Title

Methods for Optimizing Mechanistic and Predictive Models of Human Disease

Author

Mester, Rachel

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERITY OF CALIFORNIA

Los Angeles

Methods for Optimizing Mechanistic and Predictive Models of Human Disease

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy in Biomathematics

by

Rachel Shoshana Mester

2024

ABSTRACT OF THE DISSERTATION

Methods for Optimizing Mechanistic and Predictive Models of Human Disease

by

Rachel Shoshana Mester

Doctor of Philosophy in Biomathematics

University of California, Los Angeles, 2024

Professor Bogdan Pasaniuc, Chair

A major goal of the biomathematics discipline is to optimize mathematical models for biological processes. This optimization can take on various forms; finding the appropriate model that fits available data, allows for accurate inference, and is computationally feasible is no easy task and requires an understanding of both the biological processes at hand and the mathematics behind each potential model or algorithm. In this dissertation, I seek to understand how mathematical modeling choices affect our ability to understand human disease. I study infectious, cancerous, and polygenic disease from a variety of computational perspectives. First, I apply methods for differential sensitivity analysis in biological models for both cancerous and infectious disease spread. I compare prediction accuracy for existing first-order methods and propose a second-order method with enhanced flexibility both in terms of the model for which it is applied and the programming environment available. Second, I compare statistical approaches for uncovering genetics of complex disease in admixed populations, using likelihood ratio tests to understand how

to incorporate local ancestry in genome wide association studies to achieve the highest power. Third, I utilize machine learning methods to reduce diagnostic delay for patients across the University of California Health system. I adapt a logistic regression model to find patients likely to have common variable immune deficiencies from one health system to five health systems. I also adapt this algorithm from the immunology realm to the cardiology realm to predict cardiac amyloidosis. Along the way, I use this context to study automated feature selection, longitudinal feature engineering, and observational bias in electronic health record data.

The dissertation of Rachel Shoshana Mester is approved.

Kenneth L. Lange

Loes Marlein Olde Loohuis

Harold Joseph Pimentel

Bogdan Pasaniuc, Committee Chair

University of California, Los Angeles

2024

To my parents, who showed me how fun learning can be.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Bogdan Pasaniuc. Thank you for taking me on as your graduate student and helping corral my diverse interests into cohesive research projects. Thank you for giving me the opportunity to create work that has a real-world impact and for providing me with opportunities to share my work with others.

Next, I would like to thank Ken Lange for being a mentor to me both in research for *Differential methods for assessing sensitivity in biological models* and throughout my PhD. Taking your courses was an eye-opening and fascinating experience and I strive to bring your creativity for problem solving to everything that I've done.

Thank you to my other committee members, Harold Pimentel and Loes Olde Loohuis. Harold, you have always had my back and I appreciate so much your willingness to spend time with me to help me figure out what I want to do with my life and how to get there. Loes, I have always admired your research program and I appreciate your helpful feedback regarding my projects and proposals.

I'd like to acknowledge the co-authors of *Differential methods for assessing sensitivity in biological models*, Alfonso Landeros and Chris Rackauckas. Alfonso, thank you for helping me through the rollercoaster that is a first research project, and for your help in implementing the multi-threaded versions of differential sensitivity methods. Thank you to Chris Rackauckas for your expertise in automatic, forward, and adjoint differentiation, as well as the time you spent helping me implement Julia packages for each.

I'd also like to acknowledge the co-authors of *Impact of cross-ancestry genetic architecture on GWASs in admixed populations*, Kangcheng Hou, Yi Ding, Kathryn Burch, Arjun Bhattacharya, Gillian Meeks, and Brenna Henn. Kangcheng, your calm mastery of the

technicalities of admixed genetics was invaluable to me and I appreciate the foundation that your work has provided for me. Yi and Kathy, in addition to the technical expertise you provided your friendship and support as senior members of the lab helped me get through all the tough times of graduate school. Arjun, thank you for helping me work through the statistical details required for this project. Thank you to Gillian Meeks and Brenna Henn for your insightful questions regarding ancestry-specific allele frequencies and for sharing your expertise.

I will always be grateful to Ruth Johnson for creating PheNet and being so gracious as I took over its stewardship after she graduated. Ruthie, thank you for always being available to help transition this project and for being such an incredible friend in the process. I would also like to acknowledge the contributions of Manish Butte, Alexis Stevens, Aaron Chin, Veronica Tozzo, and Kristen Boulier to the third chapter of this dissertation. Manish, your vision of using a data driven approach to diagnose CVID is what made this possible. Alexis, thank you for being willing to spend hours diving into the details of chart review to ensure this project's success. Aaron, thank you for your clinical expertise and your help implementing antibiotic-specific infection windows in the UCHDW. Veronica, thank you for jumping into this crazy project, providing me with support, and taking on additional tasks related to this grant so that I could focus on my dissertation. Kristen, thank you so much for the time you spent with me explaining cardiac amyloidosis and compiling helpful clinical indications of the disease. You always keep me honest in my quest to use computational methods in ways that are helpful within clinical workflows.

Thank you so much to the other members of the Bogdan Lab, most notably Sandra Lapinska and Ella Petter. Sandra, it's been so fun sitting next to you in lab every day, and I appreciate our spontaneous chats, scientific or otherwise. Ella, thank you for always being there to help me with

editing a paper, practicing a presentation, or any of the other thankless tasks that are vital to research. I appreciate how you've helped cheer me on, in good times and in bad.

Thank you to the Biomathematics program at UCLA. Eric Sobel, thank you for your guidance as I navigated this program, and Jeannette Papp thank you for suggesting I get involved in high school science fairs. You helped me discover my love for mentorship which is a big part of my career goals today. Thank you to my cohort Mariana Harris, Christine Kling, and Gary Zhou. It was so fun studying with you all during our first year and it has been so cool to see the different directions we've taken with our research. I never would have made it past my qualification exams without you!

I'd like to extend a special acknowledgement to Janet Sinsheimer. Janet's support and advice was something that I didn't learn to fully appreciate until it was gone. I am so glad that I was able to experience her mentorship in the early years of my PhD and this department is not the same without her.

I would like to acknowledge my funding sources. I have been funded in part by the University of California Regents Eugene V. Cota Robles Award (2019-2020, 2022-2023), the National Institutes of Health award no. T32HG002536 (2020-2022), and the National Library of Medicine award no. T15LM013976 (2023-2024).

Thank you to my friends at and from UCLA. Gabe Hassler, Sam Christensen, Paheli Desai-Chowdhry, and Vicky Kelley, thank you for your guidance as alumni of the Biomath program and for your friendship. I can always count on you and on Emily Maciejewski to go on fun adventures to help me get out of my own head.

Outside of UCLA, I would like to thank my community of friends in Los Angeles. I am so grateful to Kayla Imhoff, Steven Halling, Ken Miller, Sophia Charan, Jacob Wasserman, Danielle

Ayalon, Elise Boretz, Kaitlyn Sever, Myra Meskin, and Ben Gurin for being my home away from home and for helping me appreciate life outside of the computer.

Thank you to my friends from afar, Emily Brown, Marissa Maimone and Aidan Mehigan. Thank you for letting me stay on your couches and for coming to stay on my couch while I've been in this foreign land of Los Angeles.

Thank you Mom, Dad, and Emma for being such an incredible family. You've helped me so much during the ups and downs of grad school, whether I needed a shoulder to cry on or a celebration planned, you have always been there for me when I needed you.

Last but not least, thank you to Gina Rozner for being an incredible partner. Your patience and support in the last year have been integral to my ability to get my work done and knowing that after a day in the lab I get to come home to you brings a smile to my face every day.

VITA

Degrees

2019-2020 M. S. Biomathematics
University of California, Los Angeles, Los Angeles, CA

2012-2016 B.S. Applied Mathematics
Columbia University, New York, NY

Research and Teaching Positions

2021 – 2024 Graduate Research Assistant, Department of Pathology and Laboratory Medicine
University of California, Los Angeles, Los Angeles, CA

2022 Teaching Assistant, Mathematics for Life Scientists (LS 30B)
University of California, Los Angeles, Los Angeles, CA

Publications

Johnson R, Stephens AV, **Mester R**, Knyazev S, Kohn LA, Freund MK, Bondhus L, Hill BL, Schwarz T, Zaitlen N, Arboleda VA. Electronic health record signatures identify undiagnosed patients with Common Variable Immunodeficiency Disease. *Science Translational Medicine. 2024 May 1;16(745):eade4510.*

Mester R, Hou K, Ding Y, Meeks G, Burch KS, Bhattacharya A, Henn BM, Pasaniuc B. Impact of cross-ancestry genetic architecture on GWASs in admixed populations. *The American Journal of Human Genetics. 2023 June 24.*

Hou K, Ding Y, Xu Z, Wu Y, Bhattacharya A, **Mester R**, Belbin GM, Buyske S, Conti DV, Darst BF, Fornage M. Causal effects on complex traits are similar for common variants across segments of different continental ancestries within admixed individuals. *Nature Genetics. 2023 Mar 20:1-0.*

Mester, R., Landeros, A., Rackauckas, C., & Lange, K. (2022). Differential methods for assessing sensitivity in biological models. *PLoS computational biology*, *18*(6), e1009598.

Hou K, Bhattacharya A, **Mester R**, Burch KS, Pasaniuc B. (2021). On powerful GWAS in admixed populations. *Nature genetics, 53(12):1631-3.*

Grants and Awards

Fall 2023 – Spring 2025	Biomedical Data Sciences for Precision Health Equity Training Grant
June 2023	American Journal of Human Genetics Featured Article
February 2023	Society of Industrial and Applied Mathematics Computational Science and Engineering Student Travel Grant
October 2022	American Society of Human Genetics Reviewers' Choice Abstract
Fall 2020 – Spring 2022	Genomic Analysis and Interpretation Training Grant
Fall 2019	Eugene V. Cota Robles Fellowship

Talks and Posters

April 2024	*Impact of cross-ancestry genetic architecture on GWAS in admixed populations,* RECOMB-Genetics Research Highlights Talk
December 2023	*Impact of cross-ancestry genetic architecture on GWAS in admixed populations*, Computational Medicine Quarterly Department Meeting Invited Talk
February 2023	*Differential Methods for Assessing Sensitivity in Biological Models*, Society of Industrial and Applied Mathematics' Conference on Computational Science and Engineering Contributed Talk
April 2022	*How Does Genetic Architecture Influence Disease Mapping in Admixed Populations?* lightning talk National Human Genome Research Institute conference lightning talk
various	*Leveraging >8 million patient records across University of California to understand common variable immunodeficiencies,* poster presented at UC Health Conference: From Data to Action: Driving Healthcare Innovation and American Society of Human Genetics Annual Meeting
various	*Impact of heterogeneity-by-ancestry on GWAS in admixed populations*, poster presented at UCLA Quantitative and Computational Biology Collaboratory Retreat, American Society of Human Genetics Annual Meeting, Pathology Research and Clinical Excellence Day, National Human Genome Research Institute Training and Career Development Annual Meeting

1 INTRODUCTION

For any field of applied mathematics, the choice of a model to fit the problem has a profound impact on the utility of the model. Does the choice of model accurately reflect the known mechanisms behind the problem at hand? Is fitting the data computationally feasible? Can the model be used for inference and/or prediction? These are the types of questions that must be asked before a particular model is implemented to solve a problem. The domain of human disease is no different, and when creating models in this discipline one must consider additional obstacles, such as the different scales in which human biology operates and the bias that is often present in observational data. In this dissertation, I address some open questions in the field of the mathematical, statistical, and computational modeling of human disease. In each chapter, I delve into a different category of human disease and a different type of mathematical modeling.

In Chapter 2, I focus on infectious and cancerous diseases. At first glance, these types of human disease may seem disparate, however both infectious and cancerous processes can be modeled as the collective behavior of individual agents (people in the case of infection and cells in the case of cancer) which can have a state of healthy or diseased. In this chapter, published as *Mester R, Landeros A, Rackauckas C, Lange K. Differential methods for assessing sensitivity in biological models. PLoS computational biology. 2022*[1.1], we apply stochastic models and dynamical systems to these processes. Next, we implement a variety of differential sensitivity algorithms to ascertain the advantages and trade-offs of each for these types of models. Sensitivity analysis is a process used to understand how a change in parameter value will impact the outcomes of the model. It is useful for applications such as fitting data, understanding uncertainty, and

prioritizing interventions. Differential sensitivity refers to the use of derivatives to analytically calculate a measure of model sensitivity to parameters.

We assessed three different types of differential sensitivity algorithms, including forward differentiation[1,2], adjoint differentiation[1,3], and the complex perturbation method[1,4]. We demonstrate the impact of applying differential sensitivity to a variety of biological models, from the concepts of cancer to infectious disease. These biological models also span the deterministic and stochastic model spaces. We compare the various aspects of the performances of these algorithms, including computational speed, memory, time to convergence, and precision of prediction power. We also consider the implementation benefits of each of these methods, including whether the algorithm requires differentiability of the base model, whether it can be parallelized, and whether it can be implemented in a multi-threaded way. In addition, we consider second-order methods for differential sensitivity in the same categories, including the proposal of a novel second-order complex perturbation method. My contributions to this work include implementing the complex perturbation methods in Julia, application of these methods to the six biological models we compared, computation of performance metrics, and analysis of results.

In Chapter 3, I turn to common diseases in humans. Specifically, I consider genome wide association studies (GWASs), which are statistical tests with the goal of identifying genetic variation in the form of single nucleotide polymorphisms (SNPs) which have a statistically significant correlation to the disease status of the carrier. From the construction of polygenic risk scores to the identification of biological mechanisms of disease, the genetics community has embraced GWAS as an important tool for almost 20 years, and GWAS has proven to be foundational to prioritizing genomic loci for further study. But ensuring that every person can be an ultimate beneficiary of GWAS is still an unfinished task. While we now know that the accuracy

of a polygenic risk score suffers when applied to a population distinct from the training population[1,5], the majority of GWAS studies are performed on populations of European ancestries. Specifically, admixed individuals (traditionally defined as individuals with recent ancestry from two or more continents) have been underrepresented in GWAS studies. While over 35% of US individuals self-report as having admixed ancestry, less than 5% of individuals in genomic studies as of 2020 were admixed[1,6]. In addition to the extremely important goal of including all persons in personalized medicine, including admixed individuals properly in GWAS will also improve our ability to detect genetic signals. Before GWAS became the norm for genetic association studies, admixture mapping was a successful method of finding genetic loci that correlate with phenotypes[1,7]. The recent mixing of ancestry present in the genomes of admixed individuals results in chromosomes that contain segments that can be mapped back to an ancestral population. The ancestry information of a particular segment of an admixed genome is known as the local ancestry for that segment of the genome. Genomic loci may harbor different linkage patterns and different allele frequencies depending on local ancestry. These differences linked to local ancestry are the key to admixture mapping. In the last 10 to 15 years, a variety of statistical methods have been developed to both leverage the benefits of GWAS and utilize the additional information that local ancestry provides. I assess the different types of situations in which utilizing local ancestry in GWAS is beneficial. We compare putative causal SNPs with differing allele frequencies by ancestry, different background linkage disequilibrium levels, and different levels of causal effect size heterogeneity to find the regions in this genetic architecture space in which GWAS methods that explicitly account for local ancestry have improved power over simpler, standard GWAS methods. This chapter was published as *Mester R, Hou K, Ding Y, Meeks G, Burch KS, Bhattacharya A, Henn BM, Pasaniuc B. Impact of cross-ancestry genetic architecture on GWASs*

in admixed populations. The American Journal of Human Genetics. 2023[1.8]. My contributions to this work include writing simulations, analyzing results, and writing the manuscript with the help of all co-authors.

In Chapter 4, I shift attention to the phenotypic presentation of uncommon human diseases. Instead of statistically correlating genetic variation with disease status and predicting risk of future disease onset, I use phenotypic variables from patients' electronic health records to infer their current underlying disease status. Some diseases, such as common variable immunodeficiencies (CVID), which is the focus in this chapter, often go undiagnosed for years after a patient first starts exhibiting symptoms[1.9]. This diagnostic delay is due to a combination of factors including the heterogeneous nature of the disease both in terms of presentation (symptoms can vary drastically between patients) and in terms of body system affected (patients may seek care from different doctors for different symptoms, resulting in a constellation of care in which no individual physician has the full picture of the disease). Using electronic health record (EHR) data (the data that results directly from patient interactions with the medical system such as doctor visits, diagnoses, medication prescriptions, lab results, and demographic information), we seek to find patients likely to have CVID for the purposes of referral to immunology. In *Johnson R, Stephens AV, Mester R, Knyazev S, Kohn LA, Freund MK, Bondhus L, Hill BL, Schwarz T, Zaitlen N, Arboleda VA. Electronic health record signatures identify undiagnosed patients with common variable immunodeficiency disease. Science Translational Medicine. 2024*[1.10], we use machine learning to create PheNet, an algorithm that outputs a patient's probability to have an underlying CVID diagnosis, and apply it in the University of California, Los Angeles (UCLA) deidentified data repository (DDR). My contributions to this work include completing computational work and figure preparations for the revisions of the manuscript.

Next, we extend PheNet from the UCLA health system to the rest of the University of California health system, using the University of California Health Data Warehouse (UCHDW). The UCHDW includes EHR data from the University of California health systems at Los Angeles, San Francisco, Davis, San Diego, and Irvine. In adapting PheNet to the UCHDW, we must assess the impact of covariates on the model. In the original PheNet algorithm, we trained our data on cases and controls that we matched based on age, sex, race, and time in the EHR. Now, each site has a different distribution of these covariates amongst the general population as well as a different distribution of the covariates within our case cohort. We introduce covariates into the model to reduce the bias that can be introduced from observational data.

Additionally, we were interested in finding a mathematically rigorous method for feature selection. In the original PheNet model, we used phecodes[1.11], a mapping of International Classification of Disease (ICD) codes which groups similar phecodes together for the purposes of statistical analysis, that were known to be clinically relevant to CVID. In this next iteration, we introduced a likelihood ratio test to independently assess whether adding each additional phecode to the model increased the likelihood of the model significantly. An additional important part of feature selection is the transformation of raw data into useful information. We consider the longitudinal aspect of EHR data to be a potentially important aspect of a patient's medical history and explored methods to incorporate it into PheNet.

Last, we investigate whether PheNet is extendable to other diseases. Cardiac amyloidosis is a form of amyloidosis, a disease that is caused by abnormal proteins building up in the body. While cardiac amyloidosis can cause heart failure, it is often missed even after a patient experiences a cardiac event. Patients with transthyretin cardiac amyloidosis are often eligible for tafamidis, a treatment that can help prevent a worsening of the disease. Thus, it is important to reduce the

diagnostic delay of cardiac amyloidosis to help improve prognosis for these patients. Previously, Huda et al[4.12] developed an algorithm to predict which heart failure patients had cardiac amyloidosis. In this last piece of the chapter, we use PheNet to differentiate between patients with and without cardiac amyloidosis at the point of first incidence of a heart failure diagnosis code.

Overall, in these next iterations of PheNet we have adapted PheNet for use in a new database (the UC Health Data Warehouse), introduced ridge regression as our statistical model for classification, used antibiotics data to calculate recurrence features from longitudinal electronic health record data, leveraged data driven feature selection for both the binary and recurrence features, and applied the entire process to a completely new phenotype. We have learned that including more data in our model improves model performance, even though that data comes from heterogeneous datasets. We have also learned that while clinical feature selection and data-driven feature selection result in similar performance in our CVID cohort, data-driven feature selection can be useful to fill in the gaps of clinical knowledge. Furthermore, we have learned that it is possible to infer episodes of infection from electronic health records. And finally, we have learned that we can apply this pipeline to additional phenotypes without much additional intervention.

My contributions to this work to extend PheNet to the UCHDW, improve feature selection and apply PheNet to cardiac amyloidosis were the study design, implementation of the methods and application in the UCHDW, and analysis of the results. This work is ongoing and may be published in the future.

1.1 REFERENCES

1.1 Mester, R., Landeros, A., Rackauckas, C., & Lange, K. (2022). Differential methods for assessing sensitivity in biological models. *PLoS computational biology*, *18*(6), e1009598.

1.2 Revels J, Lubin M, Papamarkou T. Forward-mode automatic differentiation in Julia. arXiv preprint arXiv:1607.07892. 2016 Jul 26.

1.3 Innes M. Don't unroll adjoint: Differentiating ssa-form programs. arXiv preprint arXiv:1810.07951. 2018 Oct 18.

1.4 Martins JR, Sturdza P, Alonso JJ. The complex-step derivative approximation. ACM Transactions on Mathematical Software (TOMS). 2003 Sep 1;29(3):245-62.

1.5 Martin AR, Kanai M, Kamatani Y, Okada Y, Neale BM, Daly MJ. Clinical use of current polygenic risk scores may exacerbate health disparities. Nature genetics. 2019 Apr;51(4):584-91.

1.6 Mills MC, Rahal C. The GWAS Diversity Monitor tracks diversity by disease in real time. Nature genetics. 2020 Mar;52(3):242-3.

1.7 Chakraborty R, Weiss KM. Admixture as a tool for finding linked genes and detecting that difference from allelic association between loci. Proceedings of the National Academy of Sciences. 1988 Dec;85(23):9119-23.

1.8 Mester R, Hou K, Ding Y, Meeks G, Burch KS, Bhattacharya A, Henn BM, Pasaniuc B. Impact of cross-ancestry genetic architecture on GWASs in admixed populations. The American Journal of Human Genetics. 2023 Jun 1;110(6):927-39.

1.9 C. A. Slade, J. J. Bosco, T. B. Giang, E. Kruse, R. G. Stirling, P. U. Cameron, F. Hore-Lacy, M. F. Sutherland, S. L. Barnes, S. Holdsworth, S. Ojaimi, G. A. Unglik, J. De

Luca, M. Patel, J. McComish, K. Spriggs, Y. Tran, P. Auyeung, K. Nicholls, R. E. O'Hehir, P. D. Hodgkin, J. A. Douglass, V. L. Bryant, M. C. van Zelm, Delayed diagnosis and complications of predominantly antibody deficiencies in a cohort of Australian adults. *Front. Immunol.* **9**, 694–694 (2018).

1.10 Johnson R, Stephens AV, Mester R, Knyazev S, Kohn LA, Freund MK, Bondhus L, Hill BL, Schwarz T, Zaitlen N, Arboleda VA. Electronic health record signatures identify undiagnosed patients with common variable immunodeficiency disease. Science Translational Medicine. 2024 May 1;16(745):eade4510.

1.11 Wu P, Gifford A, Meng X, Li X, Campbell H, Varley T, Zhao J, Carroll R, Bastarache L, Denny JC, Theodoratou E. Mapping ICD-10 and ICD-10-CM codes to phecodes: workflow development and initial evaluation. JMIR medical informatics. 2019 Nov 29;7(4):e14325.

1.12 Huda, Ahsan, Adam Castaño, Anindita Niyogi, Jennifer Schumacher, Michelle Stewart, Marianna Bruno, Mo Hu, Faraz S. Ahmad, Rahul C. Deo, and Sanjiv J. Shah. 2021. "A Machine Learning Model for Identifying Patients at Risk for Wild-Type Transthyretin Amyloid Cardiomyopathy." *Nature Communications* 12 (1): 2725.

2 DIFFERENTIAL METHODS FOR ASSESSING SENSITIVITY IN BIOLOGICAL MODELS

2.1 INTRODUCTION TO SENSITIVITY ANALYSIS

In many mathematical models underlying parameters are poorly specified. This problem is particularly acute in biological and biomedical models. Model predictions can have profound implications for scientific understanding, further experimentation, and even public-policy decisions. For instance, in an epidemic some model parameters can be tweaked by societal or scientific interventions to drive infection levels down. Differential sensitivity can inform medical judgement about the steps to take with the greatest impact at the least cost. Similar considerations apply in economic modeling. Additionally, parameter estimation for model fitting usually involves differential sensitivity through maximum likelihood or least squares criteria. These optimization techniques depend heavily on gradients and Hessians with respect to parameters. While some parameter estimation methods rely on Bayesian computational techniques[2.1] rather than gradients, these techniques tend to scale poorly as the number of model parameters increases. A common way to alleviate the poor scaling of Bayesian inference is Hamiltonian Monte Carlo[2.2], which itself requires gradient calculations. Techniques for assessing sensitivity of stochastic models often rely on the gradient-dependent Fisher information matrix of the model, which is the basis for a variety of multi-step local sensitivity analysis techniques for discrete stochastic models[2.3].

Calculation of gradients and Hessians of a model can also be important in other steps of the scientific process. For example, iterative model development[2.4] involves using the Fisher information matrix to inform experimental design. Extended Kalman filtering[2.5] incorporates differential sensitivity into model construction. Regardless of the method, parameter estimation is

an important step in fitting a biological model, and the success of this step strongly impacts the ultimate utility of the model. Understanding the uses and limitations of differential sensitivity can aid in determining the identifiability of model parameters, how sensitive they are to experimental error or measurement noise, and the overall importance of their existence in the model. Finally, it is worth noting that while local sensitivity analysis is the focus of this manuscript, global sensitivity analysis often relies on local differential sensitivity estimates to inform optimal stepsizes in regional searching[2.6] or to resolve inconsistencies that arise when local sensitivity is non-monotonic[2.8].

In any case it is imperative to know how sensitive model predictions are to changes in parameter values. Unfortunately, assessment of model sensitivity can be time consuming, computationally intensive, inaccurate, and simply confusing. Most models are nonlinear and resistant to exact mathematical analysis. Understanding their behavior is only approachable by solving differential equations or intensive and noisy simulations. Sensitivity analysis is often conducted over an entire bundle of neighboring parameters to capture interactions. If the parameter space is large or high-dimensional, it is often unclear how to choose representative points from this bundle. Faced with this dilemma, it is common for modelers to fall back on varying just one or two parameters at a time. Model predictions also often take the form of time trajectories. In this setting, sensitivity analysis is based on lower and upper trajectories bounding the behavior of the dynamical system.

The differential sensitivity of a model quantity is measured by its gradient with respect to the underlying parameters at their estimated values. The existing literature on differential sensitivity is summarized in the modern references[2.8,2.9]. There are a variety of software packages that evaluate parameter sensitivity. For example, the Julia software DifferentialEquations.jl[2.10]

makes sensitivity analysis routine for many problems. Additionally, PESTO[2.11] is a current Matlab toolbox for parameter estimation that uses adjoint sensitivities implemented as part of the CVODES method from SUNDIALS[2.12]. Although the physical sciences have widely adopted the method of differential sensitivity[2.13,2.14], the papers and software generally focus on a single sensitivity analysis method rather than a comparison of the various approaches. This singular focus leaves open many questions when biologists conduct sensitivity analyses. Should the continuous sensitivity equations be used, or would automatic differentiation of solvers be more efficient on biological models? On the types of models biologists generally explore, would implicit parallelism within the sensitivity equations be beneficial, or would the overhead cost of thread spawning overrule any benefits? How close do simpler methods based on complex perturbation get to these techniques? The purpose of the current paper is to explore these questions on a variety of models of interest to computational biologists.

In the current paper we also suggest an accurate method of approximating gradients that compares favorably against forward automatic differentiation techniques, provided a model involves analytic functions without discontinuities, maxima, minima, absolute values, or any other excursion outside the universe of analytic functions. In the sections immediately following, we summarize known theory, including the important adjoint method for computing the sensitivity of functions of solutions[2.13,2.14]. Then we illustrate sensitivity analysis for a few deterministic models and a few stochastic models. Our exposition includes some straightforward Julia code that readers can adapt to their own sensitivity needs. These examples are followed by an evaluation of the accuracy and speed of the suggested numerical methods. The concluding discussion summarizes our experience, indicates limitations of the methods, and suggests new potential applications.

For the record, here are some notational conventions used throughout the paper. All functions that we differentiate have real or real-vector arguments and real or real-vector values. All vectors and matrices appear in boldface. The superscript ^{T} indicates a vector or matrix transpose. For a smooth real-valued function $f(\mathbf{x})$, we write its gradient (column vector of partial derivatives) as $\nabla f(\mathbf{x})$ and its differential (row vector of partial derivatives) as $df(\mathbf{x}) = \nabla f(\mathbf{x})^T$. If $g(\mathbf{x})$ is vector-valued with ith component $g_i(\mathbf{x})$, then the differential (Jacobi matrix) $dg(\mathbf{x})$ has ith row $dg_i(\mathbf{x})$. The chain rule is expressed as the equality $d[f \circ g(\mathbf{x})] = df[g(\mathbf{x})]dg(\mathbf{x})$ of differentials. The transpose (adjoint) form of the chain rule is $\nabla f \circ g(\mathbf{x}) = dg(\mathbf{x})^T \nabla f[g(\mathbf{x})]$. For a twice-differentiable function, the second differential (Hessian matrix) $d^2 f(\mathbf{x}) = d\,\nabla f(\mathbf{x})$ is the differential of the gradient. Finally, i will denote $\sqrt{-1}$.

2.2 METHODS FOR COMPUTING SENSITIVITY

2.2.1 Forward Method

Section 2.8.3 briefly discusses sensitivity analysis for the linear constant coefficient system $\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\beta})\mathbf{x}(t)$ of ordinary differential equations (ODEs). Sensitivity of the nonlinear system $\frac{d}{dt}\mathbf{x}(t, \boldsymbol{\beta}) = f[\mathbf{x}(t), \boldsymbol{\beta}]$ can be evaluated by differentiating the original ODE with respect to β_j, interchanging the order of differentiation, and numerically integrating the system

$$\frac{d}{dt}\frac{\partial}{\partial \beta_j}\mathbf{x}(t, \boldsymbol{\beta}) \quad = \quad \frac{\partial}{\partial \beta_j} f[\mathbf{x}(t), \boldsymbol{\beta}] + d_{\mathbf{x}} f[\mathbf{x}(t), \boldsymbol{\beta}] \frac{\partial}{\partial \beta_j}\mathbf{x}(t, \boldsymbol{\beta}).$$

This formulation of the problem depends on knowing $\mathbf{x}(t, \boldsymbol{\beta})$. In practice, one solves the system

Equation 2.1:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}(t,\boldsymbol{\beta})\\ \nabla_{\boldsymbol{\beta}}\mathbf{x}(t,\boldsymbol{\beta})\end{bmatrix} = \begin{pmatrix} f[\mathbf{x}(t),\boldsymbol{\beta}] \\ \nabla_{\boldsymbol{\beta}} f[\mathbf{x}(t),\boldsymbol{\beta}] + d_{\boldsymbol{\beta}}\mathbf{x}(t,\boldsymbol{\beta})^T \nabla_{\mathbf{x}} f[\mathbf{x}(t),\boldsymbol{\beta}] \end{pmatrix}$$

jointly, where $d_{\boldsymbol{\beta}}\mathbf{x}[t,\boldsymbol{\beta}]$ is the Jacobi matrix of $\mathbf{x}(t,\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. This is commonly referred to as forward sensitivity analysis and is carried out by software suites such as DifferentialEquations.jl and SUNDIALS CVODES[2.12]. We note that a common implementation of sensitivity analysis is to base calculations on directional derivatives. Thus, the directional derivative

$$d_{\boldsymbol{\beta}}\mathbf{x}(t,\boldsymbol{\beta})^T \nabla_{\mathbf{x}} f[\mathbf{x}(t),\boldsymbol{\beta}] = \lim_{\epsilon\to 0} \frac{f\{\mathbf{x}(t)+\epsilon\nabla_{\mathbf{x}} f[\mathbf{x}(t),\boldsymbol{\beta}],\boldsymbol{\beta}\} - f[\mathbf{x}(t),\boldsymbol{\beta}]}{\epsilon}$$

version of the forward method allows one to evolve dynamical systems without ever computing full Jacobians. The forward method can also be applied when quantities of interest are defined recursively.

2.2.2 Adjoint Methods

The adjoint method is incorporated in the biological parameter estimation software PESTO through CVODES[2.12]. This method[2.8,2.9] is defined directly on a function $g[x(\boldsymbol{\beta}),\boldsymbol{\beta}]$ of the solution of the ODE. The adjoint method introduces a Lagrange multiplier $\lambda(\boldsymbol{\beta})$, numerically solves the ODE system forward in time over $[t_0,t_n]$, then solves the system

$$d_{\boldsymbol{\beta}}\lambda(\boldsymbol{\beta}) = d_{\mathbf{x}} f[\mathbf{x}(\boldsymbol{\beta}),\boldsymbol{\beta}]\lambda(\boldsymbol{\beta}) + d_{\boldsymbol{\beta}} g[\mathbf{x}(\boldsymbol{\beta}),\boldsymbol{\beta}],$$

for $\lambda(\boldsymbol{\beta})$ in reverse time, and finally uses the introduced parameter to compute derivatives via

$$d_{\boldsymbol{\beta}} g[x(\boldsymbol{\beta}), \boldsymbol{\beta}] \quad = \quad \int_{t_0}^{t_n} \lambda\,(t, \boldsymbol{\beta}) d_{\boldsymbol{\beta}} \mathbf{x}(t, \boldsymbol{\beta}) dt.$$

The second and third stages are commonly combined by appending the last equation to the set of ODEs being solved in reverse. This tactic achieves a lower computational complexity than other techniques, which require solving an n-dimensional ODE system p times for p parameters. In contrast, the adjoint method solves an n-dimensional ODE forwards and then solves an n-dimensional and a p-dimensional system in reverse, changing the computational complexity from $\mathcal{O}(np)$ to $\mathcal{O}(n + p)$. Whether such asymptotic cost advantages lead to more efficiency on practical models is precisely one of the points studied in this paper.

Alternatively, one can find the partial derivatives using finite differences. The simplest method here is to compute a slightly perturbed trajectory $\mathbf{x}(t, \boldsymbol{\beta} + \Delta \mathbf{v})$ and form the forward differences

$$\frac{\mathbf{x}(t, \boldsymbol{\beta} + \Delta \mathbf{v}) - \mathbf{x}(t, \boldsymbol{\beta})}{\Delta}$$

at all specified time points as approximations to the forward directional derivatives of $\mathbf{x}(t, \boldsymbol{\beta})$ in the direction $\mathbf{v}$. Choosing $\mathbf{v}$ to be unit vectors along each coordinate axis gives ordinary partial derivatives. The accuracy of this crude method suffers from round-off error in subtracting two nearly equal function values. These round-off errors are in addition to the usual errors committed in integrating the differential equation numerically. Round-off errors can be ameliorated by using central differences

$$\frac{\mathbf{x}\left(t, \boldsymbol{\beta} + \frac{\Delta}{2}\mathbf{v}\right) - \mathbf{x}\left(t, \boldsymbol{\beta} - \frac{\Delta}{2}\mathbf{v}\right)}{\Delta}$$

rather than forward differences. However, the central difference method requires twice the number of computations as the forward difference method. Thus, the choice of a difference method depends on prioritization of accuracy versus computational efficiency. In small models, computational efficiency may be less of a priority, in which case central difference methods are preferred.

2.2.3 Complex Perturbation Methods

There is a far more accurate way of computing model sensitivity when the function $f[\mathbf{x}, \boldsymbol{\beta}]$ defining the ODE is analytic in the parameter vector $\boldsymbol{\beta}$ [15]. An analytic function can be expanded in a locally convergent power series around every point of its domain. This implies that the trajectory $\mathbf{x}(t, \boldsymbol{\beta})$ is also analytic in $\boldsymbol{\beta}$. For a real analytic function $g(\beta)$ of a single variable β, the derivative approximation

$$g'(\beta) \quad = \quad \frac{\text{Imag } g(\beta + \Delta i)}{\Delta} + O(\Delta^2)$$

in the complex plane avoids roundoff and is highly accurate for $\Delta > 0$ very small [16, 17]. Thus, in calculating a directional derivative of $\mathbf{x}(t, \boldsymbol{\beta})$, it suffices to (a) solve the governing ODE $\frac{d}{dt}\mathbf{x}(t, \boldsymbol{\beta}) = f[\mathbf{x}(t), \boldsymbol{\beta}]$ with $\boldsymbol{\beta} + \Delta i\mathbf{v}$ replacing $\boldsymbol{\beta}$, (b) take the imaginary part of the result, and (c) divide by Δ. To make these calculations feasible, the computer language implementing the calculations should support complex arithmetic and ideally have an automatic dispatching mechanism so that only one implementation of each function is required. In contrast to numerical integration of the joint system (equation 2.1), the complex perturbation method is much more simply parallelizable across parameters.

The following straightforward Julia routine for computing sensitivities

```
function differential(f::F, p, $\Delta$) where F
    fvalue = real(f(p)) # function value
    df = zeros(length(fvalue), length(p)) # states $\times$ parameters
    pworker = [map(complex, p) for _ in 1:Threads.nthreads()]
    Threads.@threads for j = 1:length(p)
        _p = pworker[Threads.threadid()] # thread worker array
        _p[j] = _p[j] + $\Delta$ * im # perturb parameter
        fj = f(_p) # compute perturbed function value
        _p[j] = complex(real(_p[j]), 0.0) # reset parameter
        df[:,j] .= imag(fj) ./ $\Delta$ # fill in jth partial
    end
```

takes advantage of the simplicity of multithreading the complex perturbation method by parameter. This function requires a function $f(\mathbf{p})\colon \mathbb{R}^n \mapsto \mathbb{R}^m$ of a real vector $\mathbf{p}$ declared as complex. The perturbation scalar Δ should be small and real, say 10^{-10} to 10^{-12} in double precision. If the parameters p_j vary widely in magnitude, then a good heuristic is to perturb p_j by $p_j di$ instead of di. The returned value df is an $m \times n$ real matrix. The Julia commands *real* and *complex* effect conversions between real and complex numbers, and Julia substitutes \$\rm im\$ for $i = \sqrt{-1}$. We will employ these functions later in some case studies.

