.1007/s10109-005-0148-5. URL https://doi.org/10.1007/s10109-005-0148-5.

[30] Zhihui Wang, Joseph D. Butner, Romica Kerketta, Vittorio Cristini, and Thomas S. Deisboeck. Simulating cancer growth with multiscale agent-based modeling. *Seminars in Cancer Biology*, 30:70–78, February 2015. ISSN 1044-579X. doi: 10.1016/j.semcancer.2014.04.001. URL http://www.sciencedirect.com/science/article/pii/S1044579X14000492.

[31] Zhihui Wang, Le Zhang, Jonathan Sagotsky, and Thomas S Deisboeck. Simulating non-small cell lung cancer with a multiscale agent-based model. *Theor Biol Med Model*, 4:50, December 2007. ISSN 1742-4682. doi: 10.1186/1742-4682-4-50. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2259313/.

[32] Xiaoqiang Sun, Le Zhang, Hua Tan, Jiguang Bao, Costas Strouthos, and Xiaobo Zhou. Multi-scale agent-based brain cancer modeling and prediction of TKI treatment response: Incorporating EGFR signaling pathway and angiogenesis. *BMC Bioinformatics*, 13:218, August 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-218. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3487967/.

[33] Kieran Smallbone, Robert A. Gatenby, Robert J. Gillies, Philip K. Maini, and David J. Gavaghan. Metabolic changes during carcinogenesis: Potential impact on invasiveness. *Journal of Theoretical Biology*, 244(4):703–713, February 2007. ISSN 0022-5193. doi: 10.1016/j.jtbi.2006.09.010. URL http://www.sciencedirect.com/science/article/pii/S0022519306004115.

[34] Joachim von Eichborn, Anna Lena Woelke, Filippo Castiglione, and Robert Preissner. VaccImm: simulating peptide vaccination in cancer therapy. *BMC Bioinformatics*, 14:127, April 2013. ISSN 1471-2105. doi: 10.1186/1471-2105-14-127. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3651379/.

[35] Pier-Luigi Lollini, Santo Motta, and Francesco Pappalardo. Discovery of cancer vaccination protocols with a genetic algorithm driving an agent based simulator. *BMC Bioinformatics*, 7(1):352, July 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-352. URL https://doi.org/10.1186/1471-2105-7-352.

[36] Michelle L. Wynn, Paul M. Kulesa, and Santiago Schnell. Computational modelling of cell chain migration reveals mechanisms that sustain follow-the-leader behaviour. *J R Soc Interface*, 9(72):1576–1588, July 2012. ISSN 1742-5689. doi: 10.1098/rsif.2011.0726. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3367809/.

[37] Gary R. Mirams, Christopher J. Arthurs, Miguel O. Bernabeu, Rafel Bordas, Jonathan Cooper, Alberto Corrias, Yohan Davit, Sara-Jane Dunn, Alexander G. Fletcher, Daniel G. Harvey, Megan E. Marsh, James M. Osborne, Pras Pathmanathan, Joe Pitt-Francis, James Southern, Nejib Zemzemi, and David J. Gavaghan. Chaste: An Open Source C++ Library for Computational Physiology and Biology. *PLOS Computational Biology*, 9(3):e1002970, March 2013. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002970. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002970. Publisher: Public Library of Science.

[38] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tatara, Charles M. Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1):3, March 2013. ISSN 2194-3206. doi: 10.1186/2194-3206-1-3. URL https://doi.org/10.1186/2194-3206-1-3.

[39] Maciej H. Swat, Gilberto L. Thomas, Julio M. Belmonte, Abbas Shirinifard, Dimitrij Hmeljak, and James A. Glazier. Chapter 13 - Multi-Scale Modeling of Tissues Using CompuCell3D. In Anand R. Asthagiri and Adam P. Arkin, editors, *Methods in Cell Biology*, volume 110, pages 325–366. Academic Press, January 2012. ISBN 0091-679X. doi: 10.1016/B978-0-12-388403-9.00013-8. URL http://www.sciencedirect.com/science/article/pii/B9780123884039000138.

[40] Alexis N Prybutok, Jason Y Cain, Joshua N Leonard, and Neda Bagheri. Fighting fire with fire: Deploying complexity in computational modeling to effectively characterize complex biological systems. *Current Opinion in Biotechnology*, 75:102704, June 2022. ISSN 0958-1669. doi: 10.1016/j.copbio.2022.102704.

[41] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, November 2002. ISSN 1476-4687. doi: 10.1038/nature01254. URL https://www.nature.com/articles/nature01254. Number: 6912 Publisher: Nature Publishing Group.

[42] Vipin Narang, James Decraene, Shek-Yoon Wong, Bindu S. Aiswarya, Andrew R. Wasem, Shiang Rong Leong, and Alexandre Gouaillard. Systems immunology: a survey of modeling formalisms, applications and simulation tools. *Immunol Res*, 53(1):251–265, September 2012. ISSN 1559-0755. doi: 10.1007/s12026-012-8305-7. URL https://doi.org/10.1007/s12026-012-8305-7.

[43] Adriana Tomic, Ivan Tomic, Yael Rosenberg-Hasson, Cornelia L. Dekker, Holden T. Maecker, and Mark M. Davis. SIMON, an Automated Machine Learning System, Reveals Immune Signatures of Influenza Vaccine Responses. *J Immunol*, 203(3):749–759, August 2019. ISSN 0022-1767. doi: 10.4049/jimmunol.1900033. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6643048/.

[44] Leonard A. Harris, Justin S. Hogg, José-Juan Tapia, John A. P. Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P. Sheehan, and James R. Faeder. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*, 32(21):3366–3368, November 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw469. URL https://doi.org/10.1093/bioinformatics/btw469.

[45] Gaelle Letort, Arnau Montagud, Gautier Stoll, Randy Heiland, Emmanuel Barillot, Paul Macklin, Andrei Zinovyev, and Laurence Calzone. PhysiBoSS: a multi-scale agent-based modelling framework integrating physical dimension and cell signalling. *Bioinformatics*, 35(7):1188–1196, April 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty766. URL https://doi.org/10.1093/bioinformatics/bty766.

[46] Jessica S. Yu and Neda Bagheri. Agent-based models predict emergent behavior of heterogeneous cell populations in dynamic microenvironments. *Front. Bioeng. Biotechnol.*, 8, 2020. ISSN 2296-4185. doi: 10.3389/fbioe.2020.

00249. URL https://www.frontiersin.org/articles/10.3389/fbioe.2020.00249/abstract. Publisher: Frontiers.

[47] Mengyuan Zhao, Wenying He, Jijun Tang, Quan Zou, and Fei Guo. A comprehensive overview and critical evaluation of gene regulatory network inference technologies. *Briefings in Bioinformatics*, 22(5):bbab009, September 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab009. URL https://doi.org/10.1093/bib/bbab009.

[48] Joe G. Greener, Shaun M. Kandathil, Lewis Moffat, and David T. Jones. A guide to machine learning for biologists. *Nat Rev Mol Cell Biol*, 23(1):40–55, January 2022. ISSN 1471-0080. doi: 10.1038/s41580-021-00407-0. URL https://www.nature.com/articles/s41580-021-00407-0. Number: 1 Publisher: Nature Publishing Group.

