INTERPRETABLE MACHINE LEARNING IN PLANT GENOMES: STUDIES IN MODELING AND UNDERSTANDING COMPLEX BIOLOGICAL SYSTEMS

By

Christina Brady Azodi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Plant Biology—Doctor of Philosophy

2019

ABSTRACT

INTERPRETABLE MACHINE LEARNING IN PLANT GENOMES: STUDIES IN MODELING AND UNDERSTANDING COMPLEX BIOLOGICAL SYSTEMS

By

Christina Brady Azodi

Complex systems are ubiquitous in genetics and genomics. From the regulation of gene expression to the genetic basis of complex traits, we see that complex networks of diverse cellular molecules underpin the natural world. Driven by technological advances, today's researchers have access to large amounts of omics data from diverse species. At the same time, improvements in computer processing and algorithms have produced more powerful computational tools. Taken together, these advances mean that those working at the interface of data science and biology are poised to better model and understand complex biological systems. The research in this dissertation demonstrates how a data-driven approach can be used to better understand three complex systems: (1) transcriptional response to single and combined heat and drought stress in *Arabidopsis thaliana*, (2) the genetic basis of flowering time, a complex trait, in *Zea mays,* and (3) the social basis for opinions and beliefs about biotechnology products.

To study the first system, we generated models of the *cis*-regulatory code from information about DNA sequence and additional omics levels using both classic machine learning and deep learning algorithms. We identified 1,061 putative *cis*-regulatory elements associated with different patterns of response to single and combined heat and drought stress and found that information about additional levels of regulation, especially chromatin accessibility and known transcription factor binding, improved our models of the *cis*-regulatory code. To study the second system, we generated phenotype prediction models for flowering time, height, and yield based on either genetic markers or transcript levels at the seedling stage. We found

that, while genetic marker-based models performed better than transcript level-based models, models that integrated both types of data performed best. Furthermore, transcript-based models were more useful for finding genes known to be associated with flowering time, highlighting how using additional levels of omics data can improve our ability to understand the genetic basis of complex traits. Finally, to study the third system, we integrated 29 characteristics about a person (e.g. age, political ideology, education, values, environmental beliefs) into a machine learning model that would predict an individual's beliefs and opinions about five different types of biotechnology products (e.g. biofortification, biopharmaceuticals). While this approach was particularly usefully for identifying individuals that were broadly supportive of biotechnology, finding characteristics of individuals with negative or conditional (i.e. support product A, but not B) opinions was more challenging, highlighting the complexity of public opinions about biotechnology.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who have supported, guided, and challenged me throughout my graduate training. My advisor and mentor, Dr. Shin-Han Shiu, has graciously provided me with support when I needed it and space when I wanted it—allowing me to grow into an independent scientists with confidence and (usually) poise. His intentionality in cultivating a diverse and supportive lab has also benefited me and everyone lucky enough to call themselves a Shiu Lab member or alumni.

I have had the opportunity to work with many excellent collaborators during my training. Dr. Gustavo de los Campos was a wealth of knowledge about all things statistics and has always encouraged me to have bold aspirations. Dr. Andrew McCarren and Dr. Mark Roantree generously hosted me for five months in their lab at Dublin City University and helped me become a stronger data scientist and interdisciplinary researcher. Dr. Thomas Dietz supported my efforts to turn my Environmental Science Policy Program capstone project into a peer reviewed publication and shared priceless nuggets of academic wisdom every step of the way. Dr. Yuying Xie, Dr. Jiliang Tang, and Dr. Yuning Hao and introduced me to state-of-the-art modeling approaches that have the potential to help answer fundamental questions in genetics. My interactions with them has reminded me to always continue learning and innovating. Finally, my committee members, Dr. Yuying Xie, Dr. David Lowry, and Dr. David Kramer, provided me with excellent guidance on my dissertation.

Many of my most joyful moments in graduate school were celebrating victories (both small and large) with my peers and collaborators—research is a team sport. I would like to thank all of my teammates who made every day more enjoyable: Bethany Moore, Siobhan Cusack, Dr.

Peipei Wang, Dr. Sahra Uygun, Dr. Nick Panchy, Dr. Johnny Lloyd, Dr. Ming-Jung Liu, Jeremy Pardo, Birte Schwarz, Dr. Ben Lucker, and Dr. Peter Neofotis.

This dissertation would not have been possible without support from Michigan State University's Institute for Cyber-Enabled Research. Having access to a free, well-supported, high-performance computer cluster has been absolutely critical for my dissertation. I would also like to thank the Department of Plant Biology and the PLB Graduate Student Organization for providing me with an academic and social home.

I would like to thank my parents for their everlasting support and for always encouraging me to do what makes me happy. I would like to thank my sister and my friends for consoling me when science is hard and celebrating with me when science is going well. And finally, I would like to thank my partner Dom Del Ponte, who's levelheadedness and unwavering support have helped me grow from failures and aim even higher. I am unbelievably grateful that you have been with me for every step of my PhD and I cannot wait to begin our next adventure together! See you down under!

TABLE OF CONTENTS

LIST OF FIGURES

KEY TO ABBREVIATIONS

ANN	Artificial neural network
ANOVA	Analysis of variance
BA	Bayes-A
BB	Bayes-B
BL	Bayesian LASSO
BP	Base pairs
BRR	Bayesian ridge regression
BT	Biotechnology
CNN	Convolutional neural network
CNS	Conserved noncoding sequence
DAP	DNA affinity purification
DBH	Diameter at breast height
DE	Wood density
DHS	DNaseI hypersensitive site
DM	Diallel mating
DNA	Deoxyribonucleic acid
DT	Developmental timing
ELU	Exponential linear unit
EN	Ensemble
FC	Fold change
FDR	False discovery rate

FNR	False negative rate
FPR	False positive rate
FT	Flowering time
GBS	Genotype by sequencing
GDD	Growing degree days
GEO	Gene expression omnibus
GM	Grain moisture
GMO	Genetically modified organism
GO	Gene ontology
GP	Genomic prediction
GTB	Gradient tree boosting
HSD	Honestly significant difference
HT	Height
LASSO	Least absolute shrinkage and selection operator
LIM	Local interpretable model-agnostic explanations
LOFO	Leave-one-feature-out
MAS	Marker assisted selection
ML	Machine learning
MSE	Mean squared error
NAM	Nested association mapping
NEP	New ecological paradigm
PCC	Pearson's correlation coefficient
pCRE	putative *cis*-regulatory element

PWM	Position weight matrix
QTL	Quantitative trail loci
RBF	Radial basis function
RF	Random Forest
RKHS	Reproducing kernel Hilbert space
RNA	Ribonucleic acid
rrBLUP	Ridge regression best linear unbiased predictor
SELU	Scaled exponential linear unit
SNP	Single nucleotide polymorphism
ST	Standability
STEM	Science, technology, engineering, math
SVM	Support vector machine
SVR	Support vector regression
TAIR	The Arabidopsis Information Resource
TAMO	Tools for Analysis of Motifs
TF	Transcription factor
TFBM	Transcription factor binding motif
TPR	True positive rate
TSS	Transcription start site
UTR	Untranslated region
WTS	Willingness to sacrifice
YLD	Yield

CHAPTER ONE: INTRODUCTION

1.1 Introduction

In 1990, James D. Watson wrote, "when finally interpreted, the genetic massages encoded within our DNA molecules will provide the ultimate answers to the chemical underpinnings of human existence" (Watson 1990). And in fact, sequencing and assembling the first human genome spurred many novel discoveries in genetics, genomics, and human disease (Collins *et al.* 2003; Hood and Rowen 2013). However, genome sequences feel short of providing what Watson referred to as the "ultimate answers", as scientists began to appreciate the extent to which complicating factors, such as gene-by-environment interactions (i.e. GxE) and gene-by-gene interaction (i.e. epistasis), impact trait variation (Ku *et al.* 2010). Today, we have access to tens of thousands of genome sequences from species ranging from humans to fungi. With advances in transcriptomics, epigenomics, proteomics, and metabolomics (i.e. omics), we also have access to increasing amount of information about cellular molecules besides DNA. As scientists work to decode all of this information, our picture of "the chemical underpinnings of human existence" continues to become more complex (Huang *et al.* 2017; Pinu *et al.* 2019).

Fortunately, complexity is not an adverse quality in biological systems. Rather, complexity is tightly associated with robustness (i.e. the ability of a biological system to persist under perturbations) and evolvability (Carlson and Doyle 2002; Kitano 2004; Whitacre 2010). These qualities are especially vital in plant biological systems because, as sessile organisms, plants have to constantly adjust and evolved to their changing environment (Anderson *et al.* 2011). While beneficial for biological systems, greater complexity does mean that modeling and decoding important biological systems is a challenging task.

To meet the challenges of modeling complex biological systems with large amounts of heterogeneous (e.g. multi-omics) data, biologists are turning more to machine learning. Machine learning has been described as a tool that allows computers to learn patterns from data without being explicitly programmed (Samuel 1959). From this description we can define a few terms and concepts important for machine learning. First, the collection of all of the learned patterns from a particular dataset is called a machine learning model. Next, the input data from which the model learns is made up of a number of examples (i.e. instances) for which we have information about different characteristics (i.e. feature) and, in the case of supervised machine learning, the values that we want to predict (i.e. the label). Finally, instead of having to define the nature of the relationship between a set of features and the label (e.g. linear, exponential, A interacts with B), the ability to learn without being explicitly programmed means that machine learning models learn these relationships from the examples provided. In addition to reducing the influence of human bias on a model, this also means that machine learning models are able to represent more complex systems.

While the capacity of machine learning models to outperform classical statistical models has led to their increased use for modeling complex biological systems (Tarca *et al.* 2007; Ma *et al.* 2014; Libbrecht and Noble 2015; Angermueller *et al.* 2016; Chicco 2017; Cuperlovic-Culf 2018), decoding these models can be more difficult (Lipton 2018; Guidotti *et al.* 2018). To improve our ability to interpret machine learning models and thus gain novel biological insights into complex systems that are modeled well by these algorithms, Chapter 2 is a review of interpretable machine learning. The review highlights different types of strategies that can be used to better understand the patterns that a machine learning model has learned. While the strategies reviewed can be used to interpret machine learning models built to learn any system,

the review focuses on how these strategies can be used to better understand genetics and genomics.

The chapters following this review, present my work on using machine learning to predict and understand complex systems (Figure 1.1). Chapter 3 focuses on elucidating how model plant, *Arabidopsis thaliana*, regulates its response to single and combined heat and drought stress at the genetic level. Here, I use machine learning to integrate DNA sequence information (i.e. known and putative regulatory elements) and additional regulatory information (e.g. chromatin accessibility and histone marks) into models that are predictive of different patterns of response to single and combined heat and drought stress. These models were then interpreted in order to identify the regulatory elements and additional regulatory information driving these predictions.

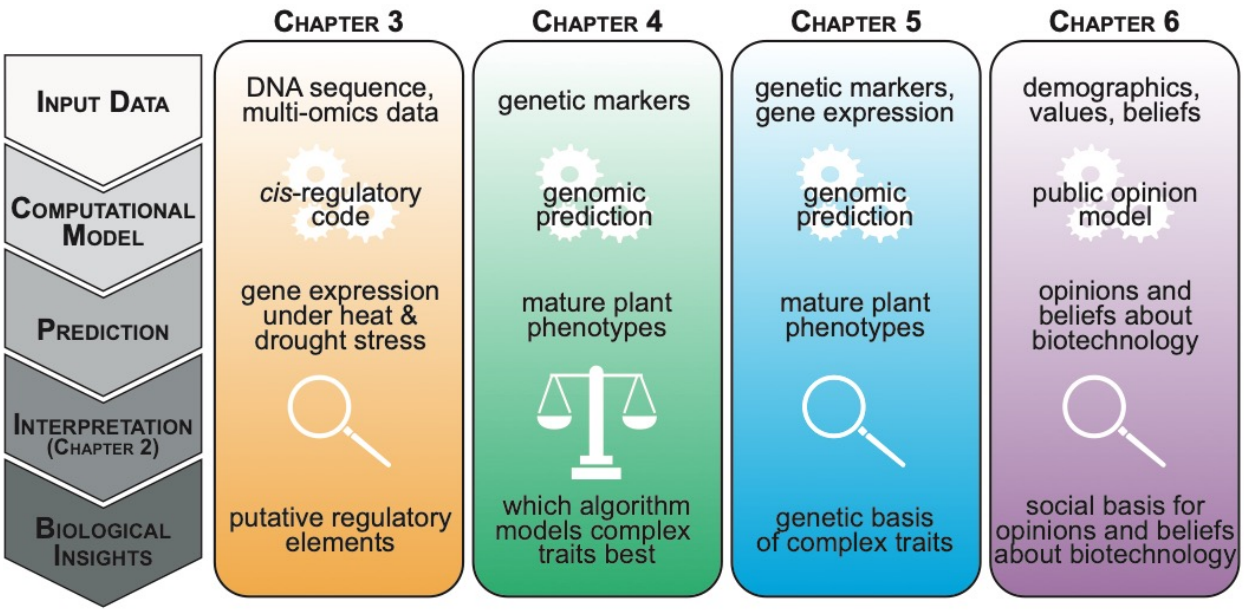


Figure 1.1. Overview of the content of this dissertation.

Chapters 4 and 5 focus on using genetic information to predict mature plant phenotypes (e.g. yield and flowering time), an approach known as genomic prediction. Chapter 4 provides a thorough comparison of the performance of different algorithms for the genomic prediction of 18

different traits across six diverse plant species. This study focuses on how non-linear algorithms, including classic machine learning and deep learning based algorithms, compare to the linear-regression based algorithms that were first used for genomic prediction (Meuwissen *et al.* 2001; de los Campos *et al.* 2013). Then, Chapter 5 describes my work to use both genetic markers and gene expression levels from seedlings to predict mature phenotypes in *Zea mays*. These genomic prediction models were compared to determine which type of data was the most useful for predicting traits. In addition, because a great deal is already known about the genetic basis of flowering time in *Z. mays,* we used a set of known flowering time genes as a benchmark to determine if genetic marker-based or gene expression-based genomic prediction models were better for helping us understand the genetic basis of flowering time.

While up to this point, the focus has been on modeling and improving our understanding of complex systems in plant genetics and genomics, complex systems are ubiquitous in other fields (De Laurentiis *et al.* 2016; Kapsar *et al.* 2019). Toward demonstrating this, Chapter 6 describes my work using interpretable machine learning to better understand the social basis for public opinion about biotechnology. Here I use factors including age, gender, religion, politics, personal values, and environmental beliefs, to predict an individual's beliefs and options about five different types of biotechnology products. This chapter is the product of my capstone research for the Environmental Science Policy specialization.

REFERENCES

REFERENCES

Anderson, J. T., J. H. Willis, and T. Mitchell-Olds, 2011 Evolutionary genetics of plant adaptation. Trends in Genetics 27: 258–266.

Angermueller, C., T. Pärnamaa, L. Parts, and O. Stegle, 2016 Deep learning for computational biology. Molecular Systems Biology 12: 878–16.

de los Campos, G., J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, and M. P. L. Calus, 2013 Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding. Genetics 193: 327–345.

Carlson, J. M., and J. Doyle, 2002 Complexity and robustness. PNAS 99: 2538–2545.

Chicco, D., 2017 Ten quick tips for machine learning in computational biology. BioData Mining 10: 35.

Collins, F. S., M. Morgan, and A. Patrinos, 2003 The Human Genome Project: Lessons from Large-Scale Biology. Science 300: 286–290.

Cuperlovic-Culf, M., 2018 Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling. Metabolites 8: 4.

De Laurentiis, V., D. V. L. Hunt, and C. D. F. Rogers, 2016 Overcoming Food Security Challenges within an Energy/Water/Food Nexus (EWFN) Approach. Sustainability 8: 95.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti *et al.*, 2018 A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys 51: 1–42.

Hood, L., and L. Rowen, 2013 The Human Genome Project: big science transforms biology and medicine. Genome Med 5: 79.

Huang, S., K. Chaudhary, and L. X. Garmire, 2017 More Is Better: Recent Progress in Multi-Omics Data Integration Methods. Front. Genet. 8:.

Kapsar, K. E., C. L. Hovis, R. F. Bicudo da Silva, E. K. Buchholtz, A. K. Carlson *et al.*, 2019 Telecoupling Research: The First Five Years. Sustainability 11: 1033.

Kitano, H., 2004 Biological robustness. Nat Rev Genet 5: 826–837.

Ku, C. S., E. Y. Loy, A. Salim, Y. Pawitan, and K. S. Chia, 2010 The discovery of human genetic variations and their use as disease markers: past, present and future. J Hum Genet 55: 403–415.

Libbrecht, M. W., and W. S. Noble, 2015 Machine learning applications in genetics and genomics. Nature Publishing Group 16: 321–332.

Lipton, Z. C., 2018 The Mythos of Model Interpretability. ACM Queue 16:.

Ma, C., H. H. Zhang, and X. Wang, 2014 Machine learning for Big Data analytics in plants. Trends in Plant Science 1–11.

Meuwissen, T. H. E., B. J. Hayes, and M. E. Goddard, 2001 Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps. Genetics 1–11.

Pinu, F. R., D. J. Beale, A. M. Paten, K. Kouremenos, S. Swarup *et al.*, 2019 Systems Biology and Multi-Omics Integration: Viewpoints from the Metabolomics Research Community. Metabolites 9:.

Samuel, A. L., 1959 Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development 3: 210–229.

Tarca, A. L., V. J. Carey, X. Chen, R. Romero, and S. Drăghici, 2007 Machine Learning and Its Applications to Biology. PLOS Computational Biology 3: e116.

Watson, J. D., 1990 The human genome project: past, present, and future. Science 248: 44–49.

Whitacre, J. M., 2010 Degeneracy: a link between evolvability, robustness and complexity in biological systems. Theoretical Biology and Medical Modelling 7: 6.

CHAPTER TWO: A LIGHT IN THE BLACK BOX: INTERPRETABLE MACHINE LEARNING FOR GENETICISTS

2.1 Abstract

As we move further into the Era of Big Data, geneticists are turning more and more to machine learning (ML) to make sense of the deluge of omics data now available. Machine Learning is a subfield of artificial intelligence that focuses on generating models that learn from data without being explicitly programmed. ML models are well suited to address challenges unique to genetic and genomic data including high dimensionality (e.g. predicting trait values from millions of genetic markers), complex systems (e.g. mapping gene regulatory networks), and high order interactions (e.g. identifying epistatic interactions). While the complexity of ML models is what makes them so powerful, it also makes them difficult to interpret. Fortunately, researchers have developed strategies to make the inner workings of machine learning models understandable *to humans*, and in doing so have made it possible to derive novel biological insights from ML models. In this review, we discuss what types of strategies for interpreting ML models are available, how they work, and how they can be used in a biological context. Finally, we describe challenges and promising future directions in interpretable ML for biology.

2.2 Why is interpretable machine learning needed?

Thanks to the advances in technology and reduced cost to generate data, biologists are now living in the Era of Big Data (Marx 2013; Stephens *et al.* 2015). In this Era, big data will drive progress in biological fields ranging from population genetics (Schrider and Kern 2018) to precision medicine (Alyass *et al.* 2015). But big data also presents researchers with new challenges, such as how to derive biological understanding from large amounts of heterogeneous data (e.g. multi-omics data) and how to model highly complex systems (e.g. gene regulation and protein folding). In order to address the challenge of harnessing big data to answer biological questions, bioinformaticians and computational biologists are now turning to machine learning

(ML; Figure 2.1) (Tarca *et al.* 2007; Ma *et al.* 2014; Libbrecht and Noble 2015; Angermueller *et al.* 2016; Chicco 2017; Cuperlovic-Culf 2018). Arthur Samuel, a pioneer of machine learning, described it as a "field of study that gives computers the ability to learn without being explicitly programmed" (Samuel 1959). One common criticism faced by those employing ML in biology and elsewhere is that the ML models are "Black Boxes". While this term lacks a precise definition (Lipton 2018), broadly it means that only model inputs and outputs, but not the internal logic, can be understood *by a human*. However, not all ML models are equally "Black Boxes". For example, the internal logic of ML models based off decision trees is inherently interpretable. On the other hand, the internal logic of a deep learning model, which could be made up of hundreds or hundreds of thousands of connections and hidden variables, is much more of a black box.

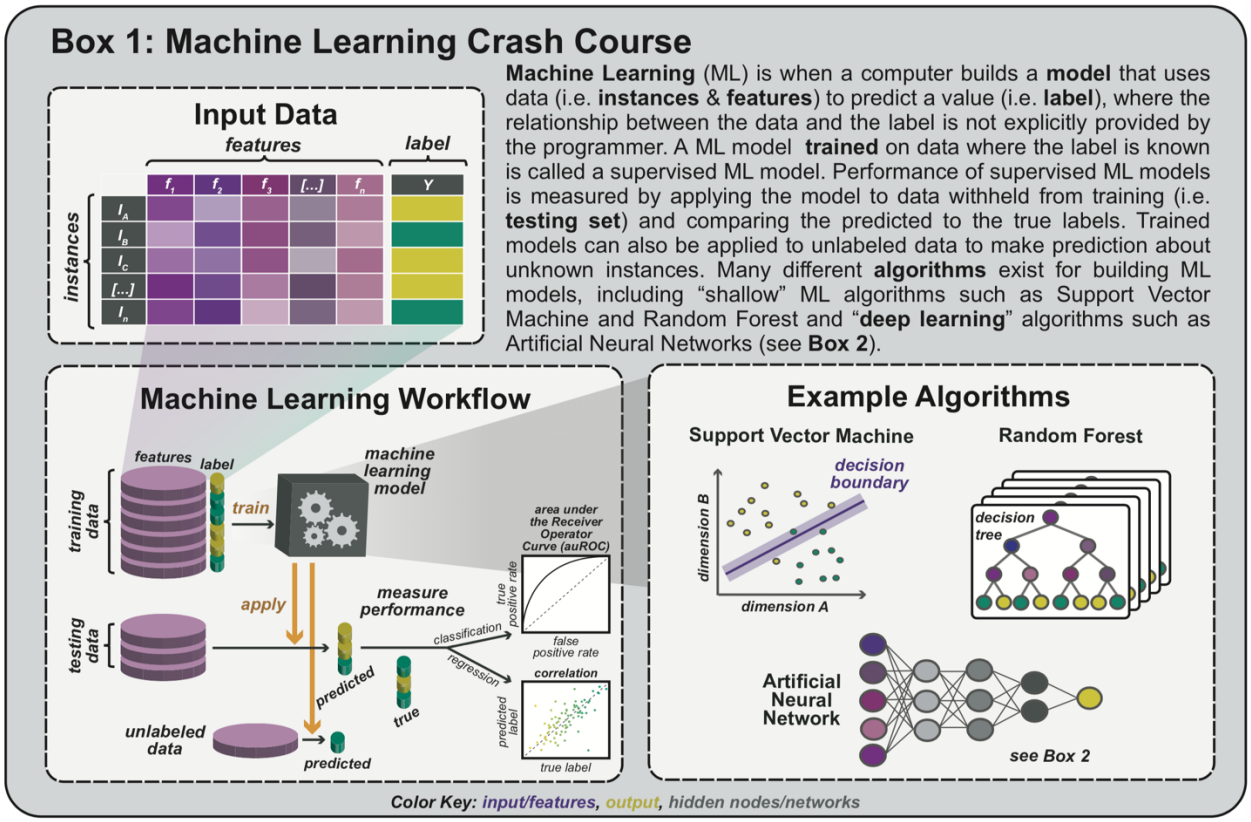


Figure 2.1. Machine Learning Crash Course.

There are three major reasons why ML model interpretation is important: troubleshooting, novel insights, and trust (Figure 2.2). First, models rarely perform best without tweaking or troubleshooting and understanding why mispredictions are made is essential for determining if there were mistakes or biases in the input data or if there were issues with how the model trained. Second, an ML model with impressive performance may have identified biologically interesting patterns in the data. However, scientists could not learn such biological insights without interpreting the model. Finally, we are not keen to trust things if we do not understand how and why they work and without trust ML models will not be used to their full

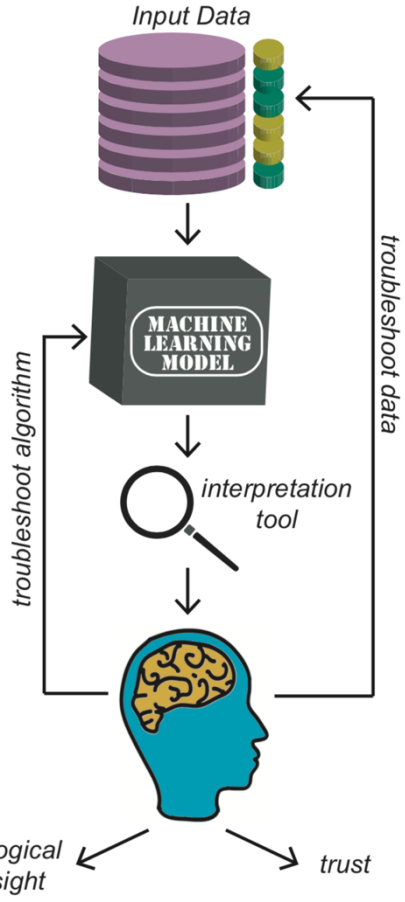


Figure 2.2. Why interpretable machine learning?

Interpretability of machine learning models is important for facilitating troubleshooting, making biological insights, and trust.

potential. For example, ML models have the potential to improve our ability to diagnose and treat diseases. However, doctors will be reluctant to trust models that have not been or cannot be readily interpreted because they won't understand their medical basis and may worry the models are capturing artifacts (Miller 2017). Furthermore, many patients would be reluctant to undergo treatment without knowing why that treatment was selected for them. Therefore, model interpretability, or the ability to understand what logic is driving a model's performance, is critical for the those using ML for biology.

2.3 Overview of strategies for interpretable machine learning

Interpretable ML is an emerging focus among data scientists. Here we will review three strategies for interpretability—probing, surrogate, and perturbing (Figure 2.3)—that have been used to interpret ML models in biology. Probing strategies involve dissecting the inner structure of a trained model. Surrogate strategies involve training classically interpretable models that estimate an ML model's predictions. Perturbing strategies involve measuring the change in model performance before and after disturbing features or instances (Guidotti *et al.* 2018; Molnar 2019).