A recent extension[2.18] of the complex perturbation method facilitates accurate approximation of second derivatives. The relevant formula is

Equation 2.2:

$$\frac{\partial^2}{\partial \beta_j^2} g(\boldsymbol{\beta}) \quad = \quad \frac{\text{Imag}\left[g\left(\boldsymbol{\beta} + e^{\pi i/4}\Delta \mathbf{e}_j\right) + g\left(\boldsymbol{\beta} - e^{\pi i/4}\Delta \mathbf{e}_j\right)\right]}{\Delta^2} + O(\Delta^4),$$

where $e^{\pi i/4} = (1+i)/\sqrt{2}$. Roundoff errors can now occur but are usually manageable. Here we present a novel result for how to extend the complex perturbation method to approximate mixed partials. Our derivation is condensed into the following equations

$$
\begin{aligned}
\Delta g\left[\mathbf{x}+e^{\pi i/4}\left(\mathbf{e}_j+\mathbf{e}_k\right)\right] &\approx g(\mathbf{x})+e^{\pi i/4} dg(\mathbf{x})\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right) \\
&\quad +\frac{i}{2}\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right)^{\mathsf{T}} d^2 g(\mathbf{x})\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right) \\
&\quad +\frac{e^{\pi 3/4}}{6} d^3 g\left[\mathbf{x};\Delta^3\left(\mathbf{e}_j+\mathbf{e}_k\right)^3\right] \\
g\left[\mathbf{x}-e^{\pi i/4}\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right)\right] &\approx g(\mathbf{x})-e^{\pi i/4} dg(\mathbf{x})\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right) \\
&\quad +\frac{i}{2}\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right)^{\mathsf{T}} d^2 g(\mathbf{x})\Delta\left(\mathbf{e}_j+\mathbf{e}_k\right) \\
&\quad -\frac{e^{\pi 3/4}}{6} d^3 g\left[\mathbf{x};\Delta^3\left(\mathbf{e}_j+\mathbf{e}_k\right)^3\right].
\end{aligned}
$$

This approximation is accurate to order $O(\Delta^6)$ and allows us to infer that

Equation 2.3:

$$
\begin{aligned}
\frac{\text{Imag } g\left[\boldsymbol{x}+e^{\pi i/4\Delta}\left(\boldsymbol{e_j}+\boldsymbol{e_k}\right)\right]+g\left[\boldsymbol{x}-e^{\pi i/4\Delta}\left(\boldsymbol{e_j}+\boldsymbol{e_k}\right)\right]}{\Delta^2} &= \\
\left(\boldsymbol{e_j}+\boldsymbol{e_k}\right)^{\mathsf{T}} d^2 g(\boldsymbol{x})\left(\boldsymbol{e_j}+\boldsymbol{e_k}\right)+O\left(\Delta^4\right) &= \\
\frac{\partial^2}{\partial\beta_j^2} g(\boldsymbol{\beta})+\frac{\partial^2}{\partial\beta_k^2} g(\boldsymbol{\beta})+2\frac{\partial^2}{\partial\beta_j\partial\beta_k} g(\boldsymbol{\beta})+O\left(\Delta^4\right)
\end{aligned}
$$

Since we can approximate $\frac{\partial^2}{\partial\beta_j^2} g(\boldsymbol{\beta})$ and $\frac{\partial^2}{\partial\beta_k^2} g(\boldsymbol{\beta})$, we can now approximate $\frac{\partial^2}{\partial\beta_j\,\partial\beta_k} g(\boldsymbol{\beta})$ to order $O(\Delta^4)$. These approximations are derived in the Section 2.8.1.

The Julia code for computing second derivatives

```
function hessian(f::F, p, $\Delta$) where F
  d2f = zeros(length(p), length(p)) # hessian
  dp = $\Delta$ * (1.0 + 1.0 * im) / sqrt(2)
  for j = 1:length(p) # compute diagonal entries of d2f
    p[j] = p[j] + dp
    fplus = f(p)
    p[j] = p[j] - 2 * dp
    fminus = f(p)
```

```
    p[j] = complex(real(p[j]), 0.0) # reset parameter
    d2f[j, j]  = imag(fplus + fminus) / $\Delta$^2
  end
  for j = 2:length(p) # compute off diagonal entries
    for k = 1:(j - 1)
      (p[j], p[k]) = (p[j] + dp, p[k] + dp)
      fplus = f(p)
      (p[j], p[k]) = (p[j] - 2 * dp, p[k] - 2 * dp)
      fminus = f(p)
      (p[j], p[k]) = (complex(real(p[j]), 0.0), complex(real(p[k]), 0.
0))
      d2f[j, k]  = imag(fplus + fminus) / $\Delta$^2
      d2f[j, k] = (d2f[j, k] - d2f[j, j] - d2f[k, k]) / 2
      d2f[k, j] = d2f[j, k]
    end
  end
  return d2f
end
```

operates on a scalar-valued function $f(u)$ of a real vector **p** declared as complex. The second-order complex perturbation method can also be multithreaded by parameter, provided the unmixed second partials are computed prior to the mixed ones. Because roundoff error is now a concern, the perturbation scalar Δ should be in the range 10^{-3} to 10^{-6} in double precision. The returned value d^2f is a symmetric matrix.

2.2.4 Automatic Differentiation

Another technique one can use to calculate the derivatives of model solutions is to differentiate the numerical algorithm that calculates the solution. This can be done with computational tools collectively known as automatic differentiation[2.19]. Forward mode automatic differentiation is performed by carrying forward Jacobian-vector products at each successive calculation. This is accomplished by defining higher-dimensional numbers, known as dual numbers[2.20], coupled to primitive functions $f(\mathbf{x})$ through the action

$$f(\mathbf{a} + \mathbf{b}\epsilon) \quad = \quad f(\mathbf{a}) + \epsilon df(\mathbf{a})\mathbf{b}.$$

Here ϵ is a dimensional marker, similar to the complex i, which is a two-dimensional number. For a composite function $f = f_2 \circ f_1$, the chain rule is $df(\mathbf{a})\mathbf{b} = df_2[f_1(\mathbf{a})]df_1(\mathbf{a})\mathbf{b}$. The ith column of the Jacobian appears in the expression $f(\mathbf{x} + \mathbf{e}_i\epsilon) = f(\mathbf{x}) + \epsilon\nabla_i f(\mathbf{x})$. Since computational algorithms can be interpreted as the composition of simpler functions, one need only define automatic differentiation on a small set of base cases (such as $+$, $*$, sin, and so forth, known as the primitives) and then apply the accepted rules in sequence to differentiate more elaborate functions. The ForwardDiff.jl package[2.20] in Julia accomplishes this by defining dispatches for such primitives on a dual number type and provides convenience functions for easily extracting common objects like gradients, Jacobians, and Hessians. Hessians are calculated by layering automatic differentiation twice on the same algorithm to effectively take the derivative of a derivative.

In this form, forward mode automatic differentiation shares many similarities to the complex perturbation methods described above without the requirement that the extension of $f(\mathbf{x})$ be complex analytic. At every stage of the calculation $f(\mathbf{x})$ must be differentiable, a weaker yet still restrictive assumption. Conveniently, automatic differentiation allows for arbitrarily many derivatives to be calculated simultaneously. By defining higher-dimensional dual numbers that act independently via

$$f\left(\mathbf{a} + \sum_i b_i\,\epsilon_i\right) \quad = \quad f(\mathbf{a}) + \sum \epsilon_i df(\mathbf{a})\mathbf{b}_i,$$

one can calculate entire Jacobians in a single function call $f(\mathbf{a} + \sum_i \mathbf{e}_i\,\epsilon_i)$. This use of higher-dimensional dual numbers is a practice known as chunking. Chunking reduces the number of

primal (non-derivative) calculations required for computing the Jacobian. Because the ForwardDiff.jl package uses chunking by default, we will investigate the extent to which this detail is applicable in biological models.

2.3 CASE STUDIES

We now explore applications of differential sensitivity to a few core models in oncology and epidemiology.

2.3.1 CARRGO Model

The CARRGO model[2.21] was designed to capture the tumor-immune dynamics of CAR T-cell therapy in glioma. The CARRGO model generalizes to other tumor cell-immune cell interactions. Its governing system of ODEs

$$\begin{aligned} \frac{dx}{dt} &= \rho x\left(1-\frac{y}{\gamma}\right)-\kappa_1 xy \\ \frac{dy}{dt} &= \kappa_2 xy-\theta y \end{aligned}$$

follows cancer cells x as prey and CAR T-Cells y as predators. This model captures Lotka-Volterra dynamics with logistic growth of the cancer cells. Our numerical experiments assume the parameter values and initial conditions

$$\begin{aligned} \kappa_1 &= 6\times10^{-9}/(\text{day}\times\text{cell}), \quad \kappa_2 = 3\times10^{-11}/(\text{day}\times\text{cell}), \\ \theta &= 1\times10^{-6}/\text{day}, \quad \rho = 6\times10^{-2}/\text{day}, \quad \gamma = 1\times10^{9}\ \text{cells}, \\ x_0 &= 1.25\times10^{4}\ \text{cells}, \quad y_0 = 6.25\times10^{2}\ \text{cells} \end{aligned}$$

suggested by Sahoo et al.[2.21].

A traditional sensitivity analysis hinges on solving the system of ODEs and displaying the solutions at a chosen future time across an interval or rectangle of parameter values. Figure 2.1 shows how $x(t)$ and $y(t)$ vary at $t = 1000$ days under joint changes of κ_1 and κ_2, where κ_1 is the rate at which cancer cells are destroyed in an interaction with an immune cell, and κ_2 is the rate at which immune cells are recruited after such an interaction. This type of analysis directly portrays how a change in one or two parameters impacts the outcome of the system. Surprisingly, the number of cancer cells $x(t)$ depends strongly on κ_2 but only weakly on κ_1. In contrast, the number of immune cells $y(t)$ depends comparably on both parameters, perhaps because the initial population of immune cells is much smaller than the initial population of cancer cells.

There are limitations to this type of sensitivity analysis. How many solution curves should be examined? What time is most informative in displaying system changes? Is it necessary to compute sensitivity over such a large range of parameters when the trends are so clear? These ambiguities cloud our understanding and require far more computing than is necessary. Differential sensitivity successfully addresses these concerns. Gradients of solutions immediately yield approximate solutions in a neighborhood of postulated parameter values. The relative importance of different parameters in determining species levels can be determined from inspection of the gradient. Furthermore, modern software easily delivers the gradient along entire solution trajectories. There is no need to solve for an entire bundle of neighboring solutions.

Differential assessment is far more efficient. The required calculations involve solving an expanded system of ordinary differential equations just once under the automatic differentiation method or solving the system once for each parameter under the complex perturbation method. Either way, the differential method is much less computationally intensive than the traditional method of solving the ODE system over an interval for each parameter or over a rectangle for each

pair of parameters. Here is our brief Julia code for computing sensitivity via the complex perturbation method.

```
using DifferentialEquations, Plots


function sensitivity(x0, p, d, tspan)
  problem = ODEProblem{true}(ODE, x0, tspan, p)
  sol = solve(problem, saveat = 1.0) # solve ODE
  (lp, ls, lx) = (length(p), length(sol), length(x0))
  solution = Dict{Int, Any}(i => zeros(ls, lp + 1) for i in 1:lx)
  for j = 1:lx # record solution for each species
    @views solution[j][:, 1] = sol[j, :]
  end
  for j = 1:lp
    p[j] = p[j] + d * im # perturb parameter
    problem = ODEProblem{true}(ODE, x0, tspan, p)
    sol = solve(problem, saveat = 1.0) # resolve ODE
    p[j] = complex(real(p[j]), 0.0) # reset parameter
    @views sol .= imag(sol) / d # compute partial
    for k = 1:lx # record partial for each species
      @views solution[k][:,j + 1] = sol[k, :]
    end
  end
  return solution
end

function ODE(dx, x, p, t) # CARRGO model
  dx[1] = p[4] * x[1] * (1 - x[1] / p[5]) - p[1] * x[1] * x[2]
  dx[2] = p[2]* x[1] * x[2] - p[3] * x[2]
end

p = complex([6.0e-9, 3.0e-11, 1.0e-6, 6.0e-2, 1.0e9]); # parameters
x0 = complex([1.25e4, 6.25e2]); # initial values
(d, tspan) = (1.0e-13, (0.0,1000.0)); # step size and time interval in
days
solution = sensitivity(x0, p, d, tspan); # find solution and partials
CARRGO1 = plot(solution[1][:, 1], label = "x1", xlabel= "days",
ylabel = "cancer cells x1", xlims = (tspan[1],tspan[2]))
```

```
CARRGO2 = plot(solution[1][:, 2], label = "d1x1", xlabel= "days",
ylabel = "p1 sensitivity", xlims = (tspan[1],tspan[2]))
```

In the Julia code the parameters κ_1, κ_2, θ, ρ, and γ and the variables x and y exist as components of the vector **p** and **x**, respectively. The two plot commands construct solution curves for cancer and its sensitivity to perturbations of κ_1.

Figure 2.2 reinforces the conclusions drawn from the heatmaps, but more clearly and quantitatively. Additionally, differential sensitivity allows for the assessment of the sensitivity over the course of time, rather than just at a single time or small set of times. For example, the sensitivity of x with respect to γ in this model exhibits both large positive and large negative values over the course of time. Measured as the difference in x caused by a difference in γ at our end-time $t = 1000$, these effects tend to cancel each other out and fail to communicate the impact of the parameter γ on x at intermediate times. In brief, the scaled sensitivity of cancer cells x is much more dependent on carrying capacity γ later in the simulation, while the model sensitivity to birth rate ρ is most pronounced around the earlier time $t = 200$.

2.3.2 Deterministic SIR Model

The deterministic SIR model follows the number of infectives $I(t)$, the number of susceptibles $S(t)$, and the number of recovereds $R(t)$ during an epidemic. These three subpopulations satisfy the ODE system

$$\begin{aligned} \frac{d}{dt}S &= -\eta I\frac{S}{N} \\ \frac{d}{dt}I &= \eta I\frac{S}{N} - \delta I \\ \frac{d}{dt}R &= \delta I, \end{aligned}$$

where η is the daily infection rate per encounter and δ is the daily rate of progression to immunity per person. For SARS-CoV-2, current estimates[2.22] of η range from 0.0012 to 0.48, and estimates of δ range from 0.0417 to 0.0588[2.23]. As an alternative to solving the extended set of differential equations, we again use the complex perturbation method to evaluate parameter sensitivities.

The following Julia code for the complex perturbation method reuses the generic sensitivity function from the CARRGO model example.

```
function ODE(dx, x, p, t) # Covid model
  N = 3.4e8 # US population size
  dx[1] = - p[1] * x[2] * x[1] / N
  dx[2] = p[1] * x[2] * x[1] / N - p[2] * x[2]
  dx[3] = p[2] * x[2]
end

p = complex([0.2, (0.0417 + 0.0588) / 2]); # parameters
x0 = complex([3.4e8, 100.0, 0.0]); # initial values
(d, tspan) = (1.0e-10, (0.0, 365.0)) # 365 days
solution = sensitivity(x0, p, d, tspan);
Covid = plot(solution[1][:, :], label = ["x1" "d1x1" "d2x1"],
        xlabel = "days", xlims = (tspan[1],tspan[2]))
```

Our parameter choices roughly capture measurements for the COVID-19 virus from early in the pandemic[2.22,2.23]. Figure 2.3 plots the susceptible curve and its sensitivities. In this case all three curves conveniently occur on comparable scales. Figure 2.3 captures not only the pronounced parameter sensitivity early in the pandemic but also the symmetry between the δ and η parameters.

2.3.3 Second-Order Expansions of Solution Trajectories

In predicting nearby solution trajectories, the second-order Taylor expansion

Equation 2.4:

$$f(\mathbf{x}+\mathbf{v}) \quad \approx \quad f(\mathbf{x}) + df(\mathbf{x})\mathbf{v} + \frac{1}{2}\mathbf{v}^t d^2 f(\mathbf{x})\mathbf{v}$$

improves accuracy over the first-order expansion

Equation 2.5:

$$f(\mathbf{x}+\mathbf{v}) \quad \approx \quad f(\mathbf{x}) + df(\mathbf{x})\mathbf{v}.$$

The improved accuracy achieved by including second-order terms often justifies their computation. The complex perturbation method permits straightforward computation of second derivatives via approximations (equation 2.2) and (equation 2.3). The DiffEqSensitivity.jl and ForwardDiff.jl packages implement both adjoint and forward difference methods for computing the second derivatives of differential equation systems. Figure 2.4 displays predicted trajectories for the SIR model using the complex perturbation method when all parameters p_i are replaced by $p_i(1 + U_i)$, where each U_i is chosen uniformly from $(-0.25,0.25)$. The figure vividly confirms the improvement in accuracy in passing from a first-order to a second-order approximation. More improvement becomes evident as the non-linearity of the solution trajectory increases.

For example, the middle panel of Figure 2.4 shows that the solution trajectory of infected individuals bends dramatically with a change in parameters. This behavior is much better reflected in the second-order prediction compared to the first-order prediction, which over-corrects at the peak. The Euclidean distance between the actual and predicted trajectories at the sampled time points is about 25.4 in the first-order case and only about 9.06 in the second-order case, a reduction of over 60% in prediction error. By contrast, the trajectory of the recovered individuals steadily increases in a much more linear fashion. The bottom panel of Figure 4 shows that the first-order

prediction now remains reasonably accurate over a substantial period. Even so, the discrepancy between the predicted solutions grows so that by day 100 the Euclidean distance between the first-order prediction and the actual trajectory exceeds 154, compared to about 34.0 for the second-order prediction. Thus, calculating second-order sensitivity is helpful in both highly non-linear systems and systems with long time scales.

2.3.4 Stochastic SIR Model

We now illustrate sensitivity calculations in the stochastic SIR model. This model postulates an original population of size n with i infectives and s susceptibles. The parameters δ and η again capture the rate of progression to immunity and the infection rate per encounter. Since extinction of the infectives is certain, we focus on the time to elimination of the infectives. It is also convenient to follow the vector (i, n), where $n = i + s$ is the sum of the number of infectives i plus the number of susceptibles s. The mean time t_{in} to elimination of all infectives satisfies the recurrence

Equation 2.6:

$$t_{in} = \frac{1}{i\delta + i\left(\frac{n-i}{N}\right)\eta} + \frac{i\delta}{i\delta + i\left(\frac{n-i}{N}\right)\eta} t_{i-1,n-1} + \frac{i\left(\frac{n-i}{N}\right)\eta}{i\delta + i\left(\frac{n-i}{N}\right)\eta} t_{i+1,n}$$

for $0 < i < n$ together with the boundary conditions

$$t_{ii} = \sum_{j=1}^{i} \frac{1}{j\delta} \quad \text{and} \quad t_{0n} = 0.$$

The expression for t_{ii} stems from adding the expected time for the $i \to i-1$ transition, plus the expected time $i-1 \to i-2$, and so forth. This system of equations can be solved recursively for $i = n, n-1, \ldots 0$ starting with $n = 1$. Once the values for a given n are available, n can be incremented, and a new round is initiated. Ultimately the target size $n = N$ is reached. Taking partial derivatives of the recurrence (equation 2.6) yields a new system of recurrences that can also be solved recursively in tandem with the original recurrence. The complex perturbation method is easier to implement and comparable in accuracy to the partial derivative method.

Another important index of the SIR process is the mean number of infectives m_{in} ever generated starting with i initial infectives and n total people. These expectations can be calculated via the recurrences

Equation 2.7:

$$m_{in} = \frac{i\delta}{i\delta + i\left(\frac{n-i}{N}\right)\eta}\left(m_{i-1,n-1} + 1\right) + \frac{i\left(\frac{n-i}{N}\right)\eta}{i\delta + i\left(\frac{n-i}{N}\right)\eta} m_{i+1,n}$$

for $0 < i < n$ together with the boundary conditions

$$m_{ii} = i \quad \text{and} \quad m_{0n} = 0.$$

One can compute the sensitivities of the m_{in} to parameter perturbations in the same way as the t_{in}. Here is the Julia code for the two means and their sensitivities via the complex perturbation method. Note how our earlier differential function plays a key role.

```
function SIRMeans(p)
    (delta, eta) = (p[1], p[2])
    M = zeros(typeof(p[1]),(N+1, N+1)) # mean matrix
    T = similar(M) # time to extinction matrix
```

```
    for n = 1:N # recurrence relations loop
        for j = 0:(n-1)
            i = n - j
            a = i * delta # immunity rate
            if i == n # initial conditions
                M[i+1, n+1] = i
                T[i+1, n+1] = T[i, i] + 1 / a
            else
                b = i * (n - i) * eta / N  # infection rate
                c = 1 / (a + b)
                M[i+1, n+1] = a * c * (M[i, n] + 1) + b * c * M[i+2, n
+1]
                T[i+1, n+1] = c * (1 + a * T[i, n] + b * T[i+2, n+1])
            end
        end
    end
    return [M[:, N+1]; T[:, N+1]]
end

p = complex([0.2, (0.0417 + 0.0588) / 2]); # delta and beta
(N, d) = (100, 1.0e-10);
@time (f, df) = differential(SIRMeans, p, d);
```

The left column of Figure 2.5 displays a heatmap of the expected total number of individuals infected and the right column displays a heatmap of the expected days to extinction of the infection process. Rows 2 and 3 show the sensitivities of these quantities to the η and δ parameters in the stochastic SIR model.

It is interesting to compare results from differential sensitivity to estimates from stochastic simulations. To see the difference in accuracy, we calculated the average number of individuals infected and the average time to extinction by stochastic simulation using the software package BioSimulator.jl[2.24]. Table 2.1 records the analytic and simulated means of these outcomes in the SIR model. As Table 2.1 indicates, the simulated means over $r = 100$ runs are roughly comparable to the analytic means, but the standard errors of the simulated means are large. Because the

standard errors decrease as $\frac{1}{\sqrt{r}}$, it is difficult to achieve much accuracy by simulation alone. In more complicated models, simulation is so computationally intensive and time consuming that it is nearly impossible to achieve accurate results. Of course, the analytic method is predicated on the existence of an exact solution or an algorithm for computing the same.

Parameter sensitivities inform our judgment in interesting and helpful ways. For example, derivatives of both the total number of infecteds and the time to extinction with respect to η are very small except in a narrow window of the η parameter. This suggests that we focus further simulations, sensitivity analysis, and possible interventions on the region of parameter space where η falls in these windows. Derivatives with respect to δ also depend mostly on η except at very small values of δ. These conclusions are harder to draw from noisy simulations alone.

2.3.5 Branching Processes

Branching process models offer another opportunity for checking the accuracy of sensitivity calculations. For simplicity we focus on birth-death-migration processes[2.25]. These are multi-type continuous-time processes[2.26,2.17] that can be used to model the early stages of an epidemic over a finite graph with n nodes, where nodes represent cities or countries. On node i we initiate a branching process with birth rate $\beta_i > 0$ and death rate $\delta_i > 0$. The migration rate from node i to node j is $\lambda_{ij} \geq 0$. All rates are per person, and each person is labeled by a node. Let $\lambda_i = \sum_{j \neq i} \lambda_{ij}$ be the sum of the migration rates emanating from node i. Given this notation, the mean infinitesimal generator of the process is the matrix

$$\mathbf{\Omega} = \begin{pmatrix} \beta_1 - \delta_1 - \lambda_1 & \lambda_{12} & \cdots & \lambda_{1,n-1} & \lambda_{1n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{n1} & \lambda_{n2} & \cdots & \lambda_{n,n-1} & \beta_n - \delta_n - \lambda_n \end{pmatrix}$$

The entries of the matrix $e^{t\mathbf{\Omega}} = \left[m_{ij}(t)\right]$ represent the expected number of people at node j at time t starting from a single person of type i at time 0. The process is irreducible when the pure migration process corresponding to the choice $\beta_i = \delta_i = 0$ for all i is irreducible. Equivalently, the process is irreducible when the graph representing transition probabilities is strongly connected. Henceforth, we assume the process is irreducible and let $\mathbf{\Gamma}$ denote the mean infinitesimal generator of the pure migration process. The process is subcritical, critical, or supercritical depending on whether the dominant eigenvalue ρ of $\mathbf{\Omega}$ is negative, zero, or positive.

To determine the local sensitivity of ρ to a parameter θ[2.26,2.27], suppose its left and right eigenvectors $\mathbf{v}$ and $\mathbf{w}$ are normalized so that $\mathbf{vw} = 1$. Differentiating the identity $\mathbf{\Omega w} = \rho\mathbf{w}$ with respect to θ yields

$$\left(\frac{\partial}{\partial\theta}\mathbf{\Omega}\right)\mathbf{w} + \mathbf{\Omega}\frac{\partial}{\partial\theta}\mathbf{w} \quad = \quad \left(\frac{\partial}{\partial\theta}\rho\right)\mathbf{w} + \rho\frac{\partial}{\partial\theta}\mathbf{w}.$$

If we multiply this by $\mathbf{v}$ on the left and invoke the identities $\mathbf{v\Omega} = \rho\mathbf{v}$ and $\mathbf{vw} = 1$ we find that

$$\frac{\partial}{\partial\theta}\rho \quad = \quad \mathbf{v}\left(\frac{\partial}{\partial\theta}\mathbf{\Omega}\right)\mathbf{w}.$$

Because $\frac{\partial}{\partial\delta_i}\mathbf{\Omega} = -\frac{\partial}{\partial\beta_i}\mathbf{\Omega}$, it follows that an increase in δ_i has the same impact on ρ as the same decrease in β_i. The sensitivity of $\mathbf{v}$ and $\mathbf{w}$ can be determined by an extension of this reasoning [28]. The extinction probabilities e_i of the birth-death-migration satisfy the system of algebraic equations

Equation 2.8:

$$e_i \quad = \quad \frac{\delta_i}{\beta_i + \delta_i + \lambda_i} + \frac{\beta_i}{\beta_i + \delta_i + \lambda_i}e_i^2 + \sum_{j\neq i}\frac{\lambda_{ij}}{\beta_i + \delta_i + \lambda_i}e_j$$

for all i. This is a special case of the vector extinction equation

$$\mathbf{e} \quad = \quad P(\mathbf{e}) \quad = \quad \begin{pmatrix} P_1(\mathbf{e}) \\ \vdots \\ P_n(\mathbf{e}) \end{pmatrix}$$

for a general branching process with offspring generating function $P_i(\mathbf{x})$ for a type i person[2.29]. For a subcritical or critical process, $\mathbf{e} = \mathbf{1}$. For a supercritical process all $e_i \in (0,1)$. Iteration is the simplest way to find $\mathbf{e}$. Starting from $\mathbf{e}_0 = \mathbf{0}$, the vector sequence $\mathbf{e}_n = P(\mathbf{e}_{n-1})$ satisfies

$$\mathbf{0} \quad \leq \quad \mathbf{e}_n \quad \leq \quad \mathbf{e}_{n+1} \quad \leq \quad \mathbf{e}$$

and converges to a solution of the extinction equations. Here all inequalities apply component-wise.

To find the differential[2.28] of the extinction vector $\mathbf{e}$ with respect to a vector $\boldsymbol{\theta}$ of parameters, we assume that the branching process is supercritical and resort to implicit differentiation of the equation $\mathbf{e}(\boldsymbol{\theta}) = P[\mathbf{e}(\boldsymbol{\theta}), \boldsymbol{\theta}]$. The chain rule gives

$$d_{\boldsymbol{\theta}}\mathbf{e} \quad = \quad d_{\mathbf{e}}P(\mathbf{e}, \boldsymbol{\theta}) d_{\boldsymbol{\theta}}\mathbf{e} + d_{\boldsymbol{\theta}}P(\mathbf{e}, \boldsymbol{\theta}).$$

This equation has the solution

Equation 2.9:

$$d_{\boldsymbol{\theta}}\mathbf{e} \quad = \quad [\mathbf{I}_n - d_{\mathbf{e}}P(\mathbf{e}, \boldsymbol{\theta})]^{-1} d_{\boldsymbol{\theta}}P(\mathbf{e}, \boldsymbol{\theta}).$$

The indicated inverse does, in fact, exist. Alternatively, one can compute an entire extinction curve $\mathbf{e}(t)$ whose component $e_i(t)$ supplies the probability of extinction before time t starting from a single person of type i. This task reduces to solving the ODE for $\frac{d}{dt}\mathbf{e}(t)$ by the methods previously discussed.

The following Julia code computes the sensitivities of the extinction probability for a two-node process by the complex perturbation method.

```
using LinearAlgebra

function extinction(p)
  types = Int(sqrt(1 + length(p)) - 1) # length(p) = 2 * types + types
^2
  (x, y) = (zeros(Complex, types), zeros(Complex, types))
  for i = 1:500 # functional iteration
    y = P(x, p)
    if norm(x - y) < 1.0e-16 break end
    x = copy(y)
  end
  return y
end

function P(x, p) # progeny generating function
  types = Int(sqrt(1 + length(p)) - 1) # length(p) = 2 * types + types
^2
  delta = p[1: types]
  beta = p[types + 1: 2 * types]
  lambda = reshape(p[2 * types + 1:end], (types, types))
  y = similar(x)
  t = delta[1] + beta[1] + lambda[1, 2]
  y[1] = (delta[1] + beta[1] * x[1]^2 + lambda[1, 2] * x[2]) / t
  t = delta[2] + beta[2] + lambda[2, 1]
  y[2] = (delta[2] + beta[2] * x[2]^2 + lambda[2, 1] * x[1]) / t
  return y
end

delta = complex([1.0, 1.75]); # death rates
beta = complex([1.5, 1.5]); # birth rates
lambda = complex([0.0 0.5; 1.0 0.0]); # migration rates
p = [delta; beta; vec(lambda)]; # package parameter vector
(types, d) = (2, 1.0e-10)
@time (e, de) = differential(extinction, p, d)
```

To adapt the code to a different branching process model, one simply supplies the appropriate progeny generating function and necessary parameters.

The average number a_{ij} of infected individuals of type j ultimately generated by a single initial infected individual of type i is also of interest. The matrix $\mathbf{A} = \left(a_{ij}\right)$ of these expectations can be calculated via the matrix equation

Equation 2.10:

$$\mathbf{A} \quad = \quad (\mathbf{I}_n - \mathbf{F})^{-1},$$

where $\mathbf{F}$ is the offspring matrix

$$\mathbf{F} \quad = \quad \begin{pmatrix} \frac{2\beta_1}{\beta_1 + \delta_1 + \lambda_1} & \frac{\lambda_{12}}{\beta_1 + \delta_1 + \lambda_1} & \cdots & \frac{\lambda_{1,n-1}}{\beta_1 + \delta_1 + \lambda_1} & \frac{\lambda_{1n}}{\beta_1 + \delta_1 + \lambda_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\lambda_{n1}}{\beta_n + \delta_n + \lambda_n} & \frac{\lambda_{n2}}{\beta_n + \delta_n + \lambda_n} & \cdots & \frac{\lambda_{n,n-1}}{\beta_n + \delta_n + \lambda_n} & \frac{2\beta_n}{\beta_n + \delta_n + \lambda_n} \end{pmatrix}.$$

One can determine the local sensitivity of the expected numbers of total descendants by differentiating the equation $\mathbf{A} = (\mathbf{I}_n - \mathbf{F})^{-1}$. The result

Equation 2.11:

$$d_{\boldsymbol{\theta}}\mathbf{A} \quad = \quad (\mathbf{I}_n - \mathbf{F})^{-1} d_{\boldsymbol{\theta}}\mathbf{F} (\mathbf{I} - \mathbf{F})^{-1},$$

depends on the sensitivity of the expected offspring matrix $\mathbf{F}$. Julia code for the complex perturbation method with two nodes follows.

```
function particles(p) # mean infected individuals generated
  types = Int(sqrt(1 + length(p)) - 1) # length(p) = 2 * types + types
^2
  delta = p[1: types]
```

```
  beta = p[types + 1: 2 * types]
  lambda = reshape(p[2 * types + 1:end], (types, types))
  F = complex(zeros(types, types))
  t = delta[1] + beta[1] + lambda[1, 2]
  (F[1, 1], F[1, 2]) = (2 * beta[1] / t, lambda[1, 2] / t)
  t = delta[2] + beta[2] + lambda[2, 1]
  (F[2, 1], F[2, 2]) = (lambda[2, 1] / t, 2 * beta[2] / t)
  A = vec(inv(I - F)) # return as vector
end

delta = complex([1.0, 1.75]); # death rates
beta = complex([1.5, 1.5]); # birth rates
lambda = complex([0.0 0.5; 1.0 0.0]); # migration rates
p = [delta; beta; vec(lambda)]; # package parameter vector
(types, d) = (2, 1.0e-10)
@time (A, dA) = differential(particles, p, d)
```

2.4 RESULTS

We now measure the accuracy, computational speed, and prediction error for adjoint, forward mode, and complex perturbation methods. To account for the variety of settings encountered by biologists, we include two additional ODE models in our comparisons. The ROBER model describes chemical reactions typical of enzymatic behavior[2.30] and furnishes an example of a stiff ODE system. More information on the ROBER model can be found in Section 2.8.2. To compare the three methods in a high-dimensional ODE model, we turn to the mammalian cell cycle (MCC) model. Our MCC model is a simplified version of the original MCC model constructed by Gerard and Goldbetor[2.31], as explained in more detail in Section 2.8.2. The model comprises 11 equations and 15 parameters and captures aspects of cell reproduction and cycling mediated by chemical signaling via cell-state dependent proteins such as tumor repressors, transcription factors, and other DNA replication checkpoints. The model relies on cell state as opposed to cell mass and nicely replicates sequential progression along the cell cycle.

2.4.1 Accuracy

It is important to understand how close computed differential sensitivities are to true differential sensitivities. Unfortunately, the latter are almost always unavailable for ODE models. For the stochastic SIR and branching process models, true sensitivities are well matched by the approximate sensitivities delivered by the complex perturbation methods, provided the complex perturbation is small enough[2.32]. As a proxy for comparison to true values in ODE models, one can compute the Euclidean distance between sensitivities delivered by the complex perturbation method and the methods relying on the chain rule. In general, we find that these distances are very small.

For the forward and adjoint sensitivities of non-stiff ODEs such as the SIR and CARRGO models, it is known that as one decreases the tolerance of the underlying ODE solver, the solution and its sensitivities converge to their true values[2.33]. To demonstrate that the same behavior occurs in our cases, we compute the sensitivities $\frac{\partial}{\partial \eta} S$ of the SIR model and $\frac{\partial}{\partial p_1} x_1$ of the ROBER model at $t = 1000$ using the adjoint, forward, and complex perturbation methods at a variety of tolerances ranging from 1×10^{-2} to 1×10^{-8}.

Figure 2.6 shows that all three method types (adjoint, forward, and complex perturbation) ultimately converge. In the non-stiff case (the SIR model), the adjoint method requires a step size of 1.0 to converge, while the stiff case (the ROBER model) requires a much smaller step size of 0.1 to converge. Each method converges at a different rate and potentially from a different direction. In the case of a relatively small, non-stiff model, the complex perturbation method converges more quickly (and at a higher tolerance) than the other methods. Notably, when the tolerance for the adjoint method is too weak the error rate increases more dramatically than for the

forward method. This behavior becomes even more pronounced if we consider a stiff ODE model such as ROBER. In this case it is worth noting that the forward and complex perturbation methods converge, albeit under a more stringent tolerance. The adjoint method however struggles to converge for the ROBER model unless the step size is decreased to 0.1 (shown in the figure). While the smaller step size does allow the adjoint method to converge even in the stiff case, this smaller step size is much more computationally intensive and, in many cases, may be infeasible.

2.4.2 The Speed versus Accuracy Trade-off

The trade-off between computational speed and accuracy is relevant to solving ODE systems whether they are stiff or not. Figure 2.7 displays the time versus error trade-off for both the SIR (non-stiff) and ROBER (stiff) models. In this case, error is calculated as the Euclidean distance between the derivatives calculated at various error tolerances and the derivatives calculated at a strict tolerance of 1×10^{-8} (for the SIR model) and 1×10^{-5} (for the ROBER model). We chose these tolerances as the strictest possible that are numerically realistic for each model. Figure 2.6 demonstrates that our choices are strict enough for the methods to reach convergence. We display errors versus time in a log-log plot averaged over compartments and parameters and normalized by length of time. We do not include the adjoint method in this comparison due to its difficulties in convergence and large computational cost.

Figure 2.7 figure demonstrates the clear trade-off between speed and accuracy in both the stiff (ROBER) and non-stiff (SIR) cases. In both cases, the forward method can be computed more quickly for equal errors than the complex perturbation method. As expected, the ROBER model has a less steep slope compared to the SIR model, indicating that the returns in accuracy grow more slowly per time invested for a stiff ODE system.

2.4.3 Computational Speed

Speed is an important attribute of any computational method, especially when it is performed without the benefit of computational clusters or distributed computing resources. Our speed comparisons offer a first look at the efficiency gains possible with multithreading. In implementing multithreading for both the complex perturbation and forward mode methods, we call the Polyester.jl package to compute each partial derivative in a separate thread. All computations were done in Julia version 1.7.1 on a Windows operating system with an Intel Core i7-8565U CPU.

In addition to multithreading, the forward method as implemented in ForwardDiff.jl package provides the capability of multichunking. This involves splitting the equations in each system into different chunks to be solved separately. While forward methods do benefit from chunking, this tactic is unavailable in many packages outside of ForwardDiff.jl or outside of the Julia language. For biologists who depend on other packages and computer languages, it may be more pertinent to focus on the non-chunked results for the forward method.

Table 2.2 records the computational speed of the complex perturbation, forward, and adjoint methods (and their multithreaded and multi-chunked versions, as applicable) for four ODE systems models (SIR, CARRGO, ROBER, and MCC). Our comparisons of the first-order methods show that the forward and complex perturbation methods perform comparably, while the adjoint method performs orders of magnitude slower than the other two. The fastest method is the multichunked forward method, with the complex perturbation method a close second for the simpler ODE systems such as SIR and CARRGO. For the stiff (ROBER) and large (MCC) models however, the complex perturbation method falls further behind the multichunk forward mode method. This could be expected from the larger gap between the time versus accuracy curves in

the ROBER model as compared with the SIR model and illustrated in Figure 2.6. However, it is noteworthy that naive implementations of forward mode differentiation lack the advantage of chunking and are consequently slower than the complex perturbation method.

The adjoint method also has the worst time performance of the second-order methods by orders of magnitude. Both the forward and complex perturbation methods performed well in all four ODE systems models, with the complex perturbation method performing particularly well in models where the number of parameters is not large compared to the number of equations.

While multi-threading usually decreases computational time for both first-order and second-order methods, it does not decrease computational time by as wide of a margin as expected. Many of the solver methods for stiff ODEs rely on BLAS operations that are already internally optimized by running on multiple threads. Explicitly multi-threading sensitivity methods therefore restricts the number of threads available for BLAS operations, adversely affecting their performance. In addition to the reduced efficiency of BLAS operations, multi-threading incurs a start-up cost for each thread. These start-up costs may overshadow the benefits of multi-threading if the amount of computation per thread is not high enough. Multi-threaded methods require more allocations than other methods, and thus require more garbage collection. While time spent on garbage collection varies, we find that garbage collection can take over twice as much computational time in multi-threaded methods than in their single-threaded counterparts. Thus, multi-threading can only really start to improve computational efficiency when these additional costs are small compared to the cost of each computation. Multi-threading may even be less efficient in some cases.