[49] John S. Tsang, Pamela L. Schwartzberg, Yuri Kotliarov, Angelique Biancotto, Zhi Xie, Ronald N. Germain, Ena Wang, Matthew J. Olnes, Manikandan Narayanan, Hana Golding, Susan Moir, Howard B. Dickler, Shira Perl, Foo Cheung, Baylor HIPC Center, and CHI Consortium. Global analyses of human immune variation reveal baseline predictors of postvaccination responses. *Cell*, 157(2):499–513, April 2014. ISSN 1097-4172. doi: 10.1016/j.cell.2014.03.031.

[50] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Ake Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nat Biomed Eng*, 4(8):827–834, August 2020. ISSN 2157-846X. doi: 10.1038/s41551-020-0578-x. URL https://www.nature.com/articles/s41551-020-0578-x. Number: 8 Publisher: Nature Publishing Group.

[51] Zicheng Hu, Alice Tang, Jaiveer Singh, Sanchita Bhattacharya, and Atul J. Butte. A robust and interpretable end-to-end deep learning model for cytometry data. *Proceedings of the National Academy of Sciences*, 117(35):21373–21380, September 2020. doi: 10.1073/pnas.2003026117. URL https://www.pnas.org/doi/10.1073/pnas.2003026117. Publisher: Proceedings of the National Academy of Sciences.

[52] Amlan Talukder, Clayton Barham, Xiaoman Li, and Haiyan Hu. Interpretation of deep learning in genomics and epigenomics. *Briefings in Bioinformatics*, 22(3):bbaa177, May 2021. ISSN 1477-4054. doi: 10.1093/bib/bbaa177. URL https://doi.org/10.1093/bib/bbaa177.

[53] Dongshunyi Li, Jun Ding, and Ziv Bar-Joseph. Identifying signaling genes in spatial single-cell expression data. *Bioinformatics*, 37(7):968–975, April 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa769. URL https://doi.org/10.1093/bioinformatics/btaa769.

[54] Sierra M Barone, Alberta GA Paul, Lyndsey M Muehling, Joanne A Lannigan, William W Kwok, Ronald B Turner, Judith A Woodfolk, and Jonathan M Irish. Unsupervised machine learning reveals key immune cell subsets in COVID-19, rhinovirus infection, and cancer therapy. *eLife*, 10:e64653, August 2021. ISSN 2050-084X. doi: 10.7554/eLife.64653. URL https://doi.org/10.7554/eLife.64653. Publisher: eLife Sciences Publications, Ltd.

[55] Kathryn E. Yost, Ansuman T. Satpathy, Daniel K. Wells, Yanyan Qi, Chunlin Wang, Robin Kageyama, Katherine L. McNamara, Jeffrey M. Granja, Kavita Y. Sarin, Ryanne A. Brown, Rohit K. Gupta, Christina Curtis, Samantha L. Bucktrout, Mark M. Davis, Anne Lynn S. Chang, and Howard Y. Chang. Clonal replacement of tumor-specific T cells following PD-1 blockade. *Nat Med*, 25(8):1251–1259, August 2019. ISSN 1546-170X. doi: 10.1038/s41591-019-0522-3. URL https://www.nature.com/articles/s41591-019-0522-3. Number: 8 Publisher: Nature Publishing Group.

[56] Vanessa Jurtz, Sinu Paul, Massimo Andreatta, Paolo Marcatili, Bjoern Peters, and Morten Nielsen. NetMHCpan-4.0: Improved Peptide-MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *J Immunol*, 199 (9):3360–3368, November 2017. ISSN 1550-6606. doi: 10.4049/jimmunol.1700893.

[57] Daniel K. Wells, Marit M. van Buuren, Kristen K. Dang, Vanessa M. Hubbard-Lucey, Kathleen C. F. Sheehan, Katie M. Campbell, Andrew Lamb, Jeffrey P. Ward, John Sidney, Ana B. Blazquez, Andrew J. Rech, Jesse M. Zaretsky, Begonya Comin-Anduix, Alphonsus H. C. Ng, William Chour, Thomas V. Yu, Hira Rizvi, Jia M. Chen, Patrice Manning, Gabriela M. Steiner, Xengie C. Doan, Tumor Neoantigen Selection Alliance, Taha Merghoub, Justin Guinney, Adam Kolom, Cheryl Selinsky, Antoni Ribas, Matthew D. Hellmann, Nir Hacohen, Alessandro Sette, James R. Heath, Nina Bhardwaj, Fred Ramsdell, Robert D. Schreiber, Ton N. Schumacher, Pia Kvistborg, and Nadine A. Defranoux. Key Parameters of Tumor Epitope Immunogenicity Revealed Through a Consortium Approach Improve Neoantigen Prediction. *Cell*, 183(3):818–834.e13, October 2020. ISSN 1097-4172. doi: 10.1016/j.cell.2020.09.015.

[58] Franz-Georg Wieland, Adrian L. Hauber, Marcus Rosenblatt, Christian Tönsing, and Jens Timmer. On structural and practical identifiability. *Current Opinion in Systems Biology*, 25:60–69, March 2021. ISSN 2452-3100. doi: 10.1016/j.coisb.2021.03.005. URL https://www.sciencedirect.com/science/article/pii/S245231002100007X.

[59] Joseph Walpole, Jason A. Papin, and Shayn M. Peirce. Multiscale computational models of complex biological systems. *Annu Rev Biomed Eng*, 15:137–154, 2013. ISSN 1545-4274. doi: 10.1146/annurev-bioeng-071811-150104.

[60] Joseph J. Muldoon, Yishan Chuang, Neda Bagheri, and Joshua N. Leonard. Macrophages employ quorum licensing to regulate collective activation. *Nat Commun*, 11(1):878, February 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-14547-y. URL https://www.nature.com/articles/s41467-020-14547-y. Number: 1 Publisher: Nature Publishing Group.

[61] Anna Coulibaly, Anja Bettendorf, Ekaterina Kostina, Ana Sofia Figueiredo, Sonia Y. Velásquez, Hans-Georg Bock, Manfred Thiel, Holger A. Lindner, and Maria Vittoria Barbarossa. Interleukin-15 Signaling in HIF-1α Regulation in Natural Killer Cells, Insights Through Mathematical Models. *Front Immunol*, 10:2401, October 2019. ISSN 1664-3224. doi: 10.3389/fimmu.2019.02401. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6805776/.

[62] Gustavo Barbosa Libotte, Fran Sérgio Lobato, Gustavo Mendes Platt, and Antônio J. Silva Neto. Determination of an optimal control strategy for vaccine administration in COVID-19 pandemic treatment. *Computer Methods and Programs in Biomedicine*, 196:105664, November 2020. ISSN 0169-2607. doi: 10.1016/j.cmpb.2020.105664. URL https://www.sciencedirect.com/science/article/pii/S0169260720314978.

[63] Xiao Hou, Song Gao, Qin Li, Yuhao Kang, Nan Chen, Kaiping Chen, Jinmeng Rao, Jordan S. Ellenberg, and Jonathan A. Patz. Intracounty modeling of COVID-19 infection with human mobility: Assessing spatial heterogeneity with business traffic, age, and race. *Proceedings of the National Academy of Sciences*, 118(24):e2020524118, June 2021. doi: 10.1073/pnas.2020524118. URL https://www.pnas.org/doi/10.1073/pnas.2020524118. Publisher: Proceedings of the National Academy of Sciences.