Interpretation strategies can also be defined based on if they are applicable to all algorithms (i.e. model-agnostic) or only to one or a subset of algorithms (i.e. model-specific) (see Figure 2.1 for example algorithms). Finally, interpretation strategies can be either global or local. Global interpretation involves explaining the overall relationship between features and labels. While local interpretation strategies focus on explaining the prediction of an individual instance. For example, imagine you train an ML model to predict if a gene (i.e. the instance) is up-regulated when the organism is exposed to an environmental toxin (i.e. the label) based on the presence or absence of a set of known regulatory sequences (i.e. the features). A global

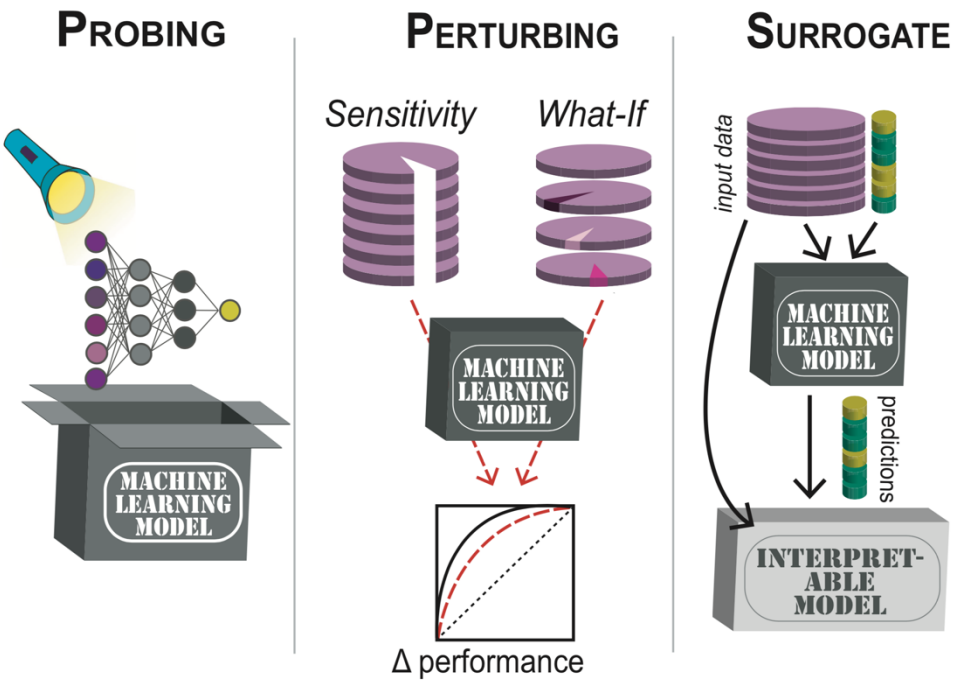


Figure 2.3. Overview of strategies in interpretable machine learning.

ML models can be interpreted by probing trained models (gray box), training interpretable surrogate models (white box), or perturbing the input data (purple data stack) and measuring the change in performance. Probing strategies are characterized by which algorithms (shown here: deep learning model) they are used for. Surrogate models can be trained to represent global or local predictions. Perturbing strategies, which are algorithm agnostic, are characterized based on if a feature (i.e. Sensitivity analysis) or an instance (i.e. What-If analysis) is perturbed.

interpretation strategy would tell you how important regulatory sequence X was for predicting up-regulation across all of the genes in your dataset. In contrast, a local interpretation strategy would tell you how important regulatory sequence X was for predicting gene Y as up-regulated. This example will be used repeatedly throughout the review to explain various concepts in ML and interpretable ML.

We should emphasize that, because ML models identify association through correlation, ML interpretation strategies do not identify causal relationships between input features and labels. For example, if putative regulatory sequence X is present in the promoter region of all up-regulated but no non-up-regulated genes, X is considered highly correlated with up-regulation, but we cannot claim X is responsible for that response.

2.4 Probing strategies for interpreting machine learning models

Training a machine learning model involves identifying the best set of parameters to predict the label of interest (e.g. is the gene up-regulated). After training is complete, those parameters can be probed to better understand what the model learned. Probing strategies tend to provide global interpretations of trained models. However, some strategies for probing deep learning models provide local interpretations (e.g. DeepLIFT, see description below). Because the structure and type of parameters learned by ML models vary by algorithm, probing strategies are algorithm-specific. In the following sections, we will discuss how different types of ML algorithms can be probed and provide examples of how these strategies have been used to answer fundamental questions in biology.

2.4.1 Probing Support Vector Machine models

Support Vector Machine (SVM) is an algorithm that finds the hyperplane that best separates instances by their label (for classification tasks) or best approximates the label values (for predicting continuous labels). Using predicting gene up-regulation as an example, the hyperplane would lie in a multi-dimensional space defined by the presence or absence of the regulatory sequences (i.e. features) and would separate genes (i.e. instances) that are up-regulated from those that were not up-regulated (i.e. the label). (Figure 2.4A). These models identify linear relationships between features and labels, but they can be modified to identify

non-linear relationships by using a non-linear function (e.g. polynomial) to map the data into a non-linear feature space. While there are advanced methods for probing non-linear SVM models (Barakat and Bradley 2010; Rasmussen *et al.* 2011), in biological applications of SVM, only linear SVM models are typically interpreted with probing strategies.

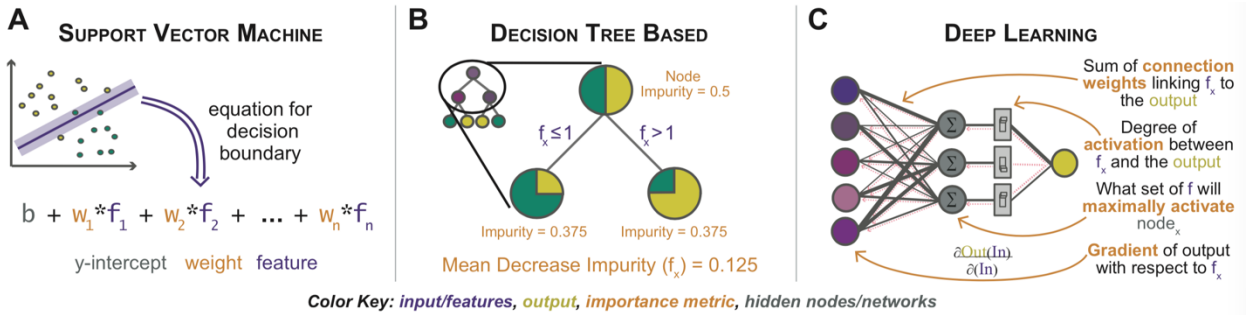


Figure 2.4. Detailed overview of the probing strategies.

Probing strategies used to understand the predictions made by **(A)** support vector machine, **(B)** decision tree based, and **(C)** deep learning models. Input features (e.g. genetic markers, environmental variables, image pixels) are shown in purple, the labels (e.g. state of differential expression, diagnosis) are shown in green/yellow, and the interpretation strategies (e.g. coefficient weight, mean decrease impurity, gradient) are shown in orange.

A linear SVM model is probed by extracting the trained weights (i.e. coefficients) that define the hyperplane (Figure 2.4A). These weights directly represent the relationship between the feature and the label, making their interpretation relatively straight forward. For example, Ronen *et al.* trained a linear SVM model to classify simulated populations as being under positive or negative selection, using genetic markers as features (Ronen *et al.* 2013). They found genetic markers with large, positive weights in their SVM model (indicating strong positive selection) were also found to be associated with positive selection by traditional population genetics statistical tests (e.g. Tajima's *D* and Fay and Wu's *H*) with positive selection. However,

SVM probing strategies, like many other interpretation strategies covered in this review, need to be interpreted with caution because they can provide an incomplete picture of what features are important for the model. For example, two highly correlated features will split the coefficient weight between them, effectively reducing the importance of each feature by half. Or a feature with a highly non-linear effect may not be assigned a high coefficient weight by a linear SVM model and will therefore be missed in the interpretation (see Challenges and Opportunities).

2.4.2 Probing decision tree-based models

A decision tree is a set of true/false questions nested in a hierarchical structure. They are inherently interpretable because the content and order of each true/false question can be directly observed from the tree and the path for each instance through a decision tree can be traced. For example, for predicting gene up-regulation using regulatory sequences as features, the first question in a trained decision tree can be, "is regulatory sequence X present?" and if the answer is yes then the second question is, "is regulatory sequence Y present?" From this we can infer that, based on the trained model, regulatory sequence X best separates up-regulated from non-up-regulated genes, and that if regulatory sequence X is present, the presence or absence of regulatory sequence Z is the next most useful information. How well a given true/false question separates up-regulated from non-up-regulated genes can be quantified, for example by measuring the change in node impurity, or how many instances of a different class are present in a node, before and after a particular feature is used to separate the genes (Figure 2.4B). Where a regulatory sequence that is present in up-regulated but not in non-up-regulated gene promoters would be able to split the genes perfectly, resulting in a large decrease in node impurity, and therefore interpreted as being very important.

Single decision trees often do not perform well at predicting complex biological patterns. Instead, tree-based algorithms that take advantage of ensemble methods tend to perform better in biological applications (Rokach 2016). In ensemble methods, many decision trees are trained on small subsets of the data to produce many "weak" models that when combined provide one strong model (e.g. Random Forest, Gradient Tree Boosting, Extra-Tree, etc. (Breiman 2001)). However, the "forest" of decision trees is more complicated to interpret. One way to probe ensemble decision-tree based models is average the decrease in node impurity every time a feature is uses in a tree in the ensemble, providing a mean decrease in impurity (a.k.a. Gini Importance) score for each feature. For example, Uygun *et al.* used the mean decrease impurity to determine which DNA motifs (i.e. features) best classified genes as differentially expressed or not differentially expressed after a model plant, *Arabidopsis thaliana*, was exposed to high salinity stress (Uygun *et al.* 2017).

In addition to facilitating intuitive model interpretation, the hierarchical structure of decision tree-based models allows them to inherently model interactions between features—and those interactions can be probed. Given the complexity of many biological systems (e.g. neuronal networks, gene regulatory networks, protein-protein interactions), interpretation strategies that can pinpoint important interactions are useful. Using iterative Random Forest, a tool for finding stable feature interactions in RF models (Basu *et al.* 2018), Vervier and Michaelson identified interactions between genomic, transcriptomic, and epigenomic features that were predictive of deleterious genetic variants (Vervier and Michaelson 2018). They found that the local GC content and the distance to the nearest expression Quantitative Trait Loci for a genetic variant were consistently found to interact that was important for driving the model performance.

As with SVM based importance scores, feature importance scores and interactions from decision-tree based models should be interpreted with caution. For example, mean decrease impurity tends to be inflated for continuous over categorical features, categorical features with a large number of categories over those with few categories, and continuous features on a large scale (e.g. range = 0-100) over those on a small scale (e.g. range = 0-1) (Strobl *et al.* 2007). Therefore, this strategy should only be used when the input feature types are relatively uniform. In addition, while tree-based models are readily interpretable, deep learning algorithms outperform even ensemble decision tree-based methods at predicting complex patterns, in biology and in other fields. Deep learning algorithms also benefit from the ability to learn from raw data (e.g. whole DNA sequence), rather than user defined features (e.g. enriched sequence motifs) and are therefore being turned to more and more by the ML community.

2.4.3 Probing deep learning networks

In statistics, there is often a tradeoff between predictability and interpretability, and this is certainly the case for deep learning (Figure 2.5). Given that deep learning models have been shown to outperform classic ML models (e.g. regression, decision trees) in applications ranging from machine translations to computer vision (LeCun *et al.* 2015; Guo *et al.* 2016; Banerjee *et al.* 2017), there has been a substantial effort to develop new methods to interpret these complex models. Because interpreting neural networks is an active area of research, there is yet a consensus on the best interpretation methods. Here we describe three general strategies for probing deep learning models (Supplemental Table 2.1) and provide specific examples of how some of these strategies have been used in biology.

Like the coefficient weights and the Gini Importance scores for SVM and RF models, respectively, similar metrics indicating feature importance for neural networks can be derived by

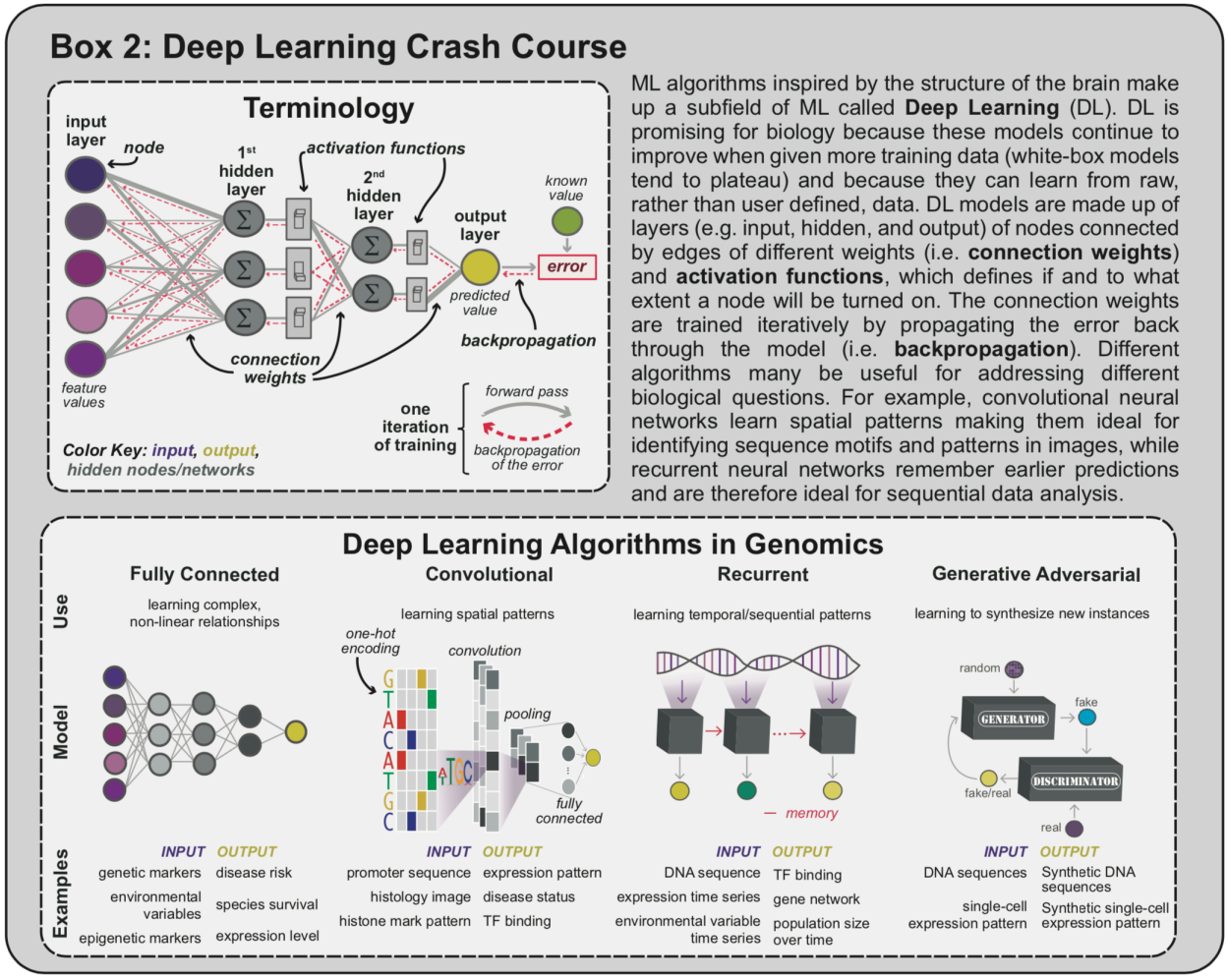


Figure 2.5. Deep Learning Crash Course.

probing three different components of a trained network: (1) the weights of the connections between nodes in different layers of the network, (2) the gradient of the output with respect to the input, and (3) the activation level at every connection for a given instance (Figure C). In the first category, connection weight-based methods quantify the overall relationship between each feature and the output by summing the connection weights from input-to-hidden, hidden-to-hidden, and hidden-to-output nodes for each input feature (Garson 1991; Olden and Jackson 2002). For example, following the path through the neural network in Figure 2.5 between an

input feature (e.g. i_1) and the output layer, you notice the paths for different input features have different sized connection weights (represented by the widths of the gray lines). If we were to use some handy linear algebra to sum the connection weights along the path through the network for each feature, we could quantify the extent to which some features (e.g. i_1) are more important for predicting the value of the label in the output layer than others (e.g. i_3). This approach was used to determine which of 179 microRNAs (i.e. input features) had the highest connection weights to the expression level of Smad7 (i.e. the label), a gene that is involved in disrupting a signaling process that gets up-regulated in patients with breast cancer (Manzanarez-Ozuna *et al.* 2018).

The second category of neural network importance scores are gradient-based scores (sometimes referred to as Saliency). In the case of interpretable deep learning, the gradient refers to the change in the predicted label value as the values of an input feature is changed. This is not to be confused with the use of the word gradient in describing how neural networks are trained (i.e. gradient descent). These scores are calculated using a handy trick from calculus, the partial derivative, which measures the change in the output (i.e. the predicted label value) due to making tiny changes in the input (i.e. the feature value). A feature with a large gradient (e.g. i_1) is one where a small change in the input feature value would result in a big change in the prediction (Simonyan *et al.* 2013). Kelley *et al.* used this approach to identify putative distal regulatory sequences in genomic regions where positive and negative gradient-based importance score peaks represented enhancer and silencer regions, respectively (Kelley *et al.* 2018). Although weigh-based and gradient-based approaches can provide interpretations of deep learning models, there are situations where their applications can be misleading. For example, connection weight-based importance scores are not directly comparable when features are on different scales, they

underestimate importance when positive and negative weights cancel each other out, and they can overestimate importance when connections with large weights are rarely activated (i.e. rarely turned on) (REF). Similarly, gradient-based importance scores are not useful when input features are categorical (e.g. true/false, present/absent) or when the signal from an input feature is saturated (i.e. small changes in the feature value will not change the prediction) (Shrikumar *et al.* 2017).

The third category, activation-based interpretation strategies, such as DeepLIFT (Deep Learning Important FeaTures), avoid these limitations and have therefore become popular for interpreting deep learning models in recent biological applications (Zuallaert *et al.* 2017; Shrikumar *et al.* 2017; Washburn *et al.* 2019). DeepLIFT works by comparing the activation level of nodes in the network (i.e. the output value of a node after it has passed through the activation function) when a reference instance is input into the trained model compared to when an instance of interest is used. In models where the input is a DNA sequence, the reference instance could be a randomly shuffled DNA sequence or the background nucleotide frequency. Then, for example, given that a DeepLIFT interpretation is unique to the null instance and the instance of interest that are selected, it provides a local interpretation. For example, using DeepLIFT. This approach is robust when features are on different scales or when connection weights are large but rarely activated because it relays on activation rather than connection weights. Furthermore, it is robust against categorial features and saturated nodes because it looks for important differences between the null and the instance of interest.

In addition to the three strategies described above for probing the importance of each feature to the prediction overall, another promising strategy is to probe each hidden node to see what pattern it has learned to identify. One approach involving this type of strategy is to feed the

trained model either real or fake instances to identify which ones maximally activate a node, referred to as activation maximization**,** and associate the properties of those real or fake instance to that node. For example, if the 10 DNA sequence that maximally activate node X (i.e. cause node X to have the maximum possible output value) all contain the motif ACGGTC, one could associate that motif to node X. Other strategies to probe hidden nodes are unique to the type of deep learning algorithm. For example, Esteva *et al.* used a dimensionality reduction technique (t-Distributed Stochastic Neighbor Embedding: tSNE) to visualize the nodes in the last hidden layer of a convolutional new network trained to diagnose different types of skin cancer from photos (Esteva *et al.* 2017).

2.5 Perturbing strategies for interpreting machine learning models

Perturbing strategies involve modifying the input data and observing changes in the model output. Because modifications to the input data can be made regardless of the type of ML algorithm applied, perturbing strategies are generally model-agnostic (although there are some perturbing strategies particular to deep learning models that will not be discussed here). Next we discuss two general perturbation based strategies: sensitivity analysis and What-if methods.

2.5.1 Sensitivity Analysis

Sensitivity analysis involves modifying the input features and measuring the extent to which these modifications impact overall model performance, which provides a global measure of importance (Figure 2.6A). Feature modification is typically done by either removing (i.e. leave-one-feature-out) or permuting (e.g. set all values to the mean, or randomly shuffle) features one at a time during training and observing the change in predictive performance of the trained model. The result from a sensitivity analysis is a highly intuitive score for each feature that indicates its overall contribution as the decrease in model performance when the feature in

question is removed or permuted (Figure 2.6A). Because perturbing a feature not only impacts that feature but also other features that interacted with it, this type of analysis captures both the main and interaction effects for each feature.

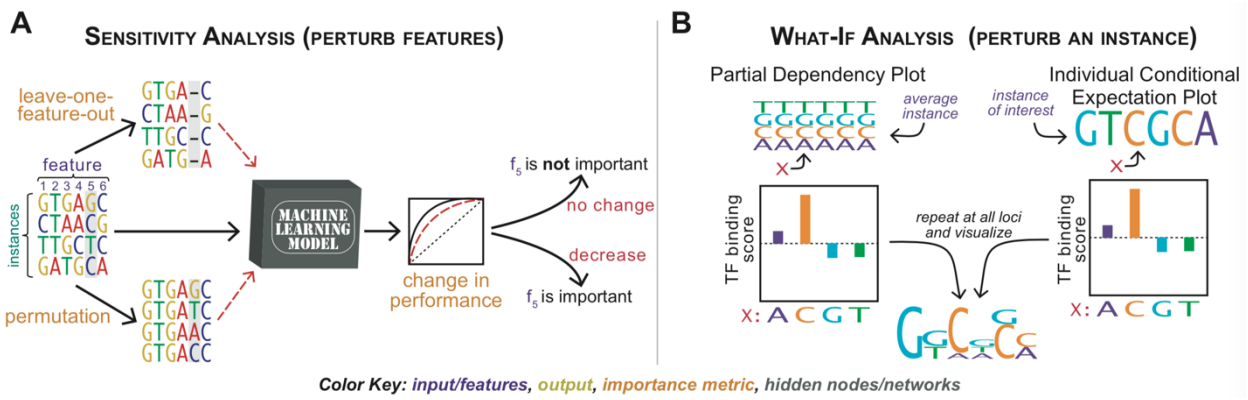


Figure 2.6. Detailed overview of the perturbing strategies.

Colors represent the same data as in Figure 2.4.

One approach for assessing feature importance using a sensitivity analysis is to compare the performance of a model trained on the original dataset compared to a model trained on the perturbed dataset (i.e. leave-one-feature-out). This approach was used to determine what sequence features were most important for identifying genomic islands containing clusters of genes acquired by horizontal gene transfer (Che *et al.* 2010). They found models trained without information about size (i.e. length of the genomic region in base pairs) had a 13% lower performance as measured by area under the Receiver Operator Characteristic curve (i.e. a plot of the true positive rate against the false positive rate at various thresholds). However, this type of sensitivity analysis can be computationally expensive as it requires training new models for every perturbed dataset. Because of the greater computational cost for training deep learning models, this type of sensitivity analysis is typically only used to interpret deep learning model when there are few input features. For example, by building convolutional neural networks

(CNN, see Figure 2.5, (Schmidhuber 2015), leave-one-feature-out was used to determine which of five histone marks were most important for predicting transcription factor binding sites (Jing *et al.* 2019). They found the H3K4me3 mark was the most important for their model predictions, which is consistent with what is known about H2K4me3 being associated with active transcription of nearby genes (Roadmap Epigenomics Consortium *et al.* 2015).

Another way to reduce the amount of computational power needed to perform a sensitivity analysis is to train one model on the full dataset, then measure how the model performance on a specific instance or a held out set of data changes when different features are perturbed. This type of approach has also been demonstrated to be well suited for interpretable ML in genetic studies because they mirror mutagenesis experiments. For example, *in silico* mutagenesis, i.e., permuting nucleotides, was used to identify which changes in the DNA sequence most impacted tissue specific gene expression (Zhou *et al.* 2018). First, they trained a series of models that first predicted an epigenomic profile (e.g. histone marks, chromatin accessibility) directly from DNA sequence and then used that epigenomic profile to predict tissue specific gene expression levels. After training their models they were able to mutate (i.e. perturb) over 140 million base pairs and measure the impact those mutations had on expression.

2.5.2 What-if Analysis

What-if methods involve modifying one or more feature values for a single instance and observing to what degree the modification impacts the prediction for that instance. Because the focus is on a specific instance, this provides a local measure of importance, as opposed to sensitivity analysis which looks at the global impact of feature space modifications on model performance. What-if interpretation strategies are also sometimes called counterfactual strategies, where counterfactuals are the name given to instances that have been modified and fed

back to the model (Wachter *et al.* 2018). Here we will discuss three What-if methods: partial dependency plots, individual conditional expectation plots, and occlusion sensitivity.

Partial dependency plots show the effect of changing the value for a specific feature on the prediction of an instance when all other feature values are averaged (Friedman 2001). For example, imagine we trained a machine learning model that predicts the likelihood that a sequence will be bound by a certain transcription factor (TF). To determine how important position #3 is for TF binding, we could generate an instance with the background nucleotide frequency at the other positions, then make a partial dependency plot showing the change in the TF binding likelihood when each nucleotide is placed at position #3 (left side; Figure 2.6B). These plots are useful because they show the magnitude, direction, and non-linearities in the relationship between a feature and the label. However, one limitation is that, dependencies can only be visualized for one or two features at a time, so these plots are typically only generated for models with few features, which are uncommon in most application of machine learning in genetics and genomics, or when a subset of features deemed important based on another interpretation strategy have already been identified (Liu and Yang 2014).