Table 2.3 compares the computational speeds of the different methods for the stochastic SIR and branching process models. As expected for the stochastic SIR model, computational speed

varies roughly quadratically with the number N of individuals in the system. In the stochastic SIR model, the complex perturbation method proves to be twice as fast as the manual differentiation of (equation 2.10) and (equation 2.8) because the latter requires a larger number of individual computations. For the branching process model however, this trend reverses since manual differentiation relies on fast linear algebra rather than iteration and avoids the overhead of complex arithmetic. The derivatives of $\mathbf{A}$ are matrix equations, and in this case forward mode differentiation even without chunking performs as well as the complex perturbation method, although it does not scale as well to larger systems ($N = 1000$). However, in the case of the derivatives of $\mathbf{e}$, which are calculated using recursion, neither implementation of forward mode differentiation can be computed as quickly as the complex perturbation method, and this difference increases with the size of the system. Other evidence not shown suggests that the complex perturbation method can reliably evaluate sensitivities where solutions depend on linear algebra and/or recurrence relations. In summary, unless derivatives are quite complicated, manual differentiation is generally more computationally efficient than either the complex perturbation method or the forward method. In computing second derivatives, we expect the tables will be turned. To their credit, the forward and complex perturbation methods do not require formulating derivatives analytically in advance and are consequentially easier to implement.

2.4.4 Prediction Error

In general, prediction error measures how well the first and second-order sensitivities capture the change in behavior of a model. Since we have previously shown that the various methods for computing differential sensitivity yield nearly the same results, prediction error is a good metric for determining the value of differential sensitivity in a particular model. We measure prediction error by the Euclidean norms

$$\text{err}_1 = \| f(\mathbf{x}+\mathbf{v}) - f(\mathbf{x}) - df(\mathbf{x})\mathbf{v} \|$$
$$\text{err}_2 = \| f(\mathbf{x}+\mathbf{v}) - f(\mathbf{x}) - df(\mathbf{x})\mathbf{v} - \frac{1}{2}\mathbf{v}^t d^2 f(\mathbf{x})\mathbf{v} \|.$$

Other norms, such as the ℓ_1 and ℓ_∞ norms, yield similar results. In the ODE models, $f(\mathbf{x})$ denotes a matrix trajectory so the Frobenius norm applies. To capture proportional prediction errors, we normalize all vector outputs by their length and all matrix outputs by the square of their length.

Prediction accuracy varies widely between models and even between parameters. As we expect however, second-order approximations are more accurate in prediction. Table 2.4 records prediction errors for each model. For the ODE systems, we see that stiffness highlights the added value of the second-order approximations. In the ROBER and CARRGO models, the second-order approximations have an order of magnitude less prediction error than the first-order approximations. However, stiffness does not appear to impact how the prediction errors grow over time. The ROBER and MCC models do not suffer from increased errors per time point after longer prediction intervals.

In the stochastic SIR model, prediction error does not seem to be compounded at all; in fact, the error per value calculated decreases in the case of $\frac{\partial M}{\partial \eta}$. In the case of branching processes with many types N and large parameter sets, it is inadvisable to compare prediction accuracy across system sizes. However, we can conclude from these results that at least the prediction error does not compound as N increases. Furthermore, prediction accuracy for the branching process models appears to vary dramatically depending on the parameter in question.

2.5 DISCUSSION

Our purpose throughout has been to demonstrate the ease and utility of incorporating differential sensitivity analysis in dynamical modeling. Because models are always approximate, and parameters are measured imprecisely, uncertainty plagues virtually all dynamical models. While improving models is incremental and domain specific, sensitivity analysis provides a handle on local parameter uncertainty across models.

Of the methods mentioned in this text, the adjoint method, forward method, and complex perturbation methods all require that the functions defining a model be differentiable in the underlying parameters. While the complex perturbation method has the additional requirement that these functions be complex analytic, it is the only method explored in this manuscript that can be extended to discrete stochastic models in addition to ODE systems. For the modeler who prefers a one size fits all approach, or who prefers to prioritize ease of implementation, we argue that the complex perturbation method should be the method of choice. In addition to its wide range of applicability, the complex perturbation method can be easily multi-threaded and requires only implementation of the component functions of the model. In contrast to the second-order complex perturbation method, forward differentiation slows dramatically in calculating a Hessian directly. It becomes competitive if one calculates the gradient of the gradient. The gradient of the gradient method is not always available natively and usually must be implemented separately as we have done in the current manuscript. Crucially, implementing a specialized forward mode method was possible due to the underlying automatic differentiation software's flexibility and support for composition.

In situations demanding computational speed, our results suggest that choosing a method tailored to a model may be pertinent. In the case of stochastic models, manually differentiating and

applying the chain rule must be balanced against the complex perturbation method, which requires less effort up front but longer processing after the derivatives have been determined. For ODE systems models, forward mode is the most computationally efficient when multichunking is available. If multichunking is not available, then the complex perturbation method has comparable speed to the forward method when run with the same tolerance. In maximizing computational efficiency, it is important to note that the use of automatic differentiation tools may require more user input for algorithm selection or multi-threading implementation. Choice of software is critical as well; not all software packages with automatic forward differentiation support chunking as implemented in the ForwardDiff.jl package and that so dramatically improves the computational efficiency of this method.

There are additional challenges to computing model sensitivity that we do not address. For example, not all models use functions that are differentiable in their parameters. Additionally, models may be differentiable yet extremely stiff, in which case the computational time for each sensitivity method discussed here will suffer disproportionally as the number of parameters grows. Furthermore, assessing global parameter sensitivity is more challenging. It can be attacked by techniques such as Latin square hypercube sampling or Sobel quasi-random sampling, but these become infeasible in high dimensions[2.34]. Given the availability of appropriate software, differential sensitivity is computationally feasible, even for high-dimensional systems.

In the case of stochastic models, traditional methods require costly and inaccurate simulation over a bundle of parameter values. Differential sensitivity is often out of the question. Current automatic differentiation systems, such as PyTorch, Zygote and ForwardDiff, treat generated random numbers as constants, and thus are not reliable methods for use in calculating differential sensitivity of model outcomes that depend on these random variables. This limits the

ability of researchers to understand a biological system and how it responds to parameter changes. If a system index such as a mean, variance, extinction probability, or extinction time can be computed by a reasonable algorithm, then differential parameter sensitivity analysis can be undertaken. We have indicated in a handful of examples how this can be accomplished.

In summary, across many models representative of computational biology, we have reached the following conclusions:

a) Forward mode, adjoint, and complex perturbation sensitivity methods all converge to the same differential sensitivity values in non-stiff models, thus offering the same level of accuracy for all methods. For stiff models, forward mode and complex perturbation methods converge but adjoint sensitivity struggles and does not achieve convergence for realistic tolerance parameters.

b) Chunked forward mode automatic differentiation and forward mode sensitivity analysis tend to be the most computationally efficient on the tested models.

c) The complex perturbation methods described in this manuscript are competitive and often outperform the unchunked version of forward mode automatic differentiation, while being less sensitive to stiffness than the adjoint methods.

d) Shared memory multi-threading of the complex perturbation and forward mode automatic differentiation methods provides a performance gain but only in high-dimensional systems.

e) Forward mode automatic differentiation method requires that each step of a calculation is differentiable. This renders it unusable for calculating the derivative of ensemble means of

discrete state models, such as birth-death processes. For these cases, the complex perturbation method outperforms manual differentiation.

f) The complex perturbation method is competitive with automatic differentiation methods in accuracy, is more straightforward to implement, and can be applied to a wider variety of methods.

These conclusions are tentative but supported by our limited number of biological case studies.

We note that the performance differences may change depending on the efficiency of the implementations. The Julia DifferentialEquations.jl library and its DiffEqSensitivity.jl package have been shown to be highly efficient, outperforming other libraries in both equation solving and derivative calculations in Python, MATLAB, C, and Fortran[2.19,2.33].

The automatic differentiation implementations in machine learning libraries optimize array operations much more than scalar operations. This can work to the detriment of forward mode AD. MATLAB or Python style vectorization improves the performance of forward mode AD sensitivity analysis by reducing interpreter overhead. Therefore, our conclusions serve as guidelines for the case where all implementations are well-optimized. For programming languages with high overheads or without compile-time optimization of the automatic differentiation passes, the balance in efficiency shifts more favorably towards the complex perturbation method.

One last point worth making is on the coding effort required by the various methods. Both automatic differentiation and the complex perturbation method have comparable accuracy when applied to systems of ODEs, with automatic differentiation having the advantage in speed when it is implemented with the additional level of parallelization provided by chunking. However, the complex perturbation method can easily be generalized to other kinds of objective functions and

may be more straightforward to implement for those less sophisticated in computer science. While automatic differentiation is the basis of many large scientific packages, the code required for the complex perturbation methods is fully contained within this manuscript and is easily transferable to other programming languages with similar dispatching on complex numbers. This hard to measure benefit should not be ignored by practicing biologists who simply wish to quickly arrive at reasonably fast code.

2.6 TABLES

Table 2.1: **SIR model outcomes**

	Calculated Mean	Simulated Mean	Simulated Standard Error
Time to Extinction	2.792×10 days	3.074×10 days	4.153 days
Number Infected	5.484×10^3 people	5.838×10^3 people	8.551×10^2 people

Comparison between the calculated and simulated means of SIR model outcomes in the stochastic SIR model simulated under the initial conditions $S_0 = 3.4 \times 10^4, I_0 = 1$ and parameter values $\eta = 0.7194, \delta = .5025$. Results for the simulated means were obtained using the BioSimulator package in Julia and averaging results over $r = 100$ runs.

Table 2.2: **ODE model computational time (μs)**

SIR model

First-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	2.252×10^2	1.688×10^3	1.377×10^4
Complex Perturbation Multithread	1.913×10^2	1.401×10^3	1.062×10^4
Forward	3.272×10^2	2.036×10^3	1.460×10^4

Forward Multithread	2.218×10^2	1.480×10^3	1.117×10^4
Forward Multichunk	1.567×10^2	9.564×10^2	7.247×10^3
Forward Multichunk Multithread	1.499×10^2	9.526×10^2	7.236×10^3
Adjoint	8.901×10^4	7.707×10^6	6.950×10^8

Second-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	7.885×10^2	5.712×10^3	5.806×10^4
Complex Perturbation Multithread	6.732×10^2	4.528×10^3	3.724×10^4
Forward	9.325×10^2	5.280×10^3	4.530×10^4
Forward Multithread	7.546×10^2	3.504×10^3	2.640×10^4
Forward Multichunk	1.742×10^2	7.601×10^2	4.541×10^3
Forward Multichunk Multithread	1.714×10^2	7.270×10^2	4.631×10^3
Adjoint	2.976×10^4	6.240×10^5	1.626×10^7

CARRGO model

First-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	3.977×10^2	2.195×10^3	2.332×10^4
Complex Perturbation Multithread	3.661×10^2	2.480×10^3	2.330×10^4
Forward	5.404×10^2	2.597×10^3	2.505×10^4
Forward Multithread	4.527×10^2	2.601×10^3	2.336×10^4
Forward Multichunk	3.759×10^2	1.661×10^3	1.417×10^4
Forward Multichunk Multithread	2.699×10^2	1.352×10^3	1.215×10^4
Adjoint	6.118×10^4	5.097×10^6	7.825×10^8

Second-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	2.039×10^3	1.245×10^4	1.469×10^5
Complex Perturbation Multithread	2.123×10^3	1.206×10^4	1.573×10^5
Forward	2.749×10^3	1.239×10^4	1.376×10^5
Forward Multithread	1.737×10^3	1.011×10^4	1.735×10^5
Forward Multichunk	1.097×10^3	4.475×10^3	5.382×10^4
Forward Multichunk Multithread	7.135×10^2	3.181×10^3	3.967×10^4
Adjoint	2.048×10^4	2.795×10^5	7.536×10^6

ROBER model

First-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	2.475×10^3	4.111×10^3	4.117×10^3
Complex Perturbation Multithread	1.549×10^3	2.600×10^3	5.016×10^3
Forward	3.029×10^3	4.544×10^3	8.271×10^3
Forward Multithread	1.726×10^3	2.905×10^3	4.766×10^3
Forward Multichunk	1.471×10^3	2.422×10^3	4.113×10^3
Forward Multichunk Multithread	1.343×10^3	2.442×10^3	3.902×10^3
Adjoint	1.456×10^8	2.656×10^9	2.069×10^{10}

Second-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	7.985×10^3	1.250×10^4	2.306×10^4
Complex Perturbation Multithread	5.157×10^3	8.868×10^3	1.763×10^4
Forward	7.422×10^3	1.101×10^4	2.291×10^4
Forward Multithread	4.062×10^3	6.131×10^3	1.403×10^4

Forward Multichunk	1.420×10^3	2.157×10^3	3.655×10^3
Forward Multichunk Multithread	1.439×10^3	2.159×10^3	3.552×10^3
Adjoint	3.669×10^7	7.388×10^8	–

Mammalian cell cycle model

First-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	2.952×10^3	2.588×10^4	8.50×10^4
Complex Perturbation Multithread	1.806×10^3	1.521×10^4	4.612×10^4
Forward	2.758×10^3	1.527×10^4	7.741×10^4
Forward Multithread	2.147×10^3	1.524×10^4	4.646×10^4
Forward Multichunk	1.071×10^3	6.806×10^4	1.709×10^4
Forward Multichunk Multithread	8.038×10^2	5.494×10^3	1.325×10^4
Adjoint	3.601×10^5	3.029×10^7	3.332×10^9

Second-order Methods	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
Complex Perturbation	3.336×10^4	4.457×10^5	1.262×10^6
Complex Perturbation Multithread	3.969×10^4	2.969×10^4	1.198×10^6
Forward	6.331×10^4	5.213×10^5	1.383×10^6
Forward Multithread	3.465×10^4	3.445×10^5	1.116×10^6
Forward Multichunk	2.257×10^4	1.392×10^5	2.886×10^5
Forward Multichunk Multithread	1.544×10^4	8.824×10^4	2.007×10^5
Adjoint	6.589×10^5	2.041×10^7	7.388×10^8

Parameters for the ODE models match those previously introduced in this manuscript. Multithread refers to parallelism across parameters. Multichunk refers to parallelism across compartments. We

invoke the Julia solvers AutoVern9(Rodas5(autodiff=false)) with a tolerance of 1×10^{-5} for the nonstiff (SIR, CARRGO, and MCC) models and Rodas4(autodiff=false) with a tolerance of 1×10^{-7} for the stiff (ROBER) model. These tolerances reflect the convergence tolerances. Continuation of computational time (μs) in ODE models. Second-order adjoint method not included for the ROBER model at t=1000 due to time constraints. For the second-order adjoint method, the ForwardDiffOverAdjoint(QuadratureAdjoint(autodiff=false) solver option was used.

Table 2.3: **Stochastic model computation time (μs)**

SIR model

$\partial M / \partial \delta$	$N = 10$	$N = 100$	$N = 1000$
Complex Perturbation	1.90×10^{1}	1.634×10^{3}	1.975×10^{5}
Manual Differentiation	3.80×10^{1}	3.879×10^{3}	4.925×10^{5}

$\partial T / \partial \delta$	$N = 10$	$N = 100$	$N = 1000$
Complex Perturbation	1.86×10^{1}	1.620×10^{3}	2.006×10^{5}
Manual Differentiation	3.45×10^{1}	4.213×10^{3}	4.875×10^{5}

Branching process model

$\partial A / \partial \delta_1$	$N = 10$	$N = 100$	$N = 1000$
Complex Perturbation	2.43×10^{3}	2.33×10^{5}	1.36×10^{8}
Manual Differentiation	1.08×10^{1}	3.75×10^{4}	4.97×10^{5}
Forward	1.79×10^{2}	2.96×10^{5}	–
Forward Multichunk	2.85×10^{1}	1.32×10^{5}	1.39×10^{9}

$\partial e / \partial \delta_1$	$N = 10$	$N = 100$	$N = 1000$
Complex Perturbation	1.04×10^3	3.44×10^4	4.25×10^6
Manual Differentiation	4.26×10^2	4.90×10^4	3.19×10^6
Forward	1.03×10^4	1.33×10^6	–
Forward Multichunk	1.23×10^3	1.12×10^6	2.27×10^9

Model parameters for stochastic SIR match those previously described in this manuscript. Parameters for the branching process model are generated randomly on the range $\beta \in [0.05,0.16]$, $\delta \in [0.05,0.19]$, and $\lambda \in [0.0003,0.00046]$. Manual differentiation relies on differentiating equations (eq.7) and (eq. 6) for the stochastic SIR model and equations (eq. 11) and (eq. 9) for the branching process model.

Table 2.4: **Prediction error results for ODE and stochastic models.**

ODE models

	$t_{end} = 10$	$t_{end} = 100$	$t_{end} = 1000$
SIR First Order	3.370×10^1	2.444×10^6	2.524×10^5
SIR Second Order	8.208×10^0	2.299×10^6	2.303×10^5
CARRGO First Order	6.195×10^{-1}	2.801×10^3	1.465×10^5
CARRGO Second Order	1.116×10^{-2}	4.956×10^2	4.217×10^4
ROBER First Order	3.205×10^{-5}	3.588×10^{-5}	1.837×10^{-5}
ROBER Second Order	1.753×10^{-6}	2.201×10^{-6}	1.039×10^{-6}
MCC First Order	3.467×10^{-4}	7.556×10^{-5}	1.542×10^{-4}
MCC Second Order	1.268×10^{-4}	1.922×10^{-5}	3.918×10^{-5}

Stochastic SIR model

	$N = 10$	$N = 100$	$N = 1000$
Total Number Infected (M) from η	2.322×10^{-3}	2.009×10^{-2}	1.241×10^{1}
Total Number Infected (M) from δ	4.456×10^{-3}	2.586×10^{-2}	3.670×10^{-2}
Time to Extinction (T) from η	1.601×10^{-3}	6.715×10^{-3}	4.046×10^{-3}
Time to Extinction (T) from δ	2.074×10^{-1}	1.599×10^{-1}	4.811×10^{-2}

Branching process model

	$N = 10$	$N = 100$	$N = 1000$
Total Number Infected (A) from β_1	3.025×10^{-2}	7.020×10^{-6}	7.234×10^{-9}
Total Number Infected (A) from δ_1	5.036×10^{-2}	8.734×10^{-5}	1.348×10^{-7}
Total Number Infected (A) from $\lambda_{1,1}$	2.402×10^{-4}	4.229×10^{-6}	4.152×10^{-8}
Extinction Probability (e) from β_1	1.119×10^{-4}	3.476×10^{-7}	7.257×10^{-10}
Extinction Probability (e) from δ_1	7.682×10^{-4}	3.776×10^{-6}	9.424×10^{-9}
Extinction Probability (e) from $\lambda_{1,1}$	5.123×10^{-5}	3.044×10^{-6}	5.800×10^{-8}

Stochastic derivatives are calculated with the complex perturbation method and ODE derivatives are calculated with the forward difference method. All predictions are for a 10% change in parameter. Parameters for the ODE and stochastic SIR models match those previously introduced in this manuscript. Parameters for the branching process model are generated randomly on the range β in $[0.05,0.16]$, λ in $[0.0003,0.00046]$, and $\delta = \beta + .03$ for calculation of a sub-critical system (A) and $\delta = \beta - .03$ for calculation of a super-critical system (e).

Table 2.S1: **Parameters in the MCC model**

Compartment	Initial Value (μ mol)	Parameter	Value
pRBc1	0.1	kpc1	0.05
pRBc2	0.05	kpc3	0.025
Cd	0.01	kcd1	0.4
Mdi	0.01	Ki8	2.0
Md	0.01	Ki7	0.1
pRB	0.0	kcd2	0.005
E2F	0.0	kdecom1	0.1
pRBp	0.0	k2d	0.1
AP1	0.0	$Cdk4_{tot}$	1.5
p27	0.0	kcom1	0.175
Mdp27	0.0	Vm2d	0.2
–	–	k1d	0.1
–	–	Vm1d	1.0
–	–	kc1	0.15

Table 2.S2: **Parameters in the ROBER model**

Compartment	Initial Value	Parameter	Value
x_1	1.0	p_1	4×10^{-2}
x_2	0.0	p_1	3×10^{7}
x_3	0.0	p_1	1×10^{4}

2.7 FIGURES

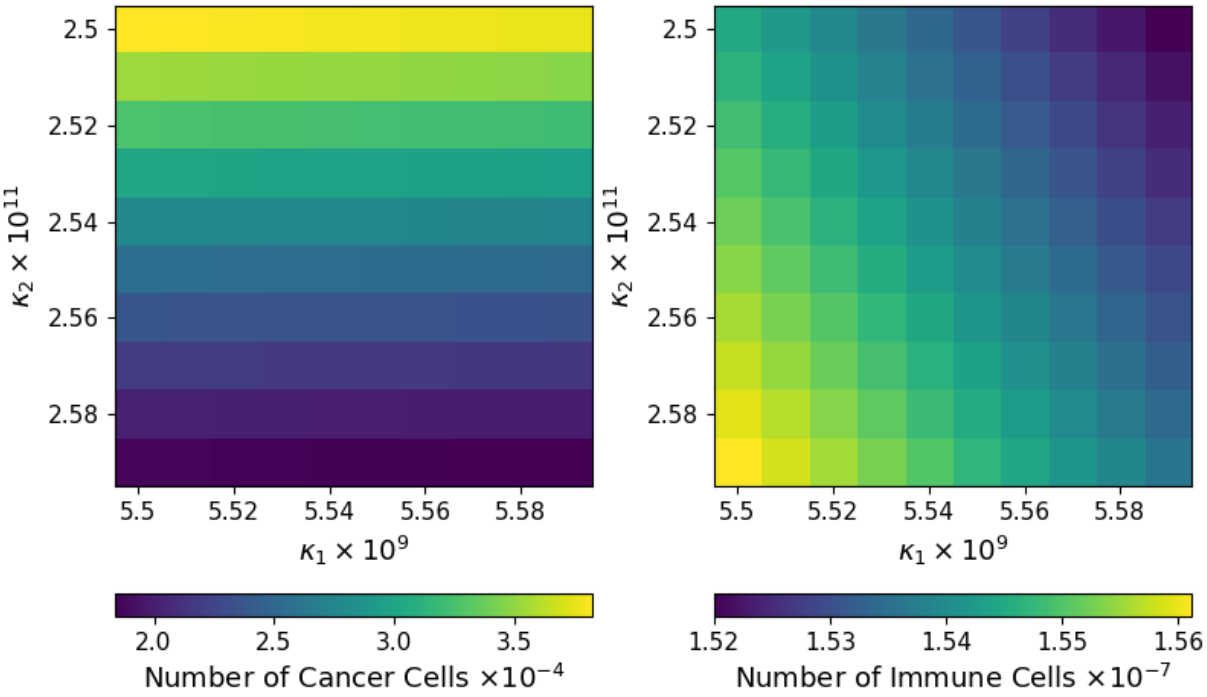

Figure 2.1: **Sensitivity of cancer and immune cells in the CARRGO model.** A heatmap representing the number of cancer cells, or $x(t)$ (left) and the number of immune cells, or $y(t)$ (right) as the parameters κ_1 (horizontal axis) and κ_2 (vertical axis) are varied. Results displayed summarize simulations of the CARRGO model with parameter values and initial conditions indicated in this section at time $t = 1000$ days.

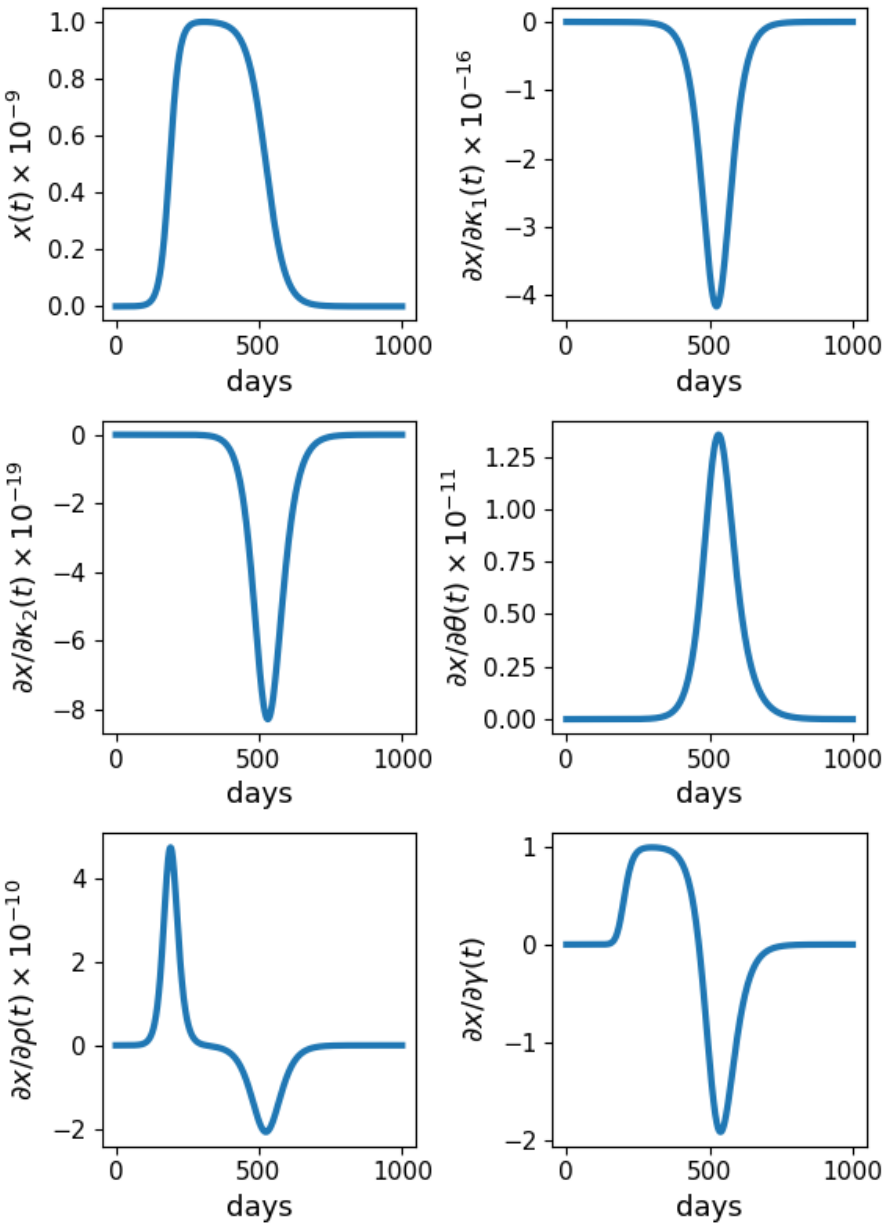

Figure 2.2: **Sensitivity of cancer cells in the CARRGO model.** Time series plots of cancer cells ($x(t)$) and the derivatives of $x(t)$ with respect to the CARRGO parameters $\kappa_1, \kappa_2, \theta, \rho, \gamma$. Results shown are for the initial conditions and parameter values defined in Figure 1 and simulated over the course of $t = 1000$ days. The complex perturbation method of sensitivity analysis is used to compute derivatives.

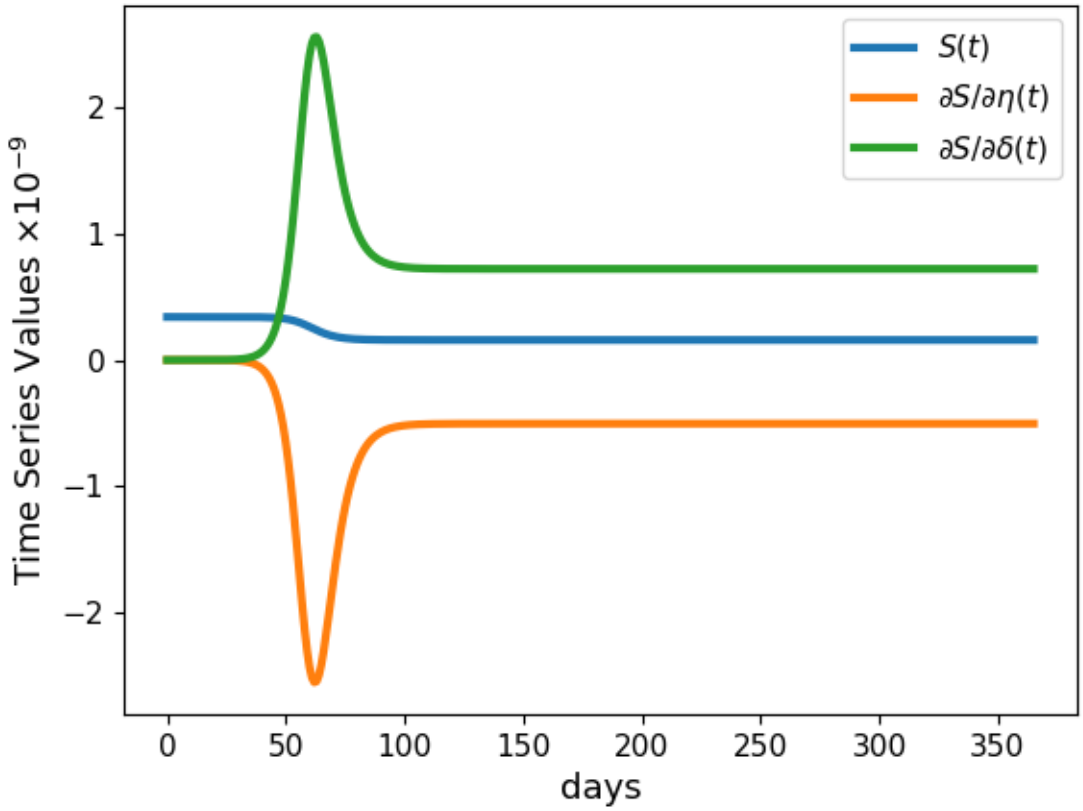

Figure 2.3: **Sensitivities of susceptibles in the Covid model.** Time series of the susceptible population ($S(t)$) and its sensitivities to the two parameters (η and δ) of the classic SIR model. Results shown are for the SIR model simulated for one year with initial conditions $S_0 = 3.4 \times 10^8$, $I_0 = 100$, $R_0 = 0$, and the parameter values $\eta = 0.7194$, $\delta = 0.5025$. Derivatives are calculated using the complex perturbation method.

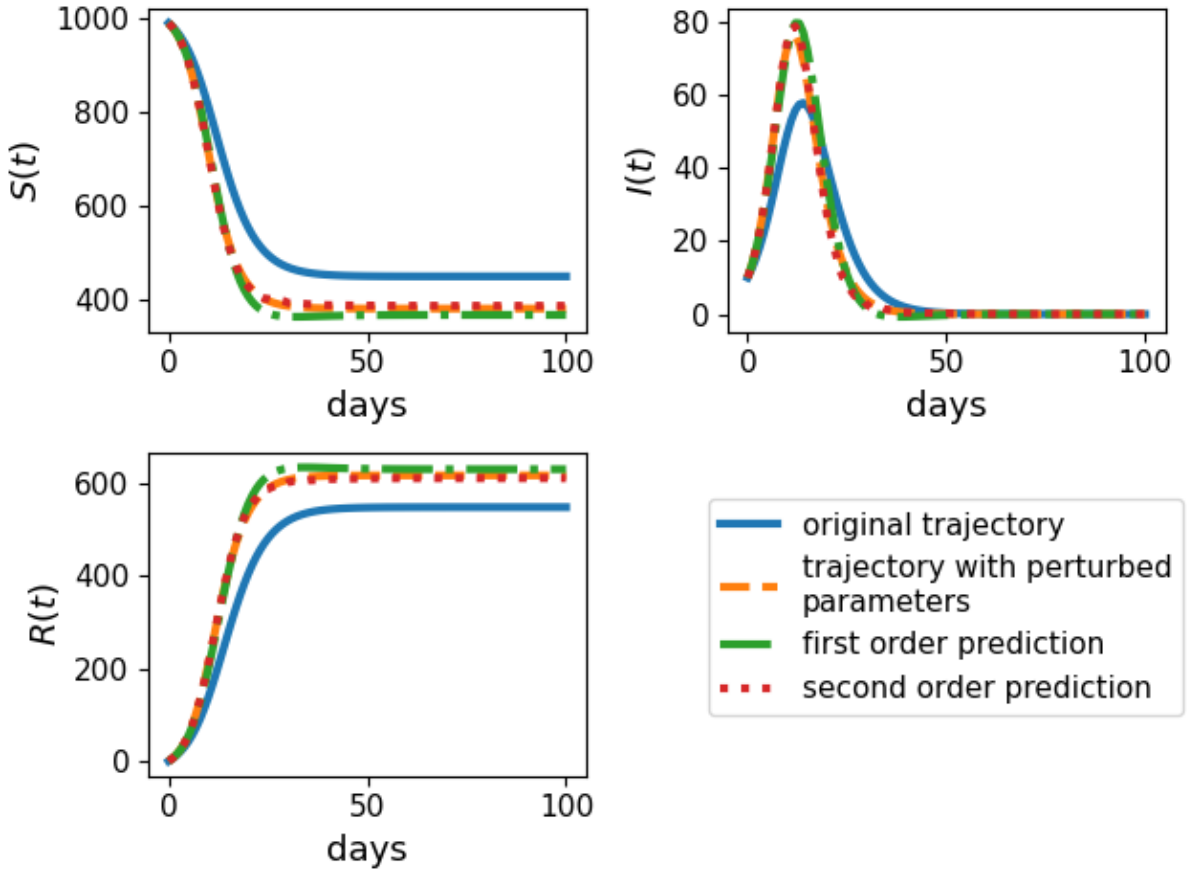

Figure 2.4: **Model trajectories for SIR model calculated using first and second differentials.** Time series plot of the SIR model simulated over $t = 100$ days with initial conditions $S_0 = 1000$ and $I_0 = 10$. Results depend on the SIR model with the original parameters from Figure 3 (original trajectory), re-simulating the SIR trajectory after perturbing the parameters by a random amount around 25% (trajectory with perturbed parameters), approximating the trajectory based on the linear expansion (eq. 5) and the first derivative calculated with the complex perturbation method (first-order prediction), and approximating the trajectory based on the quadratic expansion (eq. 4) and the first and second derivatives calculated with the complex perturbation method (second-order prediction).

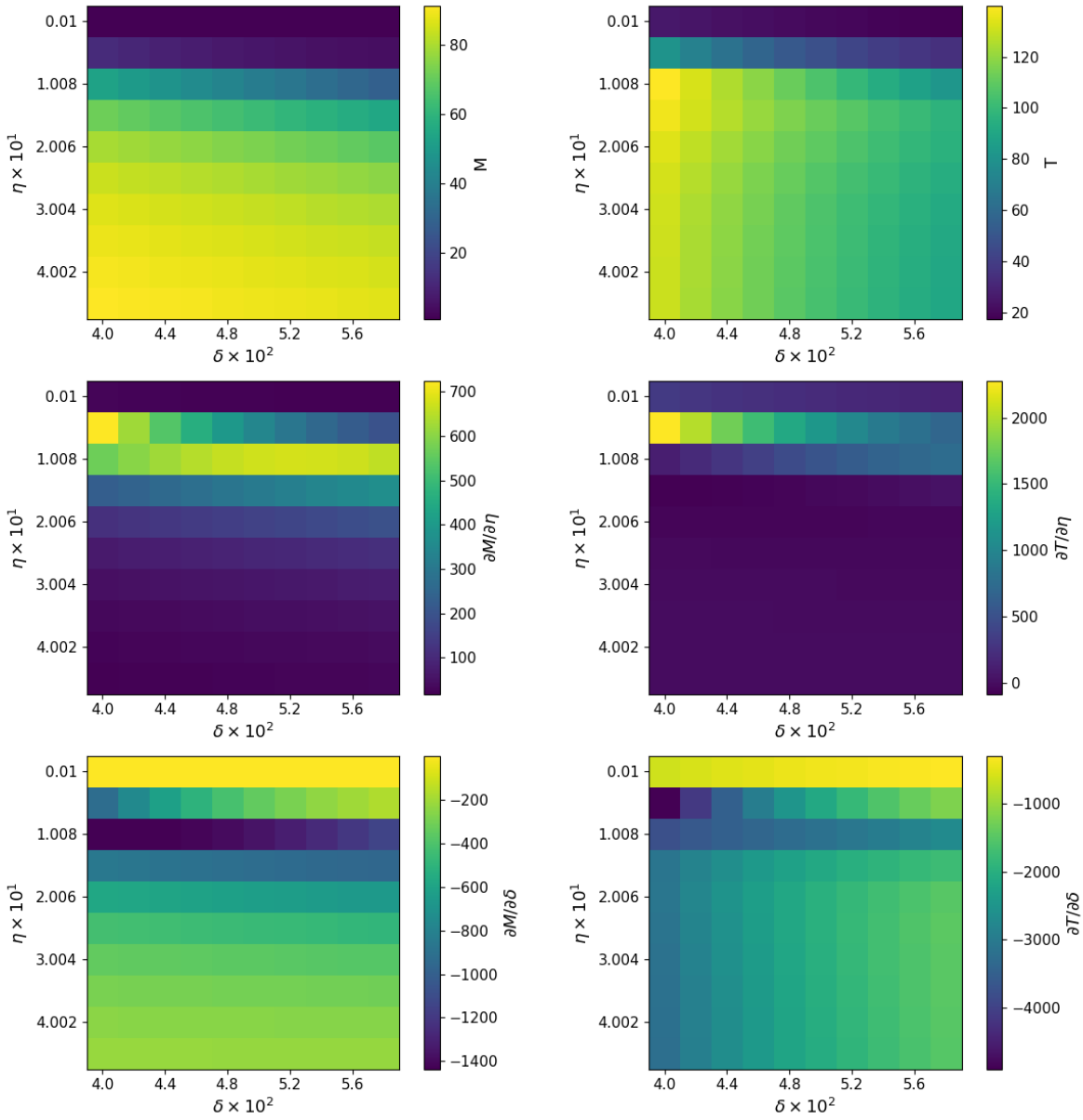

Figure 2.5: **Sensitivity of stochastic SIR model.** Heatmaps showing the mean number of infected individuals (M) at extinction, the mean time to extinction (T), and their sensitivities to the parameters η and δ for the stochastic SIR process. Sensitivities rely on the complex perturbation method to calculate derivatives and assume initial conditions $S_0 = 100$ and $I_0 = 1$.