[64] Katherine Owens and Ivana Bozic. Modeling CAR T-Cell Therapy with Patient Preconditioning. *Bull Math Biol*, 83(5):42, March 2021. ISSN 1522-9602. doi: 10.1007/s11538-021-00869-5.

[65] David D. Ho, Avidan U. Neumann, Alan S. Perelson, Wen Chen, John M. Leonard, and Martin Markowitz. Rapid turnover of plasma virions and CD4 lymphocytes in HIV-1 infection. *Nature*, 373(6510):123–126, January 1995. ISSN 1476-4687. doi: 10.1038/373123a0. URL https://www.nature.com/articles/373123a0. Number: 6510 Publisher: Nature Publishing Group.

[66] X. Wei, S. K. Ghosh, M. E. Taylor, V. A. Johnson, E. A. Emini, P. Deutsch, J. D. Lifson, S. Bonhoeffer, M. A. Nowak, and B. H. Hahn. Viral dynamics in human immunodeficiency virus type 1 infection. *Nature*, 373(6510):117–122, January 1995. ISSN 0028-0836. doi: 10.1038/373117a0.

[67] A. S. Perelson, A. U. Neumann, M. Markowitz, J. M. Leonard, and D. D. Ho. HIV-1 dynamics in vivo: virion clearance rate, infected cell life-span, and viral generation time. *Science*, 271(5255):1582–1586, March 1996. ISSN 0036-8075. doi: 10.1126/science.271.5255.1582.

[68] Alan S. Perelson. Modelling viral and immune system dynamics. *Nat Rev Immunol*, 2(1):28–36, January 2002. ISSN 1474-1741. doi: 10.1038/nri700. URL https://www.nature.com/articles/nri700. Number: 1 Publisher: Nature Publishing Group.

[69] Sahak Z. Makaryan and Stacey D. Finley. Enhancing network activation in natural killer cells: predictions from in silico modeling. *Integr Biol (Camb)*, 12(5):109–121, May 2020. ISSN 1757-9708. doi: 10.1093/intbio/zyaa008.

[70] Raman S. Ganti, Wan-Lin Lo, Darren B. McAffee, Jay T. Groves, Arthur Weiss, and Arup K. Chakraborty. How the T cell signaling network processes information to discriminate between self and agonist ligands. *Proceedings of the National Academy of Sciences*, 117(42):26020–26030, October 2020. doi: 10.1073/pnas.2008303117. URL https://www.pnas.org/doi/10.1073/pnas.2008303117. Publisher: Proceedings of the National Academy of Sciences.

[71] Jennifer A. Rohrs, Dongqing Zheng, Nicholas A. Graham, Pin Wang, and Stacey D. Finley. Computational Model of Chimeric Antigen Receptors Explains Site-Specific Phosphorylation Kinetics. *Biophys J*, 115(6):1116–1129, September 2018. ISSN 1542-0086. doi: 10.1016/j.bpj.2018.08.018.

[72] Jennifer A. Rohrs, Elizabeth L. Siegler, Pin Wang, and Stacey D. Finley. ERK Activation in CAR T Cells Is Amplified by CD28-Mediated Increase in CD3ζ Phosphorylation. *iScience*, 23(4):101023, April 2020. ISSN 2589-0042. doi: 10.1016/j.isci.2020.101023.

[73] Colin G. Cess and Stacey D. Finley. Data-driven analysis of a mechanistic model of CAR T cell signaling predicts effects of cell-to-cell heterogeneity. *Journal of Theoretical Biology*, 489:110125, March 2020. ISSN 0022-5193. doi: 10.1016/j.jtbi.2019.110125. URL https://www.sciencedirect.com/science/article/pii/S0022519319304941.

[74] Arvind K. Chavali, Erwin P. Gianchandani, Kenneth S. Tung, Michael B. Lawrence, Shayn M. Peirce, and Jason A. Papin. Characterizing emergent properties of immunological systems with multi-cellular rule-based computational modeling. *Trends Immunol*, 29(12):589–599, December 2008. ISSN 1471-4906. doi: 10.1016/j.it.2008.08.006.

[75] Minki Hwang, Marc Garbey, Scott A. Berceli, and Roger Tran-Son-Tay. Rule-Based Simulation of Multi-Cellular Biological Systems—A Review of Modeling Techniques.

Cell Mol Bioeng, 2(3):285–294, September 2009. ISSN 1865-5025. doi: 10.1007/s12195-009-0078-2. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3045734/.

[76] Alexander R.A. Anderson, Alissa M. Weaver, Peter T. Cummings, and Vito Quaranta. Tumor Morphology and Phenotypic Evolution Driven by Selective Pressure from the Microenvironment. *Cell*, 127(5):905–915, December 2006. ISSN 0092-8674. doi: 10.1016/j.cell.2006.09.042. URL http://www.sciencedirect.com/science/article/pii/S0092867406013481.

[77] Bartlomiej Waclaw, Ivana Bozic, Meredith E. Pittman, Ralph H. Hruban, Bert Vogelstein, and Martin A. Nowak. A spatial model predicts that dispersal and cell turnover limit intratumour heterogeneity. *Nature*, 525(7568):261–264, September 2015. ISSN 1476-4687. doi: 10.1038/nature14971. URL https://www.nature.com/articles/nature14971. Number: 7568 Publisher: Nature Publishing Group.

[78] Sahar Jafari Nivlouei, M. Soltani, João Carvalho, Rui Travasso, Mohammad Reza Salimpour, and Ebrahim Shirani. Multiscale modeling of tumor growth and angiogenesis: Evaluation of tumor-targeted therapy. *PLOS Computational Biology*, 17(6):e1009081, June 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009081. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009081. Publisher: Public Library of Science.

[79] Jessica S. Yu and Neda Bagheri. Modular microenvironment components reproduce vascular dynamics de novo in a multi-scale agent-based model. *Cell Systems*, 12(8):795–809.e9, August 2021. ISSN 2405-4712. doi: 10.1016/j.cels.2021.05.007. URL https://www.sciencedirect.com/science/article/pii/S2405471221001939.

[80] Chang Gong, Oleg Milberg, Bing Wang, Paolo Vicini, Rajesh Narwal, Lorin Roskos, and Aleksander S. Popel. A computational multiscale agent-based model for simulating spatio-temporal tumour immune response to PD1 and PDL1 inhibition. *J R Soc Interface*, 14(134):20170320, September 2017. ISSN 1742-5662. doi: 10.1098/rsif.2017.0320.

[81] Emma Leschiera, Tommaso Lorenzi, Shensi Shen, Luis Almeida, and Chloe Audebert. A mathematical model to study the impact of intra-tumour heterogeneity on anti-tumour CD8+ T cell immune response. *Journal of Theoretical Biology*, 538:111028, April 2022. ISSN 0022-5193. doi: 10.1016/j.jtbi.2022.111028. URL https://www.sciencedirect.com/science/article/pii/S0022519322000261.