A second limitation of partial dependency plots is that they can obscure patterns when there are interactions between features (e.g. the influence of regulatory sequence X on expression depends on the presence or absence of regulatory sequence Y) or when the effect of a feature is heterogeneous across the instances (e.g. regulatory sequence X is associated with up-regulation in some genes, but not for others where regulatory sequence X is highly methylated). To alleviate the impact of feature interactions, Goldstein *et al.* suggested that dependency plots could be generated for every instance in the datasets, instead of on one averaged instance, an approach they termed individual conditional expectation (Goldstein *et al.* 2015). For example,

instead of looking at the impact of changes in the nucleotide at position #3 when other positions are set to the background nucleotide frequency, an individual conditional expectation plot could be generated for a specific sequence (right side; Figure 2.6B).

Individual conditional expectation plots have been used to interpret deep learning models that utilize adversarial learning. Adversarial learning is when two deep learning models are trained by competing with one another, in a sort of machine learning arms-race (Dalvi *et al.* 2004; Biggio and Roli 2018). For example, Generative Adversarial Networks (GANs, see Figure 2.5) are composed of a generator, which generates simulated data, and a discriminator, which has access to the real data and tries to determine if the simulated data is real or not (Goodfellow *et al.* 2014). The generator and discriminator compete with one another, with the generator getting better at simulating real data and the discriminator getting better at spotting simulated data. This type of approach was used to better understand the diversity of gene expression patterns at the single cell level, where a generator was trained to simulate realistic single cell gene expression levels and a discriminator was trained to classify these simulations as real or not given a set of diverse, real single cell expression data (Ghahramani *et al.* 2018). To determine what patterns of gene expression were characteristic of real single cell expression, they generated individual conditional expectation plots by varying the expression level of known epidermal cell marker genes and observing the change in the prediction from the discriminator. Using this approach, Ghahramani et al. were able to train a discriminator that was sensitive to changes in the expression levels for genes that were known markers for particular cell-type states (e.g. IvI, Krt10, and Krt14 for epidermal cell state). Therefore, such a discriminator can potentially be used to identify novel markers.

Finally, What-if approaches are useful for interpreting deep learning models used to classify images and have therefore been used to better understand models generated to classify medical images. For example, convolutional neural networks were used to classify blood samples as infected or non-infected with malaria based on blood smear images (Rajaraman *et al.* 2018). To make sure these models were basing their classification off differences in parasitized regions as opposed to unrelated background signals, they "grayed out" different regions of an image and found that graying our regions with visible parasites decreased model performance more than graying out un-parasitized regions. This approach is often referred to as occlusion sensitivity because it involves excluding pixels from the image.

2.6 Surrogate strategies for interpreting machine learning models

Image you have an ML model that is truly a Black Box—meaning it cannot be probed or perturbed, or that these strategies do not provide useful information. In such a case, one interpretation strategy is to use an inherently interpretable model as a surrogate for the Black Box model. The surrogate model could be a well-established statistical model (e.g. linear regression) or it could be an ML model that is easier to interpret (e.g. decision tree). For example, to generate a surrogate for a Black Box model that can predict gene up-regulation under toxin stress with great accuracy, we would first apply the Black Box model to a set of genes to get predictions. Then we would train our selected interpretable surrogate model (e.g. logistic regression) on the same set of genes to learn the prediction from the Black Box model (i.e. the surrogate label) and interpret the logistic regression model by observing the *p*-values and/or effect size.

One major limitation of such an approach is that Black Box models are often highly complex (i.e. a highly non-linear decision boundary) and, thus, cannot be fully and accurately

learned by an interpretable surrogate. In such cases, one approach is to generate a surrogate that can more accurately learn just a portion of the Black Box model. This concept is called LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro *et al.* 2016). The theory behind this is that while the complex, non-linear decision boundary for the full model may be too complex for a surrogate model to learn, the decision boundary for one instance or a group of similar instances (e.g. genes) will likely be simple enough. To generate a LIME model, for example, an instance of interest is selected, then that instance is perturbed many times like that in sensitivity analysis, then the Black Box model is applied to those perturbed instances to generate predictions. After you have Black-Box 2.model predictions for each perturbed instance, an interpretable model is trained to learn those predictions and the interpretable model is then inspected.

Local surrogate models have been used by those interested in understanding predictions in medical settings (Nanayakkara *et al.* 2018; Wang *et al.* 2019). For example, LIME was used to learn explanations for predictions of individual mortality following cardiac arrest (Nanayakkara *et al.* 2018). They focused their interpretation on patients that were misclassified by the Black Box model (i.e. predicted to survive but did not). For example, they found for one patient that was given a 78% probability of survival due to favorable features (e.g. healthy neurologic status, lack of chronic respiratory illness), other negative features were indicators of mortality (e.g. elevated creatinine, advanced age).

2.7 Challenges and Opportunities

While a great deal of effort has gone into developing the interpretation strategies discussed above, there are still several technical and practical challenges to interpreting machine learning models in genetics.

- **Correlation**: when the input data contains features that are highly correlated, both probing and perturbing interpretation strategies can underestimate the importance of a feature by splitting the importance between the correlated features (i.e. during probing) or by compensating for one feature when it is left out or permuted (i.e. during perturbing) (Altmann *et al.* 2016).
- **Heterogeneous input space**: Multi-omics data integration is an area of computational biology receiving much attention (Pinu *et al.* 2019). However, some interpretation strategies (e.g. SVM coefficient weights, ANN weights/gradient-based probing) provide misleading results if the scale of the input features differs. This can sometimes be addressed with normalization, but when both continuous and categorial features are used together, normalization may not be an option. Therefore, interpretation strategies that can handle diverse multi-omics data are needed.
- **Heterogeneous effects**: Given the importance of non-linear effects in biology (i.e. epistasis, feedback loops, synergistic/antagonistic effects), interpretation strategies that can identify features that have important but heterogeneous effects are critical. However, some interpretation strategies (e.g. partial dependency plots, LIME) aren't able to identify such features because positive and negative signals will average out. Therefore, interpretation strategies that identify features with heterogeneous effects are needed.
- **Multiple interpretations**: For some interpretation strategies (e.g. What-if analysis, DeepLIFT), there may be more than one explanation for why an instance was predicted a certain way, which one is best?

While these challenges can at times undermine efforts to understand the biology driving a model's predictions, many of them are shared with traditional statistical methods. In addition,

these challenges are also associated with data in other fields and beyond. These challenges also represent opportunities for computational biologists to develop novel solutions. There are also many tools that have been developed to facilitate interpreting ML models using many of the strategies described in this review (Supplemental Table 2.2).

2.8 Concluding Remarks and Future Perspective

Interpretability is critical for applications of ML both within and outside of biology and will therefore likely see substantial advances in the coming years. Given its broad use, future innovations in ML interpretability will likely to come out of fields working on diverse applications ranging from self-driving cars and smart city development to targeted advertising and natural language processing. Training the next generation of biologists to be able to harness these innovations to improve our ability to derive biological insights from what have been considered "Black Box" models represents a major research and training priority in the coming decade (see Outstanding Questions).

2.9 Outstanding Questions

- How will advances in deep learning (e.g. transfer learning and multi-label learning) impact the interpretability of these models?
- Interaction effects are a common phenomenon in biological systems whether we are thinking about community dynamics, epistasis, or environmental effects. What interpretation strategies are most appropriate for finding these, often complex, interactions?
- Deep learning algorithms often outperform less complex ML algorithms (i.e. RF and SVM) when enough training data is available, however can we learn as much from interpreting deep learning models as we can from these simpler models?

- There have been few efforts to compare the robustness of interpretation strategies using biological datasets. How should biologists benchmark interpretation strategies in order to find the methods most useful for addressing biological questions?
- Interpreting ML models can be relatively involved and require extensive computational skills. To what extent can existing and new tools for interpreting ML be made accessible to biologists without extensive programming skills?

2.10 Glossary

Algorithm: The procedure taken to solve a problem/to build a model

Decision tree: A model made up of a series of branching true/false questions.

Deep Learning: A subset of ML algorithms roughly inspired by the structure of the brain that can find complex, nonlinear patterns in data.

Ensemble: A combination of multiple models that makes one prediction for each unknown sample instead of multiple predictions.

Feature: An explanatory (i.e. independent) variable during modeling.

Global interpretation: A ML interpretation that gives an explanation of the overall relationship between features and the label.

Instance: A single example or object (n) from which the model will learn or be applied to.

Interpretable: Capable of being understood by a human.

Label: The dependent variable to be predicted. Either a continuous variable for regression models or a categorical variable (i.e. class) for classification models.

Local interpretation: A ML interpretation that gives an explanation of the relationship between features and the label for one or a subset of instances.

Machine learning: Computational models that are able to learn from data without being explicitly programmed.

Model: The set of patterns learned for a specific problem, where given an input (i.e. instances with features) the model will generate an output (i.e. prediction).

Parameters: Variables in an ML model whose values are estimated/optimized during training (e.g. connection weights, tree depth, coefficient weights).

Perturbing: A family of interpretable ML strategies that measure how changes in the input data impact model predictions or performance.

Probing: A family of interpretable ML strategies that involve inspecting the structure and parameters in a trained model.

Surrogate: A family of interpretable ML strategies that involve training an inherently interpretable surrogate model to represent a black-box model.

Test set: Subset of the data used to test the performance of a trained model

Training: The process of identifying the best parameters to make up a model – the learning part in ML.

REFERENCES

REFERENCES

Altmann, A., M. S. Schröter, V. I. Spoormaker, S. A. Kiem, D. Jordan *et al.*, 2016 Validation of non-REM sleep stage decoding from resting state fMRI using linear support vector machines. NeuroImage 125: 544–555.

Alyass, A., M. Turcotte, and D. Meyre, 2015 From big data analysis to personalized medicine for all: challenges and opportunities. BMC Medical Genomics 8: 33.

Angermueller, C., T. Pärnamaa, L. Parts, and O. Stegle, 2016 Deep learning for computational biology. Molecular Systems Biology 12: 878–16.

Banerjee, S., P. Bhattacharjee, and S. Das, 2017 Performance of Deep Learning Algorithms vs. Shallow Models, in Extreme Conditions - Some Empirical Studies, pp. 565–574 in *Pattern Recognition and Machine Intelligence*, edited by B. U. Shankar, K. Ghosh, D. P. Mandal, S. S. Ray, D. Zhang, et al. Lecture Notes in Computer Science, Springer International Publishing.

Barakat, N., and A. P. Bradley, 2010 Rule extraction from support vector machines: A review. Neurocomputing 74: 178–190.

Basu, S., K. Kumbier, J. B. Brown, and B. Yu, 2018 Iterative random forests to discover predictive and stable high-order interactions. Proceedings of the National Academy of Sciences 115: 1943–1948.

Biggio, B., and F. Roli, 2018 Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition 84: 317–331.

Breiman, L., 2001 Random Forests. Machine Learning 45: 5–32.

Che, D., C. Hockenbury, R. Marmelstein, and K. Rasheed, 2010 Classification of genomic islands using decision trees and their ensemble algorithms. BMC Genomics 11: S1.

Chicco, D., 2017 Ten quick tips for machine learning in computational biology. BioData Mining 10: 35.

Cuperlovic-Culf, M., 2018 Machine Learning Methods for Analysis of Metabolic Data and Metabolic Pathway Modeling. Metabolites 8: 4.

Dalvi, N., P. Domingos, Mausam, S. Sanghai, and D. Verma, 2004 Adversarial classification, pp. 99 in *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, ACM Press, Seattle, WA, USA.

Esteva, A., B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter *et al.*, 2017 Dermatologist-level classification of skin cancer with deep neural networks. Nature 542: 115–118.

Friedman, J. H., 2001 Greedy function approximation: A gradient boosting machine. Ann. Statist. 29: 1189–1232.

Garson, D. G., 1991 Interpreting neural network connection weights. AI Expert 6: 46–51.

Ghahramani, A., F. M. Watt, and N. M. Luscombe, 2018 Generative adversarial networks simulate gene expression and predict perturbations in single cells. bioRxiv 262501.

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin, 2015 Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. Journal of Computational and Graphical Statistics 24: 44–65.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley *et al.*, 2014 Generative Adversarial Nets, pp. 2672–2680 in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, MIT Press, Cambridge, MA, USA.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti *et al.*, 2018 A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys 51: 1–42.

Guo, Y., Y. Liu, A. Oerlemans, S. Lao, S. Wu *et al.*, 2016 Deep learning for visual understanding: A review. Neurocomputing 187: 27–48.

Jing, F., S. Zhang, Z. Cao, and S. Zhang, 2019 An integrative framework for combining sequence and epigenomic data to predict transcription factor binding sites using deep learning. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1–1.

Kelley, D. R., Y. A. Reshef, M. Bileschi, D. Belanger, C. Y. McLean *et al.*, 2018 Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. 28: 739–750.

LeCun, Y., Y. Bengio, and G. Hinton, 2015 Deep learning. Nature 521: 436–444.

Libbrecht, M. W., and W. S. Noble, 2015 Machine learning applications in genetics and genomics. Nature Publishing Group 16: 321–332.

Lipton, Z. C., 2018 The Mythos of Model Interpretability. ACM Queue 16:.

Liu, Z., and J. Yang, 2014 Quantifying ecological drivers of ecosystem productivity of the early-successional boreal Larix gmelinii forest. Ecosphere 5: art84.

Ma, C., H. H. Zhang, and X. Wang, 2014 Machine learning for Big Data analytics in plants. Trends in Plant Science 1–11.

Manzanarez-Ozuna, E., D.-L. Flores, E. Gutiérrez-López, D. Cervantes, and P. Juárez, 2018 Model based on GA and DNN for prediction of mRNA-Smad7 expression regulated by miRNAs in breast cancer. Theoretical Biology and Medical Modelling 15:.

Marx, V., 2013 Biology: The big challenges of big data. Nature 498: 255–260.

Molnar, C., 2019 *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar.

Nanayakkara, S., S. Fogarty, M. Tremeer, K. Ross, B. Richards *et al.*, 2018 Characterising risk of in-hospital mortality following cardiac arrest using machine learning: A retrospective international registry study. PLOS Medicine 15: e1002709.

Olden, J. D., and D. A. Jackson, 2002 Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling 154: 135–150.

Pinu, F. R., D. J. Beale, A. M. Paten, K. Kouremenos, S. Swarup *et al.*, 2019 Systems Biology and Multi-Omics Integration: Viewpoints from the Metabolomics Research Community. Metabolites 9:.

Rajaraman, S., K. Silamut, Md. A. Hossain, I. Ersoy, R. J. Maude *et al.*, 2018 Understanding the learned behavior of customized convolutional neural networks toward malaria parasite detection in thin blood smear images. Journal of Medical Imaging 5: 1.

Rasmussen, P. M., K. H. Madsen, T. E. Lund, and L. K. Hansen, 2011 Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. NeuroImage 55: 1120–1131.

Ribeiro, M. T., S. Singh, and C. Guestrin, 2016 "Why Should I Trust You?": Explaining the Predictions of Any Classifier, pp. 1135–1144 in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, ACM Press, San Francisco, California, USA.

Roadmap Epigenomics Consortium, A. Kundaje, W. Meuleman, J. Ernst, M. Bilenky *et al.*, 2015 Integrative analysis of 111 reference human epigenomes. Nature 518: 317–330.

Rokach, L., 2016 Decision forest: Twenty years of research. Information Fusion 27: 111–125.

Ronen, R., N. Udpa, E. Halperin, and V. Bafna, 2013 Learning Natural Selection from the Site Frequency Spectrum. Genetics 195: 181–193.

Samuel, A. L., 1959 Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development 3: 210–229.

Schmidhuber, J., 2015 Deep learning in neural networks: An overview. Neural Networks 61: 85–117.

Schrider, D. R., and A. D. Kern, 2018 Supervised Machine Learning for Population Genetics: A New Paradigm. Trends in Genetics 34: 301–312.

Shrikumar, A., P. Greenside, and A. Kundaje, 2017 Learning Important Features Through Propagating Activation Differences. Proceedings of the 34 th International Conference on Machine Learning.

Simonyan, K., A. Vedaldi, and A. Zisserman, 2013 Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. International Conference on Learning Representations.

Stephens, Z. D., S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai *et al.*, 2015 Big Data: Astronomical or Genomical? PLOS Biology 13: e1002195.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn, 2007 Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics 8:.

Tarca, A. L., V. J. Carey, X. Chen, R. Romero, and S. Drăghici, 2007 Machine Learning and Its Applications to Biology. PLOS Computational Biology 3: e116.

Uygun, S., A. E. Seddon, C. B. Azodi, and S.-H. Shiu, 2017 Predictive Models of Spatial Transcriptional Response to High Salinity. Plant physiology 174: 450–464.

Vervier, K., and J. J. Michaelson, 2018 TiSAn: estimating tissue-specific effects of coding and non-coding variants. Bioinformatics 34: 3061–3068.

Wachter, S., B. D. M. Mittelstadt, and C. Russell, 2018 Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harvard Journal of Law and Technology 31:.

Wang, X., D. Wang, Z. Yao, B. Xin, B. Wang *et al.*, 2019 Machine Learning Models for Multiparametric Glioma Grading With Quantitative Result Interpretations. Front Neurosci 12:.

Washburn, J. D., M. K. Mejia-Guerra, G. Ramstein, K. A. Kremling, R. Valluru *et al.*, 2019 Evolutionarily informed deep learning methods for predicting relative transcript abundance from DNA sequence. PNAS 116: 5542–5549.

Zhou, J., C. L. Theesfeld, K. Yao, K. M. Chen, A. K. Wong *et al.*, 2018 Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat Genet 50: 1171–1179.

Zuallaert, J., M. J. Kim, Y. Saeys, and W. De Neve, 2017 Interpretable convolutional neural networks for effective translation initiation site prediction, pp. 1233–1237 in *IEEE International Conference on Bioinformatics and Biomedicine-BIBM*, IEEE.

CHAPTER THREE: MODELING THE *CIS*-REGULATORY CODE OF PLANT SINGLE AND COMBINED STRESS TRANSCRIPTIONAL RESPONSE WITH MACHINE LEARNING

3.1 Abstract

When faced with adverse environmental conditions, plants mount dynamic and diverse responses at the transcriptional level. The set of regulatory components that control these responses is called the *cis*-regulatory code. Previous studies have characterized the *cis*-regulatory code regulating response to individual stress conditions (e.g. salinity), however, the way plants regulate their response to multiple simultaneous stresses is poorly understood. Here, we use classic machine learning and deep learning approaches to model the *cis*-regulatory code of response to single and combined heat and drought stress in *Arabidopsis thaliana.* First, we trained *cis*-regulatory code models using DNA sequence-based features from gene promoters (e.g. the presence/absence of a putative *cis*-regulatory elements), that were able to predict a gene's pattern of response better than random guessing. Then, we demonstrated how integrating additional levels of regulatory information (e.g. chromatin accessibility, histone modifications) and sequence-based features from outside the promoter region (e.g. downstream of the transcriptional stop site) improved the accuracy of our *cis*-regulatory codes. We found that features based on known transcription factor binding, histone 3 lysine 9 acetylation, chromatin accessibility, and downstream DNA sequence were the most useful additions to our models. We also found that while some of the most important putative *cis*-regulatory elements for our models resembled transcription factor (TFs) binding sites (TFBMs) associated with TFs known to be involved in heat and/or drought stress, others resembled TFBMs for TFs involved in developmental or other stress pathways. This study demonstrates how an *in silico*/data driven approach can be used to generate biological insights into important complex biological systems.

3.2 Introduction

In order to survive and thrive, plants dynamically coordinate their physiology and development with their environment. Given projected increases in global temperatures (Stocker *et al.* 2013) and the frequency and severity of droughts, heat waves, and flooding (Reynolds and Ortiz 2010; Sillmann *et al.* 2013), improving our understanding of how plants regulate these dynamic changes will be useful for future efforts to breed or engineer for more resilient crops (Rabara *et al.* 2014) and for our ability to understand how a changing climate will impact diverse plant species (Nicotra *et al.* 2010). Efforts to study how plants regulate their response to a single stress improve our understanding of how plants will regulate their response. However, multiple stressors are typically present, and the response to combined stress may be different than the response to either of the stresses individually. This was demonstrated at the transcriptional level, where ~60% of *Arabidopsis thaliana* genes were found to respond to combined stress conditions in ways that are not predictable based on their responses to the stresses individually (Rasmussen *et al.* 2013). While recent efforts have been made to identify transcriptomic (Atkinson *et al.* 2013; Sewelam *et al.* 2014; Bonnet *et al.* 2017), metabolomic (Prasch and Sonnewald 2013; Georgii *et al.* 2017), or physiological (Shaar-Moshe *et al.* 2017) changes in response to combined stress, how these changes are regulated remains unclear.

There are a number of important components involved in regulating a gene's response to an environmental stress. One major component is the binding of one or more transcription factors (TFs) nearby that gene. TFs are proteins that bind to DNA and activate/repress transcription of nearby genes. Their importance for regulating transcriptional response to stress has made them targets for breeding and engineering plants for improved response to stresses, including salt (Hu *et al.* 2008), drought (Choi *et al.* 2013; Lee *et al.* 2017), drought and heat (Wu

et al. 2009; Chang *et al.* 2017) stress. In fact, some of the genes involved in the domestication of crop species were TFs (Konishi *et al.* 2006; Doebley *et al.* 2006). One approach to better understand and find the TFs driving stress induced changes in gene expression is to identify the non-coding regions of DNA, or *cis*-regulatory elements (CREs), near the transcriptional start site of a gene where TFs bind. For some TFs in model species like *A. thaliana*, the DNA sequences that a TF can bind to (TF binding motif; TFBM) have been established *in vitro* (Weirauch *et al.* 2014; O'Malley *et al.* 2016). In addition, putative CREs (pCREs) can be found computationally using enrichment-based methods based on co-expression (Zou *et al.* 2011; Ghandi *et al.* 2014). Previous studies have demonstrated that both known TFBMs and pCREs can be used to generate models that are predictive of a gene's response to different environmental conditions (Zou *et al.* 2011; Uygun *et al.* 2017; Liu *et al.* 2018). These predictive models are referred to as the *cis*-regulatory code. While such studies highlight the importance of the presence of TFBMs and pCREs for understanding transcriptional regulation, factors besides the presence or absence of a CRE can influence TF binding and therefore transcriptional response to stress. For example, TF binding can also be affected by chromatin accessibility (Huebert *et al.* 2012; He *et al.* 2012; Arvey *et al.* 2012; Wang *et al.* 2012) and histone modifications (Steinfeld *et al.* 2007; Zhu *et al.* 2012). Therefore, methods to integrate these additional layers of omics information into the *cis*-regulatory code are needed.

Here we explore the *cis*-regulatory code of transcriptional response to single and combined heat and drought stress in *A. thaliana.* Heat and drought stress were selected because they often co-occur in nature, they elicit some similar and some conflicting physiological responses in plants (Rizhsky *et al.* 2004), and because many important TFs and TF binding motifs have already been identified for these stresses individually. At the physiological level, the

effects of combined drought and heat are generally additive (Vile *et al.* 2012). However, it is unclear to what degree these responses are additive, synergistic, or antagonistic at the level of transcriptional regulation. To better understand the regulatory logic underlying single and combined stress, first, we grouped genes likely to be co-regulated based on their shared pattern of transcriptional response under single and combined heat and drought stress (Prasch and Sonnewald 2013) (Step 1; Figure 3.1). Then, we used known TFBMs and enrichment based pCREs (Step 2; Figure 3.1) to generate models of the *cis*-regulatory code controlling these different patterns of responses to single and combined heat and drought stress using machine learning. To improve our *cis*-regulatory code models and therefore our understanding of how response to single and combined stress is regulated in *A. thaliana*, we modeled complex regulatory interactions (Step 3A; Figure 3.1), used a deep learning approach to integrate additional layers regulatory information (i.e. chromatin accessibility, sequence conservation, and histone marks) into our models (Step 3B; Figure 3.1), and expanded the scope of our models by including pCREs identified outside of the promoter region (Step 3C; Figure 3.1). In addition to providing a comprehensive overview of the *cis*-regulatory code of response to single and combined heat and drought stress in *A. thaliana,* this study also exemplifies how a data-driven approach can be used to make novel discoveries in a complex system like gene regulation (Step 4; Figure1).

3.3 Results and Discussion

3.3.1 More than 50% of genes have synergistic or antagonistic responses to combined heat and drought stress

In order to study the regulation of transcriptional response to single and combined stress, we first identified groups of genes that were likely to be co-regulated based on their shared

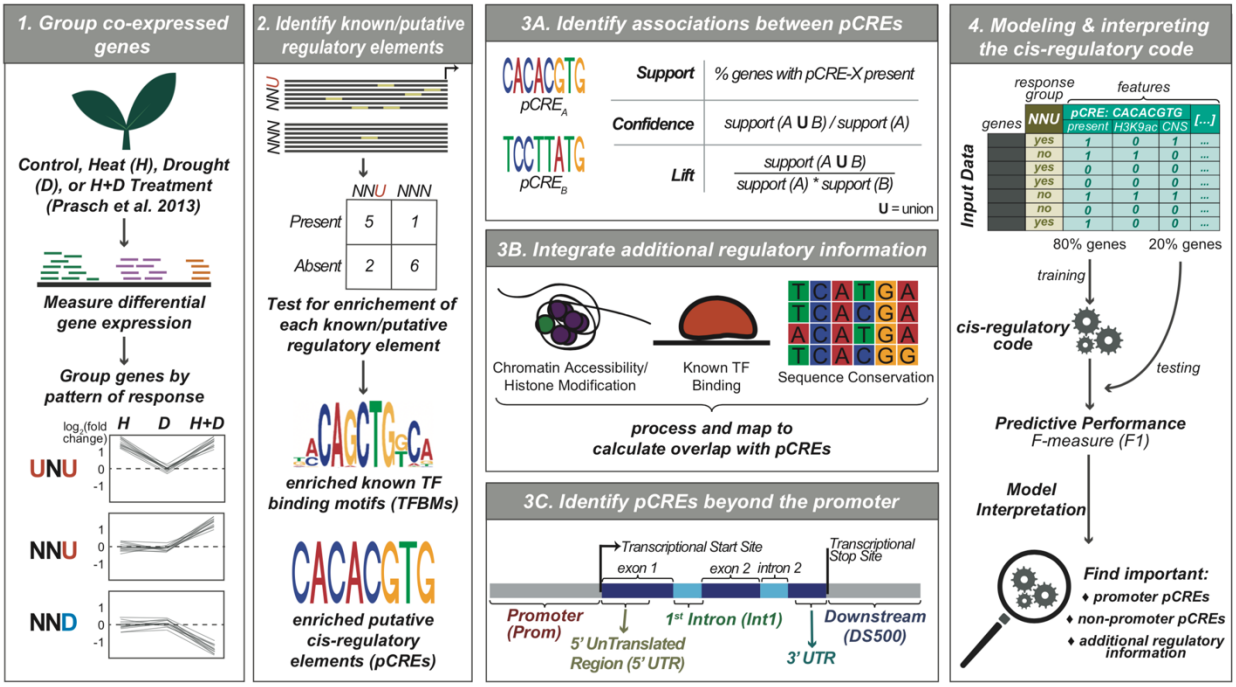


Figure 3.1. A framework for generating cis-regulatory code models.