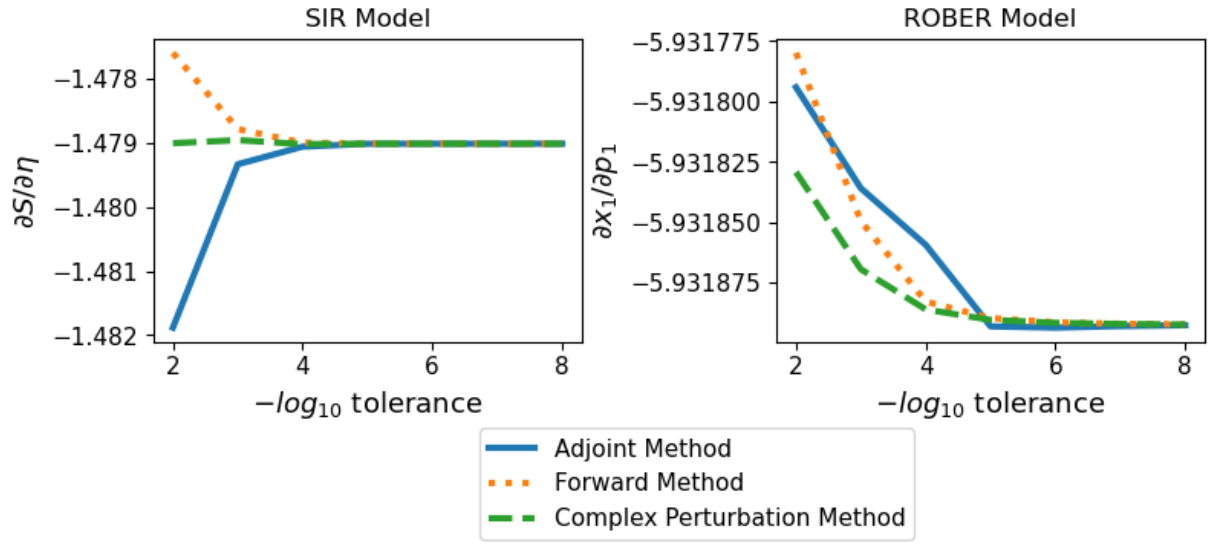

Figure 2.6: **Convergence of adjoint, forward, and complex perturbation methods for numerical sensitivities.** Convergence plot of the SIR model (top) and ROBER model (bottom) simulated over $t = 1000$ days. For SIR the initial conditions are $S_0 = 3.4 \times 10^8$, and $I_0 = 100$, and the parameters are $\eta = 0.7194$ and $\delta = 0.5025$. For ROBER the initial conditions are $x_1 = 1.0$, $x_2 = 0.0$, and $x_3 = 0.0$, and the parameters are $p_1 = 4 \times 10^{-2}$, $p_2 = 3 \times 10^7$, and $p_3 = 1 \times 10^4$. First-order sensitivities are computed via code from this manuscript (complex perturbation method), the ForwardDiff.jl package (forward method), and the Rodas4(autodiff=false) solver under the QuadratureAdjoint(autojacvec=EnzymeVJP()) sensealg protocol in the DiffEqSensitivities.jl package (adjoint method). The adjoint method requires a step size of 1.0 for the SIR model and a step size of 0.1 in the ROBER model to converge. All results are normalized by the number of time steps included in the simulation.

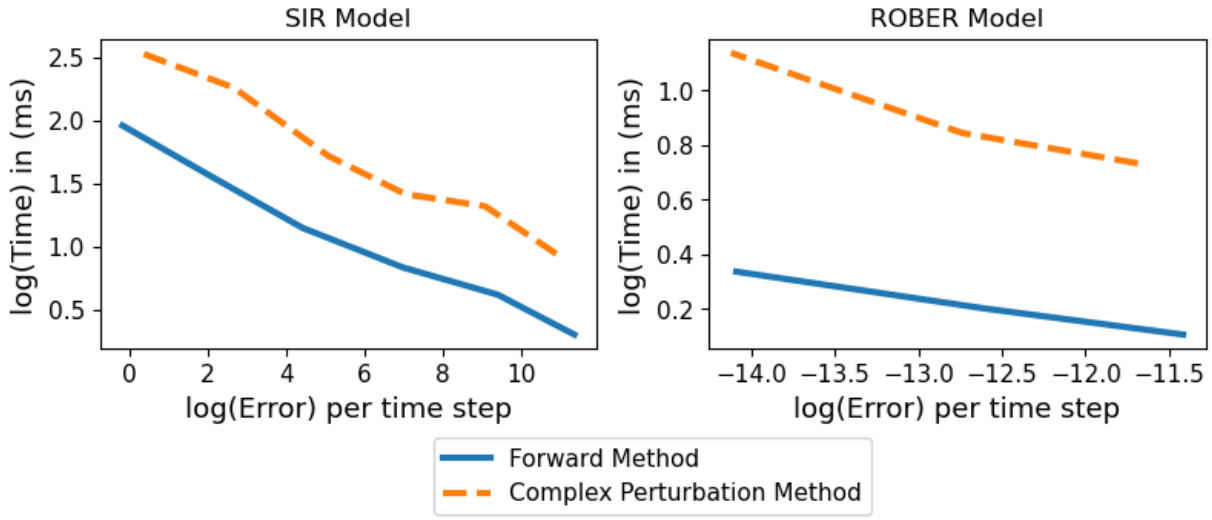

Figure 2.7: **Time vs Error of Forward and Complex Perturbation Methods for Numerical Sensitivities.** Time versus error log-log plot of the SIR model (top) and ROBER model (bottom) simulated over $t = 1000$ days. For SIR the initial conditions are $S_0 = 3.4 \times 10^8$, and $I_0 = 100$, and the parameters are $\eta = 0.7194$ and $\delta = 0.5025$. For ROBER the initial conditions are $x_1 = 1.0$, $x_2 = 0.0$, and $x_3 = 0.0$, and the parameters are $p_1 = 4 \times 10^{-2}$, $p_2 = 3 \times 10^7$, and $p_3 = 1 \times 10^4$. First-order sensitivities are computed via code from this manuscript (complex perturbation method) and the ForwardDiff.jl package (forward method). Times reported are the median times computed using the Benchmark.jl package, and errors are the Euclidean distance between the solution at the strictest tolerance (10^{-8} for SIR and 10^{-5} for ROBER) and the solution at a variety of tolerances with a maximum of 10^{-2}. All errors are normalized by the number of time steps.

2.8 SUPPLEMENT

2.8.1 Derivation of second derivative complex perturbation method

To prove the formulas for approximating partial derivatives stated in the text, we first note that any analytic function $f(\mathbf{z})$ of several variables can be expanded in a locally convergent power series about every point $\mathbf{z}$ of its open domain of definition. If we choose a real direction vector $\mathbf{v}$, then the function $g(w) = f(\mathbf{z} + w\mathbf{v})$ is locally analytic in the complex plane $\{\mathbf{z} + w\mathbf{v}: w \in \mathbb{C}\}$ and can be expanded in a power series around $w = 0$. Thus,

Equation 2.A1:

$$g(w) \quad = \quad \sum_{j=0}^{d} \frac{d^j}{dw^j} g(\mathbf{0}) \frac{w^j}{j!} + O(|w|^{d+1})$$

for any integer $d \geq 0$. Now consider the setting where $\mathbf{z}$ has real components. If $f(\mathbf{z})$ is real valued, then the derivatives $\frac{d^j}{dw^j} g(\mathbf{0})$ will be real as well. One can exploit this fact in approximating the derivatives. For example, if $w = i$, then w^j rotates among the four values 1, i, -1, and $-i$. Because the terms of the expansion (equation 2.A1) alternate between real and imaginary values, the first partial derivative formula

$$g'(\beta) \quad = \quad \frac{\text{Imag } g(\beta + \Delta i)}{\Delta} + O(\Delta^2)$$

holds. For the choice $w = e^{\pi i/4}$, the powers w^d rotate among the eight values 1, $e^{\pi i/4}$, i, $ie^{\pi i/4}$, -1, $-e^{\pi i/4}$, $-i$, and $-ie^{\pi i/4}$. The powers $(-w)^j = (-1)^j w^j$ agree in this regard except for sign. Hence, the terms in the expansion of the sum

$$g\left[\mathbf{x} + e^{\pi i/4}\Delta\left(\mathbf{e}_j + \mathbf{e}_k\right)\right] + g\left[\mathbf{x} - e^{\pi i/4}\Delta\left(\mathbf{e}_j + \mathbf{e}_k\right)\right]$$

alternately cancel and reinforce. Thus, the first five terms of the expansion are real, 0, imaginary, 0, real, 0. It follows that the imaginary part of the sum is accurate to order $O(\Delta^6)$ and that the approximations

$$\frac{\partial^2}{\partial \beta_j^2} g(\boldsymbol{\beta}) = \frac{\text{Imag}\left[g\left(\boldsymbol{\beta} + e^{\pi i/4}\Delta \mathbf{e}_j\right) + g\left(\boldsymbol{\beta} - e^{\pi i/4}\Delta \mathbf{e}_j\right)\right]}{\Delta^2} + O(\Delta^4)$$

and

$$\begin{aligned} & \frac{\text{Imag}\left\{g\left[\mathbf{x} + e^{\pi i/4}\Delta\left(\mathbf{e}_j + \mathbf{e}_k\right)\right] + g\left[\mathbf{x} - e^{\pi i/4}\Delta\left(\mathbf{e}_j + \mathbf{e}_k\right)\right]\right\}}{\Delta^2} \\ = & \left[\left(\mathbf{e}_j + \mathbf{e}_k\right)\right]^{\top} d^2 g(\mathbf{x})\left[\left(\mathbf{e}_j + \mathbf{e}_k\right)\right] + O(\Delta^4) \\ = & \frac{\partial^2}{\partial \beta_j^2} g(\boldsymbol{\beta}) + \frac{\partial^2}{\partial \beta_k^2} g(\boldsymbol{\beta}) + 2\frac{\partial^2}{\partial \beta_j \, \partial \beta_k} g(\boldsymbol{\beta}) + O(\Delta^4) \end{aligned}$$

are accurate to order $O(\Delta^4)$.

2.8.2 Additional models

The Mammalian Cell Cycle Model

The Mammalian Cell Cycle Model is a model originally described in[2.31] and simplified in the BioModels[2.35] database. This system describes the interaction of cyclin-dependent kinases (Cdk) with Cdk inhibitors, growth factors, and other proteins that regulate the development of mammalian cells. The model includes characteristics such as cell cycling, tumor repressor initiated progression control, and cell cycle completion. The ODE system representing the model is

$$\frac{dpRBc1}{dt} = kpc1 * pRB * E2F$$

$$\frac{dpRBc2}{dt} = kpc3 * pRBp * E2F$$

$$\frac{dCd}{dt} = kcd1 * AP1 + kdecom1 * Mdi$$
$$-kcom1 * Cd * \left(Cdk4_{tot} - (Mdi + Md + Mdp27)\right)$$
$$+kcd2 * E2F * \frac{Ki7}{Ki7 + pRB} * \frac{Ki8}{Ki8 + pRBp}$$

$$\frac{dMdi}{dt} = Vm2d * \frac{Md}{k2d + Md} + 2 * kcom1 * Cd * \left(Cdk4_{tot} - (Mdi + Md + Mdp27)\right)$$

$$\frac{dMd}{dt} = Vm1d * \frac{Mdi}{k1d + Mdi} + kcom1 * Cd * \left(Cdk4_{tot} - (Mdi + Md + Mdp27)\right)$$

$$\frac{dpRB}{dt} = kcd2 * E2F * \frac{Ki7}{Ki7 + pRB} * \frac{Ki8}{Ki8 + pRBp}$$

$$\frac{dE2F}{dt} = kcd2 * E2F * \frac{Ki7}{Ki7 + pRB} * \frac{Ki8}{Ki8 + pRBp}$$

$$\frac{dpRBp}{dt} = kcd2 * E2F * \frac{Ki7}{Ki7 + pRB} * \frac{Ki8}{Ki8 + pRBp}$$

$$\frac{dAP1}{dt} = kcd1 * AP1$$

$$\frac{dp27}{dt} = 0$$

$$\frac{dMdp27}{dt} = kc1 * Md * p27 + kcom1 * cd * \left(Cdk4_{tot} - (Mdi + Md + Mdp27)\right)$$

The initial values of each compartment and parameter are defined in Table 2.S1.

The ROBER Model

The ROBER Model refers to the auto-catalytic chemical reaction of Robertson as described in[2.30]. This model is often used as an example of a classic stiff ODE system encountered in biology. The ODE system this model represents is

$$\begin{aligned}\frac{dx_1}{dt} &= -p_1x_1 + p_3x_2x_3\\ \frac{dx_2}{dt} &= p_1x_1 - p_2x_2^2 - p_3x_2x_3\\ \frac{dx_3}{dt} &= p_2x_2^2\end{aligned}$$

The initial values of each compartment and parameter are defined in Table 2.S2.

2.8.3 Sensitivity of linear systems

The simplest dynamical models are governed by the linear constant coefficient differential equation $\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\beta})\mathbf{x}(t)$ with solution $\mathbf{x}(t) = e^{t\mathbf{A}(\boldsymbol{\beta})}\mathbf{x}_0$, where $\mathbf{A}(\boldsymbol{\beta})$ is any function differentiable in its parameters $\boldsymbol{\beta}$ and constant in t. The directional derivative of the matrix exponential $e^{\mathbf{B}}$ in the direction $\mathbf{V}$ can be represented by the integral

$$d_{\mathbf{V}}e^{\mathbf{B}} = \int_0^1 e^{s\mathbf{B}}\,\mathbf{V}e^{(1-s)\mathbf{B}}ds\,.$$

A simple proof of this fact appears in Example 3.2.2 of reference[36]. Setting $\mathbf{B} = t\mathbf{A}(\boldsymbol{\beta})$ and applying the chain rule leads to the partial derivative

$$\begin{aligned}\frac{\partial}{\partial\beta_j}e^{t\mathbf{A}(\boldsymbol{\beta})}\mathbf{x}(0) &= \int_0^1 e^{st\mathbf{A}(\boldsymbol{\beta})}\,t\frac{\partial}{\partial\beta_j}\mathbf{A}(\boldsymbol{\beta})e^{(1-s)t\mathbf{A}(\boldsymbol{\beta})}ds\,\mathbf{x}(0)\\ &= \int_0^t e^{s\mathbf{A}(\boldsymbol{\beta})}\frac{\partial}{\partial\beta_j}\mathbf{A}(\boldsymbol{\beta})e^{(t-s)\mathbf{A}(\boldsymbol{\beta})}ds\,\mathbf{x}(0),\end{aligned}$$

which can be laboriously evaluated by numerical integration. Simplification into a sum of exponentials is possible if $\mathbf{A}(\boldsymbol{\beta})$ is uniformly diagonalizable across all $\boldsymbol{\beta}$[26].

In practice, it is simpler to differentiate the original ODE with respect to β_j, interchange the order of differentiation, and numerically integrate the system

$$\frac{d}{dt}\frac{\partial}{\partial \beta_j}\mathbf{x}(t,\boldsymbol{\beta}) \quad = \quad \frac{\partial}{\partial \beta_j}\mathbf{A}(\boldsymbol{\beta})\mathbf{x}(t,\boldsymbol{\beta}) + \mathbf{A}(\boldsymbol{\beta})\frac{\partial}{\partial \beta_j}\mathbf{x}(t,\boldsymbol{\beta})$$

from 0 to some final value of t. The initial condition $\mathbf{x}(0,\boldsymbol{\beta}) = \mathbf{x}(0)$ remains intact, and the new condition $\nabla_{\boldsymbol{\beta}}\mathbf{x}(0,\boldsymbol{\beta}) = 0$ is added.

2.9 REFERENCES

2.1 Liepe, Juliane, et al. "A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation." *Nature protocols* 9.2 (2014): 439-456.

2.2 Neal, Radford M. "MCMC using Hamiltonian dynamics." *Handbook of markov chain monte carlo* 2.11 (2011): 2.

2.3 Gunawan, Rudiyanto, et al. "Sensitivity analysis of discrete stochastic systems." *Biophysical journal* 88.4 (2005): 2530-2540.

2.4 Gadkar, Kapil G., Rudiyanto Gunawan, and Francis J. Doyle. "Iterative approach to model identification of biological networks." *BMC bioinformatics* 6.1 (2005): 1-20.

2.5 Lillaci, Gabriele, and Mustafa Khammash. "Parameter estimation and model selection in computational biology." *PLoS computational biology* 6.3 (2010): e1000696.

2.6 Balsa-Canto, Eva, and Julio R. Banga. "AMIGO, a toolbox for advanced model identification in systems biology using global optimization." *Bioinformatics* 27.16 (2011): 2311-2313.

2.7 Marino, Simeone, et al. "A methodology for performing global uncertainty and sensitivity analysis in systems biology." *Journal of theoretical biology* 254.1 (2008): 178-196.

2.8 Cao, Yang, et al. "Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution." *SIAM journal on scientific computing* 24.3 (2003): 1076-1089.

2.9 Strang, Gilbert. *Computational science and engineering*. No. Sirsi) i9780961408817. 2007.

2.10 Rackauckas, Christopher, and Qing Nie. "Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia." *Journal of Open Research Software* 5.1 (2017).

2.11 Stapor, Paul, et al. "PESTO: parameter estimation toolbox." *Bioinformatics* 34.4 (2018): 705-707.

2.12 Hindmarsh, Alan C., et al. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers." *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005): 363-396.

2.13 Errico, Ronald M. "What is an adjoint model?." *Bulletin of the American Meteorological Society* 78.11 (1997): 2577-2592.

2.14 Granzow, Glen D. "A tutorial on adjoint methods and their use for data assimilation in glaciology." *Journal of Glaciology* 60.221 (2014): 440-446.

2.15 Il′i͡ashenko, I͡U S., et al. *Lectures on analytic differential equations*. Vol. 86. American Mathematical Soc., 2008.

2.16 Henrici, Peter. "Fast Fourier methods in computational complex analysis." *Siam Review* 21.4 (1979): 481-527.

2.17 Lange, Kenneth. *Applied probability*. Vol. 224. New York: Springer, 2003.

2.18 Lai, Kok-Lam, et al. "New complex-step derivative approximations with application to second-order kalman filtering." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2005.

2.19 Ma, Yingbo, et al. "A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions." *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, 2021.

2.20 Revels, Jarrett, et al. “Forward-mode automatic differentiation in Julia”. *arXiv preprint arXiv:*1607.07892 (2016).

2.21 Sahoo, Prativa, et al. "Mathematical deconvolution of CAR T-cell proliferation and exhaustion from real-time killing assay data." *Journal of the Royal Society Interface* 17.162 (2020): 20190734.

2.22 Toda, Alexis Akira. "Susceptible-infected-recovered (sir) dynamics of covid-19 and economic impact." *arXiv preprint arXiv:2003.11221* (2020).

2.23 Zhou, Fei, et al. "Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study." *The lancet* 395.10229 (2020): 1054-1062.

2.24 Landeros, Alfonso, et al. "BioSimulator. jl: Stochastic simulation in Julia." *Computer methods and programs in biomedicine* 167 (2018): 23-35.

2.25 Renshaw, Eric. "Birth, death and migration processes." *Biometrika* 59.1 (1972): 49-60.

2.26 Dorman, Karin S., Janet S. Sinsheimer, and Kenneth Lange. "In the garden of branching processes." *SIAM review* 46.2 (2004): 202-229.

2.27 Hautphenne, Sophie, et al. "Sensitivity analysis of a branching process evolving on a network with application in epidemiology." *Journal of Complex Networks* 3.4 (2015): 606-641.

2.28 Magnus, Jan R., and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.

2.29 Athreya, Krishna B., Peter E. Ney, and P. E. Ney. *Branching processes*. Courier Corporation, 2004.

2.30 Robertson, H. H. "The solution of a set of reaction rate equations." *Numerical analysis: an introduction* 178182 (1966).

2.31 Gérard, Claude, and Albert Goldbeter. "Temporal self-organization of the cyclin/Cdk network driving the mammalian cell cycle." *Proceedings of the National Academy of Sciences* 106.51 (2009): 21643-21648.

2.32 Squire, William, and George Trapp. "Using complex variables to estimate derivatives of real functions." *SIAM review* 40.1 (1998): 110-112.

2.33 Rackauckas, Christopher, et al. "Universal differential equations for scientific machine learning." *arXiv preprint arXiv:2001.04385* (2020).

2.34 Qian, George, and Adam Mahdi. "Sensitivity analysis methods in the biomedical sciences." *Mathematical biosciences* 323 (2020): 108306.

2.35 Malik-Sheriff, Rahuman S., et al. "BioModels—15 years of sharing computational models in life science." *Nucleic acids research* 48.D1 (2020): D407-D415.

2.36 Lange, Kenneth. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, 2016.

3 IMPACTS OF CROSS-ANCESTRY GENETIC ARCHITECTURE ON GWASS IN ADMIXED POPULATIONS

3.1 INTRODUCTION TO GWAS IN ADMIXED POPULATIONS

The success of genomics in disease studies depends on our ability to incorporate diverse populations into large-scale genome-wide association studies (GWASs).[3.1–3.4] Cohort and biobank studies are growing to reflect this diversity,[3.5–3.7] and a variety of techniques exist which incorporate populations of different continental ancestries into GWASs.[3.8] However, while admixture has been an important factor in other steps in the disease mapping process, such as fine-mapping[3.9] and estimating heritability,[3.10,3.11] individuals of mixed ancestries (admixed individuals) have largely been left out of traditional association studies. GWASs performed in admixed populations have greater power for discovery compared to similar sized GWASs in homogeneous populations.[3.12,3.13] Thus, excluding admixed individuals from association studies will not only increase health disparities, but will also disadvantage other populations. To prevent this exclusion, approaches to association studies have been developed specifically for admixed populations.[3.14–3.17] However, the impact of HetLanc (differences in estimated allelic effect sizes for risk variants between ancestry backgrounds) on GWAS methods remains under-explored. Of particular interest are recently admixed populations, defined as fewer than 20 generations of mixture between two ancestrally distinct populations. In such populations, the admixture process creates mosaic genomes comprised of chromosomal segments originating from each of the ancestral populations (i.e., local ancestry segments). Local ancestry segments are much larger than linkage disequilibrium (LD) blocks[3.18]; thus, LD patterns within each local ancestry block of an admixed genome reflect LD patterns of the ancestral population. Similarly, allele frequency estimates from segments of a particular local ancestry are expected to reflect allele frequencies of the ancestral population. Variation in local ancestry across the genome leads to variability in global

ancestry (the average of all local ancestries within a given individual). Such variability in local and global ancestries could pose a problem to GWASs in admixed populations as genetic ancestries are often correlated with socio-economic factors that also impact disease risk, thus yielding false positives in studies that do not properly correct for genetic ancestries. Because local and global ancestry are only weakly correlated,[3.19] complete control of confounding due to admixture requires conditioning on both local and global ancestry.[3.20] However, the success of admixture mapping indicates that the possibility of losing power due to over-correction for local ancestry differences is serious.[3.21,3.22]

GWASs in admixed populations are typically performed either using a statistical test that ignores local ancestry altogether (referred to in this work as ‘‘standard GWASs’’ and defined in Table 3.1) or using a test that explicitly allows for HetLanc (e.g., Tractor). The former provides superior power in the absence of HetLanc with the latter having great potential for discovery in its presence. However, these methods' relative statistical power for discovery depends on the cross-ancestry genetic architecture of the trait, i.e., which variants are causal and what are those variants' ancestry-specific frequencies, causal effects, and linkage disequilibrium patterns. For example, existing studies have found that standard GWASs can yield a 25% increase in power over Tractor[3.13] in the absence of HetLanc while Tractor has higher power when causal effects are different by more than 60%.[3.15] However, the full impact of cross-ancestry genetic architecture on GWAS power in admixed populations remains underexplored.

In this work, we use simulations to perform a comprehensive evaluation quantifying the impact of these factors on the power of GWAS approaches in admixed populations. We provide guidelines for when to use each test as a function of cross-ancestry genetic architecture. Elements of cross-ancestry genetic architecture such as allele frequencies, global ancestry ratios, and LD are

known or can be calculated in advance of a GWAS to determine which of our simulation results apply in each case. Using extensive simulations, we find that standard GWASs should be preferred when HetLanc is small or non-existent. We quantify the extent of HetLanc and the ancestry-specific allele frequency differences required for Tractor to overcome the extra degree of freedom penalty. We further validate our results using the African-European admixed population in the UK Biobank (UKBB). By examining the HetLanc of significant SNPs in the UKBB, we can understand how often it rises to a level that impacts the power of traditional GWASs.

3.2 SUBJECTS AND METHODS

3.2.1 Simulated genotypes

We simulate genotypes using the following procedure, which produces a set of genotypes made up of independent SNPs from admixed genotypes with two ancestries.

1) Draw the individual global ancestry proportion of ancestry 2, $\alpha \sim N(\theta, \sigma^2)$ for 10,000 individuals where q is the expected global ancestry proportion of ancestry 2, and σ^2 is the variance of global ancestry in the population (we use $\sigma^2 = 0.125$ to reflect the variance of global ancestry found in the UK Biobank admixed population). α is coerced between [0,1].

2) For each individual, draw a local ancestry count $l \sim Binomial(\alpha, 2)$, where l represents the local ancestry count of ancestry 2.

3) For each local ancestry, draw a genotype $g_i \sim Binomial(l, f_i)$, where f_i represents the allele frequency at local ancestry i. Allele frequencies f_i were specified for each simulation scenario according to the figure legends.

3.2.2 Simulated quantitative phenotypes with a single causal SNP

We simulate quantitative phenotypes with a single causal SNP (used in Figures 3.2C, 3.2D, 3.3A, and 3.S1–3.S8) using the following procedure.

1) Standardize genotypes so that they have a mean 0 and variance 1.

2) Given some effect sizes β_1, β_2, calculate $Var_g = Var(\beta_1 g_1 + \beta_2 g_2)$, where the variance is taken over all individuals, and Var_g represents the genetic variance component of the phenotypes.

3) Given some heritability h^2, calculate $Var_e = Var_g \frac{1-h^2}{h^2}$, where Var_e is the environmental variance component of the phenotypes. This comes from the equation $h^2 = \frac{Var_g}{Var_g + Var_e}$.

4) For each individual, draw $\epsilon \sim N(0, Var_e)$ where ϵ is the random noise to add to the phenotype to represent environmental variables.

5) Repeat for 1,000 replicates.

3.2.3 Simulated quantitative phenotypes with multiple causal SNPs

We simulate quantitative phenotypes with multiple causal SNPs using real genotypes (used in Figures 3.4, 3.S9, and 3.S10) with the following procedure.

1) Use chromosome 1 of the UK Biobank admixed African-European genotypes.

2) Given some polygenicity p (p = 100 used in Figures 3.4 and S9, p = 1, 10, 100 used in Figure 3.S10), randomly choose p SNPs to be causal.

3) Given some genetic correlation, draw effect sizes β_1, β_2, for causal SNPs chosen in step 2. Genetic correlations equal to 1.0, 0.5, and 1.0 used in Figures 3.4 and 3.S9, genetic correlation equal to 1.0 for Figure 3.S10. For more on genetic correlation, see Hou et al.[3.23]

4) Calculate $Var_g = Var(\beta_1 g_1 + \beta_2 g_2)$, where the variance is taken over all individuals, and Var_g represents the genetic variance component of the phenotypes.

5) Given some heritability h^2, calculate $Var_e = Var_g \frac{1-h^2}{h^2}$, where Var_e is the environmental variance component of the phenotypes. This comes from the equation $h^2 = \frac{Var_g}{Var_g + Var_e}$. h^2=0.5 used in Figures 3.4, 3.S9, and 3.S10.

6) For each individual, draw $\epsilon \sim N(0, Var_e)$ where ϵ is the random noise to add to the phenotype to represent environmental variables.

7) Repeat for 100 replicates.

3.2.4 Simulated case-control phenotypes

We simulate case-control phenotypes (used in Figures 3.2A and 3.2B) using the following procedure.

1 Given some SNP, ancestry-specific odds ratios β_1, β_2, and a case prevalence c, case-control phenotypes were simulated under the logistic model as in Atkinson et al.[3.15]

   a) Calculate the genetic component of the phenotype (y_g) for each individual j as $y_{g,j} = \beta_1 g_{1,j} + \beta_2 g_{2,j}$.

b) Find some intercept b such that $\overline{expit(y_g + b)} - c = logit(c)$, where the bar refers to the mean over all individuals j.

c) For each individual j, draw case status from a $Bernoulli\left(expit\left(y_{g,j} + b\right)\right)$ distribution.

d) Randomly discard control phenotypes until the case:control ratio is 1:1.

2 Repeat for 100 iterates of 1,000 replicates.

3.2.5 Real genotypes and phenotypes

For our real data analysis, we used genotypes from the UK Biobank. We limited our study to participants with admixed African-European ancestry. Overall, we had 4,327 individuals with an average of 58.9% African and 41.1% European ancestry. We used the imputed genotypes for these individuals with a total of 16,584,433 SNPs. The genotypes were mapped to the GRCh38 build and imputed to the TOPMed reference panel. We calculated the top 10 PCs for these genotypes and added these PCs as covariates to all analyses as our global ancestry component. The phenotypes we used are also from the UK Biobank and include aspartate transferase enzyme (AST), BMI, cholesterol, erythrocyte count, HDL, height, LDL, leukocyte count, lymphocyte count, monocyte count, platelet count, and triglycerides. We log transformed AST, BMI, HDL, leukocyte count, lymphocyte count, monocyte count, platelet count, and triglycerides to analyze all 12 traits as quantitative, continuous traits. We standardized all genotypes and phenotypes to be mean centered at 0.0 and have a variance of 1. This research complies with all relevant ethical regulations. The ethics committee/IRB of UKBB gave ethical approval for collection of UKBB

data. Participants signed a written consent form to be a part of the UKBB. Approval to use UKBB individual-level data in this work was obtained under application 33127.

3.2.6 Association testing on simulated genotypes

We calculate the standard GWAS and Tractor association tests on simulated data. A standard GWAS is a one degree of freedom association test that uses the model $y = \beta_g + e_\alpha + b\mathbf{1} + \epsilon$ to test against a null hypothesis that includes global ancestry (α). Tractor is a two degree of freedom association test that uses the model $y = \beta_1 g_1 + \beta_2 g_2 + e_l l + e_\alpha \alpha + b\mathbf{1} + \epsilon$ to test for $\beta_1 = 0$ and $\beta_2 = 0$ against a null hypothesis that includes local ancestry (l) and global ancestry (α). They can both be adapted to be used on case-control phenotypes by substituting logistic regression and odds ratios for linear regression and effect sizes. Additionally, they can both be adjusted for additional covariates such as age and sex. For our simulations, we used global ancestry proportions as ourmeasure of global ancestry (α) and did not need to adjust for any additional covariates such as age and sex as we did not model those factors in our simulations. For power calculations, we use a standard significance threshold of p-value < $5\text{x}10^{-8}$.

3.2.7 Association testing on real genotypes

We used admix-kit[3.43] to perform the standard GWAS and Tractor association tests on these data and extracted the p values. To determine significant SNPs, we filtered for SNPs with a standard p-value of < $5\text{x}10^{-8}$. For the Manhattan plots, we plot all SNPs with a p-value < 10^{-2} in Figure 3.5B and a p-value < 10^{-4} in Figure 3.S11 for computational plotting purposes. For Tables 3.S1 and 3.S2 and Figure 3.5A, to determine whether SNPs were part of the same locus, we grouped SNPs within a 500 kB radius and kept the most significant SNP from each test (standard GWAS and Tractor) in that locus.

3.2.8 Measures used to compare our results

In this work, we introduce several key measures that we use to compare our results. The formal definitions of these are the following.

Percent difference in power:

$$\frac{2(Power_{Standard\ GWAS} - Power_{Tractor})}{Power_{Standard\ GWAS} + Power_{Tractor}}$$

Adjusted chi square:

We take the p value from a χ^2 statistic and convert it back to a χ_1^2 statistic, regardless of the original degrees of freedom. The adjusted chi square score for a χ_1^2 is itself.

3.3 RESULTS

3.3.1 Heterogeneity by local ancestry impacts association statistics in admixed populations

HetLanc occurs when a SNP exhibits different estimated allelic effect sizes depending on its local ancestry background. HetLanc can manifest itself at causal SNPs due to genetic interactions between multiple causal variants or differential environments, although recent work suggests that the magnitude and frequency of these types of epistatic effects between causal variants is limited.[3.23] A more common form of HetLanc is observed at non-causal SNPs that tag the causal effect in a differential manner across ancestries. Differential linkage disequilibrium by local ancestry at these non-causal SNPs (tagged SNPs) can cause HetLanc even when allele frequencies and causal effect sizes are the same across ancestries. The extent to which HetLanc exists and the magnitude of these differences in effect sizes are yet uncertain.[3.22–3.38] However, the

existence of HetLanc plays an important role in the power of GWAS methods to detect associations. Consider the example in Figure 3.1 in which the allelic effect size for a tagged SNP is estimated for a phenotype in an admixed population. In this population, both the tagged SNP and the true causal SNP may exist in regions attributed to both local ancestries present in the population (Figure 3.1A). Since LD patterns differ by local ancestry, the correlation between the tagged and causal SNPs will also depend on local ancestry (Figure 3.1B). This differential correlation between tagged and causal SNPs will cause the estimated allelic effect size for the tagged SNP $\hat{\beta}_{tag,i}$ to depend on local ancestry *i* (Figure 3.1C). Thus, even for cases in which true causal effect sizes are the same across ancestries, allelic effect sizes estimated for the tagged SNP may be heterogeneous. Since GWASs cannot determine true causal effect sizes, we introduce R_{het}, a measure of HetLanc which allows for both true causal effect-size heterogeneity and LD- and allele frequency-induced estimated allelic effect-size heterogeneity.

3.3.2 Methods for association testing in admixed populations

We start with a formal definition for a full model relating genotype, phenotype, and ancestry for a single causal SNP:

Equation 3.1:

$$y = \beta_1 g_1 + \beta - 2g_2 + e_l l + e_\alpha \alpha + e_A^T A + b\mathbf{1} + \epsilon$$

where y is a phenotype, g_1 and g_2 are vectors that represent the number of alternate alleles with local ancestry 1 and 2 (such that $g_1 + g_2 = g$, the full genotype regardless of ancestry), β_1 and β_2 are ancestry-specific marginal effect sizes of the SNP, l is the vector of local ancestry counts at the locus, e_l is the effect size of l; a is a vector of global ancestry proportions, e_α is the effect size of

α, A is a matrix of additional covariates (such as age and sex), e_A^T is a vector of effect sizes for these covariates, b is the intercept term multiplied by the column vector **1**, and ε is random environmental noise.

Variability across local and global ancestries has been leveraged in various statistical approaches for disease mapping in admixed populations. One of the first methods developed for association was admixture mapping (ADM).[3.30,3.36] ADM tests for association between local ancestry and disease status in affected individuals and control subjects or in a case-only fashion. This association is achieved by contrasting local ancestry deviation with expectations from per-individual global ancestry proportions. Therefore, ADM is often under-powered especially in situations in which allele frequency at the causal variant is similar across ancestral populations.[3.31] Genotype association testing is traditionally performed using a linear or logistic regression with some standard covariates. This type of association test, referred to in this work as a standard GWAS, tests for association between genotypes and disease status while correcting for global ancestry to account for stratification.[3.17,3.32] However, neither ADM nor standard GWASs take advantage of the full disease association signal in admixed individuals. SNP1, SUM, and MIX are examples of association tests that combine local ancestry and genotype information. SNP1 regresses out local ancestry in addition to global ancestry to control for fine-scale population structure. This approach helps control for fine-scale population stratification but may remove the signal contained in local ancestry information.[3.33] SUM[3.34] combines the SNP1[3.14] and ADM statistics into a two degree of freedom test. MIX[3.14] is a case-control test that incorporates SNP and local ancestry information into a single degree of freedom test. Most recently Tractor[3.15] conditions the effect size of each SNP on its local ancestry followed by a joint test allowing for different effects on different ancestral backgrounds. This step builds the possibility of HetLanc

explicitly into the model, which may be particularly important when SNPs are negatively correlated across ancestries.[3.35] Other varieties of tests have also been developed using different types of frameworks, most notably BMIX[3.35] which leverages a Bayesian approach to reduce multiple testing burden. These statistics have been compared at length.[3.3,3.14,3.22] However, existing comparisons do not consider HetLanc, nor do they thoroughly discuss allele frequency differences across ancestries.

3.3.3 Standard GWASs have more power than Tractor in the absence of heterogeneity by ancestry

First, we use simulations to compare type I error and power for each association statistic in Table 3.1. Starting with 10,000 simulated admixed individuals based on a 50/50 admixture proportion, we simulate 1,000 case-control phenotypes with a single causal SNP (see subjects and methods). We define type I error as the percent of noncausal SNPs found to have significant associations (p-value < 0.05) for each score (see subjects and methods). Type I error is well controlled by Tractor (5.01%), SNP1 (5.01%), MIX (5.00%), and standard GWASs (5.01%) (Figure 3.2A). However, we find that type I error is not as well controlled for ADM (9.15%) and SUM (7.84%). We next calculate power to detect causal SNPs for an odds ratio of $OR_1 = OR_2 = 1.2$ (see subjects and methods). We find that SNP1 had the highest power at 42.14%. However, SNP1 was not significantly more powerful than either MIX (power 42.12%, p-value 0.878) or a standard GWAS (power 42.05%, p value 0.325, Figure 3.2B). The power of all three of these tests was significantly higher (p-value < $1x10^{-16}$) than for SUM (power = 33.4%), ADM (power = 0.039%), or Tractor (power = 31.9%). Since Tractor is a statistical test specifically designed to find SNP-trait associations with effects that are heterogeneous by local ancestry,[3.15] this loss of power is expected for Tractor when effect sizes are the same across ancestries, which is not the

genetic architecture for which Tractor was designed. We find that while these association statistics are all well controlled, power does substantially differ between them. In the absence of both HetLanc and allele frequency difference, one degree of freedom SNP association tests outperform two degree of freedom tests.