[82] Michael L. Blinov, John H. Gennari, Jonathan R. Karr, Ion I. Moraru, David P. Nickerson, and Herbert M. Sauro. Practical resources for enhancing the reproducibility of mechanistic modeling in systems biology. *Current Opinion in Systems*

Biology, 27:100350, September 2021. ISSN 2452-3100. doi: 10.1016/j.coisb.2021.06.001. URL https://www.sciencedirect.com/science/article/pii/S2452310021000445.

[83] B. G Fitzpatrick. Issues in Reproducible Simulation Research. *Bull Math Biol*, 81(1): 1–6, January 2019. ISSN 0092-8240. doi: 10.1007/s11538-018-0496-1. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6320709/.

[84] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*, 3(1):160018, March 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.18. URL https://www.nature.com/articles/sdata201618. Number: 1 Publisher: Nature Publishing Group.

[85] Andrea Bizzego, Nicole Bussola, Marco Chierici, Valerio Maggio, Margherita Francescatto, Luca Cima, Marco Cristoforetti, Giuseppe Jurman, and Cesare Furlanello. Evaluating reproducibility of AI algorithms in digital pathology with DAPPER. *PLOS Computational Biology*, 15(3):e1006269, March 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006269. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006269. Publisher: Public Library of Science.

[86] Shuming Zhang, Chang Gong, Alvaro Ruiz-Martinez, Hanwen Wang, Emily Davis-Marcisak, Atul Deshpande, Aleksander S. Popel, and Elana J. Fertig. Integrating single cell sequencing with a spatial quantitative systems pharmacology model spQSP for personalized prediction of triple-negative breast cancer immunotherapy response. *ImmunoInformatics*, 1-2:100002, October 2021. ISSN 2667-1190. doi: 10.1016/j.immuno.2021.100002. URL https://www.sciencedirect.com/science/article/pii/S2667119021000021.

[87] Colin G. Cess and Stacey D. Finley. Multi-scale modeling of macrophage—T cell interactions within the tumor microenvironment. *PLOS Computational Biology*, 16(12):e1008519, December 2020. ISSN 1553-7358. doi: 10.1371/journal.pcbi.

1008519. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008519. Publisher: Public Library of Science.

[88] Brenden K. Petersen, Jiachen Yang, Will S. Grathwohl, Chase Cockrell, Claudio Santiago, Gary An, and Daniel M. Faissol. Deep Reinforcement Learning and Simulation as a Path Toward Precision Medicine. *J Comput Biol*, 26(6):597–604, June 2019. ISSN 1557-8666. doi: 10.1089/cmb.2018.0168.

[89] Jonathan Ozik, Nicholson Collier, Justin M. Wozniak, Charles Macal, Chase Cockrell, Samuel H. Friedman, Ahmadreza Ghaffarizadeh, Randy Heiland, Gary An, and Paul Macklin. High-throughput cancer hypothesis testing with an integrated PhysiCell-EMEWS workflow. *BMC Bioinformatics*, 19(18):483, December 2018. ISSN 1471-2105. doi: 10.1186/s12859-018-2510-x. URL https://doi.org/10.1186/s12859-018-2510-x.

[90] Jonathan Ozik, Nicholson Collier, Randy Heiland, Gary An, and Paul Macklin. Learning-accelerated discovery of immune-tumour interactions. *Mol. Syst. Des. Eng.*, 4 (4):747–760, August 2019. ISSN 2058-9689. doi: 10.1039/C9ME00036D. URL https://pubs.rsc.org/en/content/articlelanding/2019/me/c9me00036d. Publisher: The Royal Society of Chemistry.

[91] Kieran Alden, Jason Cosgrove, Mark Coles, and Jon Timmis. Using Emulation to Engineer and Understand Simulations of Biological Systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1):302–315, January 2020. ISSN 1557-9964. doi: 10.1109/TCBB.2018.2843339. Conference Name: IEEE/ACM Transactions on Computational Biology and Bioinformatics.

[92] Bo Yuan, Ciyue Shen, Augustin Luna, Anil Korkut, Debora S. Marks, John Ingraham, and Chris Sander. CellBox: Interpretable Machine Learning for Perturbation Biology with Application to the Design of Cancer Combination Therapy. *Cell Syst*, 12(2): 128–140.e4, February 2021. ISSN 2405-4720. doi: 10.1016/j.cels.2020.11.013.

[93] Anthony Culos, Amy S. Tsai, Natalie Stanley, Martin Becker, Mohammad S. Ghaemi, David R. McIlwain, Ramin Fallahzadeh, Athena Tanada, Huda Nassar, Camilo Espinosa, Maria Xenochristou, Edward Ganio, Laura Peterson, Xiaoyuan Han, Ina A. Stelzer, Kazuo Ando, Dyani Gaudilliere, Thanaphong Phongpreecha, Ivana Marić, Alan L. Chang, Gary M. Shaw, David K. Stevenson, Sean Bendall, Kara L. Davis, Wendy Fantl, Garry P. Nolan, Trevor Hastie, Robert Tibshirani, Martin S. Angst, Brice Gaudilliere, and Nima Aghaeepour. Integration of mechanistic immunological knowledge into a machine learning pipeline improves predictions. *Nat Mach Intell*, 2 (10):619–628, October 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00232-8. URL https://www.nature.com/articles/s42256-020-00232-8. Number: 10 Publisher: Nature Publishing Group.

[94] Jason Y. Cain, Jacob I. Evarts, Jessica S. Yu, and Neda Bagheri. Incorporating temporal information during feature engineering bolsters emulation of spatio-temporal emergence. *Submitted to Bioinformatics*, January 2023.

[95] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J. McCarthy, Stephanie C. Hicks, Mark D. Robinson, Catalina A. Vallejos, Kieran R. Campbell, Niko Beerenwinkel, Ahmed Mahfouz, Luca Pinello, Pavel Skums, Alexandros Stamatakis, Camille Stephan-Otto Attolini, Samuel Aparicio, Jasmijn Baaijens, Marleen Balvert, Buys de Barbanson, Antonio Cappuccio, Giacomo Corleone, Bas E. Dutilh, Maria Florescu, Victor Guryev, Rens Holmer, Katharina Jahn, Thamar Jessurun Lobo, Emma M. Keizer, Indu Khatri, Szymon M. Kielbasa, Jan O. Korbel, Alexey M. Kozlov, Tzu-Hao Kuo, Boudewijn P.F. Lelieveldt, Ion I. Mandoiu, John C. Marioni, Tobias Marschall, Felix Mölder, Amir Niknejad, Lukasz Raczkowski, Marcel Reinders, Jeroen de Ridder, Antoine-Emmanuel Saliba, Antonios Somarakis, Oliver Stegle, Fabian J. Theis, Huan Yang, Alex Zelikovsky, Alice C. McHardy, Benjamin J. Raphael, Sohrab P. Shah, and Alexander Schönhuth. Eleven grand challenges in single-cell data science. *Genome Biology*, 21(1):31, February 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-1926-6.

[96] Raluca Eftimie. Grand challenges in mathematical biology: Integrating multi-scale modeling and data. *Frontiers in Applied Mathematics and Statistics*, 8, 2022. ISSN 2297-4687.