Step 1: Genes were grouped based on their pattern of differential expression under heat (H), drought (D), and H+D stress compared to control conditions. Step 2: For each response group, known TFBMs and putative *cis*-regulatory elements (pCREs) were identified based on site enrichment among response group genes (Fisher's Exact Test; p-value < 0.01). Step 3: Information was gathered about associations between pCREs, their overlap with additional regulatory information, and pCREs located outside of the promoter regions. Step 4: All of this information was combined into machine learning models of the *cis*-regulatory code and the models were interpreted to identify the most important components driving the predictions.

pattern of transcriptional response (U: Up-regulated, N: Non-responsive, D: Down-regulated) to three stress conditions: heat, drought, and combined heat and drought stress using transcriptome data from an earlier study (Prasch and Sonnewald 2013). For example, genes that were up-regulated under heat and combined stress, but not under drought alone were placed in the UNU

response group. These response groups were further categorized based on if the response to the combined stress was similar to ("independent": UNU, NUU, DND, or NDD), less than ("antagonistic": UNN, NUN, DNN, or NDN), or greater than ("synergistic": NNU or NND) the sum of the responses to the single stress conditions (Figure 3.2A). Among genes that were responsive to at least one stress, 43%, 29%, and 24% genes were in the independent, antagonistic, and synergistic response groups, respectively (Figure 3.1B; Supplemental Table 3.1). The remaining 4% of genes belonged to rare response groups (e.g. DUN, UUD) and were not considered in our analysis. Most of the genes in the independent and antagonistic response categories were responsive (up or down-regulated) to heat, rather than drought stress. The dominance of the heat response could be due to: (1) the mild nature of the drought stress (Prasch and Sonnewald 2013), (2) an overriding influence of heat stress, as heat response also dominates over salt stress (Rasmussen *et al.* 2013), or (3) the fact that the expression data is derived from leaf where osmotic stress has a lesser effect compared to root (Shen *et al.* 2017).

To determine if genes in a response group shared distinct biological functions, we tested for the enrichment of genes with different Gene Ontology (GO) terms in each response group compared to non-stress responsive genes and genes in other response groups. Overall, we found that functional overlap existed between independent and synergistic, but not antagonistic, response group genes (Figure 3.2C, Supplemental Table 3.2). For example, both independent (UNU) and synergistic (NNU), but not antagonistic (UNN and NUN) response groups were enriched for heat and reactive oxygen species response. Similarly, for the down-regulation response groups, the independent (DND) and synergistic (NND) response groups were enriched for primarily photosynthesis related GO terms (Mathur *et al.* 2014), while the antagonist response group genes (DNN) were enriched for pollen development. This highlighted that genes

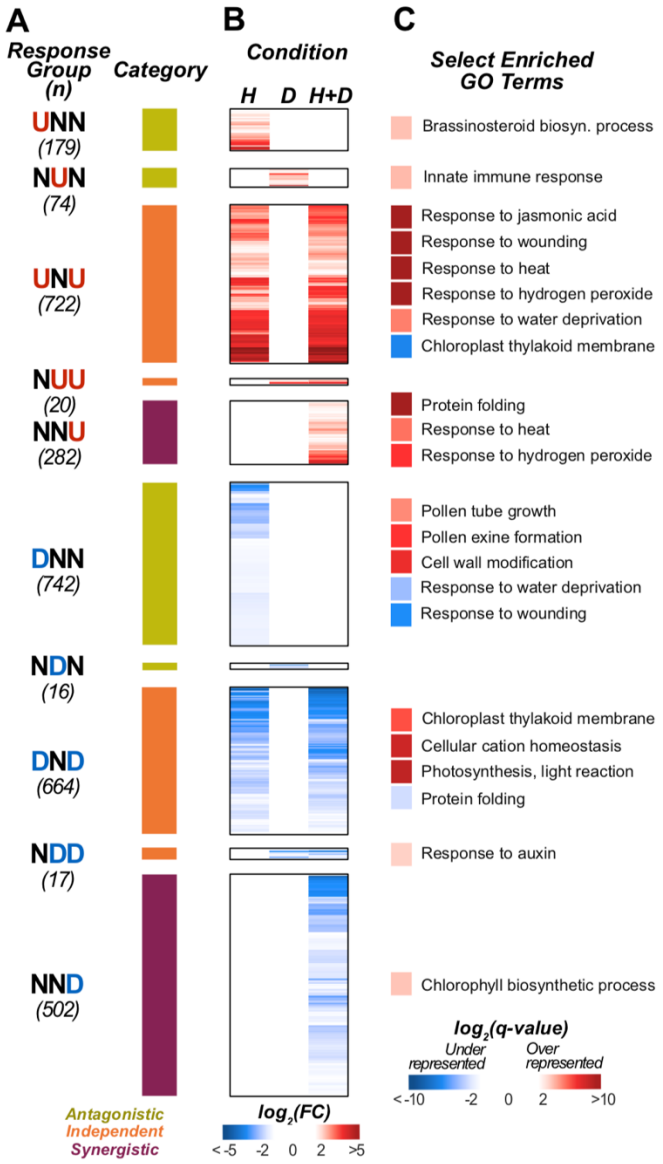


Figure 3.2. Gene expression response groups for single and combined heat and drought stress.

(A) Gene expression response groups included in the study where the three-letter response codes signify up-regulation (U), down-regulation (D), and no significant change in expression (N) ordered based on response to heat, drought, and both stresses. The number below the response group name is the number of genes in that response group that have non-overlapping promoters (1kb upstream of TSS) with neighboring genes. Colored bars designate if genes in the response

Figure 3.2 (cont'd)

group are considered to have antagonistic (yellow), independent (orange), or synergistic (purple) responses to combined stress. **(B)** The log2 Fold Change in expression under heat (H), drought (D), and H+D compared to control for each gene (X-axis), sorted by response group. If the absolute value of the Log2(FC) ≤ 1, colored white (N). **(C)** Select Gene Ontology (GO) categories that were enriched for genes belonging to the different response group compared to all other genes. GO categories with a large positive log2(*q*-value) (red) are over-represented, while those with large negative log2(*q*-value) (blue) are under-represented in that response group.

in the same functional category are not necessarily co-regulated and that antagonistic genes are not only differently regulated but also perform different biological functions.

In addition to having functional overlap, genes in the up-regulation independent and synergistic response groups were enriched for heat or water response functions (UNU & NNU, Figure 3.2C), while the antagonist response group genes were enriched for non-canonical abiotic stress response categories including brassinosteroid biosynthesis processes and innate immune response (UNN & NUN, Figure 3.2C). Brassinosteroids, for example, are most well known as cell-division and developmental regulators, but have also been implicated in heat tolerance in *Brassica juncea* (Kumar *et al.* 2010). Because an antagonistic response to combined stress means the response to the single stress was somehow counteracted, this suggests that the non-canonical functions are more tightly regulated than functions enriched in independent response group genes. This could be because the responses are detrimental in the presence of drought stress. For example, up-regulation of innate immune response (Huot et al. 2014) in NUN genes and down-

regulation of pollen development genes (De Storme and Geelen 2014) in DNN genes could be tightly regulated because aberrant responses could negatively impact fitness unnecessarily.

In summary, we found that ~55% of genes responsive to at least one stress showed either antagonistic or synergistic responses to combined heat and drought stress. Because these non-independent responses to combined stress were so prevalent, we hypothesized that a unique regulatory code must exist that is able to fine tune transcriptional response under combined heat and drought stress. We also found that genes in synergistic response groups overlapped functionally with genes in independent response groups, highlighting that genes with similar biological functions are not necessarily co-regulated.

3.3.2 Combinatorial stress response patterns can be predicted using known and putative regulatory elements

Because TFs and their binding sites regulating combinatorial stress response are yet to be identified, we set out to identify responsible TFs by taking advantage of available *in vitro* TF binding region and motif (known TFBMs) data from the DAP-seq (O'Malley *et al.* 2016) and CIS-BP (Weirauch *et al.* 2014) databases for 344 TFs. First, 197 of the 344 known TFBMs were identified as enriched in the promoter region of at least one set of response group genes (referred to as enriched TFBMs, see Methods). On average, response groups were enriched for 35 known TFBMs (range: 0-87) from 27 TF families (referred to as enriched families, Supplemental Table 3.1). In parallel, to identify regulatory sequences not covered by known TFBMs, we searched for putative *cis*-regulatory elements (pCREs) by identifying *k*-mers enriched in the promoter regions of genes in each response group compared to genes not responsive to stress (see Methods). Response groups were enriched for 68 pCREs on average (range: 7-158). These pCREs were similar to TFBMs from 22 of the 27 enriched families. This similarity was defined at two

different ways: (1) across all response groups, 13% of pCREs were significantly more similar to 36 of the 197 enriched known TFBMs compared to TFBMs from the same TF family (i.e. similar to TFBM) (Supplemental Table 3.3, see Methods), and (2) an additional 66% of pCREs were significantly more similar to an enriched TFBM compared to TFBMs from other families (i.e. similar to a TF family). The remaining 21% of pCREs were either most similar to TFBMs from TF families not significantly enriched in response group genes. Thus, the iterative *k*-mer finding approach based on co-expression recovered additional regulatory information not captured by the *in vitro* TFBM data.

To determine the extent to which known TFMBs and co-expression-based pCREs can explain combined stress response patterns (i.e. how much of the *cis*-regulatory code have we captured), we used the presence or absence of these TFBM and pCRE sites as features (i.e. independent variables) in machine learning models to classify genes as belonging to a response group or as non-responsive under any stress condition (i.e. the dependent variable). Because machine learning models need to learn from sufficient training data, we only used response groups with >20 genes. Model performance was measured by calculating the F-measure (F1) on a set of data held out from model training, where an F1=1 would be a perfect classification and an F1=0.5 would be no better than random guessing based on our approach (see Methods). Although all models performed better than random guessing (Figure 3.3A), models built using pCREs (median F1=0.64) significantly outperformed those built using known, enriched TFBMs (median F1=0.58) (paired t-test, p=3.7x10^{-4}). If we used all known TFBMs (i.e. both response group enriched and non-enriched), the model performance decreased further (median F1=0.54). These findings support the notion that pCREs contain additional regulatory information not captured by the TFBM data. This is not to say that pCREs can completely replace TFBM data

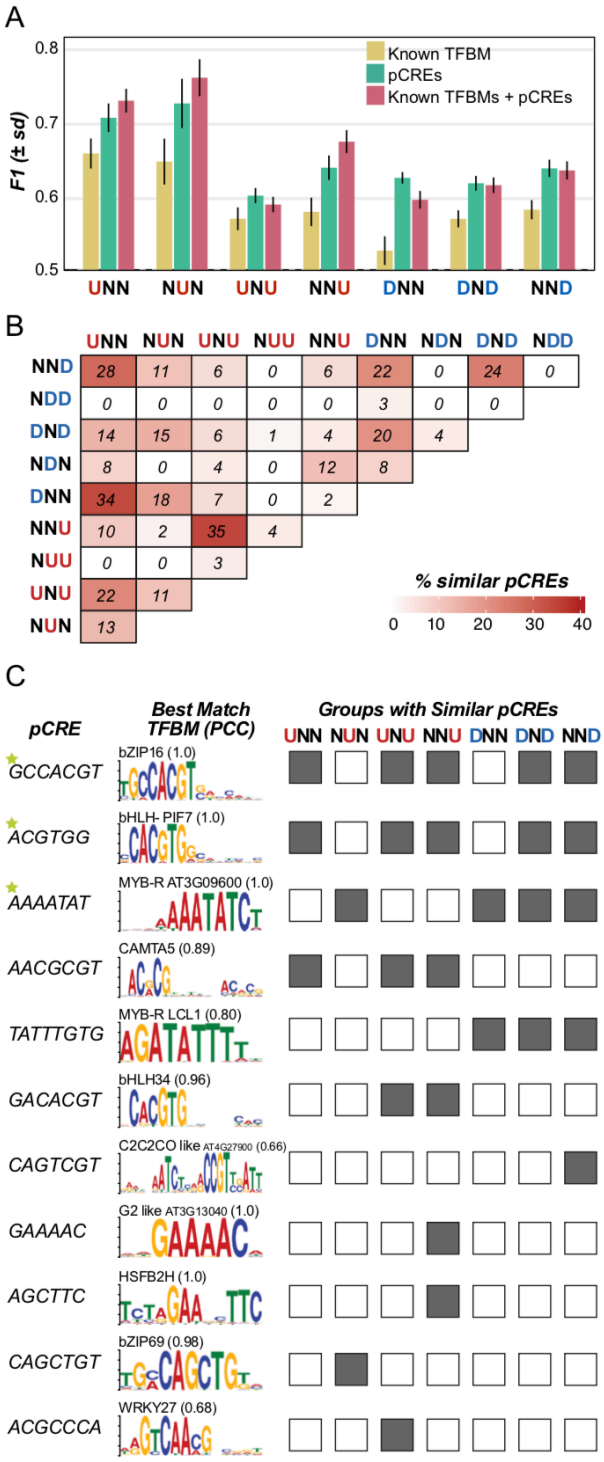


Figure 3.3. Cis-regulatory code models based on known TFBMs and pCREs.

(**A**) Predictive performance (F1) of Random Forest machine learning models using known TFBMs (yellow), pCREs (teal), or both (rose) as input features as input for predicting response

Figure 3.3 (cont'd)

group vs. non-responsive genes. **(B)** Percent of pCREs identified for each response group that are significantly similar to pCREs in other response groups based on a threshold that is the 95th percentile of the PCC distribution between 100 sets of 25 randomly generated *6*-mers (numbers in cells). Darker red indicates that the pCREs identified independently for each of the two response groups shared higher sequence similarity. (**C**) Select example pCREs (first column) and a motif logo of the known TFBM the pCRE is most similar to. The similarity scores (Pearson's Correlation Coefficient; PCC) between each pCRE and its best match are shown. The pCREs are sorted from most to least commonly enriched across response groups, where the boxes indicate what response groups the pCRE was enriched (gray) or not enriched (white) in.

because models built using the enriched TFBMs and pCREs were able to correctly classify different subsets of genes (Supplemental Figure 3.1). However, including both types of elements as features did not improve model performance compared to only using pCREs (median F1=0.64; paired t-test, p=0.51). Across the response groups, the combined models also classified genes more similarly to the pCRE-based than the enriched TFBM-based models (Supplemental Figure 3.1).

Next, we quantified the degree of overlap between pCREs identified for different response groups to assess how the *cis*-regulatory programs differ between different response patterns to single and combined stress. Two pCREs were considered overlapping if they shared a greater sequence similarity with each other than with 95% of random *6*-mers (Figure 3.3B). Using this approach, the pCRE overlap ranged from 0 to 35% between response groups, with response groups that share the same direction of response (i.e. NNU and UNU) tend to have

higher degree of overlap (R^2=0.41, p<1x10^{-4}). Interestingly, of the pCREs overlapping among the most response groups, the top three, GCCACGT, ACGTGG, and AAAATAT (stars, Figure 3.3C) were significantly similar to TFBMs associated with circadian clock TFs bZIP16, PIF7, and RVE8, respectively (Hsieh *et al.* 2012; James *et al.* 2012). PIF7 has been shown to negatively regulate *DREB1* as a means to avoid hindering plant growth by the accumulation of DREB1 when the plant is not under stress (Kidokoro *et al.* 2009). Our findings further confirm earlier studies that stress response regulation has a significant circadian clock component (Liu *et al.* 2013). Nonetheless, 65 to 100% of pCREs differed between any two response groups (Figure 3.3B). This supported the notion that there are substantially distinct regulatory mechanisms involved in different patterns of response to combined stress.

In summary, the iterative *k*-mer finding approach identified pCREs that, when used as predictive features, were better able to classify genes by their response groups than known enriched TFBMs. Over 20% of pCREs were not similar to known enriched TFBMs, indicating pCREs contain novel regulatory information. In addition, the majority of pCREs did not show significant sequence similarity with pCREs from other response groups, suggesting substantial regulatory differences. Finally, while we were able to classify genes by their response group well above random expectation, with a median F1=0.64 there was still ample room for model improvement. Thus, we next explored three strategies to improve predictions of response to single and combined stress by: (1) considering interactions between pCREs, (2) integrating dynamic multi-omics data, and (3) including pCREs located outside the proximal promoter. Because TFs frequently work in concert to regulate gene expression (Harbison *et al.* 2004; Farnham 2009), we first incorporated interactions between TFs into our models by identifying interactions between pCREs. We identified interactions between pCREs for each response group

using two statistical approaches: association Rule (aRules) and iterative Random Forest (iRF). However, pCRE pairs identified did not improve model performance when used as features alone or with pCREs (Supplemental Figure 3.2 and Supplemental Info), unlike in high salinity stress (Uygun *et al.* 2017).

3.3.3 Additional multi-omics regulatory information can improve cis-regulatory code models

To account for additional levels of regulation involved in response to single and combined heat and drought stress, we next explored adding additional information to our *cis*-regulatory code models. We included information about chromatin accessibility (DNase I Hypersensitive Sites: DHS) (Sullivan *et al.* 2015; Liu *et al.* 2018) and eight histone marks (Pfluger and Wagner 2007; Dong and Weng 2013; Stroud *et al.* 2014) because both can impact the ability of a TF to bind. In addition, because regulatory elements can experience selective pressure, information about sequence conservation across the *Brassicaceae* family (Conserved Noncoding-Sequences: CNS) (Haudry *et al.* 2013) was included because it could be informative for identifying pCREs (Guo et al.2003; Haberer et al 2006). Finally, as described above, *in vitro* TF binding regions have been identified in *A. thaliana* (O'Malley *et al.* 2016), we included these data as they may also improve our ability to identify pCREs. These data are collectively referred to as "additional regulatory information".

To determine if this additional regulatory information would improve our understanding of the *cis*-regulatory code of combined stress response patterns, we next tested if the addition of these data into our machine learning models would improve their performance. While models utilizing this additional regulatory information improved the average performance for a few response groups (i.e. NNU, DNN), overall, they did not perform significantly better than pCRE-

only models (median F1=0.66; paired t-test, p=0.062) (olive; Figure 3.4A). One possible reason for this lack of improvement could be that our machine learning algorithm (i.e. Random Forest) was not adequately integrating the additional regulatory information into the models. For example, Random Forest treats all input features, such as pCREs and histone marks as independent when they may not. To address this limitation, we applied a deep learning approach, convolutional neural network (CNN). CNNs are frequently used in image classification because when given training data (e.g. many photographs of cats) they are able to learn local patterns (e.g. triangles that resemble cat ears) and associate those patterns with what is being predicted (e.g. is there a cat in the photograph). We hypothesized we could train CNN models to look for patterns in the additional regulatory information available for each pCRE and to then associate those patterns with a response group (Figure 3.4B; see Methods). Using this approach, our ability to predict response groups increased (median F1=0.68) compared to the pCRE only models (paired t-test, p=0.002), with the largest improvements for the UNU, DNN, DND, and NNU response groups (where F1 increased by 0.069, 0.055, 0.050, and 0.046 respectively) (rose; Figure 3.4A).

3.3.4 Interpreting deep learning models provides insight into the cis-regulatory code

To understand what combinations of additional regulatory information were important for the ability of our CNN models to classify genes by their response group, we visualized and measured the importance of the trained kernels. During the process of model training, each kernel learns a particular "pattern", i.e., how much value, or weight, should be given to each type of feature (i.e. presence/absence and additional regulatory information) to best predict if a gene belongs to a response group. For example, in Figure 3.4B, kernel #1 (k_1) learned to look for pCREs that were present and that overlapped with a DAP site and with histone marks for H1 and

H7 (positive kernel weights), but not H4 or H6 (negative kernel weights) (named for illustrative purposes only). Then, each trained kernel scans across the input data and generates an output value for each pCRE based on how well it matches the pattern. For example, when k_1 was used to scan from pCRE-A down to pCRE-X, it output a large (i.e. dark) value for pCREs that match its pattern (e.g. pCRE-A) and a small value for pCREs that do not match its pattern (e.g. pCRE-D). To assess which types of features were most important (i.e. highest weighted) among kernels from CNN models for each response group, we extracted the trained kernels (i.e. a list of 12 weights) for each kernel in each replicate, clustered them into groups with similar patterns of weights, and calculated the median weight assigned to pCRE presence/absence and each additional regulatory information for each cluster (Figure 3.4C, S4; see Methods).

To measure the overall importance of each kernel, we calculated the change in model performance on the test data (i.e. data not used for training) when each kernel was zeroed out (i.e. all weights set to zero; see Methods). We then reported the median kernel importance for each kernel cluster (Figure 3.4C, S4). For example, when a kernel in the first kernel cluster for DNN was set to zero, model performance (measured using the area under the receiver operator characteristic; see Methods) dropped by > 0 .005. Note that the performance decreases are all very small, indicating the models were robust to perturbation likely because more than one kernel trained to learn important patterns. Overall, the presence or absence of the pCREs (P/A) had the highest median weights (leftmost column; Figure 3.4C). Of the additional regulatory information, DAP, H3K9ac, and DHS had the next highest kernel weights, suggesting known TF binding, the acetylation of lysine 9 on histone H3 (a hallmark of active promoters (Karmodiya *et al.* 2012)), and chromatin accessibility were consistently useful feature for predicting response to single and combined stress. Additional regulatory information were weighted differently in

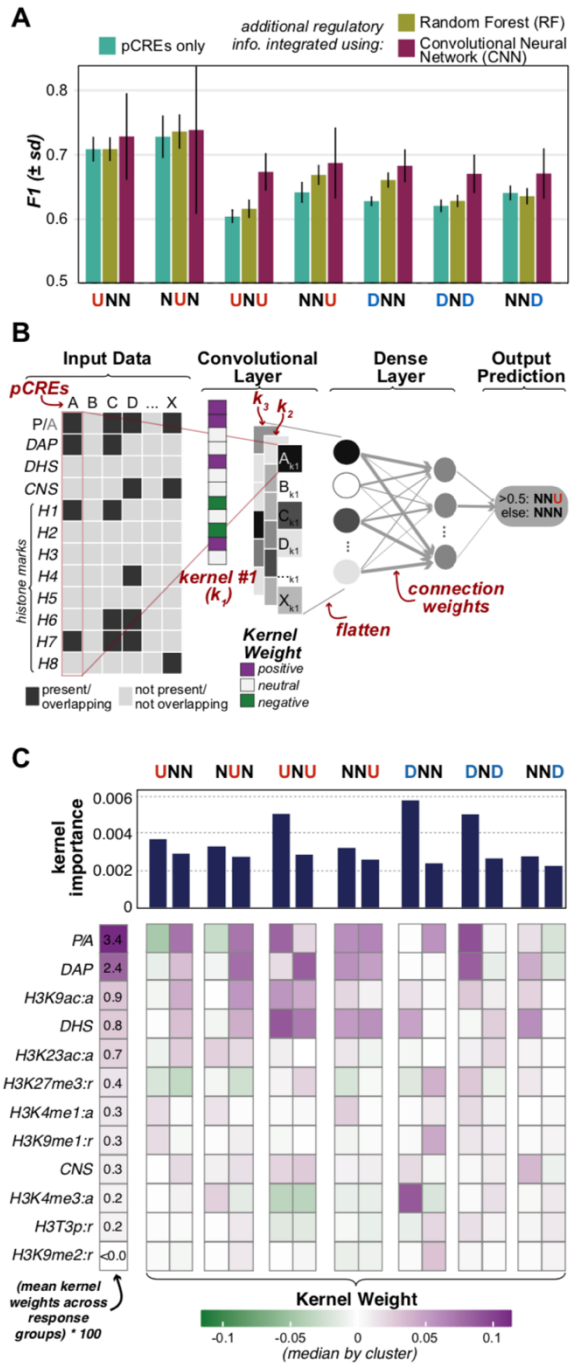


Figure 3.4. Cis-regulatory code models based on pCREs and additional multi-omics regulatory information.

(**A**) Predictive performance (F-measures (F1)) of Random Forest models using pCREs (teal, same as in Figure 3.3A) and pCREs + additional regulatory information (olive) and of

Figure 3.4 (cont'd)

Convolutional Neural Network (CNN) models using both pCREs + additional regulatory information (rose). The larger error around NUN models is due to the small number of NUN genes available for model training. **(B)** An illustration of the internal workings of the CNN models and how the trained kernels (i.e. pattern identifiers) in those models were used to understand the patterns of additional regulatory information the models were trained to identify. **(C)** Summary of results from interpreting the trained CNN models. The feature types (i.e. presence/absence (P/A) and additional regulatory information) were sorted based on the average kernel weights across all kernels trained for all response groups and replicates (first column). The remaining columns represent kernel clusters for specific response groups. For each response group, all trained kernels from all CNN replicates were clustered using hierarchical clustering with dynamic cutting (min cluster size=250 kernels). The median kernel weights and kernel importance scores are shown here for the two clusters with the highest median kernel importance for each response group. Large kernel weights (dark purple) indicate the presence of that pCRE or its overlap with the additional regulatory information was an indicator of belonging to the response group rather than NNN.

important kernel clusters for different response groups (second column and on; Figure 3.4C). This was especially true of histone mark features. For example, H3K27me3 tended to be negatively weighted in important kernel clusters for up-regulation response groups (e.g. UNN, NUN, NNU) but neutral or positively weighted in important kernel clusters for down-regulated response groups (e.g. DNN, DND). Together with the fact that H3K27me3 is known to be associated with gene silencing (Luo and Lam 2010), this finding supports the idea that lysine 27

trimethylation is involved with regulating response to single and combined heat and drought stress. However, we also found that H3K4me3 had a large positive weight for the most important DNN kernel cluster. This was unexpected given that H3K4me3 is associated with active promoters (Luo and Lam 2010) and suggests that the role of lysine 4 trimethylation in regulating single and combined heat and drought stress response may be complicated.