We next investigate how differences in causal allele frequency (CAF) impact the power of a standard GWAS and Tractor in the case where true causal effect sizes are the same. We investigate the impact of varying CAF in each ancestry independently. Using our 10,000 simulated admixed individuals from the previous experiment, we simulate 1,000 quantitative phenotypes with a single causal SNP (see subjects and methods). We calculate the power of both Tractor and a standard GWAS to find these causal SNPs and then average that power over 100 simulated genotypes with specific allele frequencies. First, we let $CAF_1 = 0.5$ and CAF_2 range from 0.0 to 1.0 with a 0.1 increment and plot power over CAF_2 (Figure 3.2C). We find that a standard GWAS and SNP1 have higher power than Tractor at all levels of CAF difference. Since Tractor has an extra degree of freedom compared to a standard GWAS and SNP1, Tractor is disadvantaged when $\beta_1 = \beta_2$. Additionally, we see that while SNP1 has (insignificantly) higher power than a standard GWAS when CAF1 = CAF_2, the power of SNP1 deteriorates as causal allele frequency difference increases. This behavior is qualitatively the same as Tractor. When CAF_1 = CAF2, a standard GWAS has 94.7% power, with Tractor at 91.1% power. However, as CAF_2 becomes more different from CAF_1, a standard GWAS maintains its power at 93.0%. By contrast, Tractor loses much of its power, with only 45.3% power when the causal allele is fixed at 100% in population 2 and only 48.1% power when the causal allele is absent in population 2. A standard GWAS maintains higher power than Tractor even at varying levels of heritability (Figures 3.S1–3.S3),

global ancestry (Figure 3.S1), effect size β (Figure 3.S2), and CAF1 (Figure 3.S3). However, the difference in power has a large range depending on the CAF difference between local ancestries.

Next we introduce percent difference in power, a one-dimensional metric to compare between these association statistics (see subjects and methods). We use this metric to visualize how varying CAF1 and CAF_2 independently impacts the power of a standard GWAS and Tractor (Figure 3.2D). The percent increase in power when using a standard GWAS over Tractor when the causal SNP is absent in population 2 is 68%. The power difference between a standard GWAS and Tractor increases as CAF difference increases. Furthermore, the lower the CAF starts out in population 1, the larger the power difference between these two statistics. Specifically, when $CAF_1 = 0.5$ and $CAF_2 = 0.1$, the difference in CAF is 0.4 and a standard GWAS has a 25% power increase over Tractor. However, when $CAF_1 = 0.4$ and $CAF_2 = 0.0$, the difference in CAF is still 0.4 but a standard GWAS has a 43% increase in power over Tractor. While these differences in power do depend on both CAF differences and absolute CAF values in both ancestries, it is worth noting that differences in power along the diagonal axis are not significant. For example, while the increase in power of a standard GWAS over Tractor is 25% when CAF1 = 0.5 and $CAF_2 = 0.1$ and the increase in power of a standard GWAS over Tractor is 26% when $CAF_1 = 0.1$ and $CAF_2 = 0.5$, the difference that occurs when switching causal allele frequencies between ancestries only has a p-value of 0.345 in this case.

While this result corroborates previous studies,[3.40–3.42] the relationship between Tractor and admixture mapping provides insight into the mechanism behind this dynamic. Mainly, as allele frequency differentiation by local ancestry increases, so does the power of the admixture mapping test statistic. In fact, ADM has no power when causal allele frequencies do not differ by ancestry but achieves up to 6.7% power when $CAF_1 = 0.0$ and $CAF_2 = 0.5$ (Figure 3.S4A). However, the

Tractor method uses the admixture mapping statistic as its null hypothesis. A stronger null hypothesis will be rejected less often than a weaker one even when the alternative hypothesis is the same, causing any test utilizing a strong null hypothesis to have less power. Thus, Tractor will have less power when its null hypothesis (ADM) has more power, which occurs in situations with high allele frequency differentiation. When allele frequencies do not differ by ancestry, Tractor achieves 91% power in our simulations. However, when $CAF_1 = 0.0$ and $CAF_2 = 0.5$, Tractor power plummets to 44% (Figure 3.S4B). SNP1, which also uses ADM as its null hypothesis, suffers from the same deterioration in power as causal allele frequency differentiation increases (Figure 3.S4C). When the causal allele frequencies are the same, SNP1's power matches that of a standard GWAS, but as causal allele frequency differentiation increases, SNP1 loses power in the same pattern as Tractor. This indicates that Tractor loses power compared to a standard GWAS due to both its additional degree of freedom and due to its choice of null hypothesis.

While high levels of allele frequency differentiation drastically decrease the power of Tractor, a standard GWAS also has a smaller decrease in power at high levels of allele frequency differentiation, from 95% at equal allele frequencies to 93% when $CAF_1 = 0.0$ and $CAF_2 = 0.5$ (Figure 3.S4D). This decrease in power is not as large as that suffered by Tractor, but it is also due to increased power of the null hypothesis at higher frequency differentiation across populations. The null hypothesis of the standard GWAS test statistic only includes global ancestry, but the power of global ancestry alone to predict a trait increases as allele frequency differentiation increases.[3.32] The idea that including global ancestry as a covariate in these analyses reduces power for SNPs with large CAF differences raises the question of how much attenuation can be expected when more exact measures of global ancestry (such as principal components) are included in the analysis. However, the overall power attenuation due to the inclusion of global ancestry is small

compared to that due to local ancestry; thus, we shift our focus back to considering local ancestry-specific effects on power.

3.3.4 Impact of HetLanc on power depends on allele frequency differences

Next, we investigate the impact of CAF differences and HetLanc on power differences between a standard GWAS and Tractor. The exact relationship between HetLanc (measured as R_{het}), CAF difference, and percent difference in power is complex (Figure 3.3A). First, there is a window when $0.5 < R_{het} < 1.5$ in which, regardless of CAF difference, HetLanc is not enough to increase the power of Tractor relative to a standard GWAS. Thus, at these ‘‘low’’ levels of HetLanc, a standard GWAS will reliably have more power than Tractor across the allele frequency spectrum. Similarly, when $R_{het} < 0.5$, there is no allele frequency difference which would increase the power of a standard GWAS relative to Tractor. This corroborates our findings that when effect sizes are in opposite directions, Tractor is expected to have improved power over standard GWASs regardless of CAF difference. We can see that it is characteristics of both standard GWASs and Tractor that drive this trend (Figure 3.S5). The power of a standard GWAS depends most strongly on the magnitude of R_{het} and is diminished the most when effect sizes are in opposite directions. By contrast, the power of Tractor depends strongly on both CAF difference and R_{het}. These two factors combine to create an asymmetric shape for the percent difference in power (Figure 3.3A). This asymmetry in power observed for the Tractor method is likely due to correlations between effective sample size, allele frequency, global ancestry, and local ancestry that can occur in an asymmetric manner when causal effect sizes and causal allele frequencies differ between local ancestries.[3.32] In these figures, we must consider that CAF_1 is held constant at 0.5 and β_2 is held constant at 1.0. For example, $R_{het} = 0.5$ corresponds to $\beta_1 = 0.5$ and $\beta_2 = 1.0$. When $CAF_1 = 0.5$ and $CAF_2 = 0.9$, most of the genetic variance from the individuals in the study will come from

ancestry 2 due to its larger causal allele frequency and larger effect size. This leaves the association for ancestry 1 with much less genetic variance to work with, and thus will lead Tractor's ability to detect an association in ancestry 1 to be under-powered. However, when $CAF_2 = 0.1$, much less of the total genetic variation in the population will come from ancestry 2, leading Tractor's power to detect association in both populations to be more balanced.

We also find that SNP1 power suffers not only when causal allele frequency differences increase but also when HetLanc increases. We additionally investigate similar scenarios for standard GWASs and Tractor with varied global ancestry proportions (Figure 3.S6), population-level CAF (Figure 3.S7), and heritability (Figure 3.S8). While the exact boundaries of these regions do differ, the overall shape of this heatmap and the conclusions mentioned above do not qualitatively change.

3.3.5 Polygenic trait simulations follow the same pattern as single causal variant simulations

We next investigate how HetLanc impacts power in polygenic traits. We consider the genotypes of individuals with African-European admixture in the UK Biobank. These individuals have an average of 58.9% African and 41.1% European ancestry over the population of 4,327 individuals. We simulate phenotypes using 100 causal SNPs along chromosome 1 and compare the power of a standard GWAS and Tractor over 100 simulations. Using real genotypes allows us to consider polygenic traits in the context of more realistic linkage disequilibrium and admixture. We now use genetic correlation[3.23] instead of R_{het} to measure HetLanc in the case of polygenic traits and separate our findings by whether or not the causal SNPs are differentiated (MAF difference > 0.2) or non-differentiated (MAF difference % 0.2).

First, we find that both standard GWASs and Tractor have relatively well-calibrated type I error rates (Figure 3.4A). At an expected false positive rate of 5%, a standard GWAS has a 5.06% false positive rate for differentiated SNPs and a 5.00% false positive rate for non-differentiated SNPs. In this situation, in which genetic correlation = 1.0 (which corresponds to zero effect size heterogeneity), Tractor has a well-calibrated false positive rate of 4.99% for differentiated SNPs, but a false positive rate of 3.35% for non-differentiated SNPs, which is significantly deflated (p-value $< 10^{-16}$).

Similar to our simulations with only a single causal SNP, a standard GWAS and Tractor each have higher power in different combinations of genetic correlation and MAF differences. When genetic correlation remains 1.0 (Figure 3.4B), a standard GWAS has 23.0% power for differentiated SNPs and 25.5% power for non-differentiated SNPs, in contrast to Tractor's 19.5% power for differentiated SNPs and 23.3% power for non-differentiated SNPs. The difference in power between differentiated SNPs and non-differentiated SNPs is significant (p-values 3.53×10^{-3} for a standard GWAS and 1.73×10^{-6} for Tractor). The difference in power between a standard GWAS and Tractor is significant as well (p-values 4.84×10^{-4} for differentiated SNPs and 3.22×10^{-4} for non-differentiated SNPs).

After we introduce HetLanc, its direction and magnitude impact which method has the most power, a result which resembles our previous findings. When effect sizes vary by ancestry but are in the same direction (genetic correlation = 0.5, Figure 3.4C), a standard GWAS has more power for differentiated SNPs (18.7% for a standard GWAS and 16.8% for Tractor, p-value 0.04), whereas Tractor has more power than a standard GWAS for nondifferentiated SNPs (18.0% for standard GWAS and 20.1% for Tractor, p-value 5.44×10^{-4}). When effect sizes are in opposite directions however (genetic correlation = 1.0, Figure 3.4D), Tractor has more power than standard

GWASs for both differentiated SNPs (4.90% for a standard GWAS and 11.5% for Tractor, p-value = 3.50×10^{-11}) and non-differentiated SNPs (1.93% for a standard GWAS and 14.7% for Tractor, p-value < 10^{-16}). We also consider the SNP1 test for these polygenic analyses. As expected, SNP1 remains well calibrated in the polygenic case but falls between Tractor and standard GWASs in terms of power when effect sizes are the same (genetic correlation = 1.0). However, when effect sizes are different (genetic correlation = 0.5 or 1.0), SNP1 performs less well than either standard GWASs or Tractor (Figure 3.S9). We also consider how the level of polygenicity impacts power in the case with genetic correlation = 1.0 (Figure 3.S10). We find that while a standard GWAS remains more powerful than Tractor when polygenicity is reduced to 10, the differences in power between a standard GWAS and Tractor do not remain significant in either the differentiated or non-differentiated case. This is likely due to the high heritability in this case since for the polygenic simulations we held $h^2 = 0.5$. Thus, in the case of 100 causal SNPs, each SNP had a $h^2 = 0.005$, which is identical to the heritability in the single causal SNP simulations. In the case of 10 causal SNPs, however, each SNP had $h^2 = 0.05$, which increased overall power, causing a necessary decrease in power difference between methods.

3.3.6 A standard GWAS finds more significant loci across 12 traits in the UK Biobank

We next seek to understand the impact of correcting for local ancestry in genetic analyses in real data. We investigate both Tractor and a standard GWAS in the same population of African-European admixed individuals from the UK Biobank. In real data, we investigate MAF (minor allele frequency) differences in lieu of CAF differences, since it is common practice to test minor alleles in real GWASs. First, we investigate MAF differences between segments of African and European local ancestry over 16,584,433 imputed SNPs. We find that the mean absolute minor

allele frequency difference of these SNPs is 0.0959, with a standard deviation of 0.115. 85.2% of them have an absolute allele frequency difference of <0.2 across local ancestry (Figure 3.S11).

Next, we investigate empirically derived values of R_{het} to determine in which region of the heatmap estimated effect sizes are likely to be found in real data (Figure 3.3B). We ran the Tractor method on 12 quantitative traits to find the actual values of R_{het} for the estimated effect sizes β_{AFR} and β_{EUR}. These traits were aspartate transferase enzyme (AST), BMI, cholesterol, erythrocyte count, HDL, height, LDL, leukocyte count, lymphocyte count, monocyte count, platelet count, and triglycerides. Then, we line up the histogram of these empirically derived values of R_{het} with the heatmap. We find that for 69.3% of all SNPs found to be significant using the Tractor test statistic, the empirical value for R_{het} is within this [-0.5, 1.5] window. While this is an estimate, we predict the true difference between estimated marginal effect sizes might be smaller than indicated by these empirical values because Tractor is more powerful in identifying SNPs with heterogeneous effect sizes. This result reflects previous findings that causal effects are similar across ancestries within admixed populations[23]. Due to this similarity in effect size, most of the significant SNPs sit in the center of the heatmap. This region of this heatmap predicts that standard GWASs will have more power than Tractor. While we cannot directly compare the standard GWAS χ_1^2 score with the Tractor χ_2^2 score due to their differing degrees of freedom, we can compare the mean adjusted χ^2 statistics. To calculate the adjusted statistic, we take the p-value from a χ^2 statistic and convert it back to a χ_1^2 statistic, regardless of the original degrees of freedom. In this way, we can compare the mean adjusted χ^2 statistic of the SNPs found to be significant in this case. We find that this statistic is significantly larger for the standard GWAS method than the Tractor method (Figure 3.S12). For significant SNPs, the mean standard GWAS χ_1^2 is 42.9, the mean adjusted Tractor χ_1^2 is 37.5, and the p-value for the difference is 2.11×10^{-4}.

In addition to assessing HetLanc directly, we can also compare the number of independent significant SNPs found by a standard GWAS and Tractor for these phenotypes. We find that while the number of independent significant SNPs varies across all traits, including when grouped by independent loci (Table 3.S1), overall, a standard GWAS finds more significant independent signals than Tractor (Figure 3.5A). We find 22 independent significant loci, with 19 loci found in a standard GWAS and 10 found in Tractor. This trend is most pronounced in HDL, in which 5 independent loci were determined to be significant by a standard GWAS compared to none for Tractor. Similarly, BMI, leukocyte count, and monocyte count also only had independent significant loci when testing using a standard GWAS as opposed to Tractor. Cholesterol and LDL had significant loci found by both standard GWAS and Tractor, with a larger number found by the standard GWAS. Height is the only trait for which Tractor identified one significant locus but not the standard GWAS. Unfortunately, our sample sizes were not large enough to detect any significant loci for platelet count, triglycerides, or lymphocyte count. All independent significant loci for these 12 phenotypes are detailed in Table 3.S2.

Additionally, we find that while a standard GWAS often finds more significant independent loci than Tractor, the two methods do not always find the same loci. Erythrocyte count is one phenotype in which we find an equal number of independent significant loci using both a standard GWAS and Tractor. However, not all loci overlap. Investigating the Manhattan plot of erythrocyte count specifically (Figure 3.5B), we see that loci on chromosome 16 are found by both a standard GWAS and Tractor. But outside of the main locus, both the standard GWAS and Tractor find separate additional significant regions. At the main locus, this Manhattan plot clearly shows that a standard GWAS has significantly smaller p-values for the same locus. Thus, in a smaller sample size only a standard GWAS would have found this important region. This example

highlights the importance of choosing the most highly powered association statistic for any given situation. Manhattan plots for other phenotypes can be found in Figure 3.S13.

3.4 DISCUSSION

In this work, we seek to understand the impact that estimated allelic effect-size heterogeneity by ancestry (HetLanc) has on the power of a GWAS in admixed populations. Our main goal is to find whether conditioning disease mapping on local ancestry leads to an increase or decrease in power. We find that HetLanc and CAF differences are the two most important factors when considering various methods for disease mapping in admixed populations. We focus on two association statistics: a standard GWAS, which ignores local ancestry, and Tractor, which conditions effect sizes on local ancestry. We find that in cases with small or absent levels of HetLanc, a standard GWAS is more powerful than Tractor in simulations of quantitative traits. This conclusion holds across a variety of global ancestry proportions and levels of SNP heritability. We find that as CAF differentiation between ancestries increases, so does the improvement of power of a standard GWAS compared to Tractor. At high HetLanc (R_{het} >1.5) or when effect sizes are in opposing directions (R_{het} < 0.5), we find that Tractor out-performs a standard GWAS. For African-European admixed individuals in the UKBB, most significant loci have both small measured HetLanc and MAF differences. We find that across 12 quantitative traits, a standard GWAS finds more significant independent loci than Tractor. Furthermore, a standard GWAS has smaller p-values for the loci that it shares with Tractor. This suggests that on smaller datasets, more of the shared loci would be found by a standard GWAS than by Tractor.

This work has several implications for GWASs in admixed populations. Our results suggest that usually, a standard GWAS adjusted for global ancestry is the most powerful way to perform

a GWAS in an admixed population. However, it may be possible to predict the comparative power of a standard GWAS and Tractor using the allele frequencies and linkage disequilibria of a specific sample. Additionally, since in real analyses a standard GWAS and Tractor often find different loci, it is important to keep both methods in mind when performing analyses. These methods prioritize different types of loci, with standard GWASs likely prioritizing loci with higher MAF differences and Tractor prioritizing loci with higher levels of HetLanc. Furthermore, our findings suggest that conditioning on local ancestry is a major factor in Tractor's loss of power in situations in which causal allele frequencies differ. Thus, the performance of a method which includes effect size heterogeneity could potentially be considerably improved if local ancestry were not included in the null hypothesis. We leave assessment of the power and calibration of this type of hybrid method for future work.

We conclude with caveats and limitations of our work. When hoping to understand these patterns of power for association statistics, there are many combinations of different elements of genetic architecture to consider. These include phenotypic factors such as environmental variance and polygenicity, as well as elements of admixture such as the number of generations of admixture and the strength of linkage disequilibrium. We could not consider them all, and thus it is likely that additional nuances to our findings exist when other factors are considered. One major element not considered in this work is case-control traits. While we chose to focus on quantitative traits in this analysis due to their simplicity and ubiquity, case-control traits are also important in medicine. It is possible that the behavior of these phenotypes will vary compared to the quantitative traits that we analyze here, both in simulations and real data. We suggest case-control traits as an interesting avenue of research for future works. Lastly, we chose to focus our analyses on standard GWASs and Tractor due to their popularity and ease of use. We compare how these methods work

‘‘out of the box’’ to provide simple and usable guidance for others. However, as discussed in the introduction to this work, a variety of other association tests exist. It is likely that in certain circumstances one of these existing methods would outperform both a standard GWAS and Tractor.

From both scientific and social perspectives, it is important that admixed populations are incorporated more effectively into genetic studies. By providing insight into the strengths and limitations of these methods, we hope to enable studies to maximize their power in admixed populations.

3.5 TABLES

Table 3.1: **Summary of GWAS association statistics**

Association statistic	Statistical test (H_0)	Assumptions on β	Ancestry-related covariates	Degrees of freedom
ADM	$e_l = 0$	-	α	1
Standard GWAS	$\beta = 0$	$\beta_1 = \beta_2 = 0$	α	1
SNP1	$\beta = 0$	$\beta_1 = \beta_2 = 0$	l, α	1
MIX	$e_l \circ \beta = 0$	$\beta_1 = \beta_2 = 0$	α	1
SUM	$\beta = 0$ and $e_l = 0$	$\beta_1 = \beta_2 = 0$	α	2
Tractor	$\beta_1 = 0$ and $\beta_2 = 0$	-	l, α	2

All tests adjust for global ancestry and can be used on binary traits, and all tests except MIX can be implemented with adjustment for additional covariates and use on quantitative traits. For more information on the comparison of standard GWAS, ADM, SUM, and MIX, see Pasaniuc et al.[3.14] and Seldin et al.[3.22] We note that while additional methods exist,[3.36–3.39] we do not focus on them in this work because they do not directly relate to Equation 3.1.

Table 3.S1: **Number of independent significant loci by phenotype**

Phenotype	**# loci standard GWAS**	**# loci tractor**	**# loci shared**
cholesterol	3	2	2
erythrocyte	3	3	2
Height	0	1	0
LDL	4	3	3
log(AST)	1	1	0
log(BMI)	1	0	0
log(HDL)	5	0	0
log(leukocyte)	1	0	0
log(lymphocyte)	0	0	0
log(monocyte)	1	0	0
log(platelets)	0	0	0
log(triglycerides)	0	0	0

Table 3.S2: **Independent significant SNPs in UKBB admixed population**

Phenotype	**SNP (reference allele / alternate allele)**	**Standard GWAS p-value**	**Tractor p-value**
cholesterol	chr1:55054772 (A / G)	3.72×10^{-8}	not significant
cholesterol	chr8:118543713 (A / T)	1.19×10^{-9}	8.31×10^{-9}
cholesterol	chr19:44908822 (C / T)	1.22×10^{-31}	2.31×10^{-30}
erythrocyte	chr16:261108 (G / A)	5.44×10^{-26}	not significant
erythrocyte	chr16:360054 (A / G)	9.15×10^{-13}	not significant
erythrocyte	chr16:50884914 (A / T)	4.92×10^{-10}	not significant
erythrocyte	chr16:117409 (C / T)	not significant	3.47×10^{-18}
erythrocyte	chr16:260355 (C / T)	not significant	6.34×10^{-18}
erythrocyte	chr16:384271 (G / A)	not significant	2.33×10^{-11}
Height	chr7:78824856 (G / A)	not significant	7.79×10^{-9}
LDL	chr1:55063542 (C / A)	2.47×10^{-11}	1.14×10^{-10}
LDL	chr1:88869866 (G / A)	3.01×10^{-8}	not significant
LDL	chr8:118543713 (A / T)	5.74×10^{-9}	2.45×10^{-8}
LDL	chr19:44908822 (C / T)	3.58×10^{-50}	6.24×10^{-49}

log(AST)	chr10:17819068 (G / A)	5.03×10^{-11}	not significant
log(AST)	chr19:17024164 (C / T)	not significant	2.30×10^{-8}
log(BMI)	chr3:196672134 (G / A)	4.83×10^{-8}	not significant
log(HDL)	chr15:76063105 (G / A)	1.54×10^{-8}	not significant
log(HDL)	chr16:56957451 (C / T)	1.34×10^{-8}	not significant
log(HDL)	chr17:58519260 (G / A)	4.30×10^{-8}	not significant
log(HDL)	chr17:58607316 (C / G)	4.94×10^{-8}	not significant
log(HDL)	chr17:58744530 (C / T)	4.94×10^{-8}	not significant
log(leukocyte)	chr14:30683993 (A / G)	4.96×10^{-8}	not significant
log(monocyte)	chr1:159092646 (G / A)	2.21×10^{-8}	not significant

3.6 FIGURES

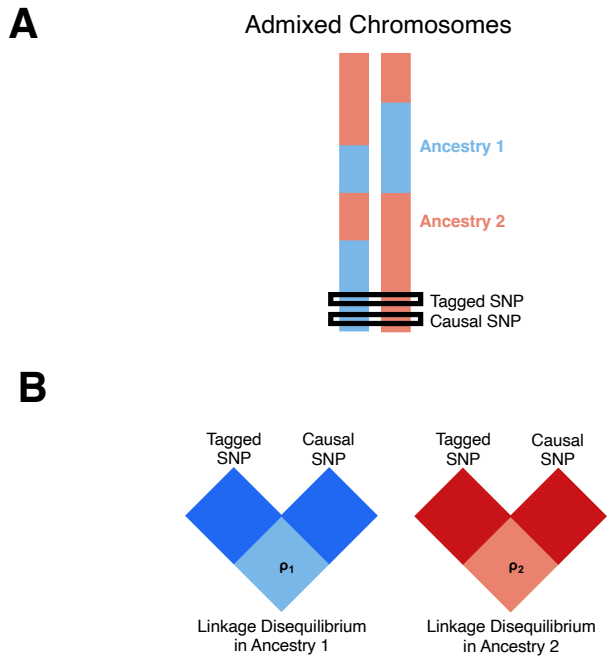

Figure 3.1: **Toy example of how differential LD by local ancestry can induce HetLanc**. (A) Admixed populations contain haplotypes with different local ancestry at the causal or tagged SNP. (B) The correlation between tagged and causal SNPs depends on their local ancestry due to differential LD by local ancestry. (C) In a GWAS, the estimated marginal SNP effect size is proportional to the true causal effect size and the correlation between the tagged and causal SNPs ($\widehat{\beta_{tag,i}} \propto \rho_i \beta_{causal,i}$) where i refers to the i^{th} ancestry).

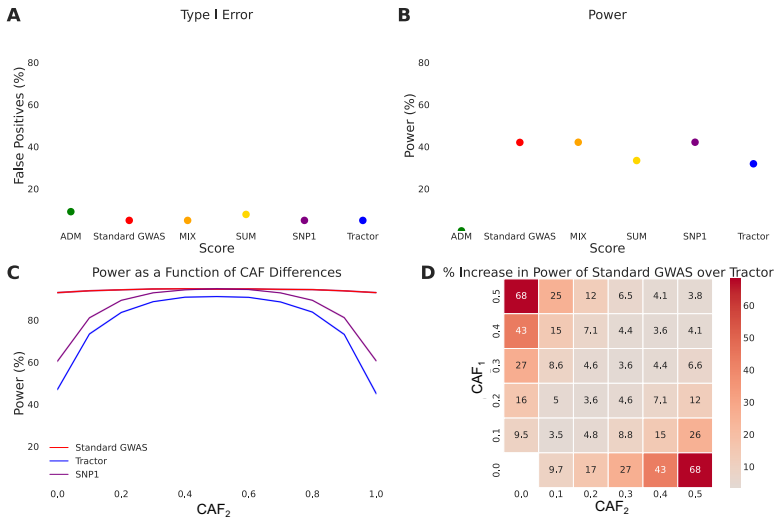

Figure 3.2: **Association statistics in the absence of HetLanc** . (A) Type I error for association statistics. Type I error calculated as the percent of null SNPs with a significant association detected. 95% confidence interval too narrow for display. (B) Power for association statistics. Power calculated as the percent of simulations to successfully recover the causal variant. Odds ratios $OR_1 = OR_2 = 1.2$. 95% confidence interval too narrow for display. (C) Power for a standard GWAS, SNP1, and Tractor as CAF_2 is varied between 0.0 and 1.0 and CAF_1 is fixed at 0.5. Power for all three methods varies as CAF difference varies. 95% confidence interval too narrow for display. (D) Heatmap of percent increase in power of a standard GWAS over Tractor when $\beta_1 = \beta_2 = 1.0$. Causal allele frequencies CAF_1 and CAF_2 varied from 0.0 to 0.5 in increments of 0.1. All simulations are for case-control (A and B) or quantitative (C and D) traits simulated 1,000 times for a population of 10,000 individuals with 100 genotypes each with global ancestry proportion 50/50. Power calculated using (A) nominal threshold p-value < 0.05, (B) Bonferroni-corrected

threshold p-value < $1x10^{-5}$, or (C and D) standard threshold p-value < $5x10^{-8}$. (A and B) Case-control traits have case-control ratio 1:1, 10% case prevalence, and $CAF_1 = CAF_2 = 0.5$. (C and D) Quantitative traits have heritability $h^2 = 0.005$. Heritability, global ancestry, causal effect size β, and overall CAF do not qualitatively impact these results (Figures S1–S3).

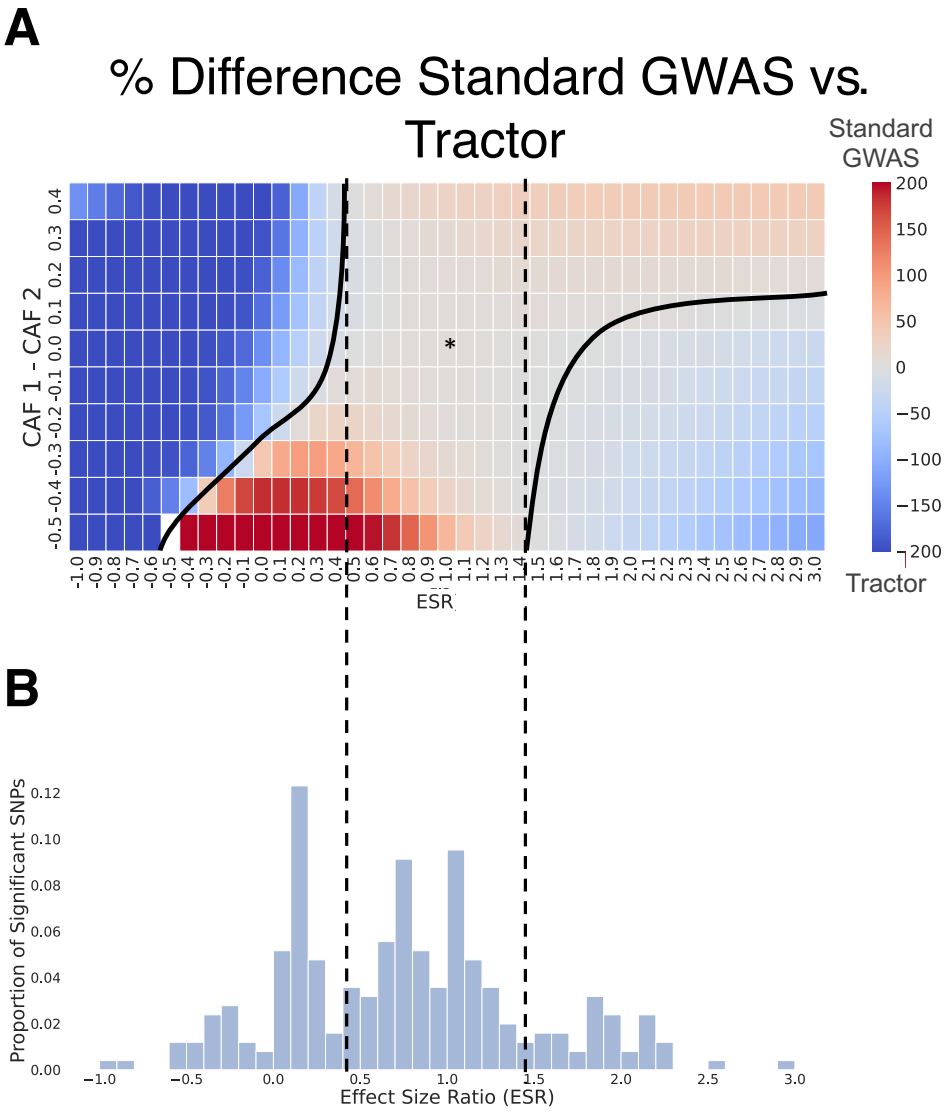

Figure 3.3: **Impact of HetLanc on percent difference in power depends on CAF difference**. (A) Heatmap of percent difference in power for a standard GWAS versus Tractor. The ‘‘*’’ indicates the center with no HetLanc or CAF difference. The solid line represents the boundary between when a standard GWAS and Tractor have higher power. The dashed line represents the region in which a standard GWAS always has higher power than Tractor. Quantitative trait simulated 1,000 times for a population of 10,000 individuals on a trait with effect size β_1 ranging from 1.0 to 3.0 in increments of 0.1, and effect size β_2 = 1.0. Global ancestry proportion 50/50, heritability at $h^2 = 0.005$, and causal allele frequencies $CAF_1 = 0.5$ and CAF_2 ranging from 0.1 to 1.0 in increments of 0.1. Power calculated using a standard threshold p-value < $5x10^{-8}$. (B) Histogram of empirical $R_{het} = \frac{\widehat{\beta_1}}{\widehat{\beta_2}}$ for significant SNPs found for 12 phenotypes in the UKBB. $\widehat{\beta_1}, \widehat{\beta_2}$ estimated using Tractor.

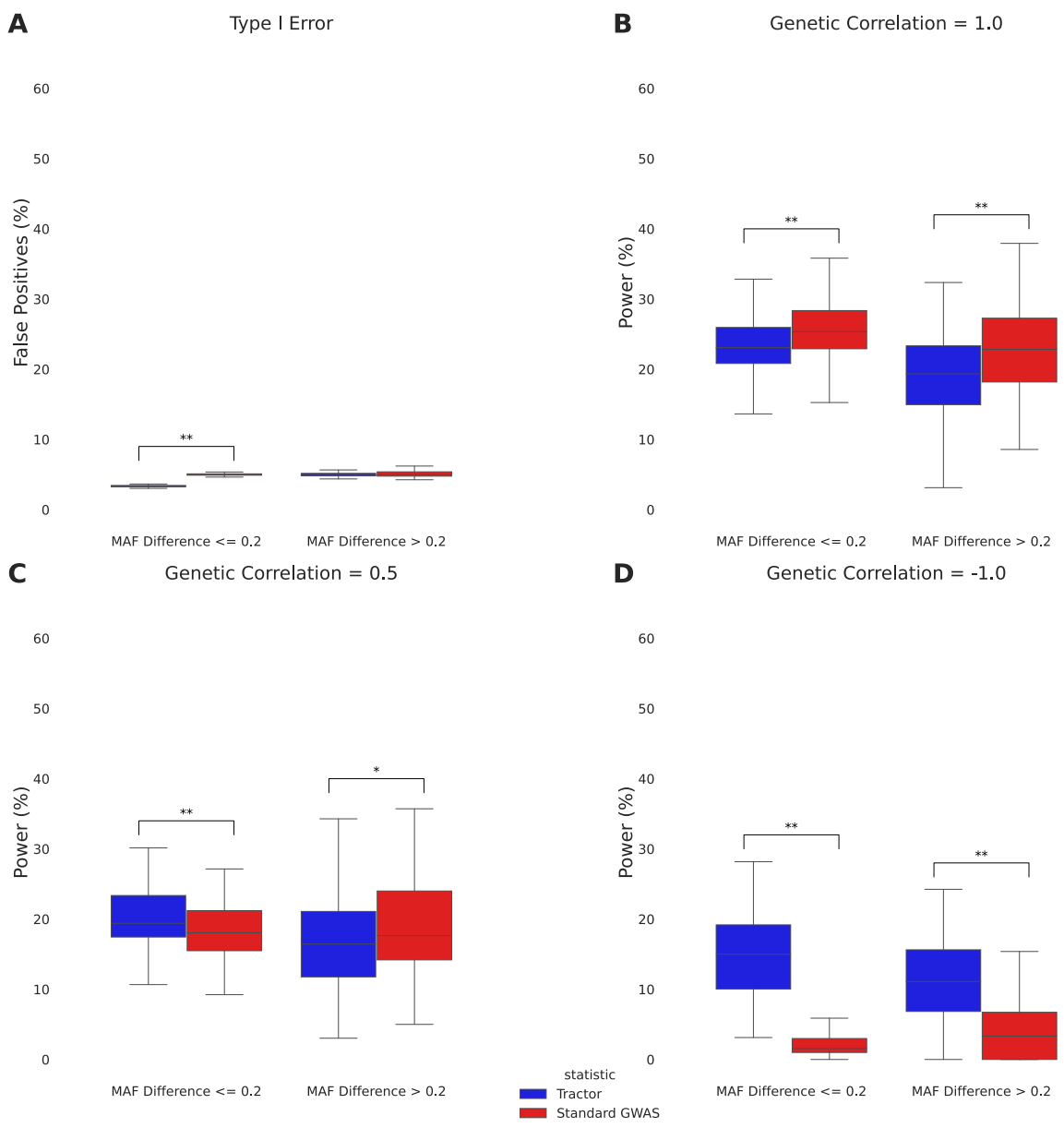

Figure 3.4: **Effect size heterogeneity in the context of polygenicity**. (A) Boxplot of type I error for Tractor and a standard GWAS split by non-differentiated (MAF difference < 0.2) and differentiated (MAF difference > 0.2) SNPs. (B) Boxplot of power for Tractor and a standard GWAS in the case of no effect size heterogeneity split by non-differentiated and differentiated

SNPs. (C) Boxplot of power for Tractor and a standard GWAS in the case of effect size heterogeneity split by non-differentiated and differentiated SNPs. (D) Boxplot of power for Tractor and a standard GWAS in the case of opposite effect sizes split by non-differentiated and differentiated SNPs. All simulations used real UKBB admixed genotypes and simulated phenotypes with 100 causal SNPs and a total additive genetic heritability of $h^2 = 0.5$ (see subjects and methods). ‘‘*’’ indicates a nominally significant p-value ($<1.28 \times 10^{-3}$). The boxes show the inter-quartile range while the whiskers show the rest of the distribution (not including outliers).

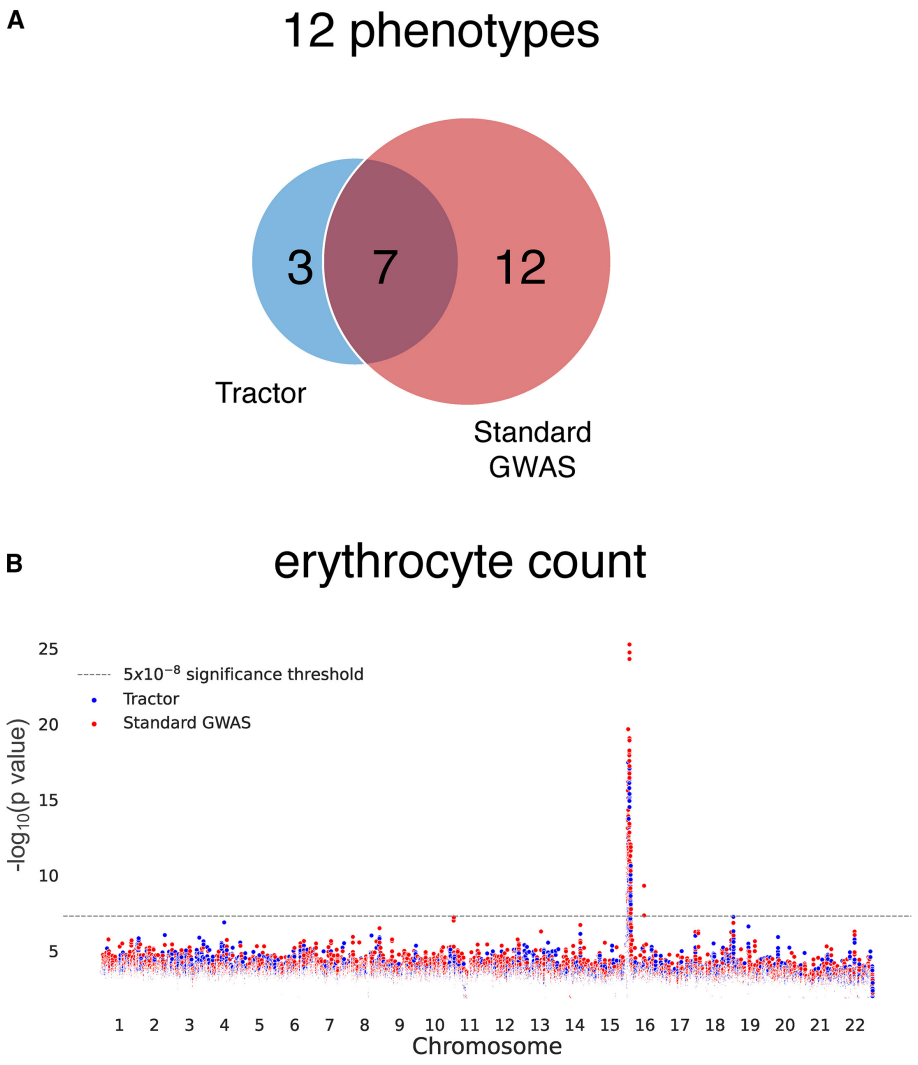

Figure 3.5: **Comparing significant SNPs found with a standard GWAS and Tractor** . (A) Venn diagram of independent significant loci found using a standard GWAS and Tractor in the UKBB across 12 quantitative traits. (B) Manhattan plot of erythrocyte count in the UKBB. Significant SNPs found with a standard GWAS shown in red and significant SNPs found with Tractor shown in blue. Manhattan plot SNPs shown filtered for p value < 0.01 and SNPs are plotted based on post-filter indices.