[97] Gabriela Bindea, Bernhard Mlecnik, Marie Tosolini, Amos Kirilovsky, Maximilian Waldner, Anna C. Obenauf, Helen Angell, Tessa Fredriksen, Lucie Lafontaine, Anne Berger, Patrick Bruneval, Wolf Herman Fridman, Christoph Becker, Franck Pagès, Michael R. Speicher, Zlatko Trajanoski, and Jérôme Galon. Spatiotemporal Dynamics of Intratumoral Immune Cells Reveal the Immune Landscape in Human Cancer. *Immunity*, 39(4):782–795, October 2013. ISSN 1074-7613. doi: 10.1016/j.immuni.2013.10.003.

[98] M. E. Johnson, A. Chen, J. R. Faeder, P. Henning, I. I. Moraru, M. Meier-Schellersheim, R. F. Murphy, T. Prüstel, J. A. Theriot, and A. M. Uhrmacher. Quantifying the roles of space and stochasticity in computer simulations for cell biology and cellular biochemistry. *MBoC*, 32(2):186–210, January 2021. ISSN 1059-1524. doi: 10.1091/mbc.E20-08-0530.

[99] Mickaël Lelek, Melina T. Gyparaki, Gerti Beliu, Florian Schueder, Juliette Griffié, Suliana Manley, Ralf Jungmann, Markus Sauer, Melike Lakadamyali, and Christophe Zimmer. Single-molecule localization microscopy. *Nat Rev Methods Primers*, 1:39, 2021. ISSN 2662-8449. doi: 10.1038/s43586-021-00038-x.

[100] Neda Bagheri, Anne E. Carpenter, Emma Lundberg, Anne L. Plant, and Rick Horwitz. The new era of quantitative cell imaging—challenges and opportunities. *Molecular Cell*, 82(2):241–247, January 2022. ISSN 1097-2765. doi: 10.1016/j.molcel.2021.12.024.

[101] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med*, 50(8):1–14, August 2018. ISSN 2092-6413. doi: 10.1038/s12276-018-0071-8.

[102] Lukas Heumos, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, Daniel C. Strobl, Juan Henao, Fabiola Curion, Herbert B. Schiller, and Fabian J. Theis. Best practices for single-cell analysis across modalities. *Nat Rev Genet*, pages 1–23, March 2023. ISSN 1471-0064. doi: 10.1038/s41576-023-00586-w.

[103] Zhiwei Ji, Ke Yan, Wenyang Li, Haigen Hu, and Xiaoliang Zhu. Mathematical and Computational Modeling in Complex Biological Systems. *BioMed Research International*, 2017:e5958321, March 2017. ISSN 2314-6133. doi: 10.1155/2017/5958321.

[104] Johannes Möller and Ralf Pörtner. Digital Twins for Tissue Culture Techniques—Concepts, Expectations, and State of the Art. *Processes*, 9(3):447, March 2021. ISSN 2227-9717. doi: 10.3390/pr9030447.

[105] Raluca Eftimie, A. Mavrodin, and Stéphane P. A. Bordas. Chapter Four - From digital control to digital twins in medicine: A brief review and future perspectives. In Stéphane P. A. Bordas, editor, *Advances in Applied Mechanics*, volume 56, pages 323–368. Elsevier, January 2023. doi: 10.1016/bs.aams.2022.09.001.

[106] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl_3):7280–7287, May 2002. doi: 10.1073/pnas.082080899.

[107] Ahmadreza Ghaffarizadeh, Randy Heiland, Samuel H. Friedman, Shannon M. Mumenthaler, and Paul Macklin. PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems. *PLOS Computational Biology*, 14(2):e1005991, February 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005991.

[108] Jessica S. Yu and Neda Bagheri. Agent-Based Models Predict Emergent Behavior of Heterogeneous Cell Populations in Dynamic Microenvironments. *Front Bioeng Biotechnol*, 8:249, 2020. ISSN 2296-4185. doi: 10.3389/fbioe.2020.00249.

[109] James M. Osborne, Alexander G. Fletcher, Joe M. Pitt-Francis, Philip K. Maini, and David J. Gavaghan. Comparing individual-based approaches to modelling the self-organization of multicellular tissues. *PLOS Computational Biology*, 13(2):e1005387, February 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005387.

[110] Cliff C. Kerr, Robyn M. Stuart, Dina Mistry, Romesh G. Abeysuriya, Katherine Rosenfeld, Gregory R. Hart, Rafael C. Núñez, Jamie A. Cohen, Prashanth Selvaraj, Brittany Hagedorn, Lauren George, Michal Jastrzebski, Amanda S. Izzo, Greer Fowler,

Anna Palmer, Dominic Delport, Nick Scott, Sherrie L. Kelly, Caroline S. Bennette, Bradley G. Wagner, Stewart T. Chang, Assaf P. Oron, Edward A. Wenger, Jasmina Panovska-Griffiths, Michael Famulare, and Daniel J. Klein. Covasim: An agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology*, 17(7): e1009149, July 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009149.

[111] Mohammad Soheilypour and Mohammad R. K. Mofrad. Agent-Based Modeling in Molecular Systems Biology. *BioEssays*, 40(7):1800020, 2018. ISSN 1521-1878. doi: 10.1002/bies.201800020.

[112] Qiyao Peng and Fred Vermolen. Agent-based modelling and parameter sensitivity analysis with a finite-element method for skin contraction. *Biomech Model Mechanobiol*, 19(6):2525–2551, December 2020. ISSN 1617-7940. doi: 10.1007/s10237-020-01354-z.

[113] Joseph M Cicchese, Elsje Pienaar, Denise E Kirschner, and Jennifer J Linderman. Applying optimization algorithms to tuberculosis antibiotic treatment regimens. *Cellular and molecular bioengineering*, 10:523–535, 2017.

[114] Ian Vernon, Junli Liu, Michael Goldstein, James Rowe, Jen Topping, and Keith Lindsey. Bayesian uncertainty analysis for complex systems biology models: Emulation, global parameter searches and evaluation of gene functions. *BMC Systems Biology*, 12 (1):1, January 2018. ISSN 1752-0509. doi: 10.1186/s12918-017-0484-3.

[115] Shangying Wang, Kai Fan, Nan Luo, Yangxiaolu Cao, Feilun Wu, Carolyn Zhang, Katherine A. Heller, and Lingchong You. Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nat Commun*, 10(1): 4354, September 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-12342-y.

[116] Minh Kieu, Hoang Nguyen, Jonathan A. Ward, and Nick Malleson. Towards real-time predictions using emulators of agent-based models. *Journal of Simulation*, 0(0):1–18, June 2022. ISSN 1747-7778. doi: 10.1080/17477778.2022.2080008.

[117] Claudio Angione, Eric Silverman, and Elisabeth Yaneske. Using machine learning as a surrogate model for agent-based simulations. *PLOS ONE*, 17(2):e0263150, February 2022. ISSN 1932-6203. doi: 10.1371/journal.pone.0263150.

[118] Chad M. Glen, Melissa L. Kemp, and Eberhard O. Voit. Agent-based modeling of morphogenetic systems: Advantages and challenges. *PLOS Computational Biology*, 15 (3):e1006577, March 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006577.