In summary, we found that the integration of additional multi-omics regulatory information into our models of the *cis*-regulatory code using CNNs improved our ability to classify genes by their pattern of response to single and combined stress. While some information (e.g. TF binding, H3K9ac) was important for all response groups, other information (e.g. H3K4me3, H3K27me3) was differentially important across the response groups. The usefulness of these data was especially surprising given some of the limitations of the data. For example, most of the data were generated either *in vitro* (e.g. DAP) or under growth conditions that do not match the transcriptome data used for this study (e.g. DHS).

3.3.5 pCREs identified outside the promoter region are predictive of response patterns

The models discussed thus far were based on features located in the proximal promoter regions typically housing regulatory sequences in plants (Yu *et al.* 2016). However, plant regulatory sequences can also be located in the 5' untranslated region (5' UTR) (Tompa 2001), first intron (Int1) (Zhang and Duff 1994), 3' UTR (Wasserman *et al.* 2000), and downstream of the transcriptional stop site (DS500). To assess the extent to which pCREs outside of the promoter regions were predictive of combined stress response patterns, the iterative *k*-mer finding approach was repeated in the 5' UTR, Int1, 3' UTR, and DS500. Then, predictive models were built using either pCREs from each region individually or in combination as features. Because sequence information was not available for all five regions for all genes (particularly 5'

and 3' UTRs), we had to remove between 47 and 587 genes from each response group to make our models comparable. Importantly, this means that the performance results from our earlier machine learning models would not be directly comparable. In order to establish a direct comparison, we also reran the iterative *k*-mer finding and modeling on the promoter region using the smaller subsets of genes.

Models built using pCREs located in promoter or, surprisingly, DS500 regions outperformed models built with pCREs from other regions (*Tukey test*; Figure 3.5A). DS500 pCREs substantially outperformed promoter pCREs for the NUN response group in terms of F1 (+0.06, Figure 3.5A), as it correctly classified 2 more genes and reduced the false positives by 14 (Supplemental Figure 3.5). Interestingly, the most predictive DS500 pCRE, ACTTTG, shares significant sequence similarity (PCC=0.92) with the known TFBM for WRKY46, which has known roles in drought response. This pCRE was not enriched in the promoter region, emphasizing the potential importance of the DS500 region for *cis*-regulation. Although the 5'UTR and 3'UTR pCREs did not perform as well as those in promoters and DS500s, they were significantly better than random expectation (t-test: p=0.02, 0.006, respectively), however Int1 pCREs were not significantly different than random (p=0.75). Because models built using pCREs from different regions were able to correctly classify different subsets of genes (Supplemental Figure 3.5), we used pCREs from all regions as features and the resulting models (the ALL column, Figure 3.5A) outperformed all single region-based models, suggesting that pCREs located beyond the promoter region are important for regulating combined stress response.

To determine if the pCREs identified from different genetic regions were unique to that region or found across regions, we identified the best matches between pCREs within and

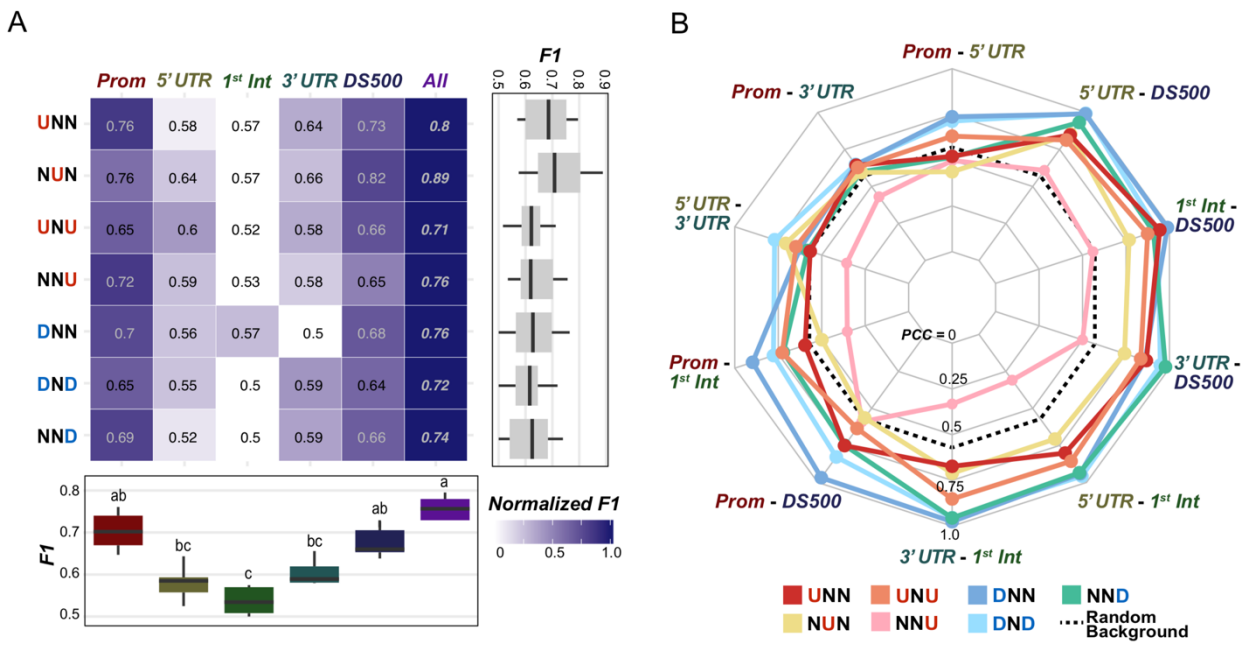


Figure 3.5. Cis-regulatory code models based on pCREs identified in putative promoter and non-promoter regions.

(**A**) Predictive performance (F1) from Random Forest models using pCREs found in the promoter, 5' UTR, first intron (1st Int), 3' UTR, downstream region (DS500), or all regions (All) as input features. The box color represents the F1 scores normalized by the F1s of each response group (the darkest blue represents the best set of input feature for each response group) with the actual F1 provided in each box. The boxplot shows the distribution of F1 scores for each region (below) and for each response group (right). Letters on the top of boxplots signify significant differences by region based on the Tukey test ($p<0.05$). (**B**) Average sequence similarity (PCC) between pCREs identified from different regions (axes) for each response group (colors). The dashed line indicates the random average similarity obtained by calculating PCC for every possible pairing of 100 sets of 25 randomly selected *6*-mers.

between different regions based on sequence similarity (Pearson's correlation coefficient, PCC, see Methods). The pCREs from most regions were more similar to each other (average PCC=0.73) than would be expected by random chance (dotted line, PCC_{95th}=0.57; Figure 3.5B). The DS500 pCREs tended to be the most similar to pCREs from other regions, especially those from the 5'UTR, 1st intron, and 3' UTR. Interestingly, the promoter pCREs tended to be the least similar to pCREs from other regions. That said, the pCREs most similar to the promoter pCREs were enriched in the DS500 (average PCC=0.73), indicating similar *cis*-regulatory mechanisms governing transcriptional regulation up and downstream of genes. In addition, pCREs from down-regulation response groups (blue color series, Figure 3.5B) tended to be more similar between regions than up-regulation response group pCREs (red color series, Figure 3.5B). This suggests that regulatory elements involved in down regulating genes are either less region specific or are more likely to be located in multiple regions around the gene. This is in sharp contrast to NNU pCREs, which were the only pCREs that were less likely to be similar (average PCC = 0.45) than random chance, suggesting the regulatory circuitry for synergistic up-regulation is specific to the promoter region.

In summary, incorporating pCREs identified outside of the proximal promoter region improved our ability to predict response to single and combined heat and drought stress. Of the five regions assessed, the DS500 pCREs performed marginally better than promoter pCREs for two of the seven response groups. Taken together, this suggests that while most of the pertinent regulatory information is in the promoter regions, additional regulatory information important for response to single and combined heat and drought stress may be located outside the promoter region.

3.3.6 The cis-regulatory code of response to single and combined heat and drought stress

We have demonstrated that adding multi-omics data and expanding our search for putative regulatory elements beyond the promoter region has improved our *cis*-regulatory code models. While these models are still not perfect, they perform well above random expectation and therefore can be used to illuminate the *cis*-regulatory code of response to single and combined heat and drought stress in *A. thaliana.* To this end, here we further characterize a subset of the most important promoter (from CNN models) and non-promoter (from Random Forest models) located pCREs identified for each of the seven response groups. The most important promoter located pCREs from the CNN models were those that, when set to zero, caused the largest decrease in model performance (see Methods). The most important pCREs from the Random Forest models are those that, when used at a node in a decision tree, were able to best separate genes by their response group (see Methods). The importance scores of pCREs based on these two approaches are in Supplemental Table 3.5, S6.

To characterize the most important promoter pCREs using the additional levels of regulatory information included in the study, we determined how much more frequently the sites of each promoter pCREs overlapped with each of the additional regulatory information in response group genes than random expected using a set of 1,000 random *6*-mers (Supplemental Table 3.5). Focusing on the top five most important pCREs from each response group, we found that these pCREs could be classified into three groups based on their degrees of overlap between their sites and the additional regulatory information (Figure 3.6A). Group 1 pCREs were unique in that, in addition to DAP and DHS, they were also much more likely to overlap with CNS than

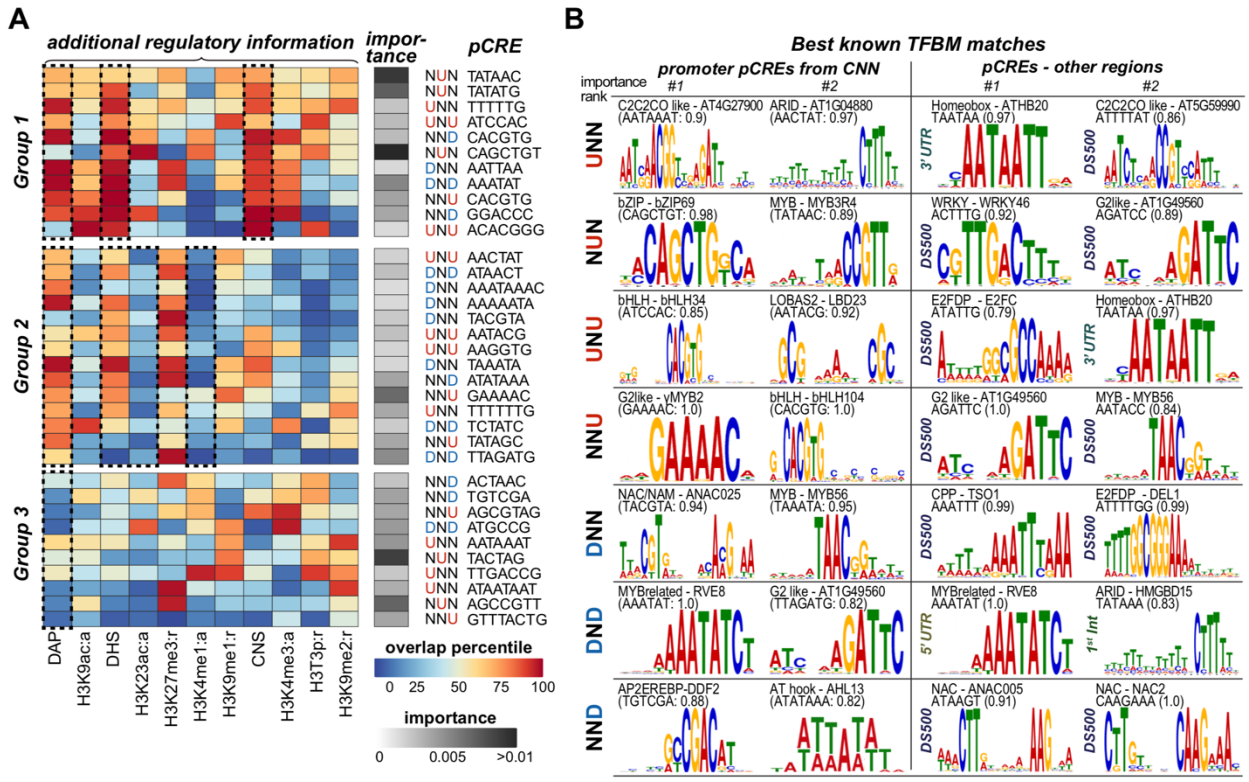


Figure 3.6. Overview of the most important pCREs for our cis-regulatory code models. (**A**) The top five most important promoter pCREs from CNN models clustered using k-means clustering (k=3) into three groups based on the pattern of overlap between their sites with additional regulatory information and sorted using hierarchical clustering. The overlap percentile refers to how frequently a pCRE overlaps with each additional regulatory information in the promoter of response group genes compared to 1000 random *6*-mers, with values in darker red signifying higher degrees of overlap compared to the random background. The importance score is the median decrease in model performance on the test set when a pCRE and its associated additional regulatory information is removed from the CNN model (i.e. larger decrease in performance means a larger importance). **(B)** The TF name, motif logo, and sequence similarity score (Pearson's Correlation Coefficient; PCC) for the known TFBMs that best match the top two promoter pCREs from the CNN model (left two columns) and the top two non-promoter

Figure 3.6 (cont'd)

pCREs from the RF model using pCREs from all five gene regions (see purple in **Figure 3.5A**) (right two columns) for each response group.

random *6*-mers (dashed boxes; Figure 3.6A), suggesting these pCREs are more highly conserved across the *Brassicaceae.* Group 2 pCREs also frequently overlapped with DAP and DHS regions, although to a lesser extent. Group 2 pCREs were also less likely to overlap histone marks associated with active transcription (e.g. H3K23ac, H3K4me1), which was interesting given that the important pCREs identified for the down-regulation response groups (i.e. DNN, DND, NND) tended to be in Group 2 (8 vs. 4 and 3 in other groups 1 and 2). Finally, Group 3 pCREs were less likely to overlap with DAP regions than random *6*-mers, suggesting these pCREs may be bound by TFs not yet included in *in vitro* binding databases.

We next characterized promoter and non-promoter pCREs by determining which were similar to known CREs and which represented putative novel CREs by (see Methods). Overall, 40.5% of promoter pCREs and 37.6% of pCREs from other regions were significantly similar to a specific known TFBM (i.e. sequence similarity (PCC) was > 95th percentile of PCCs between TFs in the same family) (Supplemental Table 3.5, S6). Focusing on the two most important promoter and non-promoter pCREs for each response group (Figure 3.6B) we found many different TFs and TF families represented. The promoter and non-promoter located pCRE for the DND models, AAATAT, is identical to the TFBM of a MYB related TF, *REVEILLE8* (*RVE8*) (Figure 3.6B), which has been proposed to be involved in a negative feedback loop regulating the circadian clock's response to temperature (James *et al.* 2012). The most important non-promoter pCRE for the NUN model, ACTTTG, is similar to TFBMs in the WRKY TF family

(PCC to *WRKY46* = 0.92), which are known to be involved in osmotic and salt stress response (Ding *et al.* 2015). The most important promoter pCRE for the NND models, TGTCGA, is similar to TFBMs in the AP2 TF family (PCC to *DDF2* = 0.88), which are known to be involved in heat, cold, and drought tolerance in *A. thaliana* (Kang *et al.* 2011). Taken together, these three examples highlight that by modeling and interpreting the *cis*-regulatory code we able to find pCREs similar to known TFBMs for TFs known to be involved in heat, drought, and combined heat and drought stress, respectively.

Interestingly, the most important pCREs for the NNU response group are not similar to TFBMs for TFs known to be involved in either heat or drought stress. For example, the most important promoter pCRE, GAAAAC is identical to the TFBM for the G2-like *γMYB2* TF, which has no know association with stress response. The second most important promoter pCRE, CACGTG is identical to the TFBM for *bHLH104*, which while known to be involved in regulating iron homeostasis in *A. thaliana* (Li *et al.* 2016), is not associated with other stresses. Similarly, the most important non-promoter pCRE for NNU, AGATTC, is identical to the TFBM AT1G49560), a G2-like family TF possibly involved in regulating flowering time. This highlights how much more work needs to be done to understand the regulation of combined heat and drought stress, with these pCREs and their associated TFs representing prime candidates for further characterization.

In summary, we found that important promoter pCREs belongs to three groups that differed in how frequently the pCREs were associated with additional regulatory information. We also found that while some of the most important pCREs found by our *cis*-regulatory code models were similar to known TFBMs bound by TFs involved in heat and/or drought stress response, others (i.e. those enriched in NNU genes) were similar to TFs with no established

association to either stress condition. Taken together, these findings highlight the complexity of the *cis*-regulatory code of response to single and combined heat and drought stress in *A. thaliana* and the need for further study.

3.4 Conclusions

Understanding how plants regulate their response to combined heat and drought stress is of great importance because of the frequency with which these stresses co-occur and the severity of their impact on our agricultural sector when they do (Rizhsky *et al.* 2004). Here we develop models of the *cis*-regulatory code regulating response to single and combined heat and drought stress in *A. thaliana.* We assessed the strength of our models by determining how well they classify genes not used for training as belonging to a response group (e.g. NNU, NND) or to the non-responsive group (i.e. NNN). We found that incorporating pCREs identified outside of the proximal promoter region and additional multi-omics regulatory information (i.e. chromatin accessibility, sequence conservation, known TF binding, and histone markers) into our models improved their performance. We also explored the use of a deep learning approach, CNN, and demonstrated that it performed better than the classical machine learning algorithm used in this study, Random Forest. Furthermore, by interpreting our *cis*-regulatory code models, we were able to provide novel biological insights, including identifying which pCREs and additional regulatory information were most important for being able to predict response to single and combined heat and drought stress.

Because our *cis*-regulatory code models are not able to perfectly predict a gene's response group, there is still more to learn about the complexities of the regulation of response to single and combined stress. One factor that is limiting our ability to model the *cis*-regulatory code is that genes in a response group are not all regulated by the same mechanisms. This issue

is compounded by the fact that samples were gathered only at a single time point a few days after the stress conditions were applied. Thus, we have only a snapshot and do not have information on whether the stress responsive genes began to respond immediately after stress initiation or later after the plants began to acclimate. A second limiting factor is that we are missing critical information about the rate of mRNA degradation. Because our picture of differential gene expression comes from measuring and comparing the steady state mRNA levels, we cannot determine if the change in gene expression is due to, for example, increase in production or a decrease in degradation. Finally, while incorporating chromatin accessibility and epigenetic mark data into our models of the *cis*-regulatory code improved their performance, these data were not ideally suited for this study because they were generated from plant at different developmental stages under different conditions than the plants used to generate the transcriptomic data used in this study (Prasch and Sonnewald 2013). We consider this a third limitation because both chromatin accessibility and epigenetic marks are dynamic, meaning they change over the course of development and in response to environmental conditions (Sullivan *et al.* 2014; King 2015).

There are numerous mechanistic possibilities for how a gene can regulate its response to combined stress. For example, a gene with a synergistic response could require the binding of two TFs in order to be expressed, each of which is only up-regulated or activated by one of the individual stresses. Alternatively, synergistic responses could be regulated by novel TFs that only bind or are only activated under the combined stress scenario. In our study we found that some pCREs were unique to individual response groups. For example, AGCTTC, which perfectly matches the TFBM identified for HSFB2H, was only enriched in genes with a synergistic up-regulation response to combined stress (NNU). However, other pCREs, especially those containing G-Box motifs, were found for multiple response groups, with pCREs from

independent and synergistic response groups having the most overlap. By studying the pCREs identified for different response patterns we can begin to understand at a global level how response to combined stress is regulated.

As high-throughput omics technologies continue to become more affordable and widely used, techniques to integrate across multiple types of omics data will become increasingly necessary. It is also critical that these techniques are interpretable so that we are able to derive from them insights into complex biological systems such as the regulation of gene expression. Here we trained classic and deep machine learning models of the *cis*-regulatory code regulating response to single and combined heat and drought stress. We then used *in silico* model interpretation strategies and were able to identify known actors in response to heat and/or drought stress in addition to putative novel actors that are prime targets for further characterization. In the future, this approach could be used to study the regulation of other developmental and stress induced responses in plants and other organisms.

3.5 Methods

3.5.1 Expression data processing, response group classification, and functional category enrichment analysis

Expression data for response to mild heat (32ºC day/28ºC night for 3 days), mild drought (30% field capacity), and combined heat and drought stress in *A. thaliana* were downloaded from NCBI Gene Expression Omnibus (GEO) (GSE46760) as normalized signal intensity values (Prasch and Sonnewald 2013). The expression data was generated using the Agilent platform and probe data was converted into TAIR10 gene identifiers using IDswop from the “agilp” package in the R environment (Chain 2012). If multiple probes were present for the same gene the mean of the probe intensities was used, unless the intensities were >20% different, in which case the

gene was excluded. Differential expression folds and associated false discovery rate (FDR) adjusted *p*-values (i.e. *q*-values) (Benjamini and Hochberg 1995) between each stress conditions and the control condition were calculated using limma (Ritchie *et al.* 2015) in the R environment.

Genes were classified as significantly up-regulated (U) if their log2 fold-change ≥ 1.0 with $q \leq 0.05$, down-regulated (D) if their log2 fold-change ≤ -1.0 with $q \leq 0.05$, or non-responsive (N) otherwise. Genes were clustered into "response groups" using a convention established by Rasmussen *et al.* (2013). Briefly, each gene was defined by its pattern of U, D, or N under heat, drought, and combined stress conditions. For example, a gene that is U under heat, D under drought, and N under combined stress was classified as in the UDN response group. To more clearly distinguish between genes belonging to a response group from genes that were considered non-stress responsive (NNN), genes were only considered NNN if they were not significantly differentially expressed (up- or down-regulated) with a log2 fold change cutoff of 0.8 under any of the three stress conditions or under any stress condition at any time point in the AtGenExpress database (http://www.weigelworld.org/resources/microarray/AtGenExpress/). P

Sequence data for the promoter, 5' UTR, 3' UTR, first intron, and downstream region for *A. thaliana* genes were downloaded from TAIR10. Genes whose promoter regions (1-kb upstream the transcriptional start site) overlapped with neighboring genes were excluded from the analysis. We tested if genes oriented in the same direction as their upstream neighboring gene were more likely to be correctly predicted than genes with partially overlapping promoter regions, but the results were not significant for most response groups (Supplemental Table 3.1), so genes oriented in any direction were kept. For the analysis of the regulatory information in

regions outside the proximal promoter, only genes that had sequence data available for all regions were included (Supplemental Table 3.4)

The enrichment of GO terms (http://www.geneontology.org/ontology/subsets/goslim_plant.obo) and metabolic pathways (http://www.plantcyc.org) in the response group genes compared to NNN genes, were determined using the Fisher's Exact test with *p*-values adjusted for multiple testing (Storey 2003). As no AraCyc terms were enriched, only GO terms were discussed.

3.5.2 Identification of known binding sites from in vitro TF binding data

Two sets of *in vitro* TF binding motif (TFBM) data were used to identify known binding sites. First, *in vitro* 200 bp binding regions for 344 TFs were collected from the DAP-Seq database (O'Malley *et al.* 2016). These 200 bp regions were derived from mapped sequencing peaks, and only peaks with a fraction of reads in peaks (FRiP) ≥ 5% were included. Second, position frequency matrices (PFMs) were obtained from the CIS-BP database for an additional 190 TFs without DAP-Seq data (Weirauch *et al.* 2014). CIS-BP PFMs were covered to Position Weight Matrices (PWM) adjusted for *A. thaliana's* AT (0.33) and GC (0.17) background using the TAMO package (Gordon et al 2005). These 190 PWMs were then mapped to the putative promoter region (within 1kb upstream of the transcription start site) of *A. thaliana* genes using Motility with a threshold of $p<1e-06$ (http://cartwheel.caltech.edu/motility/). A gene was considered to be regulated by a TF if its putative promoter region overlapped with one or more known TFBM sites. We also identified a subsets of known TFBMs that were enriched in the promoter regions of genes in a response group compared to non-responsive (NNN) genes using the Fisher's Exact test ($p<0.05$), these TFBMs are referred to as the known enriched TFBMs.

3.5.3 Computational identification of novel pCREs and comparison with known TFBMs

To identify pCREs that were not covered by the available *in vitro* TF binding data, an enrichment based computational approach was taken (referred to as the iterative *k*-mer finding approach). With this approach, modified from (Liu *et al.* 2018), all possible *6*-mers tested for enrichment in the response group gene promoters compared to NNN gene promoters using the Fisher's Exact test ($p < 0.01$). For *6*-mers that were enriched, their sequence was lengthened to all eight possible *7*-mers (e.g. ATATCG → AATATCG, TATATCG, GATATCG, CATATCG, ATATCGA, ATATCGT, ATATCGG, ATATCGC), which were then each tested for enrichment. The *k*-mer lengthening process continued until the longer *k*-mers were no longer significantly enriched. The above was repeated to find enriched pCREs in the 5' UTR, 1st intron, 3' UTR, and 500 bp downstream region.

To assess the sequence similarity between (A) the pCREs identified for different response groups, (B) between the pCREs identified in different regions, and (C) between the pCREs and all known *in vitro* TFBMs, the Pearson's Correlation Coefficients (PCC) between pCREs/TFBMs were calculated as in (Uygun *et al.* 2017). For two pCREs from (A) or from (B) to be considered similar, their PCC had to be the ≥ 95th percentile value of PCCs (i.e. > 0.78) between best matching pairs of pCREs from 100 sets of 25 random *6*-mers, where each pCRE in each set was paired with the pCRE from each of the other 99 sets with the highest PCC value. We used 25 because it was the average number of pCREs enriched in each genetic region across all response groups. To determine the degree of sequence similarity in (C), three PCC thresholds for each TFBM were calculated that range from least to most stringent. The lowest level of stringency is "better than random", where the pCRE-TFBM PCC is ≥ 95th percentile of PCCs between the TFBM and 1,000 random *k*-mers. The next level of stringency is "between family",

where the pCRE-TFBM PCC is ≥ 95th percentile of PCCs between the TFBM and TFBMs from other TF families. Finally, the highest level of stringency is "within family", where the pCRE-TFBM PCC is ≥ 95th percentile of PCCs between TFBMs from within the same family.