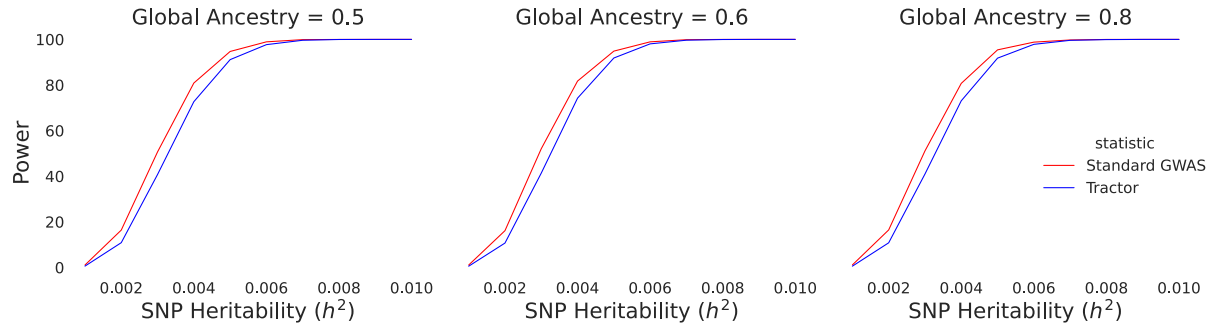

Figure 3.S1: **Global ancestry does not have a large impact on power compared to the choice of test statistic and SNP heritability.** Power curves of Standard GWAS and Tractor as SNP heritability varies. In this case where neither frequency nor causal effect size vary by local ancestry, Standard GWAS has increased power over Tractor, especially at small levels of SNP heritability. Simulation results of 1,000 replicates with N = 10,000 individuals with causal allele frequency $CAF_1 = CAF_2 = 0.5$, and causal effect sizes $\beta_1 = \beta_2 = 1.0$. 95% confidence interval too narrow for display.

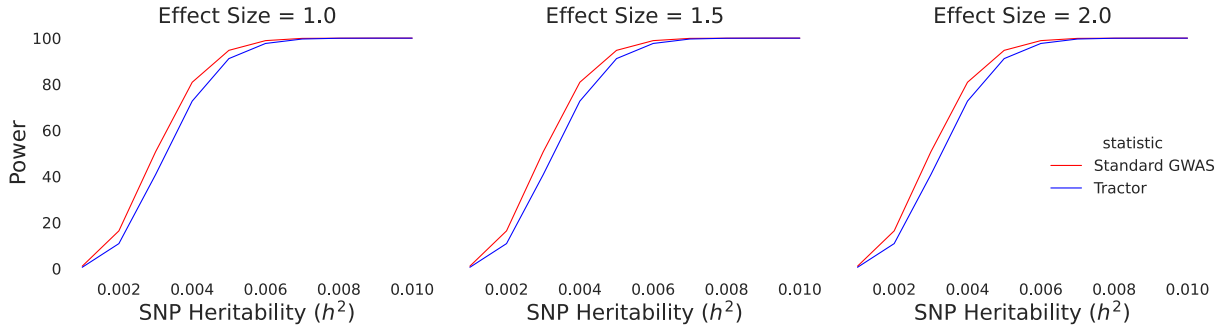

Figure 3.S2: **Effect size does not have a large impact on power compared to the choice of test statistic and SNP heritability.** Power curves of Standard GWAS and Tractor as SNP heritability varies. In this case where neither frequency nor causal effect size vary by local ancestry, Standard

GWAS has increased power over Tractor, especially at small levels of SNP heritability. Simulation results of 1,000 replicates with N = 10,000 individuals with causal allele frequency $CAF_1 = CAF_2 = 0.5$, global ancestry proportions at 50/50, and causal effect sizes $\beta_1 = \beta_2$. 95% confidence interval too narrow for display.

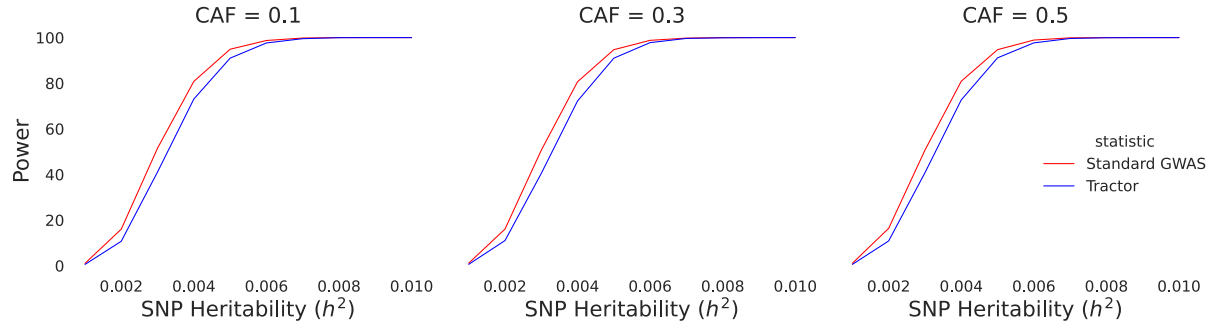

Figure 3.S3: **Causal allele frequency does not have a large impact on power compared to the choice of test statistic and SNP heritability.** Power curves of Standard GWAS and Tractor as SNP heritability varies. In this case where neither frequency nor causal effect size vary by local ancestry, Standard GWAS has increased power over Tractor, especially at small levels of SNP heritability. Simulation results of 1,000 replicates with N = 10,000 individuals with causal allele frequency $CAF_1 = CAF_2$, global ancestry proportions at 50/50, SNP heritability $h^2 = 0.005$, and causal effect sizes $\beta_1 = \beta_2 = 1.0$. 95% confidence interval too narrow for display.

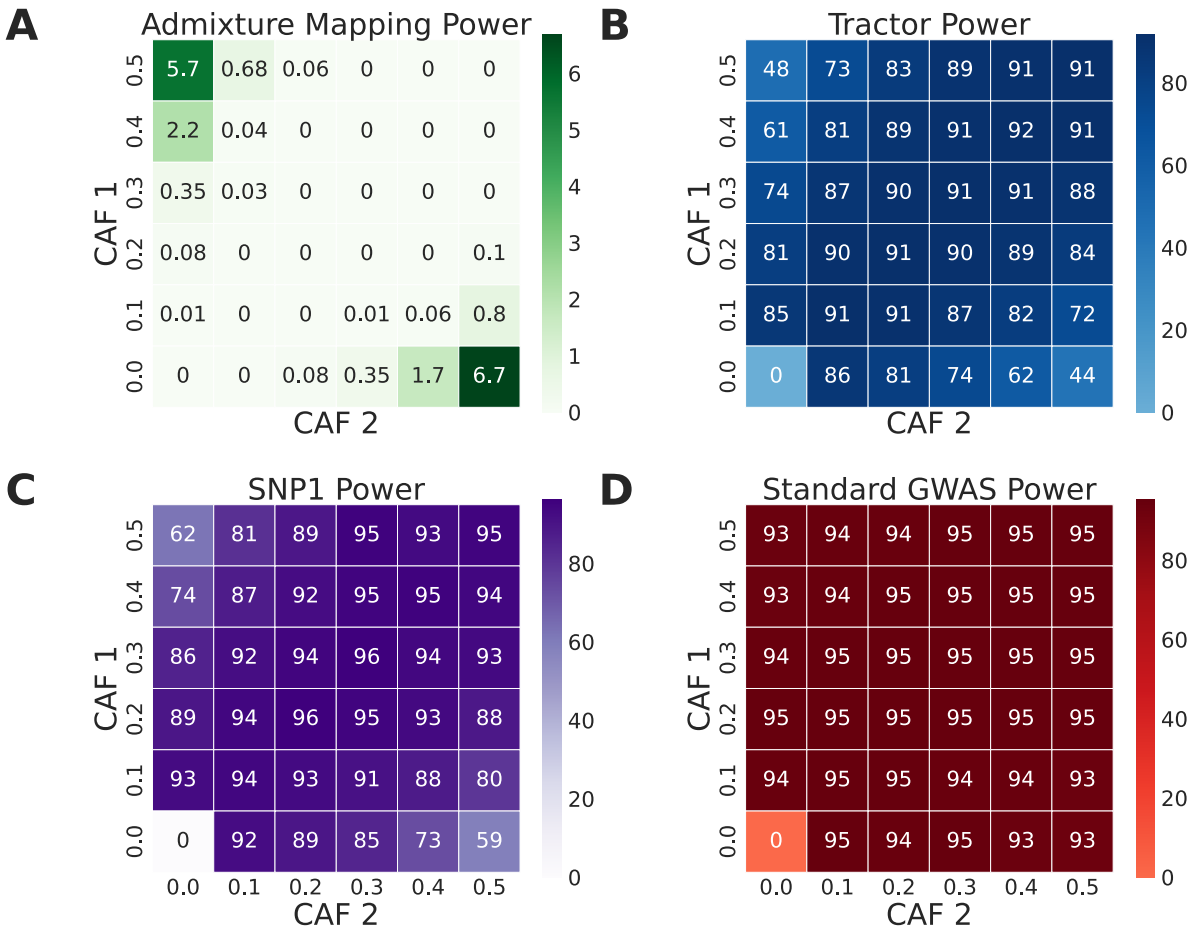

Figure 3.S4: **Association statistic power at differing levels of causal allele frequency difference. (a)** Admixture mapping has maximum power when causal allele frequency difference by local ancestry is increased. **(b)** Tractor has drastically decreased power when causal allele frequency difference by local ancestry is increased. In this case where causal effect size does not vary by local ancestry, the decrease in Tractor power at high levels of minor allele frequency difference by local ancestry is driven by the increase in power for admixture mapping, which serves as the null hypothesis against which Tractor tests SNP-level effects. **(c)** SNP1 has higher power than Tractor generally but also suffers from drastically decreased power when causal allele

frequency difference by local ancestry is increased, likely due to its identical null hypothesis. **(d)** Standard GWAS has slightly decreased power when causal allele frequency difference by local ancestry is increased. Standard GWAS does not suffer from using ADM as its null hypothesis as Tractor does, but the decrease in power is likely due to increased correlation between global and local ancestry at high levels of allele frequency difference. All panels are simulation results of 1,000 replicates with N = 10,000 individuals with global ancestry proportions at 50/50, SNP heritability h^2 = 0.005, and causal effect sizes $\beta_1 = \beta_2 = 1.0$.

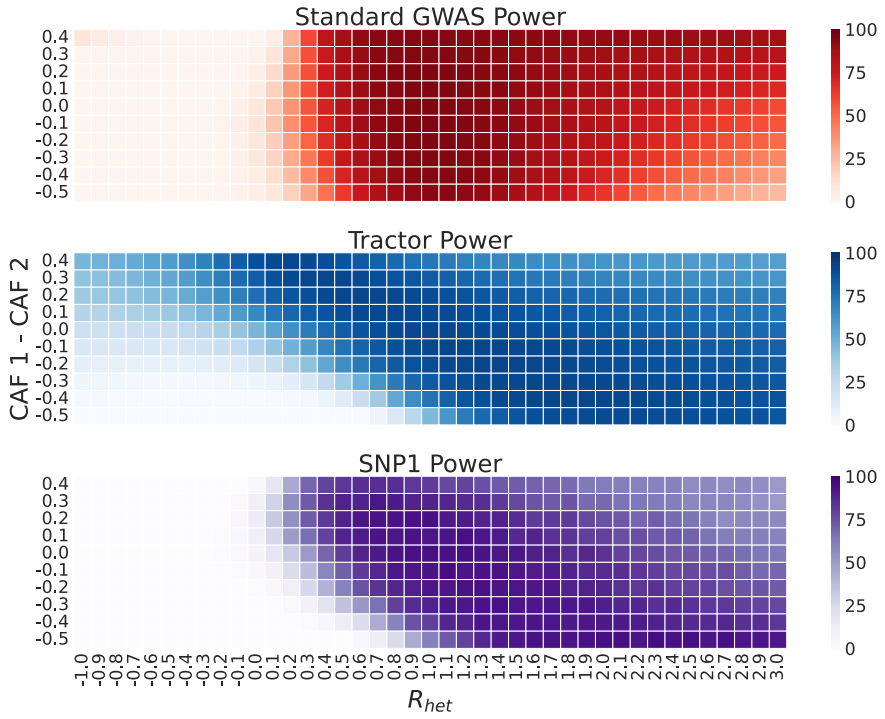

Figure 3.S5: **Impact of HetLanc and CAF difference on power of Standard GWAS, Tractor, and SNP1 individually.** As HetLanc increases, Standard GWAS power decreases, especially when causal effects are in opposite directions. CAF difference impacts Tractor and SNP1 more drastically than Standard GWAS. Simulation results of 1,000 replicates with N = 10,000 individuals with minor allele frequency CAF_1 = 0.5, global ancestry proportions at 50/50,

heritability h^2 = 0.005, and causal effect size β_2 = 1.0.

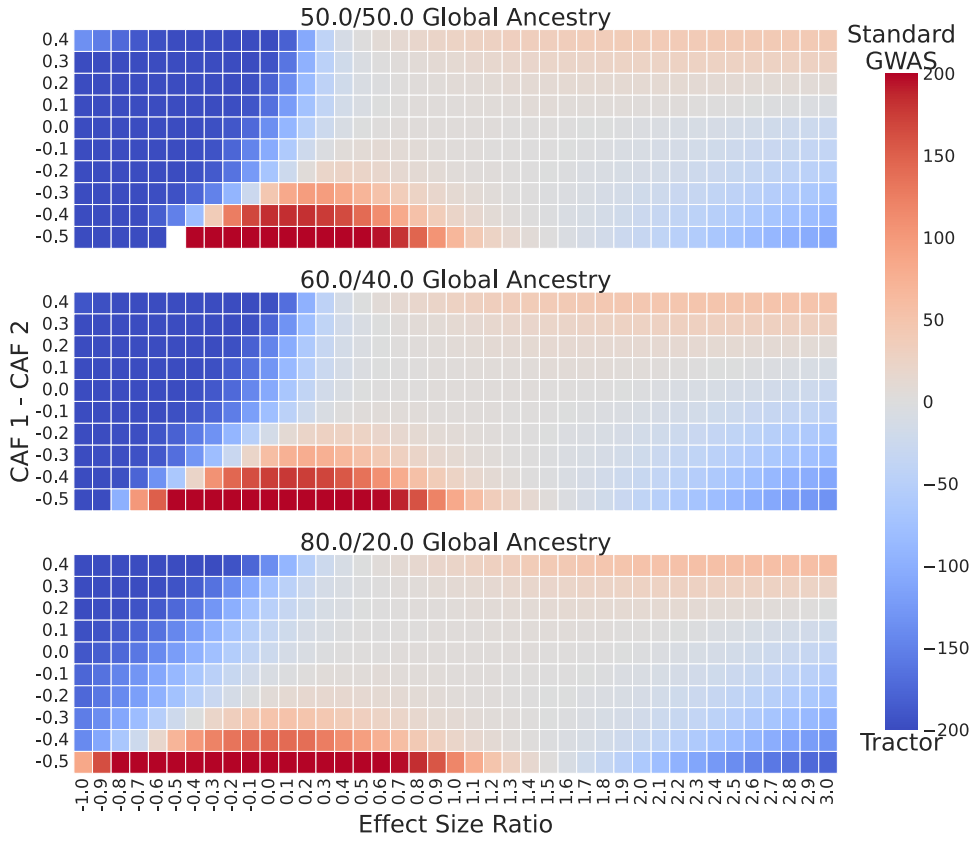

Figure 3.S6: **Impact of HetLanc and CAF difference on percent difference in power depends on global ancestry ratios**. Heatmap of percent difference in power for Standard GWAS vs Tractor. Red indicates where $\text{Power}_{\text{Standard GWAS}} > \text{Power}_{\text{Tractor}}$. As global ancestry ratios become further from 50%, the range of HetLanc and CAF difference in which Standard GWAS has more power than Tractor increases. Simulation results of 1,000 replicates with N = 10,000 individuals with minor allele frequency $\text{CAF}_1 = 0.5$, heritability h^2 = 0.005, and causal effect size $\beta_2 = 1.0$.

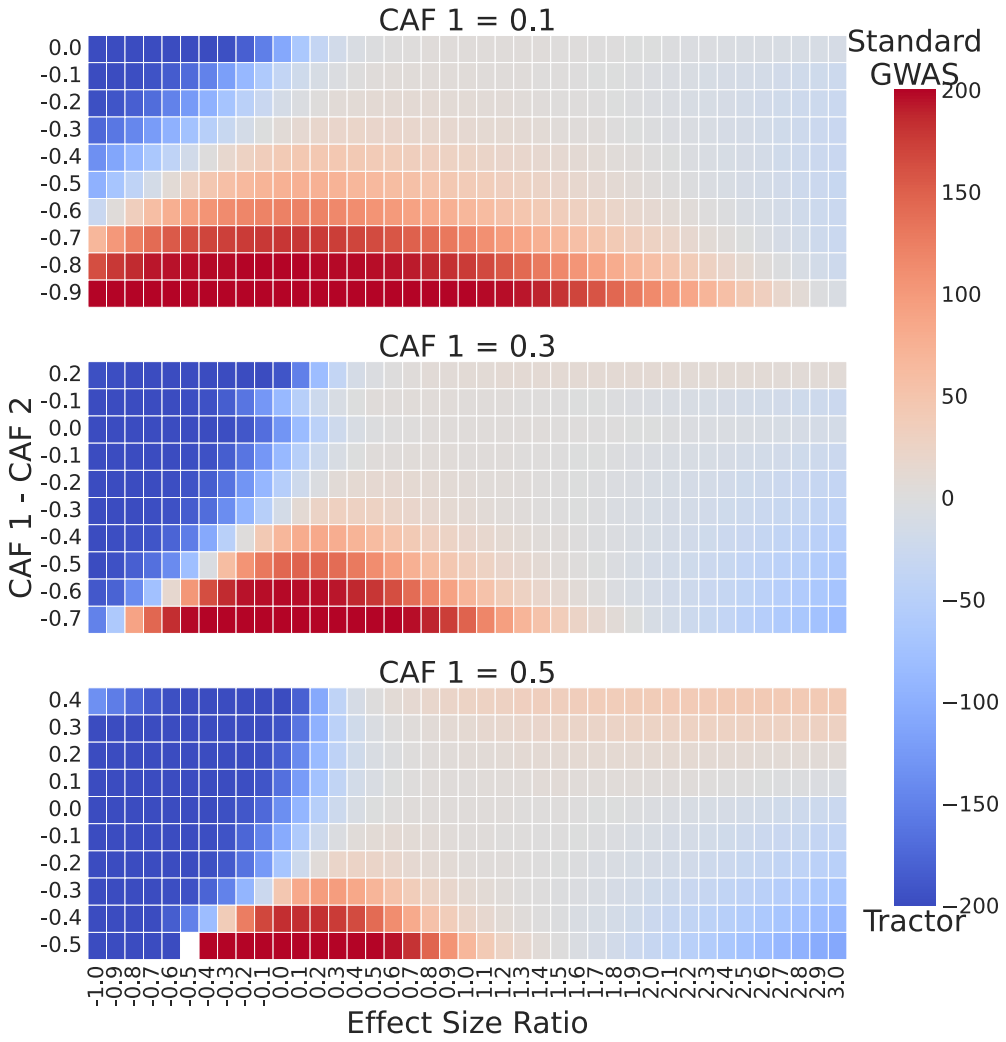

Figure 3.S7: **Impact of HetLanc and CAF difference on percent difference in power depends on CAF.** Heatmap of percent difference in power for Standard GWAS vs Tractor. Red indicates where $Power_{Standard\ GWAS} > Power_{Tractor}$. As CAF becomes further from 0.5, the range of HetLanc and CAF difference in which Standard GWAS has more power than Tractor increases. Simulation results of 1,000 replicates with N = 10,000 individuals with global ancestry

proportions at 50/50, SNP heritability h^2 = 0.005, and causal effect size β_2 = 1.0.

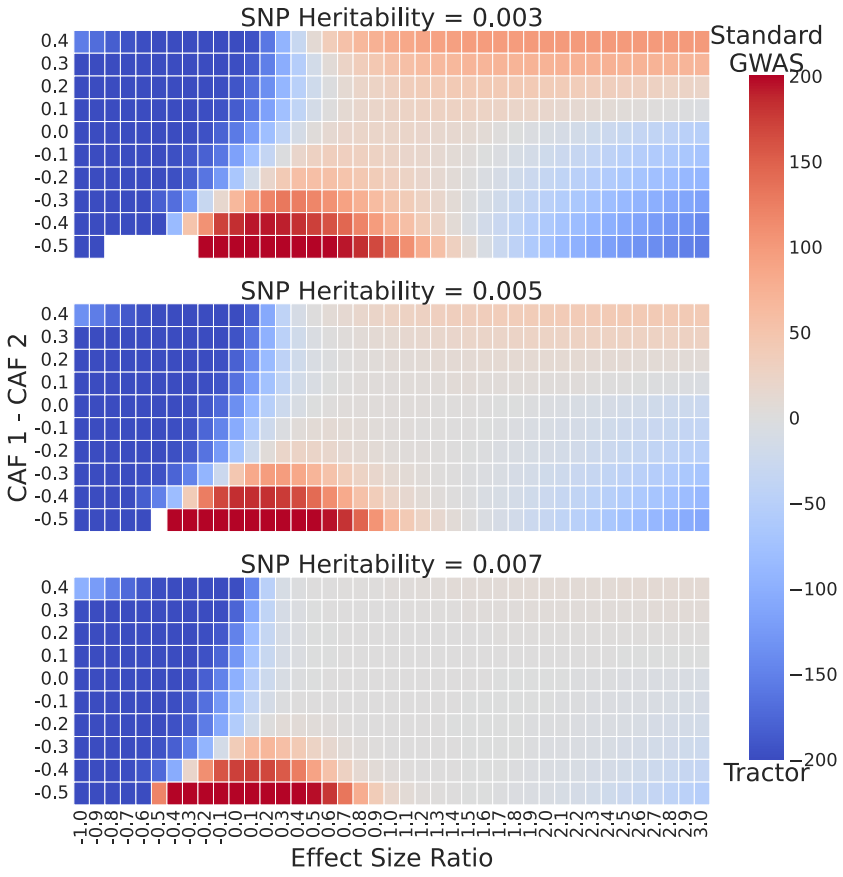

Figure 3.S8: **Impact of HetLanc and CAF difference on percent difference in power depends on heritability.** Heatmap of percent difference in power for Standard GWAS vs Tractor. Red indicates where $\text{Power}_{\text{Standard GWAS}} > \text{Power}_{\text{Tractor}}$. As heritability decreases, the percent difference in power between Standard GWAS and Tractor increases. Simulation results of 1,000 replicates with N = 10,000 individuals with causal allele frequency CAF_1 = 0.5, global ancestry proportions at 50/50, and causal effect size β_2 = 1.0.

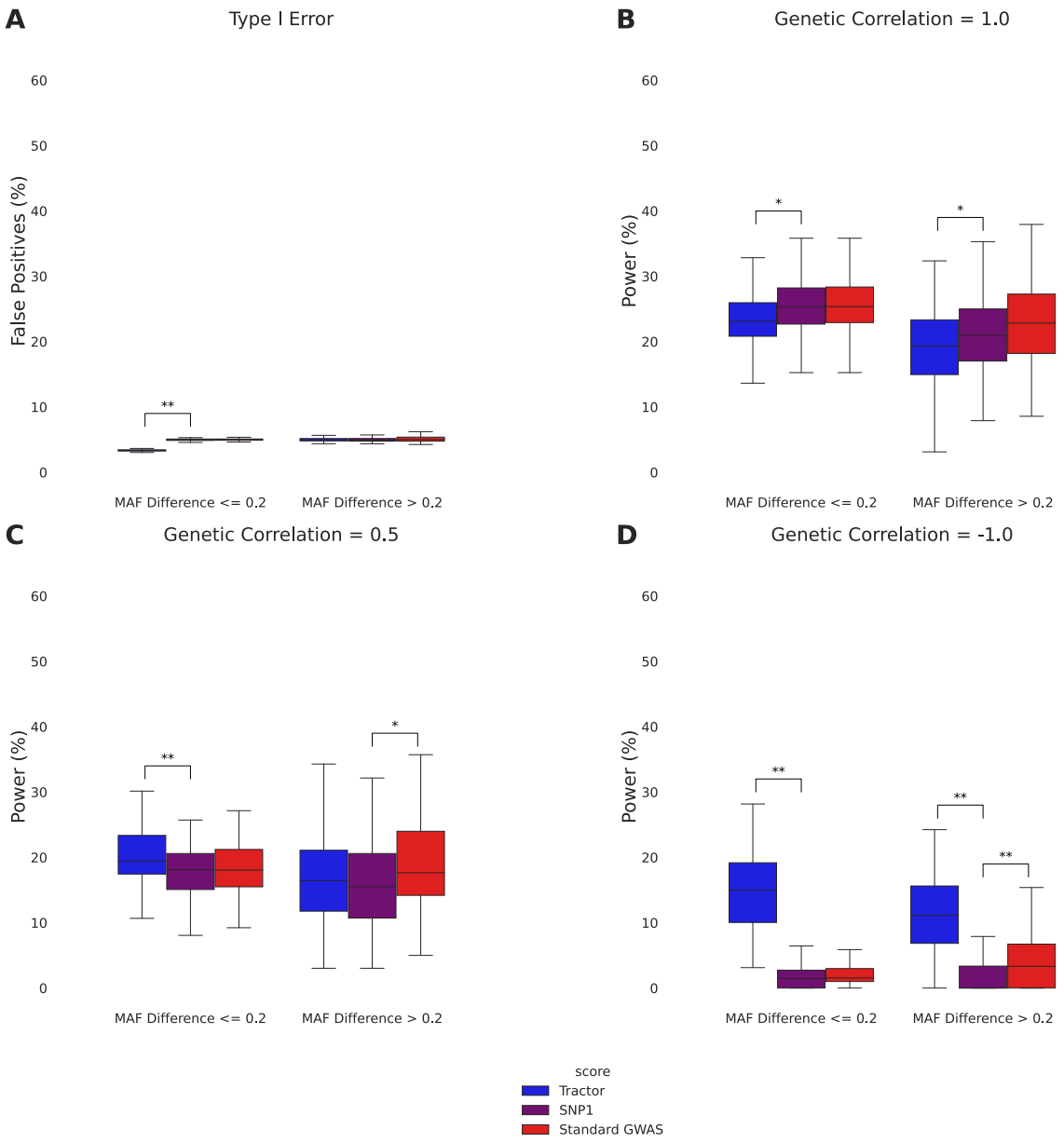

Figure 3.S9: **Effect Size Heterogeneity of Tractor, SNP1, and Standard GWAS in the Context of Polygenicity. (a)** Box plot of Type I error for Tractor, SNP1, and Standard GWAS split by non-differentiated (MAF difference ≤ 0.2) and differentiated (MAF difference > 0.2) SNPs. **(b)** Box plot of power for Tractor, SNP1, and Standard GWAS in the case of no effect size heterogeneity

split by non-differentiated and differentiated SNPs. **(c)** Box plot of power for Tractor, SNP1, and Standard GWAS in the case of effect size heterogeneity split by non-differentiated and differentiated SNPs. **(d)** Box plot of power for Tractor, SNP1 and Standard GWAS in the case of opposite effect sizes split by non-differentiated and differentiated SNPs. **(a-d)** All simulations used real UKBB admixed genotypes and simulated phenotypes with 100 causal SNPs and a total additive genetic heritability of $h^2 = 0.5$ (see methods). "*" indicates a nominally significant p-value (<0.05). "**" indicates a Bonferroni-corrected significant p-value (<1.28 x 10^{-3}).

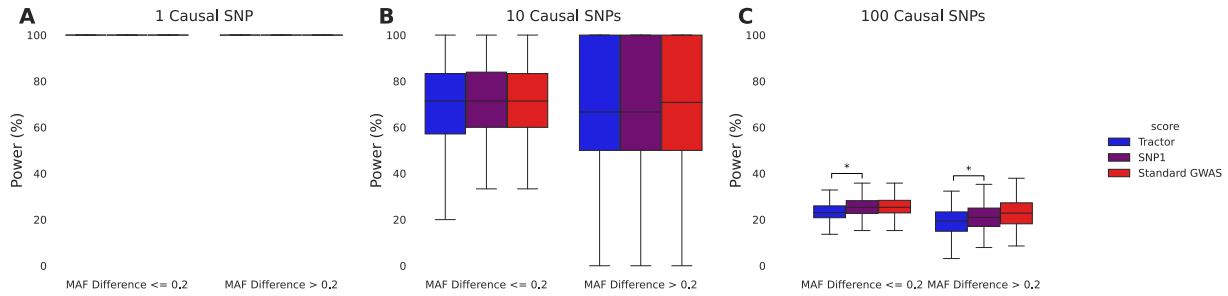

Figure 3.S10: **Effect Size Heterogeneity in the Context of Varying Levels of Polygenicity. (a)** Box plot of power for Tractor, SNP1, and Standard GWAS in the case of one causal SNP split by non-differentiated and differentiated SNPs. All methods had 100% power in this case due to a high SNP heritability of 50%. **(b)** Box plot of power for Tractor, SNP1, and Standard GWAS in the case of 10 causal SNPs split by non-differentiated and differentiated SNPs. **(c)** Box plot of power for Tractor, SNP1, and Standard GWAS in the case of 100 causal SNPs split by non-differentiated and differentiated SNPs. **(a-d)** All simulations used real UKBB admixed genotypes and simulated phenotypes with genetic correlation = 1.0 and a total additive genetic heritability of $h^2 = 0.5$ (see methods). "*" indicates a nominally significant p-value (<0.05). "**" indicates a Bonferroni-corrected significant p-value (<1.28 x 10^{-3}).

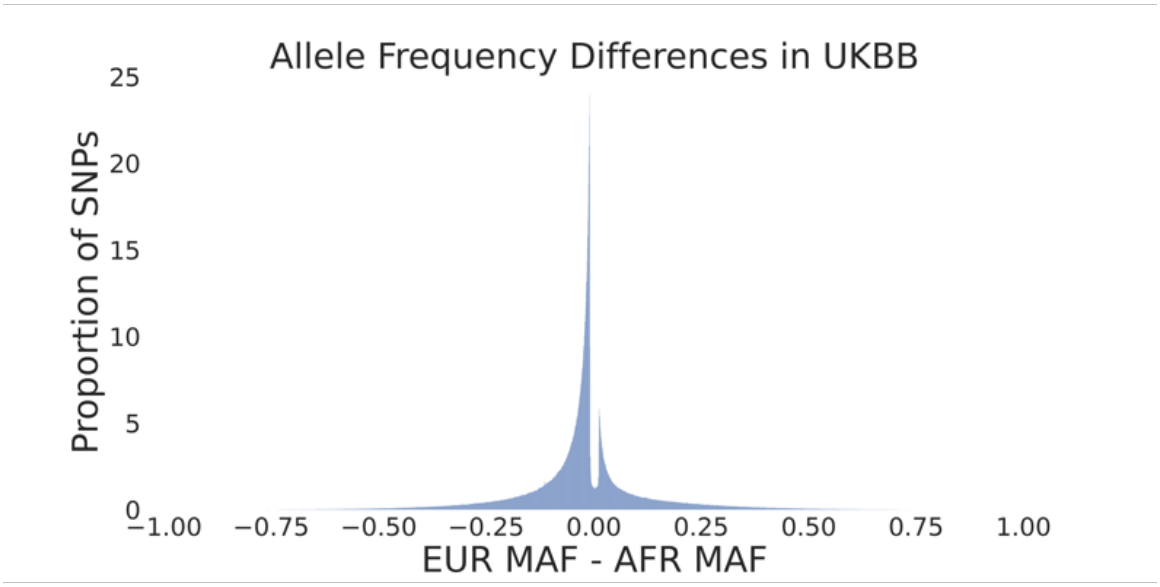

Figure 3.S11: **Minor allele frequency differences between European and African local ancestries in the African-European admixed population in the UKBB.** Minor allele frequency differences center near zero, at -2.39 x 10^{-2}, indicating only a small systematic bias towards larger minor allele frequencies in the African local ancestry segments. Mean absolute value of minor allele frequency differences is 9.59 x 10^{-2}, indicating a small average allele frequency difference, with a standard deviation of 1.15 x 10^{-1}. Study population is 4,327 individuals from the UK Biobank with on average 58.9% African and 41.1% European admixed ancestry.

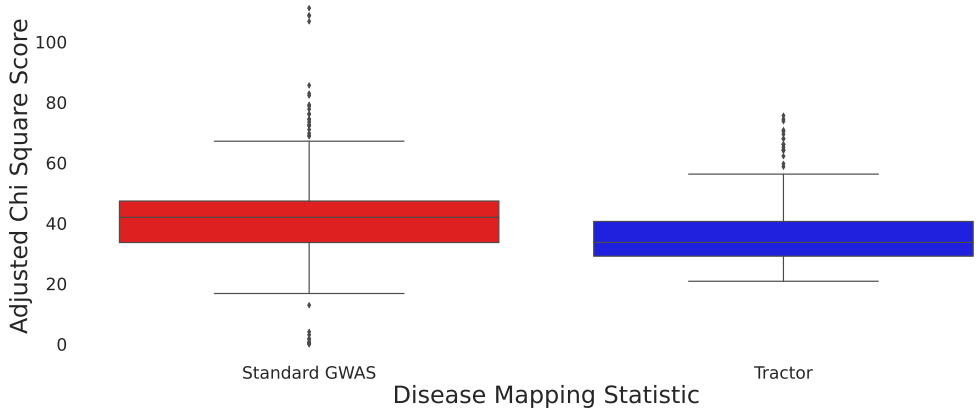

Figure 3.S12: **Adjusted Chi Square Statistics for significant SNPs for 12 traits in the UKBB.** Standard GWAS χ_1^2 is significantly larger than the Tractor statistic (adjusted from χ_2^2 to χ_1^2). Mean Standard GWAS χ_1^2 for significant SNPs is 42.9, mean Tractor χ_2^2 for significant SNPs is 37.5, p-value 2.11 x 10^{-4}. Study population is 4,327 individuals from the UK Biobank with on average 58.9% African and 41.1% European admixed ancestry. Tractor and Standard GWAS statistics computed over 16,584,433 SNPs and 12 traits including AST, BMI, cholesterol, erythrocyte count, HDL, height, LDL, leukocyte count, lymphocyte count, monocyte count, platelet count, and triglycerides. See methods for chi-square adjustment.

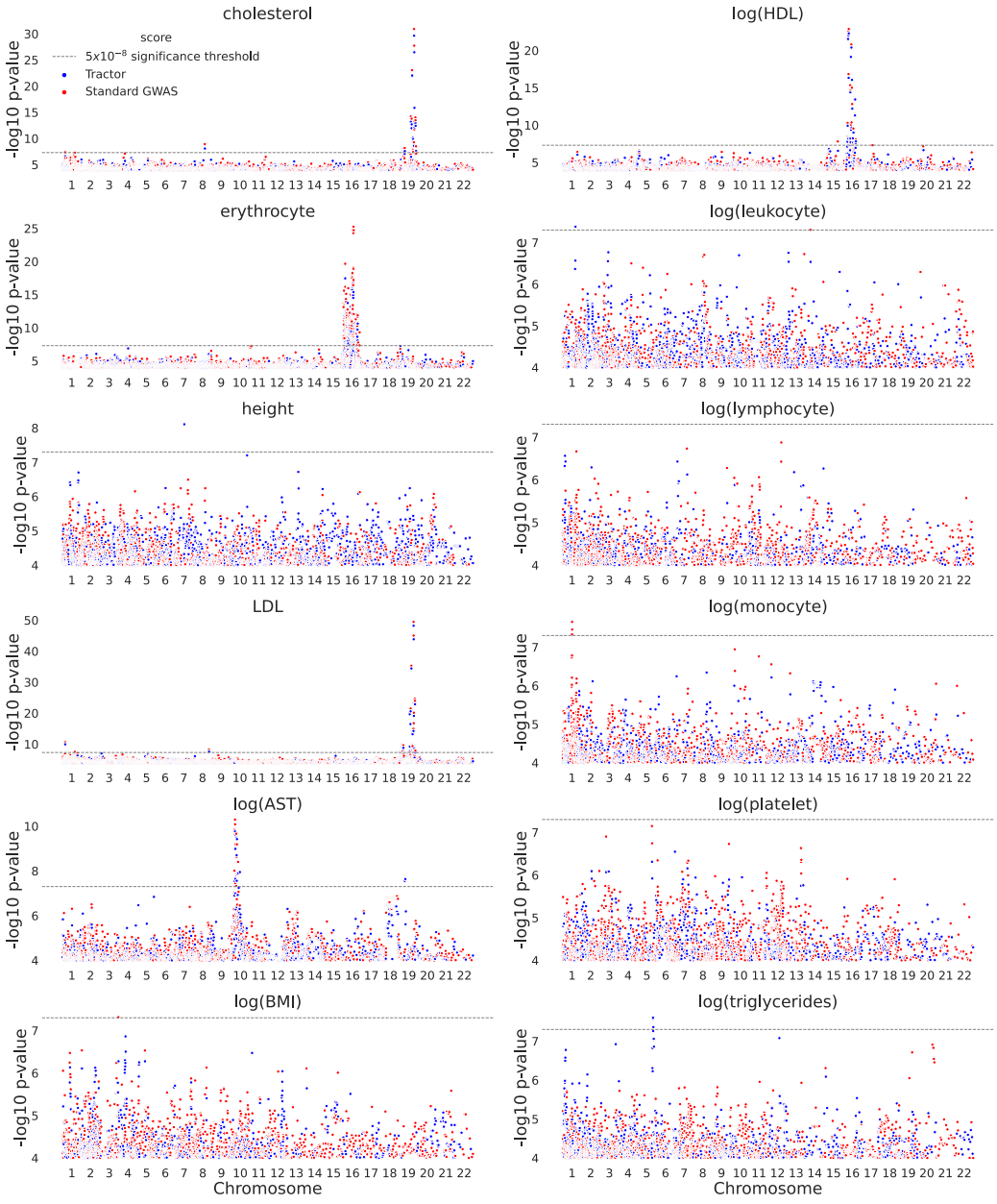

Figure 3.S13: **Manhattan plots for 12 quantitative traits in the UKBB African-European admixed population.** Study population is 4,327 individuals from the UK Biobank with on average 58.9% African and 41.1% European admixed ancestry. Manhattan plot SNPs shown filtered for p-value < 10^{-4} and SNPs are plotted based on post-filter indices.