[119] Alison Heppenstall, Andrew Crooks, Nick Malleson, Ed Manley, Jiaqi Ge, and Michael Batty. Future Developments in Geographical Agent-Based Models: Challenges and Opportunities. *Geographical Analysis*, 53(1):76–91, 2021. ISSN 1538-4632. doi: 10.1111/gean.12267.

[120] Thierry Fredrich, Michael Welter, and Heiko Rieger. Tumorcode. *Eur. Phys. J. E*, 41 (4):55, April 2018. ISSN 1292-895X. doi: 10.1140/epje/i2018-11659-x.

[121] Thierry Fredrich, Heiko Rieger, Roberto Chignola, and Edoardo Milotti. Fine-grained simulations of the microenvironment of vascularized tumours. *Sci Rep*, 9(1):11698, August 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-48252-8.

[122] Georgios A. Pavlopoulos, Maria Secrier, Charalampos N. Moschopoulos, Theodoros G. Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G. Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4(1):10, April 2011. ISSN 1756-0381. doi: 10.1186/1756-0381-4-10.

[123] Mikaela Koutrouli, Evangelos Karatzas, David Paez-Espino, and Georgios A. Pavlopoulos. A Guide to Conquer the Biological Network Era Using Graph Theory. *Frontiers in Bioengineering and Biotechnology*, 8, 2020. ISSN 2296-4185.

[124] Erin E. Peterson, Jay M. Ver Hoef, Dan J. Isaak, Jeffrey A. Falke, Marie-Josée Fortin, Chris E. Jordan, Kristina McNyset, Pascal Monestiez, Aaron S. Ruesch, Aritra Sengupta, Nicholas Som, E. Ashley Steel, David M. Theobald, Christian E. Torgersen, and Seth J. Wenger. Modelling dendritic ecological networks in space: An integrated network perspective. *Ecology Letters*, 16(5):707–719, 2013. ISSN 1461-0248. doi: 10.1111/ele.12084.

[125] Giuseppe Modica, Salvatore Praticò, Luigi Laudari, Antonio Ledda, Salvatore Di Fazio, and Andrea De Montis. Implementation of multispecies ecological networks at the regional scale: Analysis and multi-temporal assessment. *Journal of Environmental Management*, 289:112494, July 2021. ISSN 0301-4797. doi: 10.1016/j.jenvman.2021.112494.

[126] Chris S. Magnano and Anthony Gitter. Automating parameter selection to avoid implausible biological pathway models. *npj Syst Biol Appl*, 7(1):1–12, February 2021. ISSN 2056-7189. doi: 10.1038/s41540-020-00167-1.

[127] Olaf Sporns. Structure and function of complex brain networks. *Dialogues Clin Neurosci*, 15(3):247–262, September 2013. ISSN 1294-8322.

[128] Jelmer G. Kok, Alexander Leemans, Laura K. Teune, Klaus L. Leenders, Martin J. McKeown, Silke Appel-Cresswell, Hubertus P. H. Kremer, and Bauke M. de Jong. Structural Network Analysis Using Diffusion MRI Tractography in Parkinson's Disease and Correlations With Motor Impairment. *Front Neurol*, 11:841, September 2020. ISSN 1664-2295. doi: 10.3389/fneur.2020.00841.

[129] Danielle S. Bassett and Olaf Sporns. Network neuroscience. *Nat Neurosci*, 20(3): 353–364, March 2017. ISSN 1546-1726. doi: 10.1038/nn.4502.

[130] Sergio Iadevaia, Yiling Lu, Fabiana C. Morales, Gordon B. Mills, and Prahlad T. Ram. Identification of Optimal Drug Combinations Targeting Cellular Networks: Integrating Phospho-Proteomics and Computational Network Analysis. *Cancer Research*, 70(17): 6704–6714, August 2010. ISSN 0008-5472. doi: 10.1158/0008-5472.CAN-10-0460.

[131] Ivan Amat-Roldan, Annalisa Berzigotti, Rosa Gilabert, and Jaime Bosch. Assessment of Hepatic Vascular Network Connectivity with Automated Graph Analysis of Dynamic Contrast-enhanced US to Evaluate Portal Hypertension in Patients with Cirrhosis: A Pilot Study. *Radiology*, 277(1):268–276, October 2015. ISSN 0033-8419. doi: 10.1148/radiol.2015141941.

[132] A. P. Alves, O. N. Mesquita, J. Gómez-Gardeñes, and U. Agero. Graph analysis of cell clusters forming vascular networks. *R Soc Open Sci*, 5(3), March 2018. ISSN 2054-5703. doi: 10.1098/rsos.171592.

[133] Anahita Fouladzadeh, Mohsen Dorraki, Kay Khine Myo Min, Michaelia P. Cockshell, Emma J. Thompson, Johan W. Verjans, Andrew Allison, Claudine S. Bonder, and Derek Abbott. The development of tumour vascular networks. *Commun Biol*, 4(1): 1–10, September 2021. ISSN 2399-3642. doi: 10.1038/s42003-021-02632-x.

[134] Jessica S. Yu. Arcade. `https://github.com/bagherilab/ARCADE`, 2023.

[135] Alexis N. Prybutok, Jessica S. Yu, Joshua N. Leonard, and Neda Bagheri. Mapping CAR T-Cell Design Space Using Agent-Based Models. *Frontiers in Molecular Biosciences*, 9, 2022. ISSN 2296-889X.

[136] Gabor Csardi and Tamas Nepusz. The Igraph Software Package for Complex Network Research. *InterJournal*, Complex Systems:1695, November 2005.

[137] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928.

[138] I. M. Sobol. Uniformly distributed sequences with an additional uniform property. *USSR Computational Mathematics and Mathematical Physics*, 16(5):236–242, January 1976. ISSN 0041-5553. doi: 10.1016/0041-5553(76)90154-3.

[139] 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, C. J. 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, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat Methods*, 17(3):261–272, March 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2.

[140] Francois Chollet et al. Keras, 2015. URL `https://github.com/fchollet/keras`.

[141] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[142] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[143] Pablo J. Blanco, Lucas O. Müller, and J. David Spence. Blood pressure gradients in cerebral arteries: A clue to pathogenesis of cerebral small vessel disease. *Stroke Vasc Neurol*, 2(3), September 2017. ISSN 2059-8688, 2059-8696. doi: 10.1136/svn-2017-000087.

[144] Lukas Ortmann, Dawei Shi, Eyal Dassau, Francis J. Doyle, Berno J.E. Misgeld, and Steffen Leonhardt. Automated insulin delivery for type 1 diabetes mellitus patients using gaussian process-based model predictive control. In *2019 American Control Conference (ACC)*, pages 4118–4123, 2019. doi: 10.23919/ACC.2019.8815258.

[145] Elizabeth A Logsdon, Stacey D Finley, Aleksander S Popel, and Feilim Mac Gabhann. A systems biology view of blood vessel growth and remodelling. *J Cell Mol Med*, 18(8):1491–1508, August 2014. ISSN 1582-1838. doi: 10.1111/jcmm.12164. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4190897/`.