3.5.4 Sequence conservation, chromatin accessibility, and histone mark data processing and analysis

Sequence conservation the between species conservation criteria, *A. thaliana* genomic regions that overlapped with ~90,000 Conserved Non-coding Sequences (CNS) among 9 Brassicaceae species were used (Haudry *et al.* 2013). DNase I Hyper-Sensitivity (DHS) regions were downloaded from GEO (GSE53322 and GSE53324) as peaks in bed format. These regions were identified from multiple tissues and developmental stages, including roots, root hair cells, leaf, seed coat, and dark grown *A. thaliana* Col-0 seedlings at 7-days old (Sullivan *et al.* 2014). Regions associated with activation-associated histone marks (H3K4me1: SRR2001269, H3K4me3: SRR1964977, H3K9ac: SRR1964985, and H3K23ac: SRR1005405) and with repression-associated histone marks (H3K9me1: SRR1005422, H3K9me2: SRR493052, H3K27me3: SRR3087685, and H3T3p: SRR2001289) were as compiled previously (Lloyd *et al.* 2018) using data from (Stroud *et al.* 2014).

The percentage of times the sites of a pCRE overlapped with the 11 additional regulatory information (DAP-Seq, CNS, DHS, and eight histone marks) was calculated for each combination of pCRE and additional regulatory information for each response group. To determine how these overlaps were significant or not, 1,000 random, unique *6*-mers were generated and mapped to the promoter regions of response group genes, then the percentage of overlap with each combination of random *6*-mer and additional regulatory information was calculated for each response group. These overlap percentages were used to generate a

background distributions for overlap with each additional regulatory region, allowing us to convert the percent overlap scores for pCREs into percentiles along this background distribution. The percentage overlap with each additional regulatory information was also calculated for all CIS-BP motifs. Analysis of Variance (ANOVA), implemented in R v3.5.3, was used to determine if there were difference in the overlap percentage for each of the 11 additional regulatory information for each set of response group genes all pCRE, the top 10 most important pCREs (details below), the CIS-BP motifs, and the 1,000 random *6*-mers. The ANOVA *p*-values were adjusted for multiple testing (Storey 2003). Finally, post-hoc Tukey tests, implemented using the HSD.test function from the agricolae package in R, were performed on comparisons with a significant ANOVA (q-value < 0.05) to identify which groups (i.e. pCREs, top 10 pCREs, CIS-BP, or random *6*-mers) had significantly different distributions in their percent of overlap with the additional regulatory information ($p < 0.05$).

To convert the additional regulatory information into features that could be used as input to our machine learning models, a new feature was generated for each pCRE – additional regulatory information pair (e.g. pCRE-DHS), where the value of the feature was set to 1 if the pCRE was both present in the promoter region of the gene and overlapped with the additional regulatory information and set to 0 if either or both of those criteria were not met. This resulted in a total of 12 features associated with each pCRE (i.e. the original presence/absence feature + the 11 additional features).

3.5.5 Classic machine learning-based models of the cis-regulatory code

A classic machine learning algorithm called Random Forest (RF) (Breiman 2001) was used to generate models of the *cis*-regulatory code for each response group. These models were trained using a supervised learning approach, meaning they learned to predict the desired output

(e.g. does the gene belong to response group NNU or NNN?) using example instances (i.e. genes) for which they have both the input features (e.g. presence of absence of pCRE-X) and the true classification (e.g. NNU or NNN). Different sets of input features were used throughout the study, including known TFBMs, promoter pCREs, combinatorial pCRE rules (see Supplemental Methods), overlap with additional regulatory information, and non-promoter pCREs.

RF was implemented using Scikit-Learn in Python 3 (Pedregosa *et al.* 2011). To avoid training models that classify all genes as belonging to the more common response group, we balanced our input data by randomly down-sampling genes from the larger response group to match the number of genes in the smaller response group. Because the genes included in the input data can impact model training and performance, this process was replicated 100 times. To measure the performance of our models on a set of genes not seen by the model during training we used a 10-fold cross-validation scheme, where the input data was randomly divided into 10 bins, then a model was trained on bins 1-9 (i.e. the training set) and that model's performance was measured based on how will it performed on the instances in the 10^{th} bin (i.e. the validation set). This was repeated, until each bin was used as the validation set one time. To select what values to use for two important RF parameters—maximum depth [3, 5, 10, 50] and maximum features [10%, 25%, 50%, 75%, 100%, square root(100%), and log2(100%)]—a cross-validated grid search implemented using GridSearchCV from Scikit-Learn was performed on the first 10 of the 100 balanced datasets (Supplemental Table 3.7). The maximum depth parameter controls how deep each decision tree can be trained, where trees that are too shallow may not be able to capture complex patterns and trees that are too deep may overfit, meaning they would predict the training genes well, but would not generalize to predict genes not included in training well (e.g. the validation set or new genes). The maximum features parameter controls how many of the

input features each decision tree in the forest will be allowed to use, where too few will result in poor performance from individual decision trees and too many will result in most decision trees in the forest identifying the same pattern.

Model performance was evaluated using the F-measure (F1) (Bishop and Others 2006), or the harmonic mean of precision (True Positive / True Positives + False Positives) and recall (True Positives / True Positives + False Negatives), where an F1=1 would indicate all gene were perfectly classified, and an F1=0.5 would indicate the model did no better than random guessing. For each model we also determined which genes were correctly classified as belonging to a response group, *R*. Every balanced run of the model could have predicted a different subset of genes as belonging to *R*. Thus, a final classification call that a gene, *G,* belongs to group *R* was determined if the mean predicted probability of 100 balanced runs ≥ the predicted score threshold (i.e. the threshold between 0 and 1 that maximized model performance averaged over replicates). For each balanced run, we identified the predicted score that maximized the F-measure. We took the average of the predicted score maximizing F-measures for all 100 runs as the predicted score threshold. Then, models with similar F1 scores could be compared to see if they predicted a different subset of genes. Finally, the relative importance of each feature in a RF model was determined using the _importance_score function built into the Scikit-Learn implementation of RF. This function calculates feature importance as the normalized decrease in node impurity across the decision trees when that feature is used to divide a node, known as the Gini Importance (Breiman 2001).

3.5.6 Convolutional neural network-based models of the cis-regulatory code

Convolutional neural networks (CNNs), a deep learning algorithm (Breiman 2001), were tested to see if it could better integrate additional regulatory information into our models of the

cis-regulatory code. CNNs were implemented in Python 3.6 using Tensorflow 2.0 (Girija 2016). CNN models were made up of four layers: input, convolutional, dense (i.e. fully connected), and the output (i.e. the prediction) (see Figure 3.4B). The input is a 3-dimensional array [rows x columns x layers] where each layer contains data from a different gene, each column (size= # of pCREs for that response group) contains different pCREs, and each row (size=12) contains either pCRE presence/absence or overlap with additional regulatory information. The convolutional layer is composed of kernels (i.e. pattern finders) with the dimensions [12 x 1], using a stride length =1, this resulted in each kernel passing over each pCRE one time and resulting in an output with dimensions [# kernels x # pCREs]. The starting kernel weights were initialized randomly and were scaled relative to the size of the input data using Xavier Initialization (Glorot and Bengio). The output from the convolutional layer was flattened (i.e. changed the output from a 2D array to a 1D array with shape [1 x (# kernels x # pCREs)]) and then passed to the dense layer. A non-linear activation function (rectified linear units; ReLU) was applied to both the convolutional and dense layers, and a sigmoid activation function was applied to the final output layer to facilitate making a binary decision (e.g. NNU vs. NNN). Weights were optimized using the Stochastic Gradient Descent with momentum (SGDm) (momentum=0.9) as implemented in Tensorflow.

Three strategies were used to reduce the likelihood of the CNN models overfitting, where models train so specifically to the training data that they do not generalize well to new data. First, L2 regularization was applied to the kernel weights in our convolutional layer, forcing the weights to shrink toward zero. Second, dropout regularization was applied to the dense layer, meaning during each iteration of training a random subset of the dense nodes were removed. This essentially adds randomness to the model and encourages the network to learn more general

patterns in the data, rather than specific ones that may be overfit. Finally, CNNs can overfit to the training data if they are allowed to train for too many iterations. However, training for too few iterations will result in a model that has not yet converged (i.e. underfitting). To determine when to best stop training, we used an early stopping approach implemented in Keras (https://keras.io/callbacks/#earlystopping), where the training data was further split into training (90%) and validation (10%) and training stopped when model performance had not increased (min_delta = 0) for 10 iterations (patience = 10) on the validation data, with the maximum number of training iterations limited to 1,000. As with the RF models described above, CNN models were trained on balanced datasets. Because of the greater computational power needed by CNNs, instead of the cross-validation approach used for RF, the balanced data was divided into a training set (90%) and testing set (10%) and performance was measured on the testing set. Model parameters were selected using a random search across the parameter space with five-fold cross validation with ~4,800 iterations (implemented using RandomizedSearchCV in Scikit-Learn). Parameters in the search included the learning rate, the number of kernels in the convolutional layer, the number of nodes in the dense layer, the dropout rate, and the L2 regularization rate (see Table 7).

The importance of each pCRE and its associated additional regulatory information was determined by measuring the difference in model performance between the original model and a new model when the values in all rows for a pCRE column were set to zero (i.e. not present and not overlapping with the additional regulatory information) for all genes. Thus, larger positive differences indicate pCREs were important. Negative scores indicate zeroing out the pCREs in question actually improved model performance. The change in performance measured using the area under the receiver operator characteristic, rather than the F1 because it does not require the

selection of a classification threshold. The median importance scores across replicates were used to summarize the importance of each pCRE and its associated additional regulatory information. To determine what patterns the CNNs learned to identify, we extracted the weights from each kernel in the convolutional layer of our trained CNN models. Because we trained 100 CNN models, each with either 8 or 16 kernels (see Table 7), we used hierarchical clustering with dynamic branch cutting (minimum cluster size = 250) to group kernels based on the similarity of their weights and found the median weight at each position for each cluster. Kernel importance was measured as described above, where the change in model performance after a kernel's weights were set to zero (i.e. identifying no pattern) was calculated for each kernel. The median kernel importance scores across all kernels in a cluster are show.

3.5.7 Data Availability

All data used in this study are publicly available (Haudry *et al.* 2013; Prasch and Sonnewald 2013; Stroud *et al.* 2014; Weirauch *et al.* 2014; Sullivan *et al.* 2014; O'Malley *et al.* 2016). All code needed to reproduce the results from this study are available on GitHub (https://github.com/ShiuLab/Manuscript_Code/2019_CRC_HeatDrought). This repository also contains a detailed README.md file which describes our analyses in more detail, provides the commands used to generate the results in this study, and includes links to the most recent versions of the scripts used.

APPENDIX

Supplemental Information

Methods: Combinatorial pCRE rule discovery

To determine if pairs of pCREs were significantly more likely to occur in combination with each other in response group genes, we mined for association rules between our computationally identified pCREs using two statistical tests. First, the 'aRules' package (Hahsler *et al.* 2011) implemented in R, was used to identify pairs of pCREs that were found together in at least 20% of the genes in a response group (support ≥0.2), had a confidence score ≥0.5, where confidence (X⇒Y) was defined as support(X∪Y)/support(X), and had a lift score significantly > 1 ($q < 0.05$) using Fisher's Exact Test and multiple testing correction using Bonferroni. Lift (X⇒Y) is defined as support(X∪Y)/(support(X)*support(Y)) and can be interpreted as the support for the rule given the prior probability of obtaining that rule by chance. To be regarded as combinatorial pCREs, pairs also had to be on average ≥2 base pairs apart. Second, the package iterative Random Forest ('iRF': (Basu *et al.* 2018)) also implemented in R, was used to identify pCREs that formed stable interactions in the large leaf nodes of ensembles of decision trees. For each response group, iRFs were identified and compared to iRFs identified after permuting the presence/absence data, iRFs not identified after imputation and those with stability scores significantly greater than the imputed iRF stability scores were regarded as combinatorial pCREs.

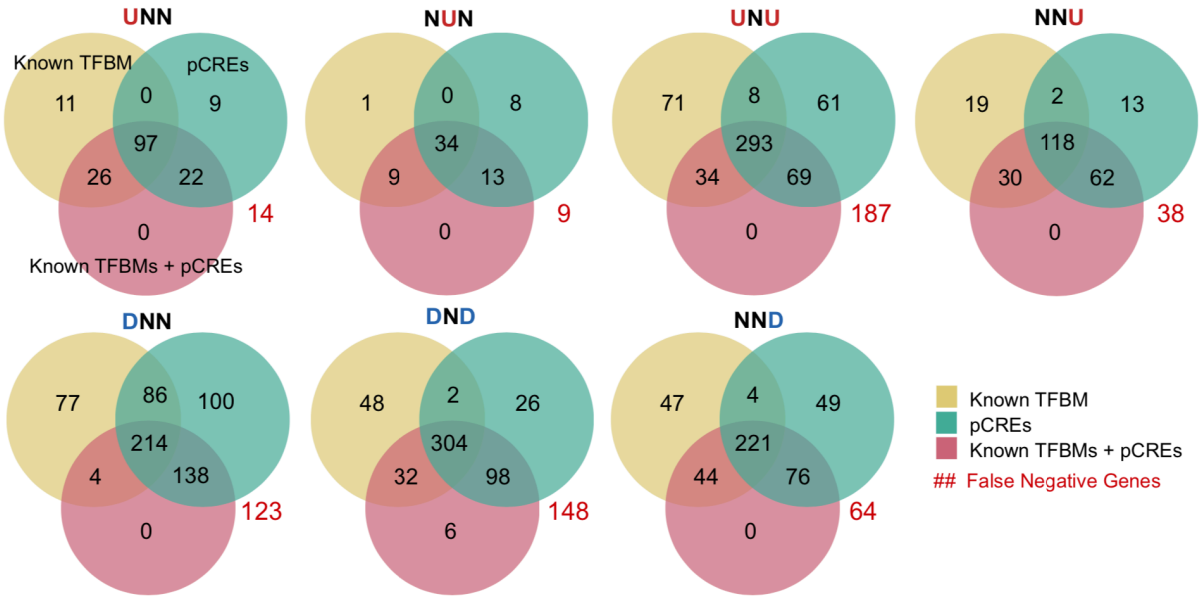
Results: Significant interactions between pCREs exist but do not improve predictive models

Given that TFs frequently work in concert (Farnham 2009), we next incorporated interactions between TFs into our *cis*-regulatory code models by generating input features to represent significant interactions between pCREs. We identified interactions between pCREs for

each response group using three approaches. The first two approaches, association Rule (aRules) and iterative Random Forest (iRF), looked for significant association between the pCREs we had already identified for each response group. The third approach was to use a combinatorial rule aware approach to identify new pairs of pCREs that were not identified by the iterative *k*-mer finding method because they were not individually enriched in the response group genes. Finally, we assessed if the inclusion of these pCRE interactions as features improved our models of the *cis*-regulatory code for each response group.

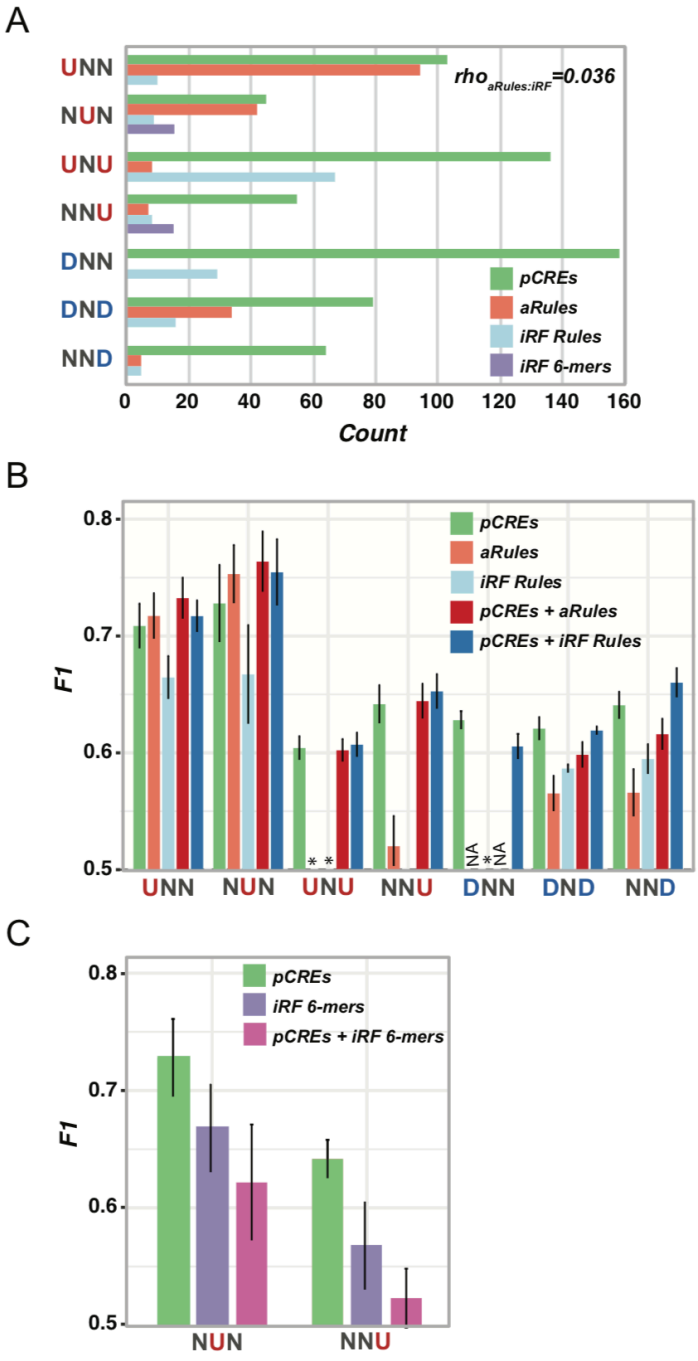
The first statistical approach, association Rules (aRules), identified pairs of pCREs that were significantly more likely to be found in the same promoter of response group genes compared to NNN genes and were found together in at least 20% of the response group genes (i.e. 20% support) (Hahsler *et al.* 2011). Because our iterative *k*-mer finding approach identified some pCREs with high sequence similarity that likely represent different regions of the same CRE (e.g. AATACT, GAATAC), we also stipulated that the average distance between pCREs in an aRule must be >2 bp. The second approach, iterative Random Forest (iRF) (Basu *et al.* 2018), identified pCREs that formed significantly more stable interactions in decision trees built to classify response group from NNN genes than in decision trees built using permuted data. While the average number of association rules across response groups was similar for both methods (27 aRules and 21 iRF rules), the number of association rules identified for each response group by each method was not correlated (Spearman's rank correlation (rho)=0.04) (Supplemental Figure 3.2A), suggesting these approaches identified different types of associations. Finally, we used iRF to identify interactions between *6*-mers from all possible *6*-mers for NUN and NNU and found 15 and 14 *6*-mer pairs, respectively.

To determine if the pCRE interactions improved our predictive models, we converted pCRE interactions into features, where the feature was given a value = 1 if both pCREs were present in the gene promoter and = 0 otherwise. When RF models were built using the interaction feature alone or in combination with the single pCRE features, model performance either did not improve or improved moderately, with aRules improving performance for UNN and NUN and iRF rules improving performance for NND (Supplemental Figure 3.2B). We hypothesized the lack of improvement was due to the fact that combinatorial rule features held less information (i.e. are sparser) than single pCRE features because two pCREs had to be present in a gene for the interaction to be considered present. Additionally, because of the hierarchical structure of the decision trees that make up the RF models, RF is able to model interaction effects without the need for explicitly coded interaction features. Finally, novel *6*-mer pairs identified using iRFs did not improve model performance when used as features alone or with pCREs (Supplemental Figure 3.3B).



Supplemental Figure 3.1. Overlap in true positive gene predictions from models using known TFBMs, pCREs, or both features as input.

The number of genes that were similarly correctly predicted by known TFBM, pCRE, and known TFBM + pCRE *cis*-regulatory code models for each response group. The number of genes in a response group that were predicted as belonging to that response group by any of the models (i.e. False Negatives) is shown in red.

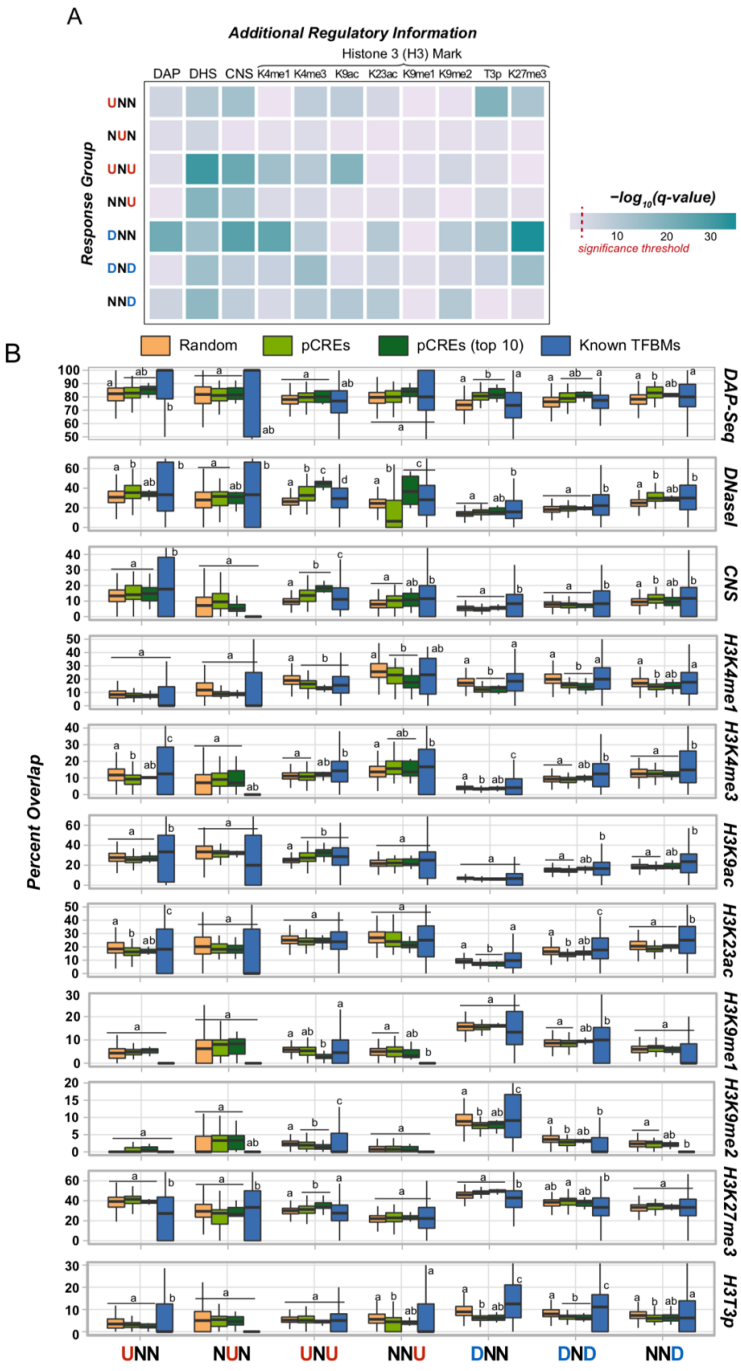


Supplemental Figure 3.2. Impact of including association rules between pCREs as model features.

(A) The number of pCREs (green), and pCRE association rules identified using aRules (pink), and iRF rules (light blue) for each response group (Y-axis). (B) Performance of RF *cis*-

Supplemental Figure 3.2 (cont'd)

regulatory code models using all single pCREs (green; as in Figure 3.3A), only aRules (pink), only iRF rules (light/), single pCREs + aRules (red), and single pCREs + iRF rules (dark blue) as input features. (C) Performance of RF *cis*-regulatory code models using *6*-mer pairs identified by iRF from a set of all possible *6*-mers as features.

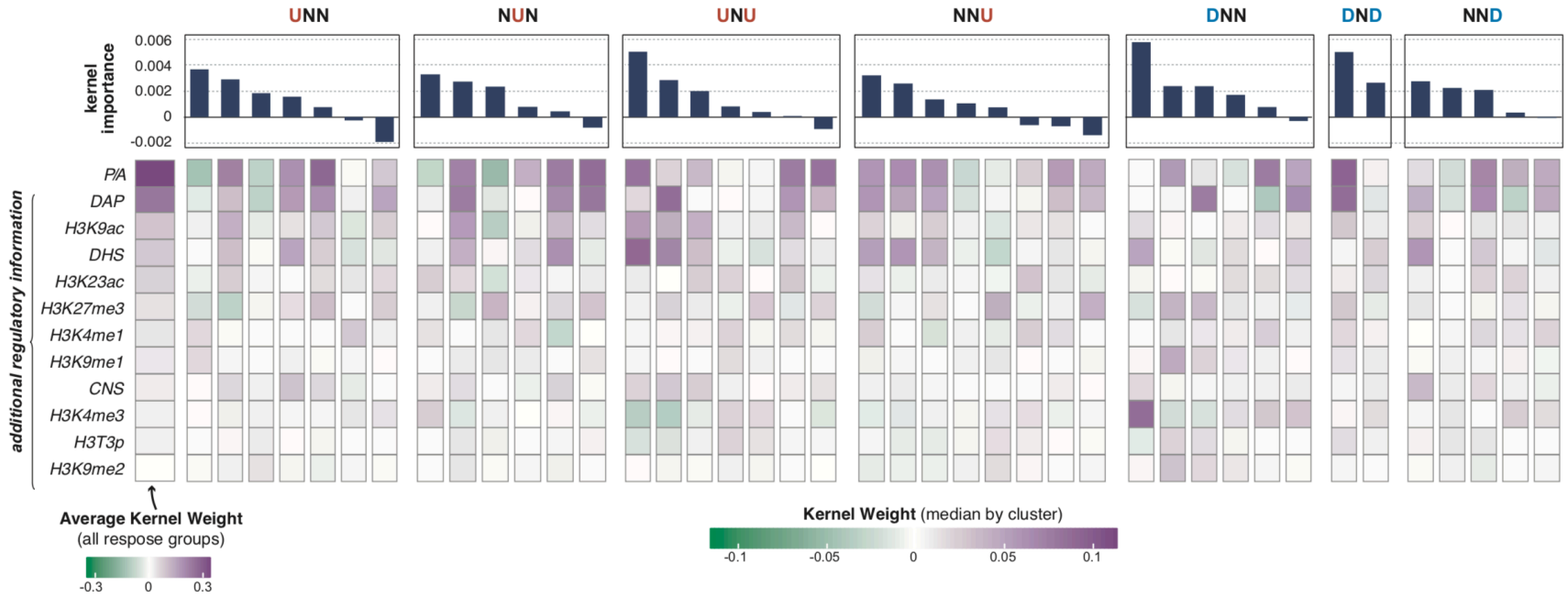


Supplemental Figure 3.3. The association of additional regulatory information with pCREs compared to random 6-mers and known TFBMs.