3.7 REFERENCES

3.1 Tian, C., Gregersen, P.K., and Seldin, M.F. (2008). Accounting for ancestry: population substructure and genome-wide association studies. Hum. Mol. Genet. 17, R143–R150.

3.2 Mills, M.C., and Rahal, C. (2020). The GWAS Diversity Monitor tracks diversity by disease in real time. Nat. Genet. 52, 242–243.

3.3 Hou, K., Bhattacharya, A., Mester, R., Burch, K.S., and Pasaniuc, B. (2021). On powerful GWAS in admixed populations. Nat. Genet. 53, 1631–1633.

3.4 Martin, A.R., Gignoux, C.R., Walters, R.K., Wojcik, G.L., Neale, B.M., Gravel, S., Daly, M.J., Bustamante, C.D., and Kenny, E.E. (2017). Human demographic history impacts genetic risk prediction across diverse populations. Am. J. Hum. Genet. 100, 635–649.

3.5 Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L.T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., et al. (2018). The UK Biobank resource with deep phenotyping and genomic data. Nature 562, 203–209.

3.6 Ramirez, A.H., Sulieman, L., Schlueter, D.J., Halvorson, A., Qian, J., Ratsimbazafy, F., Loperena, R., Mayo, K., Basford, M., Deflaux, N., et al. (2022). The All of Us Research Program: data quality, utility, and diversity. Patterns 3, 100570.

3.7 Zhou, W., Kanai, M., Wu, K.H.H., Rasheed, H., Tsuo, K., Hirbo, J.B., Wang, Y., Bhattacharya, A., Zhao, H., Namba, S., et al. (2022). Global Biobank Meta-Analysis Initiative: Powering genetic discovery across human disease. Cell Genom. 2, 100192.

3.8 Rosenberg, N.A., Huang, L., Jewett, E.M., Szpiech, Z.A., Jankovic, I., and Boehnke, M. (2010). Genome-wide association studies in diverse populations. Nat. Rev. Genet. 11, 356–366.

3.9 Qin, H., Morris, N., Kang, S.J., Li, M., Tayo, B., Lyon, H., Hirschhorn, J., Cooper, R.S., and Zhu, X. (2010). Interrogating local population structure for fine mapping in genome-wide association studies. Bioinformatics 26, 2961–2968.

3.10 Zaitlen, N., Pasaniuc, B., Sankararaman, S., Bhatia, G., Zhang, J., Gusev, A., Young, T., Tandon, A., Pollack, S., Vilhja´lmsson, B.J., et al. (2014). Leveraging population admixture to characterize the heritability of complex traits. Nat. Genet. 46, 1356–1362.

3.11 Zhong, Y., Perera, M.A., and Gamazon, E.R. (2019). On using local ancestry to characterize the genetic architecture of human traits: genetic regulation of gene expression in multiethnic or admixed populations. Am. J. Hum. Genet. 104, 1097–1115.

3.12 Lin, M., Park, D.S., Zaitlen, N.A., Henn, B.M., and Gignoux, C.R. (2021). Admixed populations improve power for variant discovery and portability in genome-wide association studies. Front. Genet. 12, 673167.

3.13 Wojcik, G.L., Graff, M., Nishimura, K.K., Tao, R., Haessler, J., Gignoux, C.R., Highland, H.M., Patel, Y.M., Sorokin, E.P., Avery, C.L., et al. (2019). Genetic analyses of diverse populations improves discovery for complex traits. Nature 570, 514–518.

3.14 Pasaniuc, B., Zaitlen, N., Lettre, G., Chen, G.K., Tandon, A., Kao, W.H.L., Ruczinski, I., Fornage, M., Siscovick, D.S., Zhu, X., et al. (2011). Enhanced statistical tests

for GWAS in admixed populations: assessment using African Americans from CARe and a Breast Cancer Consortium. PLoS Genet. 7, e1001371.

3.15 Atkinson, E.G., Maihofer, A.X., Kanai, M., Martin, A.R., Karczewski, K.J., Santoro, M.L., Ulirsch, J.C., Kamatani, Y., Okada, Y., Finucane, H.K., et al. (2021). Tractor uses local ancestry to enable the inclusion of admixed individuals in GWAS and to boost power. Nat. Genet. 53, 195–204.

3.16 Smith, M.W., and O'Brien, S.J. (2005). Mapping by admixture linkage disequilibrium: advances, limitations, and guidelines. Nat. Rev. Genet. 6, 623–632.

3.17 Price, A.L., Zaitlen, N.A., Reich, D., and Patterson, N. (2010). New approaches to population stratification in genome-wide association studies. Nat. Rev. Genet. 11, 459–463.

3.18 Korunes, K.L., and Goldberg, A. (2021). Human genetic admixture. PLoS Genet. 17, e1009374.

3.19 Kang, S.J., Larkin, E.K., Song, Y., Barnholtz-Sloan, J., Baechle, D., Feng, T., and Zhu, X. (2009). Assessing the impact of global versus local ancestry in association studies. BMC Proc. 3, 1077–S116.

3.20 Shriner, D., Adeyemo, A., Ramos, E., Chen, G., and Rotimi, C.N. (2011). Mapping of disease-associated variants in admixed populations. Genome Biol. 12, 223–228.

3.21 Peterson, R.E., Kuchenbaecker, K., Walters, R.K., Chen, C.Y., Popejoy, A.B., Periyasamy, S., Lam, M., Iyegbe, C., Strawbridge, R.J., Brick, L., et al. (2019). Genome-

wide association studies in ancestrally diverse populations: opportunities, methods, pitfalls, and recommendations. Cell 179, 589–603.

3.22 Seldin, M.F., Pasaniuc, B., and Price, A.L. (2011). New approaches to disease mapping in admixed populations. Nat. Rev. Genet. 12, 523–528.

3.23 Hou, K., Ding, Y., Xu, Z., Wu, Y., Bhattacharya, A., Mester, R., Belbin, G.M., Buyske, S., Conti, D.V., Darst, B.F., et al. (2023). Causal effects on complex traits are similar for common variants across segments of different continental ancestries within admixed individuals. Nat. Genet. 55, 549–558.

3.24 Patel, R.A., Musharoff, S.A., Spence, J.P., Pimentel, H., Tcheandjieu, C., Mostafavi, H., Sinnott-Armstrong, N., Clarke, S.L., Smith, C.J., et al.; VA Million Veteran Program (2022). Genetic interactions drive heterogeneity in causal variant effect sizes for gene expression and complex traits. Am. J. Hum. Genet. 109, 1286–1297.

3.25 Marigorta, U.M., and Navarro, A. (2013). High trans-ethnic replicability of GWAS results implies common causal variants. PLoS Genet. 9, e1003566.

3.26 Shi, H., Gazal, S., Kanai, M., Koch, E.M., Schoech, A.P., Siewert, K.M., Kim, S.S., Luo, Y., Amariuta, T., Huang, H., et al. (2021). Population-specific causal disease effect sizes in functionally important regions impacted by selection. Nat. Commun. 12, 1098–1105.

3.27 Brown, B.C., Asian Genetic Epidemiology Network Type 2 Diabetes Consortium, Ye, C.J., Price, A.L., and Zaitlen, N. (2016). Transethnic genetic correlation estimates from summary statistics. Am. J. Hum. Genet. 99, 76–88.

3.28 Galinsky, K.J., Reshef, Y.A., Finucane, H.K., Loh, P.R., Zaitlen, N., Patterson, N.J., Brown, B.C., and Price, A.L. (2019). Estimating cross-population genetic correlations of causal effect sizes. Genet. Epidemiol. 43, 180–188.

3.29 Shi, H., Burch, K.S., Johnson, R., Freund, M.K., Kichaev, G., Mancuso, N., Manuel, A.M., Dong, N., and Pasaniuc, B. (2020). Localizing components of shared transethnic genetic architecture of complex traits from GWAS summary data. Am. J. Hum. Genet. 106, 805–817.

3.30 McKeigue, P.M. (1998). Mapping genes that underlie ethnic differences in disease risk: methods for detecting linkage in admixed populations, by conditioning on parental admixture. Am. J. Hum. Genet. 63, 241–251.

3.31 Mani, A. (2017). Local ancestry association, admixture mapping, and ongoing challenges. Circ. Cardiovasc. Genet. 10, e001747.

3.32 Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. Nat. Genet. 38, 904–909.

3.33 Liu, J., Lewinger, J.P., Gilliland, F.D., Gauderman, W.J., and Conti, D.V. (2013). Confounding and heterogeneity in genetic association studies with admixed populations. Am. J. Epidemiol. 177, 351–360.

3.34 Tang, H., Siegmund, D.O., Johnson, N.A., Romieu, I., and London, S.J. (2010). Joint testing of genotype and ancestry association in admixed families. Genet. Epidemiol. 34, 783–791.

3.35 Shriner, D., Adeyemo, A., and Rotimi, C.N. (2011). Joint ancestry and association testing in admixed individuals. PLoS Comput. Biol. 7, e1002325.

3.36 Wang, X., Zhu, X., Qin, H., Cooper, R.S., Ewens, W.J., Li, C., and Li, M. (2011). Adjustment for local ancestry in genetic association analysis of admixed populations. Bioinformatics 27, 670–677.

3.37 Zhang, J., and Stram, D.O. (2014). The role of local ancestry adjustment in association studies using admixed populations. Genet. Epidemiol. 38, 502–515.

3.38 Duan, Q., Xu, Z., Raffield, L.M., Chang, S., Wu, D., Lange, E.M., Reiner, A.P., and Li, Y. (2018). A robust and powerful two-step testing procedure for local ancestry adjusted allelic association analysis in admixed populations. Genet. Epidemiol. 42, 288–302.

3.39 Chen, W., Ren, C., Qin, H., Archer, K.J., Ouyang, W., Liu, N., Chen, X., Luo, X., Zhu, X., Sun, S., and Gao, G. (2015). A generalized sequential Bonferroni procedure for GWAS in admixed populations incorporating admixture mapping information into association tests. Hum. Hered. 79, 80–92.

3.40 Simonin-Wilmer, I., Orozco-Del-Pino, P., Bishop, D.T., Iles, M.M., and Robles-Espinoza, C.D. (2021). An overview of strategies for detecting genotype-phenotype associations across ancestrally diverse populations. Front. Genet. 12, 703901.

3.41 Martin, E.R., Tunc, I., Liu, Z., Slifer, S.H., Beecham, A.H., and Beecham, G.W. (2018). Properties of global-and local-ancestry adjustments in genetic association tests in admixed populations. Genet. Epidemiol. 42, 214–229.

3.42 Qin, H., and Zhu, X. (2012). Power comparison of admixture mapping and direct association analysis in genome-wide association studies. Genet. Epidemiol. 36, 235–243.

3.43 Hou K, Gogarten S, Kim J, Hua X, Dias JA, Sun Q, Wang Y, Tan T, Atkinson EG, Martin A. Admix-kit: an integrated toolkit and pipeline for genetic analyses of admixed populations. Bioinformatics. 2024 Apr 1;40(4):btae148.

4 METHODS TO REDUCE DIAGNOSTIC DELAY FOR RARE DISEASES ACROSS THE UNIVERSITY OF CALIFORNIA HEALTH SYSTEM

4.1 INTRODUCTION TO COMMON VARIABLE IMMUNODEFICIENCIES

Common variable immunodeficiency (CVID) is one of the more common types of primary immunodeficiency[4.1]. Patients with CVID often suffer from infection, inflammation, and auto-immunity[4.2]. However, while a patient's CVID diagnosis typically follows findings of low levels of multiple immunoglobulins (Ig), that patient's journey from onset of symptoms to administration of the test to indicate the disease is often long and winding[4.3]. This type of diagnostic delay impacts both the patient and the health system in which the patient participates[4.4]. Patients with underlying CVID but without a diagnosis may not receive the care that they need or receive only care to manage symptoms of the disease individually. This delay of up to 15 years[4.5] in some cases results in increased costs, frustration, and time on the part of both patient and physician, with the patient having to additionally bear the worsening of prognosis that is consistent with diagnostic delay[4.6].

One major reason for this diagnostic delay is that CVID is in fact a group of heterogeneous human inborn errors of immunity[4.7]. As such, the clinical presentation of CVID varies widely by patient[4.8]. Thus, it may be difficult for a physician who is not a specialist in primary immunodeficiencies to recognize that a patient's symptoms are indicative of CVID. The difficulty in recognizing a patient's symptoms as that of CVID is increased if a patient does not regularly visit the same physician, instead utilizing an emergency department or network of care. Many of the symptoms of CVID include health problems that are much more common than the underlying disease itself[4.9], which is estimated to occur in ~0.004%[4.10] of the population. However, when taken as a whole, the range, severity, and recurrence of these symptoms may point to some unifying underlying cause for a patient's symptoms. Diagnostic delay can also occur due to the broad range of body systems that may be impacted by a primary immunodeficiency. From cardiologists to

dermatologists, pulmonologists to ENTs, a CVID patient may be referred to any number of specialists to treat their symptoms before they are finally referred to immunology.

These reasons for diagnostic delay all have one thing in common – the physicians involved in the patient's care either don't have a full picture of that patient's healthcare journey, or they don't have enough experience with CVID to know to look for it. This is where the introduction of machine learning in electronic health records can be helpful.

Electronic health record (EHR) data is the data that comes from a patient's interactions with the health system. Diagnosis codes, laboratory values, medication prescriptions, and other types of health information are often stored in a structured, query-able fashion. EHR data has been used to aid in diagnosis and prediction tasks for a variety of clinical phenotypes[4.11-4.13]. CVID is another disease that can benefit from the introduction of machine learning to reduce diagnostic delay.

4.2 DEVELOPING THE PHENET ALGORITHM

4.2.1 PheNet at the University of California, Los Angeles

At the University of California, Los Angeles (UCLA), the electronic health records of patients of UCLA Health are made deidentified and available for research (with patient permission). Using this data, we constructed a machine learning algorithm to find patients likely to have CVID based on their EHR data[4.14].

Utilizing the 186 known cases of CVID in the UCLA DDR, we constructed a statistical model to learn the signature of CVID in EHR data, which could then be applied to patients with unknown disease status to ascertain their probability of having CVID as an underlying disease. In this process (Figure 4.1), we first found patients with putative immunodeficiencies using the D80

ICD code in the EHR. Next, physician chart review identified 186 true CVID patients. After identifying this case cohort, we found an additional 1,106 patients for a control cohort, matching cases on age, sex, race, and amount of data available (measured as time since first recorded visit).

Once we had the full case-control cohort, we chose clinical features known to be biologically associated with CVID. Starting with the Online Mendelian Inheritance in Man (OMIM)[4.15] database, we found OMIM codes known to be related to CVID, and mapped those to Human Phenotype Ontology (HPO)[4.16] terms. From there, we mapped the HPO terms to International Classification of Disease (ICD)[4.17] codes, which we further grouped into phecodes[4.18]. In total, we extracted 34 OMIM-derived phecodes from the EHR to be used as features in our model. In addition, we included immunoglobulin G (IgG) labs. Patient values for serum IgG levels were categorically assigned to 2-low (< 600 mg/dl), 1-normal (>600 mg/dl) or 0-unknown (no test recorded).

Using our 186 cases and 1,106 controls, we constructed a cross-validation set of 5 folds of 80% training, 20% testing split. We used the cross-validation folds to determine the best-performing model, using the mean area under the receiver operating characteristics curve (AUC-ROC) as our metric. We tested different model hyperparameters including down-sampling and up-sampling rates, as well as different feature sets such as inclusion or exclusion of the IgG category, and different model types such as log-inverse, marginal logistic regression, ridge regression, and random forest models. In addition, we tested using different numbers of phecodes in addition to the 34 OMIM-derived phecodes, from 0 to 20. In isolation, we found that the best performing model had an additional 5 phecodes (AUC-ROC 0.948), included IgG as a feature (AUC-ROC 0.946), had 50% up-sampling (AUC-ROC 0.946) and no down-sampling (AUC-ROC 0.948) (Figure 4.2). In total however, we found that the best performing model included 10

phecodes in addition to the 34 OMIM phecodes. We also found that while ridge regression had the highest AUC-ROC (0.961), marginal logistic regression achieved a similarly high AUC-ROC (0.946) while remaining the most interpretable method. Since our goal was for physicians to adopt this algorithm in clinical practice, we decided to prioritize interpretability and use a marginal logistic regression model.

Next, we sampled 10,000 patients randomly from the UCLA DDR to use as controls in the model and retrained the weights for the model. With this new cohort, we retrained the model using the previously determined hyperparameters. Under this paradigm, our model (PheNet) outperformed PheRS[4.19], a recently published method for finding patients with rare disease using EHR data (Figure 4.3). By comparison, our CVID specific risk score had a higher AUC-ROC (0.95) compared to the more generalized PheRS (AUC-ROC 0.79).

Next, we applied this algorithm to the rest of the DDR patients. We found the top 100 patients scored by PheNet and compared them to a random group of 100 patients. Using physician chart review, we found that patients ranked highly by PheNet were much more likely to have a true underlying CVID (Figure 4.4). In the top 100 PheNet ranked patients, 74% were categorized as at least a 3 out of 5 likelihood score for CVID (by manual physician chart review), compared to only 10% of the randomly chosen patients.

4.2.2 PheNet in the University of California Health Data Warehouse

The University of California Health Data Warehouse (UCHDW)[4.20] is an integrated, deidentified EHR database for the combined University of California (UC) system. This database includes ICD codes, medications, lab values, demographics, and more for the 8 million+ patients

seen at the health systems affiliated with UCLA (Los Angeles), UCSF (San Francisco), UCD (Davis), UCSD (San Diego), UCI (Irvine), and UCR (Riverside).

After the success of PheNet in the DDR at UCLA, we chose to extend the reach of our algorithm to the rest of the UC health system using the UCHDW. Assembling a group of immunologists from five of the UC health systems (UCLA, UCSF, UCD, UCSD, UCI), we identified a group of 575 bona fide CVID patients at the University of California (Table 4.1). This new cohort is predominantly female and of self-identified white race. We also note that the mean age of cases at UCSF is 10 years younger than that at other sites, which is expected given UCSF's focus on pediatric medicine. Our goal was to find patients ranking highly with PheNet from each institution, handing off the top 100 from each for a further chart review by immunologists and a potential referral to an immunology clinic.

Within this new cohort, we chose to update our model training process to avoid re-using data and better separate the training and testing process (Figure 4.5). After imposing quality control on our cases to ensure we had enough data on each one, we split our cohort of 565 remaining cases into an 80% training set and 20% testing set. We then selected 10 control patients per case from the UCHDW, matched on age, sex, race, amount of available data, and now site as well. Thus, from our 565 cases we created a training cohort of 452 cases and 5,027 controls as well as a testing cohort of 113 cases and 947 controls. All model training was conducted on our training cohort, but results reported here are using the held-out testing cohort. In this way, we avoid "double dipping" and seeing inflated accuracy metrics for our model.

We also chose to introduce some additional metrics for comparison between models. In addition to the AUC-ROC previously mentioned, we also consider the area under the precision

recall curve (AUC-PR), the positive predictive value of the top 100 patients (PPV-100) and the positive predictive value of the top 113 patients (PPV-norm). We chose to include AUC-PR because it is generally considered a better metric of accuracy for a machine learning model in which the percentage of cases is small, which our percentage of 10% is considered to be[4.21]. We chose the PPV-100 metric with the application in mind; if our immunologists would be reviewing charts from the top 100 patients from their institutions, then we wanted to ensure that we optimized the top 100 patients to have the most CVID patients as possible. Similarly, PPV-norm considers a number of patients similar to our top 100 metric but is normalized so that the range for this metric is [0.0, 1.0] and thus more comparable to AUC-ROC and AUC-PR.

Finally, now that we had physician buy-in for PheNet, we were able to relax some of our conditions on interpretability. Thus, we were able to implement a ridge regression version of PheNet that had shown to have better performance during our cross-validation phase of model building. With ridge regression, we were able to increase the up-sampling to 80%. This increase in up-sampling follows from the idea that ridge regression is more calibrated for data that has higher correlations between features[4.22]. In data that is highly up-sampled, features are expected to be more highly correlated because so many of the patients (or "rows") are identical. However, up-sampling gives us the advantage of increasing our effective case:control ratio[4.23]. Thus, ridge regression provides the advantage of allowing for a larger percentage of up-sampling. Our larger dataset also allowed for additional features to be included in the model; we now include 15 phecodes in addition to the OMIM-derived phecodes as features. Table 4.2 includes the top 20 features ordered by their effect sizes included in this model. As expected, OMIM-derived phecodes dominate the list, with low IgG near the top.

We now compare the performance of the PheNet models we have trained using different methods on different data (Table 4.3). In the first row, we see the original PheNet model trained on UCLA data and tested on our UCLA-only case-control cohort. We note that under our new 80% training, 20% testing fully matched cohort, the original PheNet algorithm underperforms compared to under the paradigm in the previous section (AUC-ROC 0.67, AUC-PR 0.25, PPV-100 0.37, PPV-norm 0.28). However, the same UCLA model tested on the UC-wide testing cohort (in the second row) performs worse (AUC-PR 0.21, PPV-100 0.23). This demonstrates that the weights that were trained by PheNet on UCLA data only are not portable to the rest of the UC health system. In the last row is the UCHDW model trained on the UC-wide training cohort and tested on the UC-wide testing cohort. This model performs the best so far with an AUC-PR of 0.51 and a PPV-100 of 0.48. While this model has double the AUC-PR and PPV-100 of the previous model, there is still much room for improvement.

4.3 INCORPORATING ARTIFICIAL INTELLIGENCE

4.3.1 Feature selection using likelihood ratio tests

So far, we have utilized the clinical knowledge available in the OMIM database to ascertain 34 features for the CVID prediction model. Using this database to augment our machine learning algorithm was important because our training data was so small. However, there are some downsides to this manual form of feature selection. Manual feature selection is more labor intensive than automated feature selection, and in addition requires the presence of an expert. This requirement for time and expertise limits the scalability of implementing PheNet in additional phenotypes in the future. Additionally, it is possible that these OMIM-derived features are not actually the best features to use in a predictive model. While the symptoms described in in OMIM are undoubtedly the most common manifestations of CVID, the relative frequency of these

symptoms in individuals without CVID is not taken into account, thus potentially limiting the predictive power of these phecodes.

With the inclusion of the data from the UCHDW, the training data for our model is now over twice as large as it was previously. With this improvement in sample size, we can now consider feature selection without OMIM as a guide. The way that we implement intelligent feature selection is through the likelihood ratio test.

A likelihood ratio test capitalizes on the fact that the likelihoods of two nested models can be combined to follow a χ^2 distribution. Specifically,

$$-2ln\left[\frac{L(m_0|p_0 \in R^r)}{L(m_A|p_A \in R^n)}\right] \sim \chi^2_{n-r}$$

where $L(m_0|p_0 \in R^r)$ is the likelihood of a 'null' model m_0 given some set of size r of 'null' parameters p_0 and $L(m_A|p_A \in R^n)$ is the likelihood of an 'alternate' model m_A given some set of size n of 'alternate' parameters p_A. If the models are nesting (m_0 is a special case of m_A), then the resulting statistic follows a χ^2 distribution with n-r degrees of freedom[4.24]. In a statistical or machine learning model, the likelihood ratio test can be used for feature selection using the following procedure. First, fit m_0 (the null model) to the data. The null model can have any features that are automatically included and must be of the same form as the alternate model. In this case, I used a ridge regression model fit with only IgG lab values, since this is our strongest clinical predictor. Next, add a feature and calculate the likelihood of this alternate model. If the p-value of the likelihood ratio test falls below the significance threshold, replace the alternate model with the null model and continue to the next feature.

I next utilize feature selection by likelihood ratio test to fit PheNet, using a Bonferroni-corrected p-value of 8.18×10^{-5}. Table 4.4 demonstrates that implementing this change in PheNet results in a small increase in AUC-ROC (0.90), PPV-100 (0.51) and PPV-norm (0.49). In Table

4.6, we can see the top 20 phecodes for this model. While many of the top phecodes can be found on the original list of OMIM-derived phecodes, some additional features such as 'Influenza' and 'Methicillin resistant staphylococcus aureus' stand out.

4.3.2 Learning windows for phenotype recurrence

One way to visualize the phenotypes of this UC-wide cohort is through a fingerprint diagram (Figure 4.6). This diagram separates out each individual case as a column and groups them by site, and then contains the OMIM-derived phecodes on the y-axis. CVID cases in our cohort range from having most of the OMIM-derived phenotypes to none of them, rendering this classification task very difficult with just binary data. However, phenotype data in the EHR is not just binary. Phecodes can appear on a patient's chart any number of times, and the longitudinal aspect of the data (each entry of a phecode has a date attached to it) can be useful in telling the patient's story. If a patient has an immunodeficiency, then theoretically that patient would not only get infections such as sinusitis, pneumonia, and otitis media, but those infections should occur more often, last longer, and be more serious than for someone with normal immune function.

We decided to tackle this idea of whether how often an infection occurs (or rather, how many times an infection appears on a patient's medical record) can be collapsed into a feature that is helpful for this model. One indicator of how many times a specific infection has occurred is a count of that phecode. However, a simple count of the number of times a phecode appears in a patient's chart can be misleading. Patients may visit a physician multiple times during one episode of an infection, and even if these visits are not related to the infection, that phecode may be noted in the chart. Instead, we chose to use the idea of a rolling window to capture the number of episodes a specific phecode appears in a patient's chart.

The idea behind rolling windows is that if a patient has a phecode in their chart at least some number of days *d* after the most recent occurrence of that same phecode in their chart, then this appearance is a recurrence of the infection instead of a relapse. A recurrence of an infection will count as the start of a new episode, and the patient's episode count will increase by one. Each time the phecode appears in the chart within *d* days, the start of the window resets. In this way, we can count the number of episodes of a specific phecode a patient has documented in the EHR. However, we must first come up with an estimate for *d*. To determine the optimal *d* for calculating recurrence windows, we turn to the clinical information that antibiotic records can show us.

If we consider the antibiotic usage of all patients who have had a specific infection phecode (for example, otitis media) and take into account guidelines for first-line antibiotic usage and second-line antibiotic usage, we can calculate the time between first-line and first-line antibiotic use and first-line and second-line antibiotic use for all instances of each. The time between first-line and first-line antibiotic use can roughly be assumed to be a recurrence, while the time between first-line and second-line antibiotic use can roughly be assumed to be a relapse.

In Figure 4.7, we can see that the distributions of the length of time between first line and first line antibiotic usage differs from the distributions of the length of time between first line and second line antibiotic usage in patients with acute sinusitis or otitis media. However, we found that the median time between first line antibiotics (putative recurrence) was > 38 days and the median time between second line antibiotics (putative relapse) was < 27 days. Thus, we chose 30 days as a standard window length to use to measure recurrence of clinical features in PheNet.

Using the likelihood ratio test method of feature selection, we found 32 phecodes of the original 66 phecodes for which adding recurrence increased the likelihood of the model. We also find that overall, adding these recurrence features improves the performance of the model by a

substantial amount. Table 4.4 demonstrates that adding these additional features to PheNet results in an almost two-fold increase in AUC-PR (0.87), PPV-100 (0.83) and PPV-norm (0.75).

4.3.3 Accounting for confounders in a multi-site study

One of the major differences between the UCHDW and the DDR at UCLA is that the UCHDW is a composite of data from six different UC systems. Any time data is pooled from multiple sources, there will likely be confounding variables in the data that may introduce bias. One way to help with confounding is to include confounding variables as covariates[4.25], so that the effect sizes of the other variables in the statistical model are effectively conditioned on the existence or non-existence of that confounding variable.

Table 4.5 displays the different demographics that are present across the UCHDW sites. Each site is unique, with a diverse set of patients. The average patient age at UCSF is 36, while at the other sites the average patient age ranges from 41 to 46. The longest average record length is at UCLA, where it is 3 years as opposed to 2 elsewhere. UCR has the largest Hispanic Latino population at 29%. Unfortunately, self-identified race (SIRE) is difficult to assess given the high numbers of individuals of unknown SIRE at all sites. While the matching process present in our current algorithm causes us to not encounter confounding as much due to forced non-correlation between demographics and outcome, the resultant model from PheNet is run on the full database, in which it is impossible to ensure "matching" or lack of confounders. Thus, we chose to create an additional unmatched control set to measure the effect sizes of confounders.

After training on unmatched controls and adding in the demographic variables as covariates in the model, Table 4.4 shows that each of the metrics have improved compared to the previous model (AUC-ROC 0.98, AUC-PR 0.95, PPV-100 0.93, and PPV-norm 0.88). However, Table 4.7 shows that this improvement in performance comes with a cost. Now, only 11 phecodes, 5

recurrent phecodes, and 2 lab values add significantly enough to the model to be included. Instead, all the demographic variables (age, record length, sex, race, and site) are significant to the model.

4.4 APPLICATION TO CARDIAC AMYLOIDOSIS

4.4.1 Introduction to cardiac amyloidosis

Cardiac amyloidosis is a type of amyloidosis, a disease in which misfolded proteins are deposited in organs in the body[4.26]. Cardiac amyloidosis occurs when these misfolded proteins (amyloid plaques) are deposited in the cardiovascular system, which can cause heart failure among other symptoms. However, while cardiac amyloidosis is estimated to be a factor in heart failure in up to 25% of patients[4.27], it is not routinely tested for in the clinic. This lack of diagnosis or diagnostic delay worsens the prognosis of those that suffer from this disease[4.27].

One major reason for this diagnostic delay is that cardiac amyloidosis is often mistaken for other phenotypes[4.28]. As such, patients may see a cardiologist or be suffering from heart failure but not be tested for cardiac amyloidosis. This is especially problematic for patients with transthyretin cardiac amyloidosis, for which a drug (tafamidis) can be taken to halt the effects. Thus, patients are missing out on a potential treatment for their heart failure by not getting tested for cardiac amyloidosis.

We propose to use the PheNet and the UCHDW to aid in diagnosis through prediction for cardiac amyloidosis. By training PheNet to discern between patients with cardiac amyloidosis - induced heart failure and patients with other types of heart failure, we hope to show that PheNet has potential applications beyond CVID and can help reduce mortality from disease in a variety of contexts.

4.4.2 Study design

To train PheNet on a new phenotype, we must first create a reliable case definition. For cardiac amyloidosis, we have two options. The first option is to use the cardiac amyloidosis phecode (E85.82). However, phenotyping based on the presence or absence of a phecode can be unreliable, and it would be best if we could train on the specific type of cardiac amyloidosis for which a pharmacological intervention is possible (transthyretin cardiac amyloidosis). Thus, we chose to construct our case cohort from patients with a tafamidis prescription (N=777). Tafamidis is a drug that uniquely treats transthyretic cardiac amyloidosis and will only be prescribed once a patient has been positively diagnosed with the condition (unlike a phecode which can appear at any point in the diagnostic journey). Table 4.8 describes our cohort of cardiac amyloidosis patients, with comparisons from both methods of phenotype ascertainment to two different types of controls – controls with heart failure and controls from the general population without any heart failure diagnoses in their chart. The cardiac amyloidosis patients are expectedly older than the heart failure patients, who are in turn older than the general population. They are also mostly male, which has been noted in the literature[4.27]. We also see the expected increased prevalence of cardiac amyloidosis in the African American population; there is a genetic variant thought to be causal for this disease that has a higher frequency in populations that identify as African American. We can also see the percentages of common cardiac amyloidosis-related comorbidities, medications, and lab results.

For our controls, we use a cohort of heart failure patients who are matched with our cases based on age, sex, record length, race, site, and ethnicity. We utilize an 80/20 training/testing split of the case-control data, with each cohort consisting of 10% cases. For features, we augment

phecode information with medication information for medications clinically relevant to cardiac conditions.

4.4.3 Results

Using the PheNet framework that we developed for CVID, we trained a ridge regression model using 80% up-sampling, no down-sampling, and likelihood ratio feature selection. First, we train a null model based on age. While age is a strong predictor of cardiac amyloidosis in the general population, since our cohort is matched on age, any association that PheNet finds between age and cardiac amyloidosis would be entirely random. Thus, without a known strong clinical indicator of the disease to use as our starting feature, age stands as the ideal "null" model to test additional features against. Table 4.9 shows that, as expected, the performance of this model is only slightly better than random, with AUC-ROC = 0.53 (no signal is 0.5) and AUC-PR = 0.12 (no signal is 0.1 in this case).

Next, we use likelihood ratio feature selection to choose medication features that significantly improve the model. We aggregate medications within similar medication classes (warfarin, loop diuretics, digoxin, anti-platelets, orthostatic, and neuropathic) and consider whether a patient has ever had a medication from each class. Adding these relevant medications improved the model slightly, with an AUC-ROC = 0.63 and AUC-PR = 0.15.

Next, we added in binary phecodes features, continuing to use likelihood ratio feature selection. To encourage our model to select *for* cardiac amyloidosis (as opposed to *against* heart failure), we filtered the available phecodes to only those present in at least 2% of cardiac amyloidosis cases. Adding in these phecodes provides a huge jump in performance, up to AUC-ROC = 0.95 and AUC-PR = 0.81.

Last, we added in recurrent phecodes to the model. Nine recurrent phecodes are significant for model performance, and they improve the performance of PheNet by a small amount, keeping AUC-ROC = 0.95 and getting AUC-PR = 0.81. Figure 4.8 displays these features and their distributions across the patient set in the best performing model. We can see how sparse the medication data really is, which likely contributes to how small the impact of incorporating medication information was on the performance of the model. We also see that PheNet can find known comorbidities of cardiac amyloidosis even though its feature selection process is completely agnostic to known clinical indicators. For example, the feature "inflammatory and toxic neuropathy" has been shown to be correlated with cardiac amyloidosis status.

Currently, there is an algorithm to predict cardiac amyloidosis being used in various hospitals that was developed by Huda et al[4.29]. While this algorithm achieves AUC-ROC 0.93 in sample (claims data), it performs worse on EHR data (AUC-ROC 0.81). Our model has higher performance metrics (of the reported metrics) on an in-sample dataset, however we leave it to future work to see how this model would perform in an out-of-sample set.

4.5 DISCUSSION

In this chapter, we seek to create a machine learning algorithm that will help decrease diagnostic delay for CVID patients. Our main goal is to find whether we can correctly classify whether patients have CVID with enough precision to merit physician chart review and eventual referral of high scoring patients to immunology. Using 186 physician-reviewed CVID patients at UCLA, we created PheNet, which can classify CVID patients using marginal logistic regression and phecodes selected through the OMIM database. Next, we extend PheNet to the UC Health Data Warehouse, where we have 575 physician-identified CVID cases. We revamp our testing and training process to ensure a more realistic testing environment without reusing any data in both

training and testing. We also switch to using ridge regression due to the high correlations between phenotypes and increase our up-sampling ratio to improve model performance. We find that retraining PheNet with these changes on the UC Health Data Warehouse cohort has a higher performance that reusing the UCLA-trained weights on this new patient cohort. Next, we consider how to train PheNet in the absence of clinical knowledge such as OMIM to do the majority of feature selection for us. We introduce likelihood ratio -based feature selection to choose features for the PheNet model in a way that is agnostic of clinical knowledge or biases. We find that training PheNet to use likelihood ratio-based feature selection allows us to improve model performance. Next, we incorporate longitudinal data in our model. We utilize antibiotic usage information within the UCHDW to find optimal widow sizes with which to infer episodes of care. We use these episodes of care to define recurrent phenotypes that we use as features in PheNet. These recurrent features substantially improve the performance of PheNet for CVID. We additionally consider the impact of including covariates in our model. We switch to a non-matched control group and fit demographic covariates to our model to compare performance with the matched control group. We find that a model that is based on non-matched controls and includes demographic covariates further improves the performance of PheNet. Finally, we apply PheNet to cardiac amyloidosis, an entirely new phenotype. We train a state of the art classifier that differentiates between heart failure patients with cardiac amyloidosis and those that don't.

This work has several implications for machine learning in electronic health records. We have shown that it is possible to train a classifier using structured electronic health record data that can choose features without manual clinical input. PheNet is robust enough to handle multiple health systems beyond UCLA as well as additional phenotypes beyond CVID. Our likelihood ratio

test -based feature selection method finds clinically meaningful features in electronic health record data, and we were able to infer episodes of care well enough to use them as a predictive measure.

This work opens many possibilities of future research. When likelihood ratio testing is employed for feature selection, a p-value must be chosen as a threshold for significance. In this chapter, I used a Bonferroni-corrected p-value, which worked well. However, it is well known that phecodes are correlated with each other, which likely renders a Bonferroni correction too stringent. Finding the optimal p-value cutoff and fine-mapping the phecode signal to an actual 'causal' phecode (rather than a correlated one) would be an important step forward in intelligent feature selection for machine learning for electronic health records. Similarly, while PheNet has proven to be robust to incorporating multiple health systems' data, research has yet to show what the best way is to incorporate EHR data from multiple sources, what are the likely confounders, and how to balance preserving signal with conditioning out false positives.

I will conclude with caveats and limitations of our work. While considering rare phenotypes it is unlikely but impossible to rule out without chart review the possibility that our control cohorts contain some undiagnosed patients. If this is true, model performance may be skewed due to PheNet picking up true positives that we are calling as false positives. Additionally, we make assumptions about the 'real' environment in which we will run PheNet; if our control group is very different from the rest of the health system population, it is unclear how well our model will perform.