[146] Mary J. C. Hendrix, Elisabeth A. Seftor, Richard E. B. Seftor, Jun-Tzu Chao, Du-Shieng Chien, and Yi-Wen Chu. Tumor cell vascular mimicry: Novel targeting opportunity in melanoma. *Pharmacology & Therapeutics*, 159:83–92, March 2016. ISSN 0163-7258. doi: 10.1016/j.pharmthera.2016.01.006.

[147] F. D. Bookholt, H. N. Monsuur, S. Gibbs, and F. J. Vermolen. Mathematical modelling of angiogenesis using continuous cell-based models. *Biomech Model Mechanobiol*, 15

(6):1577–1600, 2016. ISSN 1617-7959. doi: 10.1007/s10237-016-0784-3. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5106520/.

[148] Steven A. Stacker and Marc G. Achen. The VEGF signaling pathway in cancer: the road ahead. *Chin J Cancer*, 32(6):297–302, June 2013. ISSN 1000-467X. doi: 10.5732/cjc.012.10319. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3845619/.

[149] C. Bayer and P. Vaupel. Acute versus chronic hypoxia in tumors. *Strahlenther Onkol*, 188(7):616–627, July 2012. ISSN 1439-099X. doi: 10.1007/s00066-012-0085-4. URL https://doi.org/10.1007/s00066-012-0085-4.

[150] Miguel A. S. Cavadas, Alex Cheong, and Cormac T. Taylor. The regulation of transcriptional repression in hypoxia. *Exp Cell Res*, In press, February 2017. ISSN 0014-4827. doi: 10.1016/j.yexcr.2017.02.024. URL https://research.aston.ac.uk/en/publications/the-regulation-of-transcriptional-repression-in-hypoxia. Publisher: Academic Press Inc.

[151] James P. Freyer and Robert M. Sutherland. Proliferative and Clonogenic Heterogeneity of Cells from EMT6/Ro Multicellular Spheroids Induced by the Glucose and Oxygen Supply. *Cancer Res*, 46(7):3513–3520, July 1986. ISSN 0008-5472, 1538-7445. URL https://cancerres.aacrjournals.org/content/46/7/3513. Publisher: American Association for Cancer Research Section: Basic Sciences.

[152] Antonio Brú, Sonia Albertos, José Luis Subiza, José López García-Asenjo, and Isabel Brú. The Universal Dynamics of Tumor Growth. *Biophys J*, 85(5):2948–2961, November 2003. ISSN 0006-3495. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1303573/.

[153] S. R. McKeown. Defining normoxia, physoxia and hypoxia in tumours-implications for treatment response. *Br J Radiol*, 87(1035):20130676, March 2014. ISSN 1748-880X. doi: 10.1259/bjr.20130676.

[154] Peter Vaupel and Arnulf Mayer. Tumor Hypoxia: Causative Mechanisms, Microregional Heterogeneities, and the Role of Tissue-Based Hypoxia Markers. In Qingming Luo, Lin Z. Li, David K. Harrison, Hua Shi, and Duane F. Bruley, editors, *Oxygen Transport to Tissue XXXVIII*, Advances in Experimental Medicine and Biology, pages 77–86, Cham, 2016. Springer International Publishing. ISBN 978-3-319-38810-6. doi: 10.1007/978-3-319-38810-6_11.

[155] Agnieszka Loboda, Alicja Jozkowicz, and Jozef Dulak. HIF-1 and HIF-2 transcription factors — Similar but not identical. *Mol Cells*, 29(5):435–442, May 2010. ISSN 1016-8478, 0219-1032. doi: 10.1007/s10059-010-0067-2. URL http://link.springer.com/10.1007/s10059-010-0067-2.

[156] Brian Keith, Randall S. Johnson, and M. Celeste Simon. HIF1α and HIF2α: sibling rivalry in hypoxic tumour growth and progression. *Nature Reviews Cancer*, 12(1):9–22, January 2012. ISSN 1474-1768. doi: 10.1038/nrc3183. URL https://www.nature.com/articles/nrc3183. Number: 1 Publisher: Nature Publishing Group.

[157] Lan K. Nguyen, Miguel A. S. Cavadas, Carsten C. Scholz, Susan F. Fitzpatrick, Ulrike Bruning, Eoin P. Cummins, Murtaza M. Tambuwala, Mario C. Manresa, Boris N. Kholodenko, Cormac T. Taylor, and Alex Cheong. A dynamic model of the hypoxia-inducible factor 1α (HIF-1α) network. *J Cell Sci*, 126(6):1454–1463, March 2013. ISSN 0021-9533, 1477-9137. doi: 10.1242/jcs.119974. URL https://jcs.biologists.org/content/126/6/1454. Publisher: The Company of Biologists Ltd Section: Research Article.

[158] E. N. Unemori, N. Ferrara, E. A. Bauer, and E. P. Amento. Vascular endothelial growth factor induces interstitial collagenase expression in human endothelial cells. *J Cell Physiol*, 153(3):557–562, December 1992. ISSN 0021-9541. doi: 10.1002/jcp.1041530317.

[159] C. Kut, F. Mac Gabhann, and A. S. Popel. Where is VEGF in the body? A meta-analysis of VEGF distribution in cancer. *Br J Cancer*, 97(7):978–985, October 2007. ISSN 1532-1827. doi: 10.1038/sj.bjc.6603923.

[160] Shingo Takano, Yoshihiko Yoshii, Shinichi Kondo, Hideo Suzuki, Tooru Maruno, Shizuo Shirai, and Tadao Nose. Concentration of Vascular Endothelial Growth Factor in the Serum and Tumor Tissue of Brain Tumor Patients1. *Cancer Research*, 56(9):2185–2190, May 1996. ISSN 0008-5472.

[161] Peter Vaupel and Arnulf Mayer. Hypoxia in Tumors: Pathogenesis-Related Classification, Characterization of Hypoxia Subtypes, and Associated Biological and Clinical Implications. In Harold M. Swartz, David K. Harrison, and Duane F. Bruley, editors, *Oxygen Transport to Tissue XXXVI*, Advances in Experimental Medicine and Biology, pages 19–24, New York, NY, 2014. Springer. ISBN 978-1-4939-0620-8. doi: 10.1007/978-1-4939-0620-8_3.

[162] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. MASON: A Multiagent Simulation Environment. *SIMULATION*, 81(7):517–527, 2005. doi: 10.1177/0037549705058073. URL https://doi.org/10.1177/0037549705058073. _eprint: https://doi.org/10.1177/0037549705058073.

[163] Jason Y. Cain, Jessica S. Yu, and Neda Bagheri. The in silico lab: Improving academic code using lessons from biology. *Cell Systems*, 14(1):1–6, January 2023. ISSN 2405-4712. doi: 10.1016/j.cels.2022.11.006.

[164] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. ISSN 1476-4687. doi: 10.1038/533452a. URL https://www.nature.com/articles/533452a.

[165] Krishna Tiwari, Sarubini Kananathan, Matthew G. Roberts, Johannes P. Meyer, Mohammad Umer Sharif Shohan, Ashley Xavier, Matthieu Maire, Ahmad Zyoud, Jinghao Men, Szeyi Ng, Tung V. N. Nguyen, Mihai Glont, Henning Hermjakob, and Rahuman S. Malik-Sheriff. Reproducibility in systems biology modelling. *Molecular Systems Biology*, 17(2):e9982, 2021. ISSN 1744-4292. doi: 10.15252/msb.20209982. URL https://www.embopress.org/doi/full/10.15252/msb.20209982.