(A) Results of the Analysis of Variance (ANOVA) tests use to determine if there were difference in the percent of times a sequence overlapped with each of the 11 additional regulatory information when it was present in a response group gene for all pCRE (light green), the top 10

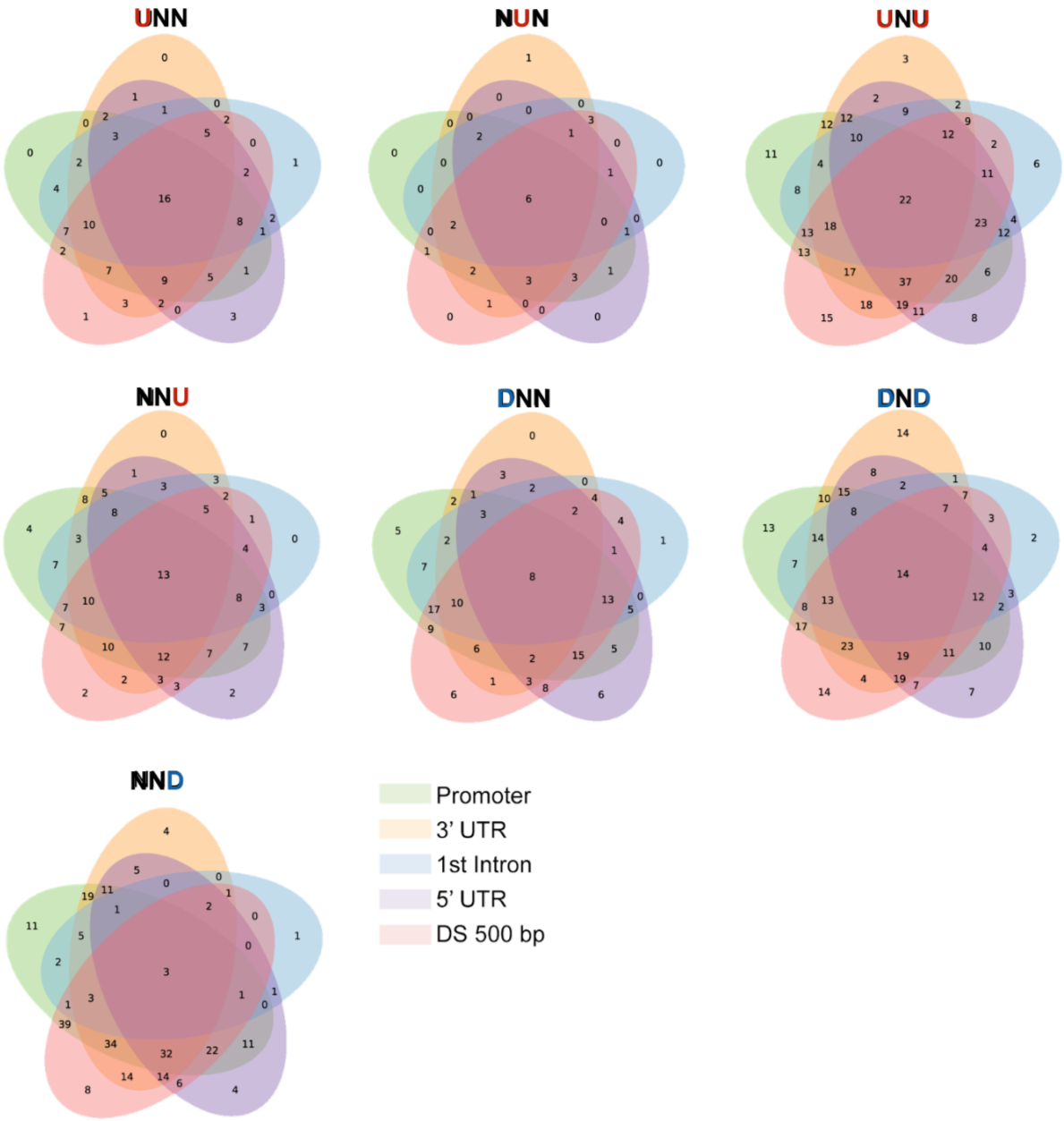
Supplemental Figure 3.3 (cont'd)

most important pCREs (based on the Gini Index from RF models; dark green), known TFBM (i.e. CIS-BP motifs; blue), and 1,000 random *6*-mers (coral), for each response groups. The results from the 77 ANOVA were corrected for multiple testing (*q*-values) and shown here as the negative log10(*q*-value). (B) The distribution of the percent overlap data used in (A). For the 59 response group - additional regulatory information pairs with significant differences in overlap (ANOVA *q*-value < 0.05), sequence groups are labeled (i.e. a, b, c) based on which groups are significantly different from each other using a post-hoc Tukey test (*p*-value < 0.05).



Supplemental Figure 3.4. Probing the trained kernels to understand the important patterns of additional regulatory information identified by CNN models.

The full results from interpreting the trained CNN models (see Figure 3.4C). The feature types (i.e. presence/absence (P/A) and additional regulatory information) were sorted based on the average kernel weights across all kernels trained for all response groups and replicates (first column). The remaining columns represent kernel clusters for specific response groups. For each response group, all trained kernels from all CNN replicates were clustered using hierarchical clustering with dynamic cutting (min cluster size=250 kernels). The median kernel weights and kernel importance scores are shown here for the resulting clusters.



Supplemental Figure 3.5. Overlap in true positive gene predictions from models using pCREs from different genetic regions.

The number of genes that were similarly correctly predicted by pCREs identified in the promoter (green), 3' untranslated region (UTR) (orange), first intron (blue), 5' UTR (purple), and downstream (500 bp; DS500; red) regions for each response group.

REFERENCES

REFERENCES

Arvey, A., P. Agius, W. S. Noble, and C. Leslie, 2012 Sequence and chromatin determinants of cell-type–specific transcription factor binding. Genome Res. 22: 1723–1734.

Atkinson, N. J., C. J. Lilley, and P. E. Urwin, 2013 Identification of genes involved in the response of Arabidopsis to simultaneous biotic and abiotic stresses. Plant physiology 162: 2028–2041.

Benjamini, Y., and Y. Hochberg, 1995 Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. J. R. Stat. Soc. Series B Stat. Methodol. 57: 289–300.

Bishop, C. M., and Others, 2006 Pattern recognition and machine learning (information science and statistics).

Bonnet, C., S. Lassueur, C. Ponzio, R. Gols, M. Dicke *et al.*, 2017 Combined biotic stresses trigger similar transcriptomic responses but contrasting resistance against a chewing herbivore in Brassica nigra. BMC Plant Biol. 17: 127.

Breiman, L., 2001 Random Forests. Machine Learning 45: 5–32.

Chain, B., 2012 agilp: Agilent expression array processing package. Internet] URL http://www. bioconductor. org/packages/release/bioc/html/agilp. html [accessed on May 2013].

Chang, Y., B. H. Nguyen, Y. Xie, B. Xiao, N. Tang *et al.*, 2017 Co-overexpression of the Constitutively Active Form of OsbZIP46 and ABA-Activated Protein Kinase SAPK6 Improves Drought and Temperature Stress Resistance in Rice. Front. Plant Sci. 8: 1102.

Choi, Y.-S., Y.-M. Kim, O.-J. Hwang, Y.-J. Han, S. Y. Kim *et al.*, 2013 Overexpression of ArabidopsisABF3 gene confers enhanced tolerance to drought and heat stress in creeping bentgrass. Plant Biotechnol. Rep. 7: 165–173.

De Storme, N., and D. Geelen, 2014 The impact of environmental stress on male reproductive development in plants: biological processes and molecular mechanisms. Plant Cell Environ. 37: 1–18.

Ding, Z. J., J. Y. Yan, C. X. Li, G. X. Li, Y. R. Wu *et al.*, 2015 Transcription factor WRKY46 modulates the development of Arabidopsis lateral roots in osmotic/salt stress conditions via regulation of ABA signaling and auxin homeostasis. Plant J. 84: 56–69.

Doebley, J. F., B. S. Gaut, and B. D. Smith, 2006 The Molecular Genetics of Crop Domestication. Cell 127: 1309–1321.

Dong, X., and Z. Weng, 2013 The correlation between histone modifications and gene expression. Epigenomics 5: 113–116.

Farnham, P. J., 2009 Insights from genomic profiling of transcription factors. Nature Reviews Genetics 10: 605–616.

Georgii, E., M. Jin, J. Zhao, B. Kanawati, P. Schmitt-Kopplin *et al.*, 2017 Relationships between drought, heat and air humidity responses revealed by transcriptome-metabolome co-analysis. BMC Plant Biol. 17: 120.

Ghandi, M., D. Lee, M. Mohammad-Noori, and M. A. Beer, 2014 Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features. PLoS Computational Biology 10: e1003711–15.

Girija, S. S., 2016 Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Glorot, X., and Y. Bengio Understanding the difficulty of training deep feedforward neural networks. 2010.

Harbison, C. T., D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac *et al.*, 2004 Transcriptional regulatory code of a eukaryotic genome. Nature 431: 99–104.

Haudry, A., A. E. Platts, E. Vello, D. R. Hoen, M. Leclercq *et al.*, 2013 An atlas of over 90,000 conserved noncoding sequences provides insight into crucifer regulatory regions. Nature Publishing Group 45: 891–898.

He, H. H., C. A. Meyer, M. W. Chen, V. C. Jordan, M. Brown *et al.*, 2012 Differential DNase I hypersensitivity reveals factor-dependent chromatin dynamics. Genome Res. 22: 1015–1025.

Hsieh, W.-P., H.-L. Hsieh, and S.-H. Wu, 2012 Arabidopsis bZIP16 Transcription Factor Integrates Light and Hormone Signaling Pathways to Regulate Early Seedling Development. The Plant Cell 24: 3997–4011.

Hu, H., J. You, Y. Fang, X. Zhu, Z. Qi *et al.*, 2008 Characterization of transcription factor gene SNAC2 conferring cold and salt tolerance in rice. Plant Mol. Biol. 67: 169–181.

Huebert, D. J., P. F. Kuan, S. Keles, and A. P. Gasch, 2012 Dynamic Changes in Nucleosome Occupancy Are Not Predictive of Gene Expression Dynamics but Are Linked to Transcription and Chromatin Regulators. Molecular and Cellular Biology 32: 1645–1653.

James, A. B., N. H. Syed, J. W. S. Brown, and H. G. Nimmo, 2012 Thermoplasticity in the plant circadian clock. Plant Signal Behav 7: 1219–1223.

Kang, H.-G., J. Kim, B. Kim, H. Jeong, S. H. Choi *et al.*, 2011 Overexpression of FTL1/DDF1, an AP2 transcription factor, enhances tolerance to cold, drought, and heat stresses in Arabidopsis thaliana. Plant Science 180: 634–641.

Karmodiya, K., A. R. Krebs, M. Oulad-Abdelghani, H. Kimura, and L. Tora, 2012 H3K9 and H3K14 acetylation co-occur at many gene regulatory elements, while H3K14ac marks a

subset of inactive inducible promoters in mouse embryonic stem cells. BMC Genomics 13: 424.

Kidokoro, S., K. Maruyama, K. Nakashima, Y. Imura, Y. Narusaka *et al.*, 2009 The phytochrome-interacting factor PIF7 negatively regulates DREB1 expression under circadian control in Arabidopsis. Plant Physiol. 151: 2046–2057.

King, G. J., 2015 Crop epigenetics and the molecular hardware of genotype × environment interactions. Frontiers in Plant Science 6: 10217–19.

Konishi, S., T. Izawa, S. Y. Lin, K. Ebana, Y. Fukuta *et al.*, 2006 An SNP caused loss of seed shattering during rice domestication. Science 312: 1392–1396.

Kumar, M., G. Sirhindi, R. Bhardwaj, S. Kumar, and G. Jain, 2010 Effect of exogenous H2O2 on antioxidant enzymes of Brassica juncea L. seedlings in relation to 24-epibrassinolide under chilling stress. Indian J. Biochem. Biophys. 47: 378–382.

Lee, D.-K., P. J. Chung, J. S. Jeong, G. Jang, S. W. Bang *et al.*, 2017 The rice OsNAC6 transcription factor orchestrates multiple molecular mechanisms involving root structural adaptions and nicotianamine biosynthesis for drought tolerance. Plant Biotechnol. J. 15: 754–764.

Li, X., H. Zhang, Q. Ai, G. Liang, and D. Yu, 2016 Two bHLH Transcription Factors, bHLH34 and bHLH104, Regulate Iron Homeostasis in Arabidopsis thaliana. Plant Physiology 170: 2478–2493.

Liu, T., J. Carlsson, T. Takeuchi, L. Newton, and E. M. Farré, 2013 Direct regulation of abiotic responses by the Arabidopsis circadian clock component PRR7. The Plant Journal 76: 101–114.

Liu, M.-J., K. Sugimoto, S. Uygun, N. Panchy, M. S. Campbell *et al.*, 2018 Regulatory Divergence in Wound-Responsive Gene Expression between Domesticated and Wild Tomato. The Plant Cell 30: 1445–1460.

Lloyd, J. P., Z. T.-Y. Tsai, R. P. Sowers, N. L. Panchy, and S.-H. Shiu, 2018 A Model-Based Approach for Identifying Functional Intergenic Transcribed Regions and Noncoding RNAs (J. Gojobori, Ed.). Molecular Biology and Evolution 35: 1422–1436.

Luo, C., and E. Lam, 2010 ANCORP: a high-resolution approach that generates distinct chromatin state models from multiple genome-wide datasets. The Plant Journal 63: 339–351.

Mathur, S., D. Agrawal, and A. Jajoo, 2014 Photosynthesis: Response to high temperature stress. Journal of Photochemistry & Photobiology, B: Biology 137: 116–126.

Nicotra, A. B., O. K. Atkin, S. P. Bonser, A. M. Davidson, E. J. Finnegan *et al.*, 2010 Plant phenotypic plasticity in a changing climate. Trends in Plant Science 15: 684–692.

O'Malley, R. C., S. C. Huang, L. Song, M. G. Lewsey, A. Bartlett *et al.*, 2016 Cistrome and Epicistrome Features Shape the Regulatory DNA Landscape. Cell 1–21.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, 2011 Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12: 2825–2830.

Pfluger, J., and D. Wagner, 2007 Histone modifications and dynamic regulation of genome accessibility in plants. Curr. Opin. Plant Biol. 10: 645–652.

Prasch, C. M., and U. Sonnewald, 2013 Simultaneous application of heat, drought, and virus to Arabidopsis plants reveals significant shifts in signaling networks. Plant physiology 162: 1849–1866.

Rabara, R. C., P. Tripathi, and P. J. Rushton, 2014 The Potential of Transcription Factor-Based Genetic Engineering in Improving Crop Tolerance to Drought. OMICS: A Journal of Integrative Biology 18: 601–614.

Rasmussen, S., P. Barah, M. C. Suarez-Rodriguez, S. Bressendorff, P. Friis *et al.*, 2013 Transcriptome responses to combinations of stresses in Arabidopsis. Plant physiology 161: 1783–1794.

Reynolds, M. P., and R. Ortiz, 2010 Adapting crops to climate change: a summary, pp. 1–8 in *Climate change and crop production*, edited by M. P. Reynolds. CABI, Wallingford.

Ritchie, M. E., B. Phipson, D. Wu, Y. Hu, C. W. Law *et al.*, 2015 limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. 43: e47.

Rizhsky, L., H. Liang, J. Shuman, V. Shulaev, S. Davletova *et al.*, 2004 When defense pathways collide. The response of Arabidopsis to a combination of drought and heat stress. Plant Physiol. 134: 1683–1696.

Sewelam, N., Y. Oshima, N. Mitsuda, and M. Ohme-Takagi, 2014 A step towards understanding plant responses to multiple environmental stresses: a genome-wide study. Plant Cell Environ. 37: 2024–2035.

Shaar-Moshe, L., E. Blumwald, and Z. Peleg, 2017 Unique Physiological and Transcriptional Shifts under Combinations of Salinity, Drought, and Heat. Plant physiology 174: 421–434.

Shen, P.-C., A.-L. Hour, and L.-Y. D. Liu, 2017 Microarray meta-analysis to explore abiotic stress-specific gene expression patterns in Arabidopsis. Bot. Stud. 58: 22.

Sillmann, J., V. V. Kharin, X. Zhang - Journal of …, and 2013, 2013 Climate extremes indices in the CMIP5 multimodel ensemble: Part 1. Model evaluation in the present climate. Wiley Online Library.

Steinfeld, I., R. Shamir, and M. Kupiec, 2007 A genome-wide analysis in *Saccharomyces cerevisiae* demonstrates the influence of chromatin modifiers on transcription. Nature Genetics 39: 303–309.

Stocker, T. F., D. Qin, G.-K. Plattner, M. Tignor, S. K. Allen *et al.*, 2013 IPCC, 2013: climate change 2013: the physical science basis. Contribution of working group I to the fifth assessment report of the intergovernmental panel on climate change.

Storey, J. D., 2003 The positive false discovery rate: a Bayesian interpretation and the q-value. Ann. Stat. 31: 2013–2035.

Stroud, H., T. Do, J. Du, X. Zhong, S. Feng *et al.*, 2014 Non-CG methylation patterns shape the epigenetic landscape in Arabidopsis. Nat. Struct. Mol. Biol. 21: 64–72.

Sullivan, A. M., A. A. Arsovski, J. Lempe, K. L. Bubb, M. T. Weirauch *et al.*, 2014 Mapping and Dynamics of Regulatory DNA and Transcription Factor Networks in A. thaliana. CellReports 8: 2015–2030.

Sullivan, A. M., K. L. Bubb, R. Sandstrom, J. A. Stamatoyannopoulos, and C. Queitsch, 2015 DNase I hypersensitivity mapping, genomic footprinting, and transcription factor networks in plants. Biochemical Pharmacology 3–4: 40–47.

Tompa, M., 2001 Identifying Functional Elements by Comparative DNA Sequence Analysis. Genome Res. 11: 1143–1144.

Uygun, S., A. E. Seddon, C. B. Azodi, and S.-H. Shiu, 2017 Predictive Models of Spatial Transcriptional Response to High Salinity. Plant physiology 174: 450–464.

Vile, D., M. Pervent, M. Belluau, F. Vasseur, J. Bresson *et al.*, 2012 Arabidopsis growth under prolonged high temperature and water deficit: independent or interactive effects? Plant Cell Environ. 35: 702–718.

Wang, J., J. Zhuang, S. Iyer, X. Lin, T. W. Whitfield *et al.*, 2012 Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. Genome Res. 22: 1798–1812.

Wasserman, W. W., M. Palumbo, W. Thompson, J. W. Fickett, and C. E. Lawrence, 2000 Human-mouse genome comparisons to locate regulatory sites. Nat. Genet. 26: 225–228.

Weirauch, M. T., A. Yang, M. Albu, A. G. Cote, A. Montenegro-Montero *et al.*, 2014 Determination and Inference of Eukaryotic Transcription Factor Sequence Specificity. Cell 158: 1431–1443.

Wu, X., Y. Shiroto, S. Kishitani, Y. Ito, and K. Toriyama, 2009 Enhanced heat and drought tolerance in transgenic rice seedlings overexpressing OsWRKY11 under the control of HSP101 promoter. Plant Cell Rep. 28: 21–30.

Yu, C.-P., J.-J. Lin, and W.-H. Li, 2016 Positional distribution of transcription factor binding sites in Arabidopsis thaliana. Sci. Rep. 6: 25164.

Zhang, G., and G. W. Duff, 1994 Intron 1 regulation of interleukin 1 beta (IL-1β) gene transcription: an alternative promoter? Cytokine 6: 564.

Zhu, Y., A. Dong, and W.-H. Shen, 2012 Histone variants and chromatin assembly in plant abiotic stress responses. BBA - Gene Regulatory Mechanisms 1819: 343–348.

Zou, C., K. Sun, J. D. Mackaluso, A. E. Seddon, R. Jin *et al.*, 2011 Cis-regulatory code of stress-responsive transcription in Arabidopsis thaliana. PNAS 1–6.

CHAPTER FOUR: BENCHMARKING PARAMETRIC AND MACHINE LEARNING MODELS FOR GENOMIC PREDICTION OF COMPLEX TRAITS[1]

[1] The work described in this chapter has been published in the following manuscript

Christina B. Azodi, Emily G. Bolger, Andrew McCarren, Mark Roantree, Gustavo de los Campos, Shin-Han Shiu (2019) Benchmarking parametric and machine learning models for genomic prediction of complex traits. *G3*. DOI: 10.1534/g3.119.400498

4.1 Abstract

The usefulness of genomic prediction in crop and livestock breeding programs has prompted efforts to develop new and improved genomic prediction algorithms, such as artificial neural networks and gradient tree boosting. However, the performance of these algorithms has not been compared in a systematic manner using a wide range of datasets and models. Using data of 18 traits across six plant species with different marker densities and training population sizes, we compared the performance of six linear and six non-linear algorithms. First, we found that hyperparameter selection was necessary for all non-linear algorithms and that feature selection prior to model training was critical for artificial neural networks when the markers greatly outnumbered the number of training lines. Across all species and trait combinations, no one algorithm performed best, however predictions based on a combination of results from multiple algorithms (i.e. ensemble predictions) performed consistently well. While linear and non-linear algorithms performed best for a similar number of traits, the performance of non-linear algorithms vary more between traits. Although artificial neural networks did not perform best for any trait, we identified strategies (i.e. feature selection, seeded starting weights) that boosted their performance to near the level of other algorithms. Our results highlight the importance of algorithm selection for the prediction of trait values.

4.2 Introduction

The ability to predict complex traits from genotypes is a grand challenge in biology and is accelerating the speed of crop and livestock breeding (Heffner *et al.* 2009; Lorenz *et al.* 2011; Jonas and de Koning 2013; Desta and Ortiz 2014). Genomic Prediction (GP, aka Genomic Selection), the use of genome-wide genetic markers to predict complex traits, was originally proposed by Meuwissen *et al*. (Meuwissen *et al.* 2001) as a solution to the limitations of Marker-

Assisted Selection (MAS) where only a limited number of previously identified markers with the strongest associations are used to select the best lines. GP is particularly well-suited for the prediction of quantitative traits controlled by many small-effect alleles (Ribaut and Ragot 2007). A major challenge in using GP is estimating the effects of a large number of makers (p) using phenotype information of a comparatively limited number of individuals (n) (i.e. $p >> n$) (Meuwissen *et al.* 2001). To address this challenge, Meuwissen *et al.* first presented three statistical methods for GP (Meuwissen *et al.* 2001). The first was a linear mixed model called ridge regression Best Linear Unbiased Prediction (rrBLUP), which uniformly shrinks the marker effects. The other two were Bayesian approaches, BayesA (BA) and BayesB (BB), which both differentially shrink the marker effects and with BB also performing variable selection. Since then, additional approaches have been shown to be useful for GP, including Least Absolute Angle and Selection Operator (LASSO) (Usai *et al.* 2009), Elastic Net (Zou and Hastie 2005), Support Vector Regression with a linear kernel (SVR_{lin}) (Moser *et al.* 2009; Xu *et al.* 2018), and additional Bayesian methods including Bayesian LASSO (BL), BayesCπ, and BayesDπ (de los Campos *et al.* 2009; Habier *et al.* 2011).

While these approaches perform well when dealing with high dimensional data (i.e. $p>>n$), they are all based on a linear mapping from genotype to phenotypes, and therefore may not fully capture non-linear effects (e.g. epistasis, dominance), which are likely to be important for complex traits (Holland 2007; Monir and Zhu 2018). To overcome this limitation, non-linear approaches, including reproducing kernel Hilbert spaces (RKHS) regression (Gianola *et al.* 2006; de los Campos *et al.* 2010), Support Vector Regression with non-linear kernels (i.e. polynomial SVR_{poly} and radial basis function SVR_{rbf} (Long *et al.* 2011; Kasnavi *et al.* 2017)), and decision tree based algorithms such as Random Forest (RF) (González-Recio and Forni 2011;

Spindel *et al.* 2015) and Gradient Tree Boosting (GTB) (González-Recio *et al.* 2013) have been applied to GP problems. In previous efforts to compare the performance of multiple linear and non-linear approaches (Heslot *et al.* 2012; Neves *et al.* 2012; Blondel *et al.* 2015; Ramstein *et al.* 2016; Roorkiwal *et al.* 2016), no single method performs best in all cases. Rather, factors such as the size of the training data set, marker type and number, trait heritability, effective population size, the number of causal loci, as well as genetic architecture (the locus effect size distribution) can all affect algorithm performance (Meuwissen 2009; Riedelsheimer *et al.* 2013; Spindel *et al.* 2015; Norman *et al.* 2018). This highlights the importance of comparing new algorithms across a diverse range of datasets.

With improvements in computing speeds, the development of graphics processing units (GPUs), and breakthroughs in algorithms for backpropagation learning (Rumelhart *et al.* 1986; Parker 1987), there has been a resurgence of research using deep learning (i.e. artificial neural networks (ANNs)) to model complex biological processes (Angermueller *et al.* 2016; Webb 2018). ANNs are a class of machine learning methods that perform layers of transformations on features to create abstraction features, known as hidden layers, which are used for predictions. The first application of ANNs for GP was presented in 2011, when Okut *et al.* trained fully connected ANNs (i.e. each node in a layer is connected to all nodes in surrounding layers) containing one hidden layer to predict body mass index in mice (Okut *et al.* 2011). Since 2011, more complex ANN architectures have been used for GP including radial basis function neural networks (González-Camacho *et al.* 2012) deep neural networks (Ehret *et al.* 2015; Bellot *et al.* 2018), deep recurrent neural networks (Pouladi *et al.* 2015), probabilistic neural network classifiers (González-Camacho *et al.* 2016, 2018), and convolutional neural networks (CNNs) (Ma *et al.* 2018). With only one exception (Bellot *et al.* 2018), these ANNs have been applied to

datasets with relatively few genetic markers (<60k), however, as sequencing continues to become less expensive, whole-genome marker datasets are becoming larger with some breeding programs generating data for hundreds of thousands of markers. Because of the internal complexity of ANN models, training an ANN with so many markers can result in sub-optimal solutions (i.e. underfitting). Therefore, it is especially important to benchmark ANNs against other GP statistical approaches on datasets with high dimensionality where underfitting may occur.