From both humanitarian and economic perspectives, diagnostic delay is a problem in our healthcare system. By providing insight into some new strategies to use machine learning to find patients with an underlying diagnosis, we hope to be able to reduce diagnostic delay and improve the health of our patients.

4.6 TABLES

Table 4.1: **UC-wide case cohort for CVID**

	UCSD cases (80)	**UCSF cases (80)**	**UCI cases (193)**	**UCLA cases (192)**	**UCD cases (20)**
Mean age (SD)	59 (15)	49 (21)	60 (18)	57 (20)	52 (20)
Mean record years (SD)	7 (4)	9 (3)	7 (4)	8 (4)	9 (3)
% male	29	50	35	29	45
% female	71	50	65	71	55
% Asian	5	1	3	2	5
% other race	5	11	4	4	5
% white	88	82	89	71	75
% unknown	2	2	6	10	5
% Hispanic or Latino	4	11	3	9	5
% not Hispanic or Latino	94	86	91	81	90
% Native Hawaiian or other Pacific Islander	0	0	1	0	0
% Black or African American	0	0	0	1	0
% American Indian or Alaska Native	0	0	0	1	0

Table 4.2 **Top 20 PheNet features and effect sizes using OMIM**

Feature	**Weight**
Primary thrombocytopenia	1.6
Acquired hemolytic anemias	1.3
Low IgG	1.1
Bronchiectasis	1.0
Failure to thrive	1.0
Meningitis	0.90
Splenomegaly	0.85
Autoimmune hemolytic anemias	0.66
Chronic sinusitis	0.65
Asthma	0.56
Other arthropathies	0.56
Diarrhea	0.43
Thrombocytopenia	0.43
Alopecia	0.26
Infections of skin/subcutaneous tissue	0.24
Pneumonia	0.23
Lymphadenitis	0.20
Hypoglycemia	0.18
Pituitary hypofunction	0.16
Chronic pharyngitis/nasopharyngitis	0.16

Table 4.3: **PheNet performance on UCHDW compared to UCLA**

	AUC-ROC	**AUC-PR**	**PPV-100**	**PPV-norm**
UCLA weights, UCLA test set	0.67	0.25	0.37	0.28
UCLA weights, UC test set	0.60	0.21	0.23	0.26
UC weights, UC test set	0.89	0.51	0.48	0.43

Table 4.4 **PheNet performance using intelligence feature selection**

	AUC-ROC	**AUC-PR**	**PPV-100**	**PPV-norm**
OMIM weights	0.89	0.51	0.48	0.43
LRT weights	0.90	0.48	0.51	0.49
LRT weights + recurrence	0.98	0.86	0.83	0.75
LRT weights + demo	0.98	0.95	0.93	0.88

Table 4.5 **Demographic factors vary across the UC health system**

	UCD (1,187,105)	**UCSF (1,943,998)**	**UCLA (2,517,440)**	**UCSD (1,194,390)**	**UCR (30,607)**	**UCI (986,844)**
Mean age (SD)	41 (24)	36 (26)	45 (23)	46 (22)	41 (22)	44 (23)
Mean record years (SD)	2 (3)	2 (3)	2 (3)	3 (3)	1 (2)	2 (3)
% male	48	47	46	45	32	45
% female	52	53	54	54	68	55
% Asian	6	11	7	8	4	13
% Black or African American	6	6	4	5	5	2
% other race	14	19	10	22	18	15
% white	38	39	43	50	35	59
% unknown	33	22	32	14	24	11
% Native Hawaiian or other Pacific Islander	1	1	0	0	0	0
% Multirace	2	3	3	2	2	2
% Hispanic or Latino	13	17	13	21	29	27
% not Hispanic or Latino	54	62	55	65	47	61

Table 4.6 **Top 20 PheNet features and effect sizes using intelligent feature selection**

Feature	**Weight**
Primary thrombocytopenia	2.0
Cushing's syndrome	1.8
Renal failure NOS	1.3
Bronchiectasis	1.2
Chronic obstructive asthma	1.00
Splenomegaly	0.90
Chronic sinusitis	0.86
Other CNS infection and poliomyelitis	0.84
Influenza	0.84

Log IgG	0.79
Chronic pain syndrome	0.73
Ventral hernia	0.69
Methicillin resistant Staphylococcus aureus	0.62
Essential tremor	0.60
Other specified diseases of hair and hair follicles	0.57
Post-inflammatory pulmonary fibrosis	0.57
Pericarditis	0.56
Diarrhea	0.53
Prurigo and lichen	0.51
Other disorders of bladder	0.47

Table 4.7 **Top 20 PheNet features and effect sizes of demographics and recurrence**

Feature	**Weight**
Low IgG	3.9
Bronchiectasis	3.3
Chronic pharyngitis/nasopharyngitis	2.9
Primary thrombocytopenia	2.8
Chronic sinusitis	2.6
Other infectious/parasitic diseases	2.0
Normalized record length	1.8
Asthma	1.5
UCI patient	1.4
Low calculated globulins	1.0
UCSD patient	0.68
White SIRE	0.56
UCLA patient	0.32
Normalized age	0.0095
UCSF patient	-0.02
Male	-0.18
Other tests	-0.27
Asian SIRE	-1.2
Attention deficit hyperactivity disorder	-1.2
Need for hormone replacement therapy (postmenopausal)	-1.3

Table 4.8 **Cardiac amyloidosis cohort in the UCHDW**

	Tafamidis cases	E85.82 cases	Heart failure patients	General population
Mean age (SD)	79 (8)	78 (10)	69 (16)	42 (24)
Mean record years (SD)	5 (4)	6 (4)	4 (4)	2 (3)
% male	89	84	56	46
% Black or African American	11	11	10	5
% white	68	67	56	44
% essential hypertension	52.2	68.1	64.7	13.6
% coronary atherosclerosis	39.5	47.3	49.0	3.7
% atrial fibrillation	52.0	64.8	40.7	2.6
% other peripheral nerve disorders	29.0	38.1	10.2	2.9
% heart failure with preserved EF (diastolic heart failure)	47.1	63.2	37.4	0.9
% loop diuretic	84.5	78.9	77.3	6.8
% neuropathic	45.2	50.7	37.4	12.5
% orthostatic	13.4	17.8	7.5	0.9
Median prealbumin (mg/dL)	24.5	18.9	20.0	25.9
Median troponin (ng/mL)	0.09	0.09	0.03	0.02

Table 4.9 **PheNet performance for cardiac amyloidosis**

	AUC-ROC	AUC-PR	PPV-100	PPV-norm
Age	0.53	0.12	0.11	0.13
Age + meds	0.63	0.15	0.16	0.15
Age + meds + phecodes	0.94	0.79	0.87	0.74
Age + meds + phecodes + recurrence	0.95	0.81	0.91	0.74

4.7 FIGURES

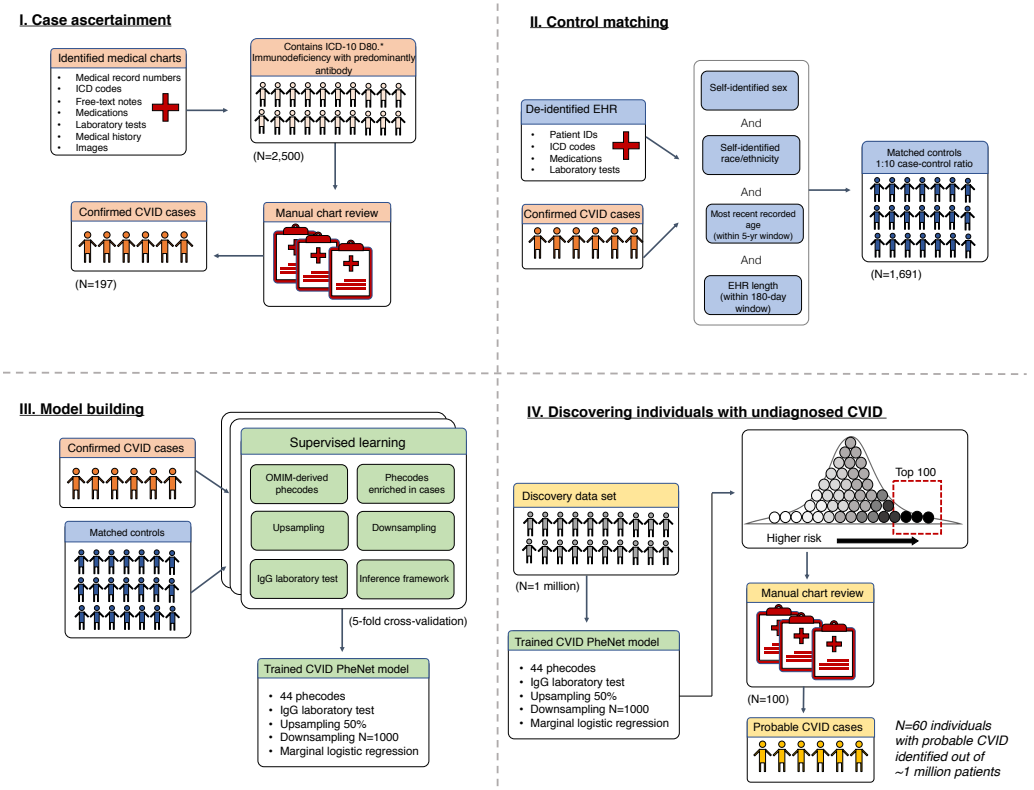

Figure 4.1: **Overview of PheNet model training and application within a discovery cohort.**

We present a visual summary of case/control cohort construction, PheNet model training, and application within a discovery dataset at UCLA Health. (**A**) The workflow for constructing a case cohort of clinically diagnosed patients with CVID from medical charts. (**B**) Criteria used to create a matched control cohort from the EHR (n = 1106). (**C**) Construction of the prediction model, including feature selection from phecodes, inclusion of laboratory values, a variety of inference frameworks, and data balancing techniques. (**D**) Example of how the PheNet model can be applied within a discovery cohort to identify patients with a high likelihood of CVID, who could then be further assessed by manual chart review to confirm diagnosis.

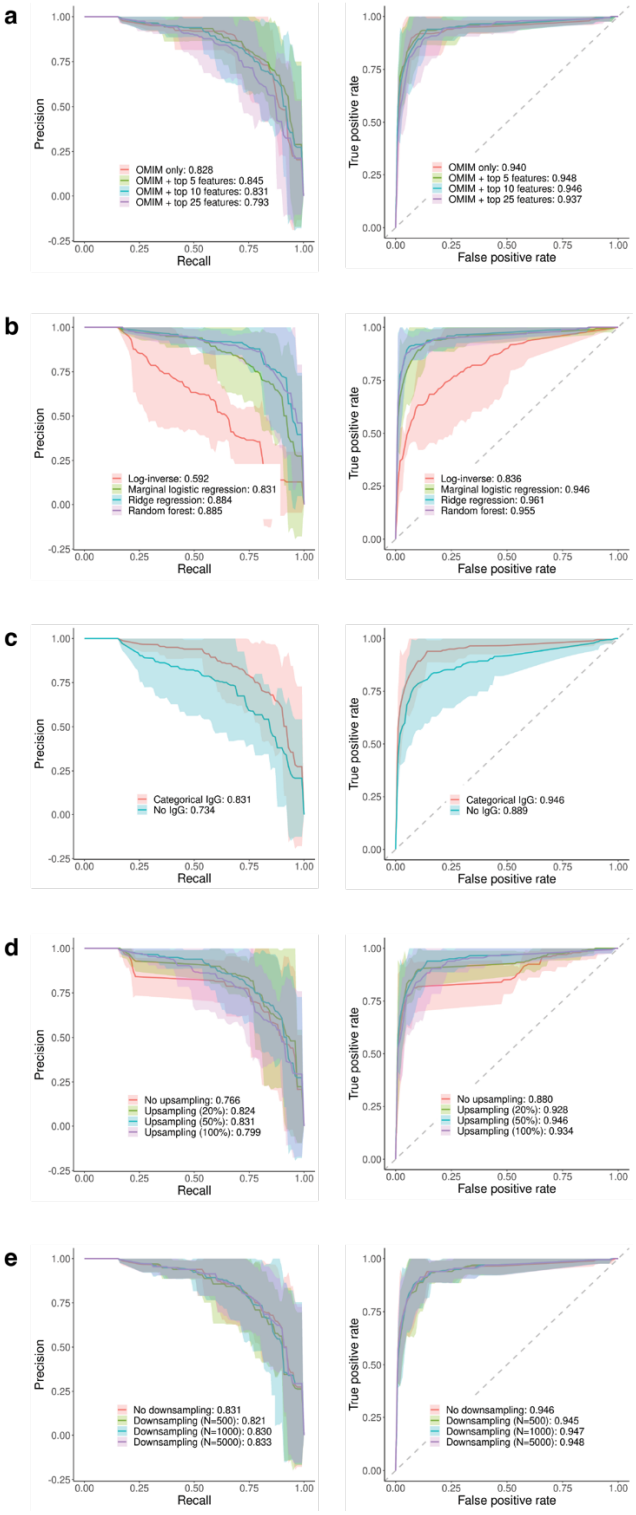

Figure 4.2: **Exploration of model parameters for PheNet**. We show AUC-Receiver Operator and AUC-Precision Recall curves for the PheNet model using matched case (N=186) and control

(N=1,106) cohorts with 5-fold cross-validation. We varied the **a**) number of additional phecode features in addition to OMIM-selected features, **b**) prediction model, **c)** inclusion of immunoglobulin G (IgG) tests, **d**) up-sampling, and **e**) down-sampling.

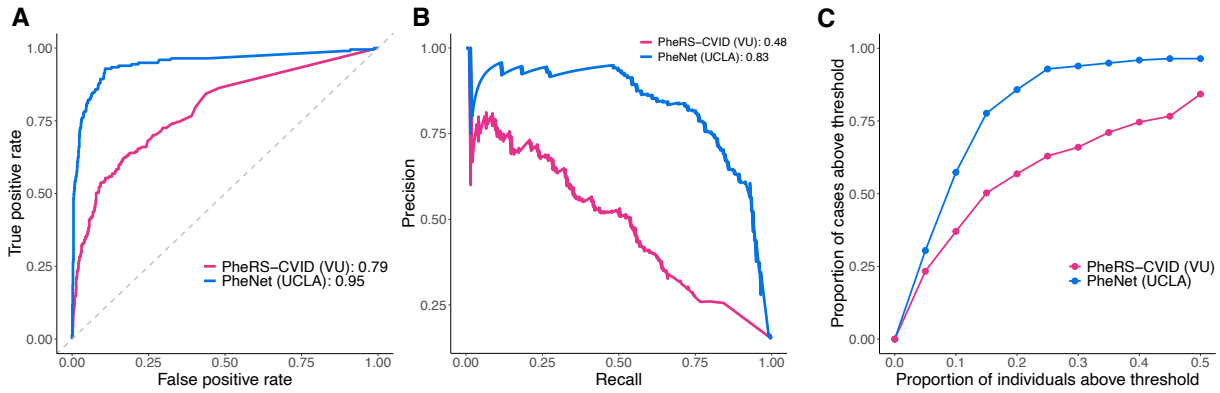

Figure 4.3: **PheNet is more accurate than existing phenotype risk scores for predicting CVID.** Performance metrics comparing the performance of PheNet and PheRS-CVID within UCLA Health population case and control cohorts. The PheNet and PheRS-CVID models were trained using weights trained from EHR data. Receiver operating characteristic (**A**) and precision-recall (**B**) curves across the different prediction models are shown. AUC is provided in the legend. (**C**) Individuals with a PheNet score of >0.90 and the proportion of CVID cases captured within the varying percentiles of PheNet and PheRS scores.

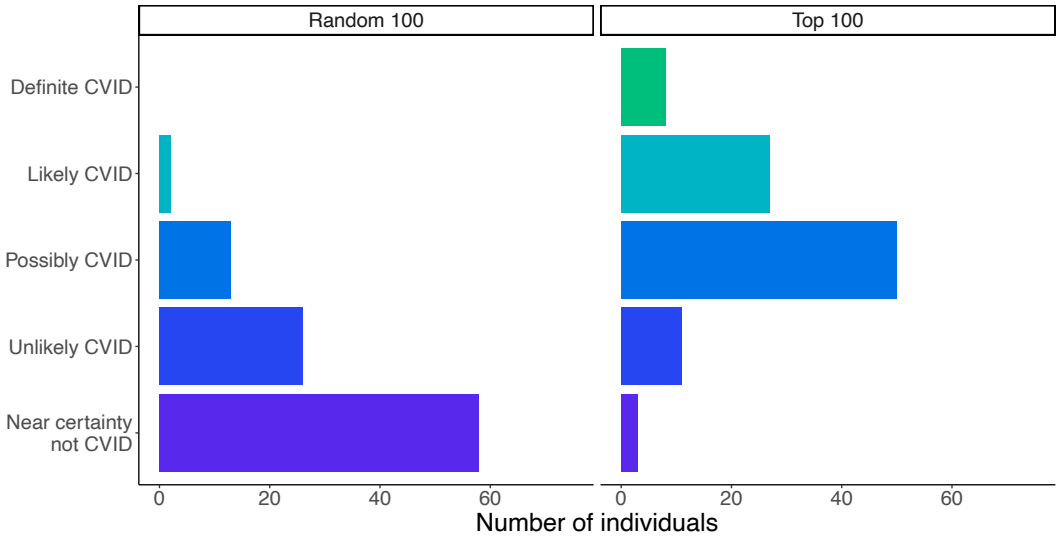

Figure 4.4: **PheNet identifies undiagnosed individuals with CVID.** CVID clinical validation scores for the top $n = 100$ individuals with the highest PheNet score and $n = 100$ randomly sampled individuals. Each individual was ranked according to an ordinal scale from 1 to 5 quantifying the likeliness of having CVID where 1 was defined as near certainty not CVID and 5 was definitive as CVID.

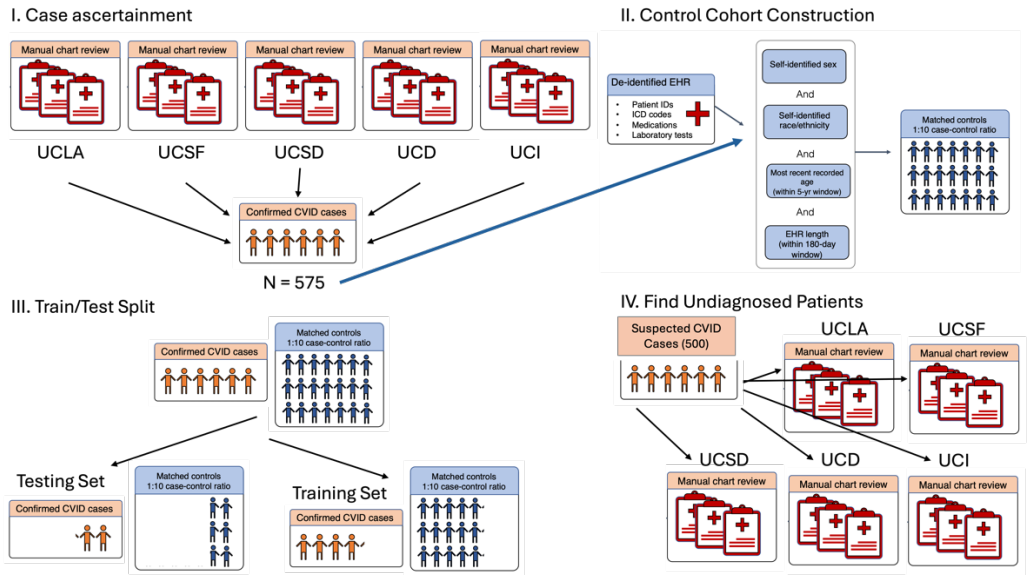

Figure 4.5 **Overview of PheNet model training and application within the UCHDW.** We present a visual summary of case/control cohort construction, PheNet model training, and

application within UCHDW. (**A**) The workflow for constructing a case cohort of clinically diagnosed patients with CVID from medical charts. (**B**) Criteria used to create a matched control cohort from the EHR. (**C**) Construction of the separate testing and training cohorts (**D**) Example of how the results of PheNet applied to the UCHDW are disseminated to each of the sites to identify patients with a high likelihood of CVID, who could then be further assessed by manual chart review to confirm diagnosis.

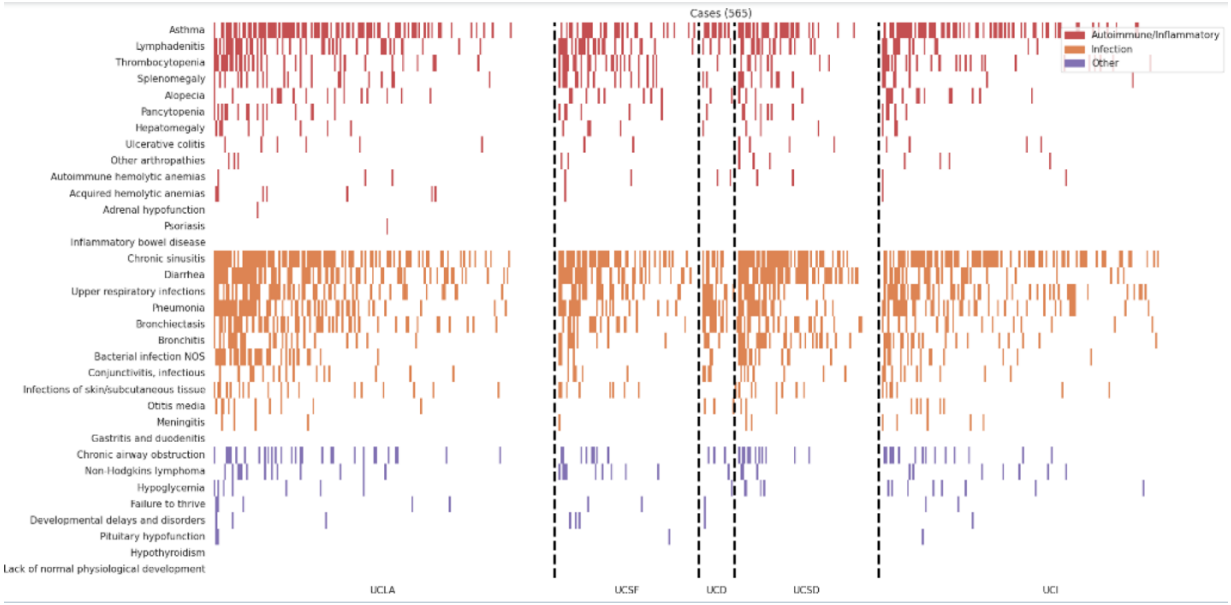

Figure 4.6 **Phenotypic fingerprint of CVID.** This phenotypic summary of OMIM phecodes in the UCHDW CVID cohort showcases the heterogeneity of the disease. Each column represents an individual separated by site, and each row represents an OMIM-derived phecode separated by phenotype category.

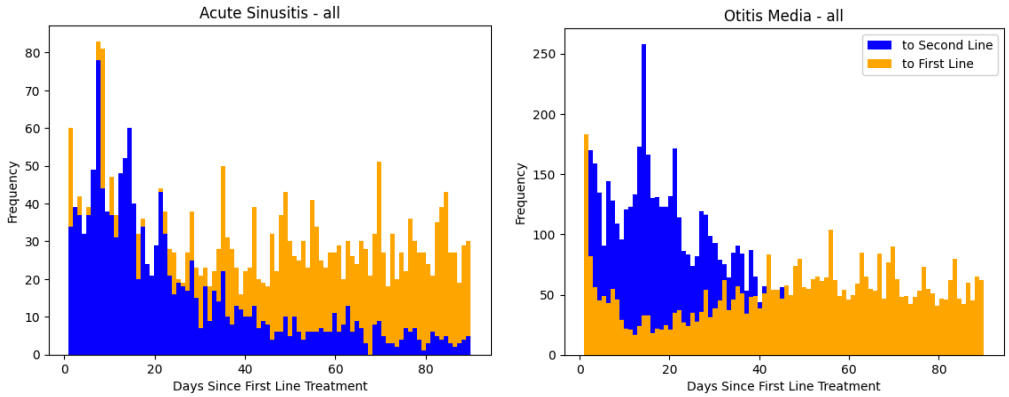

Figure 4.7 **Histogram of medication windows for recurrence.** Histogram of episode lengths between antibiotic usage in patients with acute sinusitis (left) and otitis media (right). The blue bars mark the number of days between first line antibiotics and second line antibiotics, which indicates a relapse. The orange bars mark the number of days between first line antibiotics to first line antibiotics again, which indicates a recurrence.

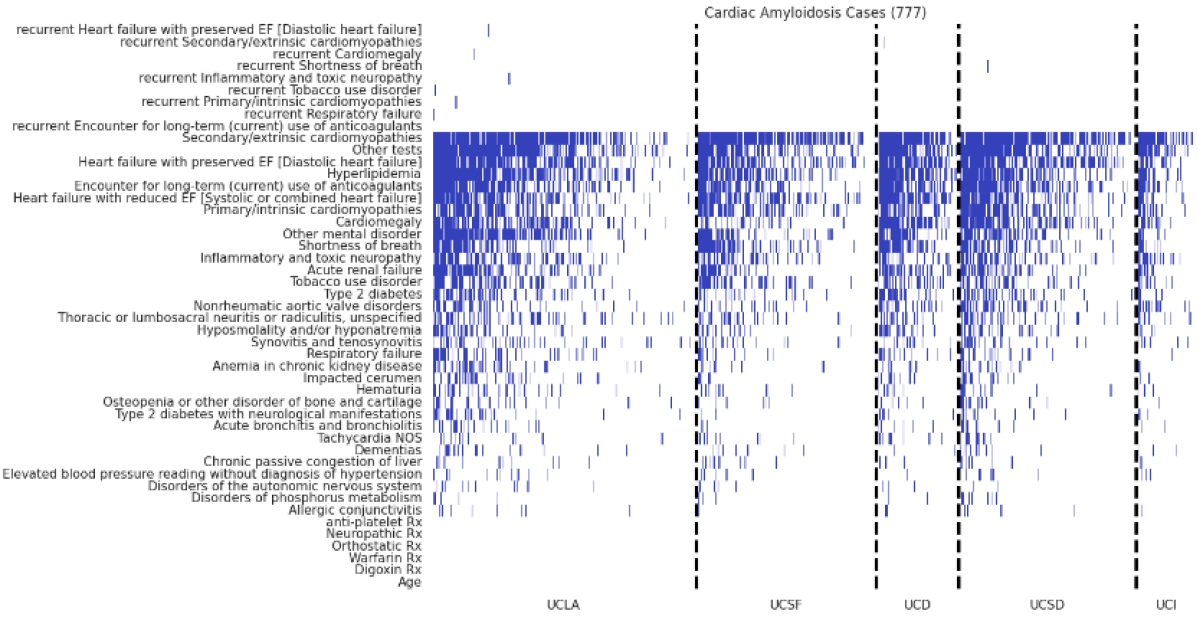

Figure 4.8 **Phenotypic fingerprint of cardiac amyloidosis.** This phenotypic summary of the model features in the UCHDW cardiac amyloidosis cohort showcases the variety of phecodes

correlated to the disease. Each column represents an individual separated by site, and each row represents a feature.

4.8 REFERENCES

4.1 S. G. Tangye, W. Al-Herz, A. Bousfiha, C. Cunningham-Rundles, J. L. Franco, S. M. Holland, C. Klein, T. Morio, E. Oksenhendler, C. Picard, A. Puel, J. Puck, M. R. J. Seppänen, R. Somech, H. C. Su, K. E. Sullivan, T. R. Torgerson, I. Meyts, Human inborn errors of immunity: 2022 Update on the classification from the International Union of Immunological Societies Expert Committee. *J. Clin. Immunol.* **42**, 1473–1507 (2022)

4.2 S. L. Silva, M. Fonseca, M. L. M. Pereira, S. P. Silva, R. R. Barbosa, A. Serra-Caetano, E. Blanco, P. Rosmaninho, M. Pérez-Andrés, A. B. Sousa, A. A. S. F. Raposo, M. Gama-Carvalho, R. M. M. Victorino, L. Hammarstrom, A. E. Sousa, Monozygotic twins concordant for common variable immunodeficiency: Strikingly similar clinical and immune profile associated with a polygenic burden. *Front. Immunol.* **10**, 2503 (2019).

4.3 Abbott JK, Gelfand EW. Common variable immunodeficiency: diagnosis, management, and treatment. Immunology and Allergy Clinics. 2015 Nov 1;35(4):637-58.

4.4 Anderson JT, Cowan J, Condino-Neto A, Levy D, Prusty S. Health-related quality of life in primary immunodeficiencies: impact of delayed diagnosis and treatment burden. Clinical Immunology. 2022 Mar 1;236:108931.

4.5 C. A. Slade, J. J. Bosco, T. B. Giang, E. Kruse, R. G. Stirling, P. U. Cameron, F. Hore-Lacy, M. F. Sutherland, S. L. Barnes, S. Holdsworth, S. Ojaimi, G. A. Unglik, J. De Luca, M. Patel, J. McComish, K. Spriggs, Y. Tran, P. Auyeung, K. Nicholls, R. E. O'Hehir, P. D. Hodgkin, J.

A. Douglass, V. L. Bryant, M. C. van Zelm, Delayed diagnosis and complications of predominantly antibody deficiencies in a cohort of Australian adults. *Front. Immunol.* **9**, 694–694 (2018).

4.6 V. Graziano, A. Pecoraro, I. Mormile, G. Quaremba, A. Genovese, C. Buccelli, M. Paternoster, G. Spadaro, Delay in diagnosis affects the clinical outcome in a cohort of cvid patients with marked reduction of iga serum levels. *Clin. Immunol.* **180**, 1–4 (2017).

4.7 J. S. Orange, J. T. Glessner, E. Resnick, K. E. Sullivan, M. Lucas, B. Ferry, C. E. Kim, C. Hou, F. Wang, R. Chiavacci, S. Kugathasan, J. W. Sleasman, R. Baldassano, E. E. Perez, H. Chapel, C. Cunningham-Rundles, H. Hakonarson, Genome-wide association identifies diverse causes of common variable immunodeficiency. *J. Allergy Clin. Immunol.* **127**, 1360–1367.e6 (2011).

4.8 C. Baloh, A. Reddy, M. Henson, K. Prince, R. Buckley, P. Lugar, 30-Year review of pediatric- and adult-onset CVID: Clinical correlates and prognostic indicators. *J. Clin. Immunol.* **39**, 678–687 (2019).

4.9 M. D. O'Sullivan, A. J. Cant, The 10 warning signs. *Curr. Opin. Allergy Clin. Immunol.* **12**, 588–594 (2012).

4.10 F. A. Bonilla, I. Barlan, H. Chapel, B. T. Costa-Carvalho, C. Cunningham-Rundles, M. T. de la Morena, F. J. Espinosa-Rosales, L. Hammarström, S. Nonoyama, I. Quinti, J. M. Routes, M. L. K. Tang, K. Warnatz, International Consensus Document (ICON): Common variable immunodeficiency disorders. *J. Allergy Clin. Immunol. Pract.* **4**, 38–59 (2016).

4.11 T. J. Morley, L. Han, V. M. Castro, J. Morra, R. H. Perlis, N. J. Cox, L. Bastarache, D. M. Ruderfer, Phenotypic signatures in clinical data enable systematic identification of patients for genetic testing. *Nat. Med.* **27**, 1097–1104 (2021).

4.12 J. M. Banda, A. Sarraju, F. Abbasi, J. Parizo, M. Pariani, H. Ison, E. Briskin, H. Wand, S. Dubois, K. Jung, S. A. Myers, D. J. Rader, J. B. Leader, M. F. Murray, K. D. Myers, K. Wilemon, N. H. Shah, J. W. Knowles, Finding missed cases of familial hypercholesterolemia in health systems using machine learning. *NPJ Digit. Med.* **2**, 23 (2019).

4.13 S.-I. Lee, S. Celik, B. A. Logsdon, S. M. Lundberg, T. J. Martins, V. G. Oehler, E. H. Estey, C. P. Miller, S. Chien, J. Dai, A. Saxena, C. A. Blau, P. S. Becker, A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nat. Commun.* **9**, 42 (2018).

4.14 Johnson R, Stephens AV, Mester R, Knyazev S, Kohn LA, Freund MK, Bondhus L, Hill BL, Schwarz T, Zaitlen N, Arboleda VA. Electronic health record signatures identify undiagnosed patients with common variable immunodeficiency disease. Science Translational Medicine. 2024 May 1;16(745):eade4510.

4.15 V. A. McKusick, Mendelian inheritance in man and its online version, OMIM. *Am. J. Hum. Genet.* **80**, 588–604 (2007).

4.16 T. Groza, S. Köhler, D. Moldenhauer, N. Vasilevsky, G. Baynam, T. Zemojtel, L. M. Schriml, W. A. Kibbe, P. N. Schofield, T. Beck, D. Vasant, A. J. Brookes, A. Zankl, N. L. Washington, C. J. Mungall, S. E. Lewis, M. A. Haendel, H. Parkinson, P. N. Robinson, The

Human Phenotype Ontology: Semantic unification of common and rare disease. *Am. J. Hum. Genet.* **97**, 111–124 (2015).

4.17 P. Wu, A. Gifford, X. Meng, X. Li, H. Campbell, T. Varley, J. Zhao, R. Carroll, L. Bastarache, J. C. Denny, E. Theodoratou, W.-Q. Wei, Mapping ICD-10 and ICD-10-CM codes to phecodes: Workflow development and initial evaluation. *JMIR Med. Inform.* **7**, e14325 (2019).

4.18 L. Bastarache, Using phecodes for research with the electronic health record: From PheWAS to PheRS. *Annu. Rev. Biomed. Data Sci.* **4**, 1–19 (2021).

4.19 L. Bastarache, Using phecodes for research with the electronic health record: From PheWAS to PheRS. *Annu. Rev. Biomed. Data Sci.* **4**, 1–19 (2021).

4.20 University of California Health Data Warehouse (UCHDW) https://intranet.uchdw.uchealth.edu/data/index.html

4.21 Jeni LA, Cohn JF, De La Torre F. Facing imbalanced data--recommendations for the use of performance metrics. In2013 Humaine association conference on affective computing and intelligent interaction 2013 Sep 2 (pp. 245-251). IEEE.

4.22 Eliot M, Ferguson J, Reilly MP, Foulkes AS. Ridge regression for longitudinal biomarker data. The International Journal of Biostatistics. 2011 Sep 27;7(1):000010220215574679135 3.

4.23 Liu S, Ong ML, Mun KK, Yao J, Motani M. Early prediction of sepsis via smote upsampling and mutual information based downsampling. In2019 Computing in Cardiology (CinC) 2019 Sep 8 (pp. Page-1). IEEE.

4.24 Vuong QH. Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica: journal of the Econometric Society. 1989 Mar 1:307-33.

4.25 Bursac Z, Gauss CH, Williams DK, Hosmer DW. Purposeful selection of variables in logistic regression. Source code for biology and medicine. 2008 Dec;3:1-8.

4.26 Martinez-Naharro A, Hawkins PN, Fontana M. Cardiac amyloidosis. Clinical Medicine. 2018 Apr 4;18(Suppl 2):s30.

4.27 Witteles, R. M. et al. Screening for transthyretin amyloid cardiomyopathy in everyday practice. JACC Heart Fail 7, 709–716 (2019).

4.28 Rapezzi C, Lorenzini M, Longhi S, Milandri A, Gagliardi C, Bartolomei I, Salvi F, Maurer MS. Cardiac amyloidosis: the great pretender. Heart failure reviews. 2015 Mar;20:117-24.

4.29 Huda, Ahsan, Adam Castaño, Anindita Niyogi, Jennifer Schumacher, Michelle Stewart, Marianna Bruno, Mo Hu, Faraz S. Ahmad, Rahul C. Deo, and Sanjiv J. Shah. 2021. “A Machine Learning Model for Identifying Patients at Risk for Wild-Type Transthyretin Amyloid Cardiomyopathy.” *Nature Communications* 12 (1): 2725.

5 CONCLUSION

In this dissertation, I have studied infectious, complex, and rare disease through the lenses of agent-based, statistical, and machine learning models. In Chapter 2, I showed how local differential sensitivity analysis can be applied to biological models ranging from the deterministic SIR model to branching processes. I also develop a novel method to calculate mixed second derivatives of a model outcome with respect to its parameters, and I implement this new method in addition to multithreaded and first order methods of the same type in the Julia programming language. I then benchmark these methods for accuracy, precision, and speed

In Chapter 3, I study how to achieve the highest possible power for GWAS in admixed populations. I consider admixture mapping, standard GWAS, and Tractor, and show that because the alternative hypothesis of admixture mapping is the same as the null hypothesis of Tractor, Tractor will have lower power precisely when admixture mapping has high power (i.e., when ancestry-specific allele frequency differences are highest). Next, I use simulations to investigate ancestry-specific allelic effect size heterogeneity and find regions of heterogeneity in which Tractor, standard GWAS, or an allele-frequency-difference-dependent choice has the highest power. Finally, I consider real admixed genomes in the UK Biobank and compare the distribution of ancestry-specific allelic effect size heterogeneity with my findings for the power for GWAS.

In Chapter 4, I create a machine learning model to find patients likely to have an underlying condition using electronic health record data. First, I adapt an existing machine learning method (PheNet) for use in the UC Health Data Warehouse, incorporating data from five university health systems, updating the statistical methodology to ridge regression on the way. Next, I implement data-driven feature selection, using likelihood ratio tests to iteratively add clinical features to a model. I also used data-driven feature selection to add recurrence features to the model, which we

inferred from longitudinal health record data using information about antibiotic usage. Last, I applied the new machine learning model to a new phenotype using ridge regression, data driven feature selection, and recurrence, to come up with an out of the box method to find undiagnosed cardiac amyloidosis patients after a heart failure.

In the future, I am interested in continuing to study how to best model electronic health record data. Our database, the UC Health Data Warehouse, is comprised of six university health systems, all with their own unique patient populations and internal policies. Understanding this structure and the best ways to incorporate six potentially disparate health systems into one machine learning model is an important problem to be able to overcome. Additionally, while CVID is a heterogeneous disease, the heterogeneity of the control population is in some ways much larger. Methods that can appropriately account for and potentially discard this type of variation will increase the portability of any method trained on electronic health record data.

Overall, this dissertation has served to add to the fields of differential sensitivity analysis, GWAS in admixed populations, and machine learning for electronic health records. It is my hope that the works described can help improve human health and equity in the future.