[166] Marcin MiA kowski, Witold M. Hensel, and Mateusz Hohol. Replicability or reproducibility? on the replication crisis in computational neuroscience and sharing only relevant detail. *J Comput Neurosci*, 45(3):163–172, 2018. ISSN 1573-6873. doi: 10.1007/s10827-018-0702-z. URL https://doi.org/10.1007/s10827-018-0702-z.

[167] Mathieu Boudreau, Jean-Baptiste Poline, Pierre Bellec, and Nikola Stikov. On the open-source landscape of PLOS computational biology. *PLOS Computational Biology*, 17(2):e1008725, 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008725. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008725.

[168] Massimiliano Bonomi, Giovanni Bussi, Carlo Camilloni, Gareth A. Tribello, Pavel Banáš, Alessandro Barducci, Mattia Bernetti, Peter G. Bolhuis, Sandro Bottaro, Davide Branduardi, Riccardo Capelli, Paolo Carloni, Michele Ceriotti, Andrea Cesari, Haochuan Chen, Wei Chen, Francesco Colizzi, Sandip De, Marco De La Pierre, Davide Donadio, Viktor Drobot, Bernd Ensing, Andrew L. Ferguson, Marta Filizola, James S. Fraser, Haohao Fu, Piero Gasparotto, Francesco Luigi Gervasio, Federico Giberti, Alejandro Gil-Ley, Toni Giorgino, Gabriella T. Heller, Glen M. Hocky, Marcella Iannuzzi, Michele Invernizzi, Kim E. Jelfs, Alexander Jussupow, Evgeny Kirilin, Alessandro Laio, Vittorio Limongelli, Kresten Lindorff-Larsen, Thomas Lohr, Fabrizio Marinelli, Layla Martin-Samos, Matteo Masetti, Ralf Meyer, Angelos Michaelides, Carla Molteni, Tetsuya Morishita, Marco Nava, Cristina Paissoni, Elena Papaleo, Michele Parrinello, Jim Pfaendtner, Pablo Piaggi, GiovanniMaria Piccini, Adriana Pietropaolo, Fabio Pietrucci, Silvio Pipolo, Davide Provasi, David Quigley, Paolo Raiteri, Stefano Raniolo, Jakub Rydzewski, Matteo Salvalaglio, Gabriele Cesare Sosso, Vojtěch Spiwok, Jiří Šponer, David W. H. Swenson, Pratyush Tiwary, Omar Valsson, Michele Vendruscolo, Gregory A. Voth, Andrew White, and The PLUMED consortium. Promoting transparency and reproducibility in enhanced molecular simulations. *Nat Methods*, 16(8):670–673, 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0506-8. URL https://www.nature.com/articles/s41592-019-0506-8.

[169] Leandro Watanabe, Tramy Nguyen, Michael Zhang, Zach Zundel, Zhen Zhang, Curtis

Madsen, Nicholas Roehner, and Chris Myers. iBioSim 3: A tool for model-based genetic circuit design. *ACS Synth. Biol.*, 8(7):1560–1563, 2019. doi: 10.1021/acssynbio.8b00078. URL https://doi.org/10.1021/acssynbio.8b00078.

[170] Benjamin D. Lee. Ten simple rules for documenting scientific software. *PLOS Computational Biology*, 14(12):1–6, 12 2018. doi: 10.1371/journal.pcbi.1006561. URL https://doi.org/10.1371/journal.pcbi.1006561.

[171] Maureen A. Carey and Jason A. Papin. Ten simple rules for biologists learning to program. *PLOS Computational Biology*, 14(1):1–11, 01 2018. doi: 10.1371/journal.pcbi.1005871. URL https://doi.org/10.1371/journal.pcbi.1005871.

[172] James T. Yurkovich, Benjamin J. Yurkovich, Andreas Dräger, Bernhard O. Palsson, and Zachary A. King. A padawan programmer's guide to developing software libraries. *Cell Systems*, 5(5):431–437, Nov 2017. ISSN 2405-4712. doi: 10.1016/j.cels.2017.08.003. URL https://doi.org/10.1016/j.cels.2017.08.003.

[173] Markus List, Peter Ebert, and Felipe Albrecht. Ten simple rules for developing usable software in computational biology. *PLOS Computational Biology*, 13(1):1–5, 01 2017. doi: 10.1371/journal.pcbi.1005265. URL https://doi.org/10.1371/journal.pcbi.1005265.

[174] Haley Hunter-Zinck, Alexandre Fioravante de Siqueira, VÃ¡leri N. VÃ¡squez, Richard Barnes, and Ciera C. Martinez. Ten simple rules on writing clean and reliable open-source scientific software. *PLOS Computational Biology*, 17(11):1–9, 11 2021. doi: 10.1371/journal.pcbi.1009481. URL https://doi.org/10.1371/journal.pcbi.1009481.

[175] Jeffrey T. Leek and Roger D. Peng. Opinion: Reproducible research can still be wrong: Adopting a prevention approach. *PNAS*, 112(6):1645–1646, 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1421412111. URL https://www.pnas.org/content/112/6/1645.

[176] Pjotr Prins, Joep de Ligt, Artem Tarasov, Ritsert C. Jansen, Edwin Cuppen, and Philip E. Bourne. Toward effective software solutions for big biology. *Nat Biotechnol*, 33(7):686–687, 2015. ISSN 1546-1696. doi: 10.1038/nbt.3240. URL https://www.nature.com/articles/nbt.3240.

[177] Sarah M Keating, Dagmar Waltemath, Matthias König, Fengkai Zhang, Andreas Dräger, Claudine Chaouiya, Frank T Bergmann, Andrew Finney, Colin S Gillespie, Tomáš Helikar, Stefan Hoops, Rahuman S Malik-Sheriff, Stuart L Moodie, Ion I Moraru, Chris J Myers, AurÃ©lien Naldi, Brett G Olivier, Sven Sahle, James C Schaff, Lucian P Smith, Maciej J Swat, Denis Thieffry, Leandro Watanabe, Darren J Wilkinson, Michael L Blinov, Kimberly Begley, James R Faeder, Harold F

Gómez, Thomas M Hamm, Yuichiro Inagaki, Wolfram Liebermeister, Allyson L Lister, Daniel Lucio, Eric Mjolsness, Carole J Proctor, Karthik Raman, Nicolas Rodriguez, Clifford A Shaffer, Bruce E Shapiro, Joerg Stelling, Neil Swainston, Naoki Tanimura, John Wagner, Martin Meier-Schellersheim, Herbert M Sauro, Bernhard Palsson, Hamid Bolouri, Hiroaki Kitano, Akira Funahashi, Henning Hermjakob, John C Doyle, Michael Hucka, and SBML Level 3 Community members. Sbml level 3: an extensible format for the exchange and reuse of biological models. *Molecular Systems Biology*, 16(8):e9110, 2020. doi: https://doi.org/10.15252/msb.20199110. URL `https://www.embopress.org/doi/abs/10.15252/msb.20199110`.