GP has yielded promising results for breeders. However, a comprehensive comparison of GP algorithms, particularly ANNs, on a wide range of GP problems is missing (Figure 4.1A). Here we compared the ability of 12 GP algorithms (see Methods, Figure 4.1B) to predict a diverse range of physiological traits in six plant species (maize, rice, sorghum, soy, spruce, and switchgrass; Figure 4.1C). These six data sets (referred to as the benchmark data sets) represent a wide range of GP data types, with the size of the training data set ranging from 327 to 5,014 individuals, and 4,000 to 332,000 markers derived from array-based approaches or sequencing. Compared to the linear algorithms included in the study, the non-linear algorithms, especially ANNs, require more pre-modeling tuning (e.g. hyperparameter selection, feature selection). Therefore, before comparing algorithm performance across all 18 combinations of species and traits, we first focused on predicting plant height in each species in order to establish best practices for model building. Because ANNs are underrepresented in GP comparison studies and our first attempts to use ANNs for GP performed relatively poorly, we focus on methods to improve ANN performance, including reducing model complexity using feature selection and combining relationships learned from linear algorithms into the more complex ANN architectures (i.e. a seeded ANN approach and convolutional layers (i.e. CNNs)). Then, using

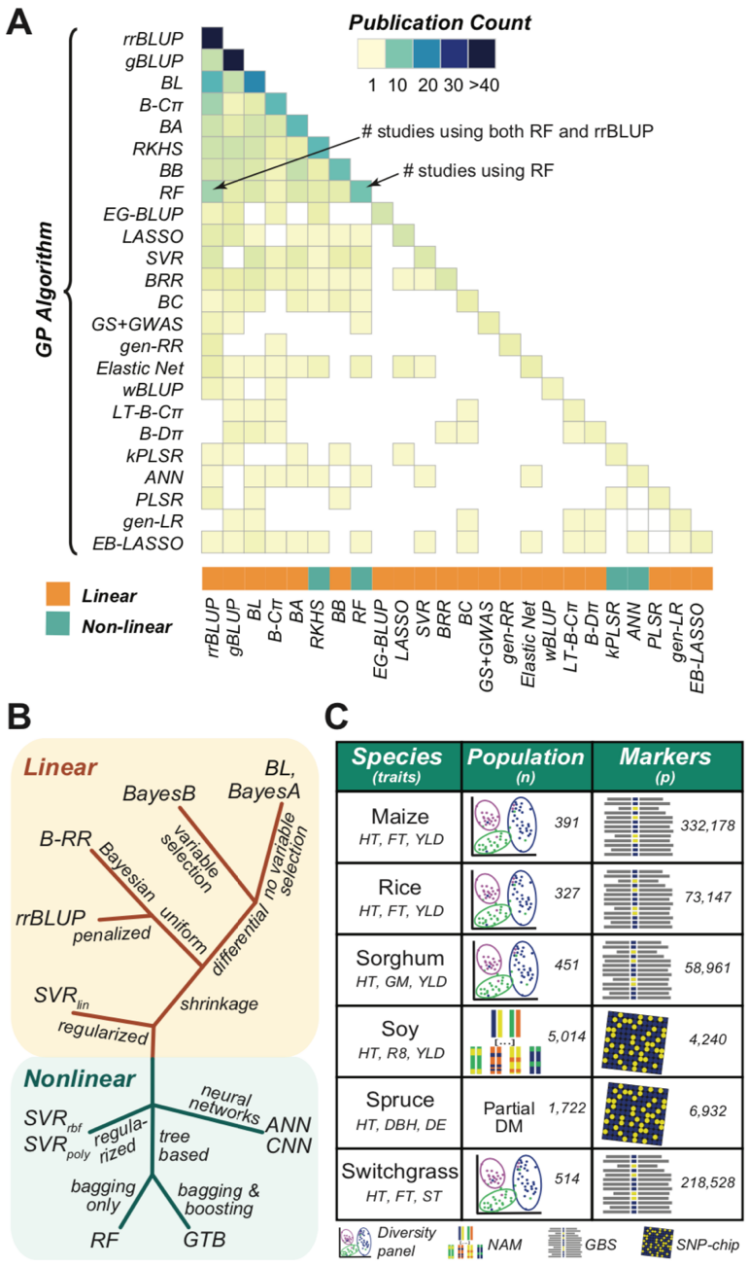


Figure 4.1. Algorithms used and compared in past GP studies and algorithms and data included in the GP benchmark.

(A) Number of times a GP algorithm was utilized (diagonal) or directly compared to other GP algorithms (lower triangle) out of 91 publications published between 2012-2018 (Supplemental Table 4.1). GP algorithms were included if they were utilized in >1 study. (B) A graphical representation of the GP algorithms included in the study and their relationship to each other.

Figure 4.1 (cont'd)

Colors designate if the algorithm identifies only linear (orange) or linear and non-linear (green) relationships. The placement of each algorithm on the tree designates (qualitatively) the relationship between different algorithms. The labels at each branch provide more information about how algorithms in that branch differ from others. rrBLUP, ridge regression Best Linear Unbiased Predictor; BRR, Bayesian Ridge Regression; BA, BayesA; BB, BayesB; BL, Bayesian LASSO; SVR, Support Vector Regression (kernel type: lin, linear; poly, polynomial; rbf, radial basis function); RF, Random Forest; GTB, Gradient Tree Boosting; ANN, Artificial Neural Network. (C) Species and traits included in the benchmark with training population types and sizes and marker types and numbers for each dataset. NAM: Nested Association Mapping. DM: partial diallel mating. GBS: genotyping by sequencing. SNP: single nucleotide polymorphism. HT: height. FT: flowering time. YLD: yield. GM: grain moisture. R8: time to R8 developmental stage. DBH: diameter at breast height. DE: wood density. ST: standability.

lessons learned from predicting height, we compared the performance of all GP algorithms across all species and traits.

4.3 Materials and Methods

4.3.1 Genotype and phenotype data

Genotypic data from six plant species were used to predict 3 traits from each species (Fig 1C). The maize phenotypic (Hansey *et al.* 2011) and genotypic (Hirsch *et al.* 2014) data were from the pan-genome population, maize trait values were averaged over replicate plots. The rice data were from elite breeding lines from the International Rice Research Institute irrigated rice breeding program (Spindel *et al.* 2015), and dry season trait data averaged over four years were

used. The sorghum data were generated from sorghum lines from the US National Plant Germplasm System grown in Urbana, IL (Fernandes *et al.* 2017) and trait values were averaged over two blocks for this study. The soybean data were generated from the SoyNAM population containing recombinant inbred lines (RILs) derived from 40 biparental populations (Xavier *et al.* 2016). The white spruce data were obtained from the SmartForests project team, using a SNP-chip developed by Quebec Ministry of Forest Wildlife and Parks (Beaulieu *et al.* 2014). Switchgrass phenotypic (Lipka *et al.* 2014) and genotypic (Evans *et al.* 2017) data were generated from the Northern Switchgrass Association Panel (Evans *et al.* 2015) which contains clones or genotypes from 66 diverse upland switchgrass populations.

The genotype data was obtained in the form of biallelic SNPs with missing marker data already dropped or imputed by the original authors. Marker calls were converted when necessary to [-1,0,1] corresponding to [aa, Aa, AA] where A was either the reference or the most common allele. Genome locations of maize SNPs were converted from assembly AGPv2 to AGPv4, with AGPv2 SNPs that did not map to AGPv4 being removed, leaving 332,178 markers for the maize analysis. Phenotype values were normalized between 0 and 1. Lines with missing phenotypic value for any of the three traits were removed.

4.3.2 Genomic selection algorithms

To assess what statistical approaches are most frequently used for genomic selection, we conducted a literature search of papers applying genomic selection methods to crop or simulated data from January 2012-February 2018. We recorded what statistical approach(es) was(were) applied in each study (Supplemental Table 4.1), allowing us to calculate both the total number of times an approach had been applied and how many times any two approaches were directly compared (Fig 1A). Based on the results from this literature search, nine commonly used

statistical approaches were included in this study: rrBLUP, Bayes A (BA), Bayes B (BB), Bayesian LASSO, Bayesian-RR, RF, SVR with a linear kernel (SVR_{lin}), SVR with polynomial kernel (SVR_{poly}), SVR with radial basis function kernel (SVR_{rbf}). Three additional machine learning approaches, gradient tree boosting (GTB), artificial neural networks (ANN), and convolutional neural networks (CNN), were also included because of their ability to model non-linear relationships.

Most linear algorithms were implemented in R packages rrBLUP (Endelman 2011) and BGLR (for Bayesian methods including BRR: Bayesian RR, BA: Bayes A, BB: Bayes B, and BL: Bayesian LASSO) (Pérez and de los Campos 2014). These algorithms vary in what approach they use to address the p >> n problem (Figure 4.1B), for example rrBLUP performs uniform shrinkage on all marker coefficients to reduce variance of the estimator, while BB performs differential shrinkage of the marker coefficients and variable selection. The differences between these algorithms have been thoroughly reviewed previously (de los Campos *et al.* 2013). Models for Bayesian methods were trained for 12,000 iterations using a burn-in of 2,000. Non-linear algorithms (SVR_{poly}, SVR_{rbf}, RF, and GTB) and SVR_{lin} were implemented in python using the Scikit-Learn library (Pedregosa *et al.* 2011). For SVR algorithms, the marker data is mapped into a new feature space using linear or non-linear kernels (i.e. poly, rbf) and then linear regression within that feature space is performed with the goal of minimizing error outside of a margin of tolerated error. The RF algorithm works by averaging the predictions from a "forest" of bootstrapped regression trees, where each tree contains a random subset of the lines and of the markers (Breiman 2001). Related to RF, GTB algorithm uses the principle of boosting (Friedman 2001) to improve predictions from weak learners (i.e. regression trees) by iteratively updating

the learners to minimize a loss function, therefore generating better weak learners as training progresses.

Artificial Neural Networks (ANNs) were implemented in python using TensorFlow (Girija 2016). The input layer for the ANNs contained the genetic markers for an individual (x; Figure 4.1B), the nodes in the hidden layers were all fully connected to all nodes in the previous and following layers (i.e. Multilayer Perceptron). A non-linear activation function (selected during the grid search, see below) was applied to each node in the input and hidden layers, except the last hidden layer, which was connected with a linear function to the output layer, the predicted trait value (y). To reduce the likelihood of vanishing gradients, when the error gradient, which controls the degree to which the weights are updated during each iteration of training, becomes so small the weights stop updating thus halting model training, in the ANN, the starting weights (w) were scaled relative to the number of input markers using the Xavier Initializer (Glorot and Bengio). Weights were then optimized using the Adam Optimizer (Kingma and Ba 2014) with a learning rate selected by the grid search (described below). To determine the optimal stopping time for training (i.e. number of epochs), an early stopping approach was used (Prechelt 1998), where the training set was further divided into training and validation, and early stopping occurred when the change in mean squared error (MSE) for the validation set was < 0.1% for 10 epochs using a 10 epoch burn-in. Occasionally, due to poor random initialization of weights, the early stopping criteria would be reached before the network started to converge and the resulting network would predict the same trait value for every line. When this was observed in the validation set the training process was repeated starting with new initialized weights.

Convolutional Neural Networks (CNNs) were implemented in Python 3.6 using Tensorflow 2.0. The input layer for the CNNs consisted of the genetic markers for an individual

one-hot-encoded so that each possible allele at each locus was represented as present or absent. Because of the large size of the possible hyperparameter space (Supplemental Table 4.2), a randomized search (using RandomizedSearchCV from Scikit-Learn with 5 folds) was performed on rice for predicting height on one replicate, and the best combination of hyperparameters (lowest average mean squared error) from this one search was used for all other species, traits, and replicates. The input data first passed through a convolutional layer, followed by a maximum pooling layer, a dropout layer, a dense (i.e. fully connected) layer, a batch normalization layer, and finally to the output layer containing one node with the predicted trait value. The EarlyStopping function in Keras (https://keras.io/callbacks/#earlystopping) was used to avoid overfitting (min_delta = 0, patience = 10). To reduce the time and memory requirements, CNN models were trained using a batch size = 100 and run for a maximum of 1,000 epochs. As with ANN models, if the early stopping criteria was reached before the network started to converge, the model would be re-run starting with new initialized weights.

To incorporate predictions from multiple algorithms into one summary prediction, an ensemble approach was used where the ensemble predicted trait value was the mean predicted trait value from 11 algorithms (EN_{11}: rrBLUP, BRR, BA, BB, BL, SVR, SVRpoly, SVRrbf, RF, GTB, ANN) or five algorithms (EN_5: rrBLUP, BL, SVRpoly, RF, ANN). The subset of five consisted of algorithms with differing statistical bases, where rrBLUP represented penalized methods, BL represented the Bayesian approaches, SVRpoly represented non-linear regularized functions, RF represented decision tree based methods, and ANN represented the deep learning approach. This ensemble predicted trait value was then compared to the true trait values to generate performance metrics. A Repeated Measures Analysis of variance (ANOVA)

implemented in R was used to compare model performance, where performance of each model on each replicate test set were considered related.

4.3.3 Hyperparameter grid search using cross-validation

To obtain the best possible results from each algorithm, a grid search approach was used to determine the combination of hyperparameters that maximized performance for each trait/species combination. No hyperparameter needed to be defined for rrBLUP, BL, or BRR. For rrBLUP, the R package estimates the regularization and kernel parameters from the data. For BL or BRR, parameters for these Bayesian regression methods were also estimated from the data. Between one and five hyperparameters were tested for the remaining algorithms (Supplemental Table 4.2).

To avoid biasing our hyperparameter selection, an 80/20 training/testing approach was used, where 20% of the lines were held out from each model as a testing set and the grid search was performed on the remaining 80% of training lines. For RF, SVR_{lin}, SVR_{poly}, SVR_{rbf}, and GTB algorithms, 10 replicates of the grid search were run using the GridSearchCV function from Scikit-Learn with 5-fold cross validation. Ten replicates of the grid search were also run for ANN models, where for each replicate 80% of the training data was randomly selected for training the network with each combination of hyperparameters and the remaining 20% used to select the best combination. This whole process (train/test split, grid search) was replicated 10 times, with a different 20% of lines selected as the test set for each replicate. ANOVA implemented in R was used to determine which hyperparameters significantly impacted model performance for each species.

4.3.4 Assessing Predictive Performance

The predictive performance of the models was compared using two metrics. For the grid search analysis, the mean squared error (MSE) between the predicted ($\hat{Y}$) and the true (Y) trait value was used. For the model comparisons, Pearson correlation coefficient (*r*) between the predicted ($\hat{Y}$) and the true trait value (Y) was used as it is the standard metric for GP performance (Heffner *et al.* 2009; Heslot *et al.* 2012; Riedelsheimer *et al.* 2013). It was computed using the cor() function in R for rrBLUP and the Bayesian approaches or the numpy corrcoef() function in Python for the ML and ANN approaches. Only predicted trait values for lines from the test set were considered when calculating *r*. Summary performance metrics (% of best *r,* rank, variance) were calculated using the mean predictive performance (*r*) across all replicates for each GP algorithm for each species/trait combination.

4.3.5 Feature Selection

The top 10, 50, 100, 250, 500, 1000, 2000, 4000, and 8000 markers were selected using three different feature selection algorithms: Random Forest (RF), Elastic Net (EN), and BayesA (BA). RF and EN feature selection were implemented in Scikit-Learn and BA was implemented in the BGLR package in R. The EN feature selection algorithm requires tuning of the hyperparameter that controls the ratio of the L1- and L2- penalties (e.g. L1:L2 = 1:10 = 0.1). Because the L1 penalty function performs variable selection by shrinking some coefficients to zero, we started with an initial weight on the L1 penalty of 0.1 and then, if fewer than 8,000 markers remained after variable selection, we reduced it in steps of 0.02 until that criteria was met (a 4,000 marker threshold was used for spruce and soy, which only had 6,932 and 4,240 markers available, respectively).

To avoid bias during feature selection, the 80:20 training/testing approach described above was used, where feature selection was performed on the training data and the ultimate performance of models built using the selected markers was scored on the testing set. This was repeated for all 10 testing sets. A repeat measures ANOVA was conducted to compare feature selection algorithms, the number of features selected, and GP algorithms (i.e. independent variables) on model performance (i.e. dependent variable) where replicates were considered repeat measures as they used the same testing set. One-sided, paired Wilcoxon Signed-Rank tests were conducted to determine if model performance (i.e. dependent variable) increased after feature selection (all vs. top 4,000 for soy and spruce, all vs. top 8,000 for other species) (i.e. independent variable). Resulting *p*-values were corrected for multiple testing (*q*-value) (Benjamini and Hochberg 1995).

4.3.6 Initializing ANN starting weights seeded from other GP algorithms

In addition to building ANNs with randomly initialized starting weights, we tested the usefulness of seeding the starting weights with information from other GP algorithms (i.e. rrBLUP, BB, BL, or RF). This is an ensemble-like approach in that it utilizes multiple algorithms to make a final prediction. Ensemble approaches often perform better than single algorithm approaches (Dietterich 2000). First, after the data was divided into training, validation, and testing sets and, for species with large p:n ratios (i.e. maize, rice, sorghum, switchgrass) the top 8,000 markers were selected, we applied a GP algorithm (rrBLUP, BB, BL, or RF) to the training data. From that model we extracted the coefficients/importance scores assigned to each marker and used those as the starting weights for 25% of the nodes in the first hidden layer. We also tested seeding starting weights for 50% of the nodes to predict height in all 6 species but found this significantly increased the model error (MSE) on the validation set (ANOVA; *p*-

value= 0.04), so only results from seeding 25% were included. Because we still needed to reduce the likelihood of vanishing gradients, described above, we manually adjusted the scale of the coefficients/importance scores to match the distribution of the starting weights assigned the remaining 75% of the nodes in the first hidden layer by Xavier Initialization. Finally, to reduce bias in the ANN, random noise was introduced to the seeded nodes by multiplying each starting weight with a random number from a normal distribution with a mean =0 and the standard deviation equal to the standard deviation of weights from Xavier Initialization.

After the training data was used to determine these seeded starting weights, it was used to train the ANN model, the validation set was used to select the best set of hyperparameters and the early stopping point. Then the final trained model was applied to the testing set and performance metrics were calculated. A repeat measures ANOVA was conducted to test if the seeded or the unseeded ANN models (i.e. independent variable) differed in the amount of variation (standard deviation) in model performance across replicates (i.e. dependent variable), with each species acting as a repeat measurement.

4.3.7 Data and Code Availability

For reproducibility, all six datasets along with training/testing designations are available on Dryad (https://doi.org/10.5061/dryad.xksn02vb9) and scripts to run all of the algorithms included in this study on GitHub for future benchmarking. All code used in this study is available on GitHub (https://github.com/ShiuLab/Manuscript_Code/tree/master/2019_GP_Comparison). A README file is included, which provides detailed instructions on how to use the code to generate GP models. Supplemental material available at FigShare (https://doi.org/10.25387/g3.9855590).

4.4 Results

4.4.1. Hyperparameter grid search is critical, particularly among non-linear algorithms

We selected six linear and five non-linear algorithms (note, CNNs are discussed separately) to compare their performance in GP problems (see Methods). While some model parameters can be estimated from the data (de los Campos *et al.* 2013), other parameters, referred to as hyperparameters, have to be user-defined (Chapelle *et al.* 2002; Kuhn and Johnson 2013). This was the case for eight of the algorithms in our study: BA, BB, SVR_{lin}, SVR_{poly}, SVR_{rbf}, RF, GTB, and ANN. For these algorithms we conducted a grid search to evaluate the prediction accuracy of models using every possible combination of hyperparameter values (for lists of hyperparameters, see Supplemental Table 4.2). To produce unbiased estimates of prediction accuracy the grid search was performed within the training set so that no data from the testing set was used to select hyperparameter values. Then we used the best set of hyperparameters from the grid search to build models using genotype and phenotype data from six plant species. This allowed us to compare the predictive performance of all algorithms included in the benchmark datasets.

To determine which hyperparameters significantly impacted model performance, we tested for changes in model performance (mean squared error; MSE) across the hyperparameter space for each algorithm/species/trait combination using Analysis of Variance (ANOVA). The degrees of freedom hyperparameter for BA and BB, both linear algorithms, that influences the shape of the prior density of marker effects (de los Campos *et al.* 2013) had no significant impact on model performance (ANOVA: p-value= 0.41~1.0; Supplemental Table 4.3). Other parameters for the Bayesian algorithms were determined using rules built into the BGLR package that account for factors such as phenotypic variance and the number of markers (p)

(Pérez and de los Campos 2014) and were therefore not considered in our grid search. However, 15 of 16 of the hyperparameters tested for the non-linear algorithms significantly impacted performance in at least one species (Supplemental Table 4.3, Supplemental Figure 4.1A-C). Using height in maize as an example, we found that SVR_{poly} algorithm performed better (i.e. lower MSE) using 2nd degree polynomials compared to using up to 3rd degree polynomials (*p*-value = $1*10^{-21}$, Figure 4.2A). For RF-based models, the maximum depth (max depth) of decision trees allowed significantly impacted performance (*p*-value = $1*10^{-3}$, Supplemental Table 4.3), with shallower trees typically performing better (Figure 4.2B). This pattern was also observed in RF models predicting height for rice, spruce, and soy (*p*-value= $1*10^{-66}$~$5*10^{-4}$, Supplemental Table 4.3, S1B Figure). Because shallower decision trees are less complex, they tend not to overfit, suggesting the best hyperparameters for RF are those that reduce overfitting. The only hyperparameter from the non-linear algorithms that did not impact performance was the rate of dropout (a useful regularization technique to avoid overfitting) for ANN models, where there was no significant change in model performance when two different rates (10% and 50%) were used (*p*-value= 0.24 ~ 0.97, Supplemental Table 4.3).

4.4.2 ANN is the most significantly impacted by hyperparameter choice

Hyperparameters for SVR_{lin}, SVR_{poly}, SVR_{rbf}, RF, and GTB tended to have moderate effects on MSE, while ANN hyperparameters often caused substantial changes in MSE (Figure 4.2A-C; S1A-C Figure). Across the six species, the median variance in MSE across the hyperparameter space for ANN was $6*10^{6}$, but ranged from $3*10^{-3}$- 0.1 for the other GP algorithms (S1D Figure) For example, for predicting height in maize, SVR_{poly} models built using the 2nd degree polynomial outperformed those built using the 3rd degree polynomial with a decrease in MSE ~ 0.05 (Figure 4.2A), while for ANN models, hyperparameter combinations

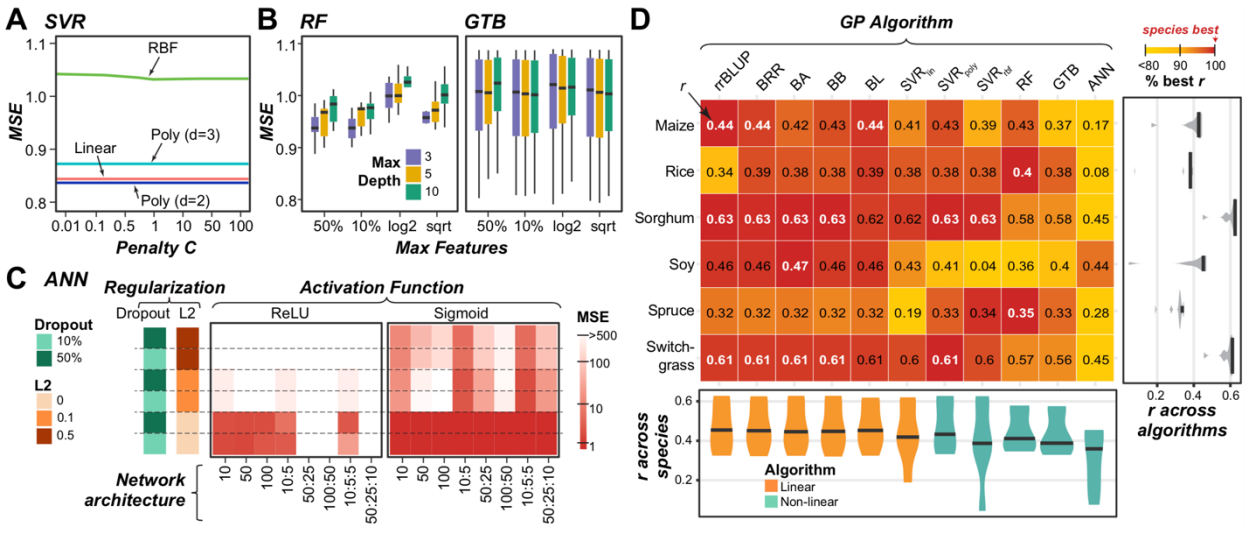


Figure 4.2. Grid search results for height in maize and overall GP algorithm performance for predicting height across species.

(A) Average of mean squared error (MSE) over hyperparameter space (penalty, *C*) for Support Vector Regression (SVR) based models predicting height in maize. SVR_{rbf} and SVR_{poly} results are shown using gamma=$1x10^{-5}$ and $1x10^{-4}$, respectively. Poly: polynomial. RBF: Radial Basis Function. (B) Distribution of MSEs across hyperparameter space for Random Forest (RF; left) and Gradient Tree Boosting (GTB; right) as the maximum features available to each tree (Max Features) and maximum tree depth (color) change. GTB results are shown using a learning rate = 0.01. (C) Average MSE across hyperparameter space for ANN models with different network architectures, degrees of regularization (dropout or L2), using either the Rectified Linear Unit (ReLU; left) or Sigmoid (right) activation function. (D) Mean performance (Pearson's Correlation Coefficient: *r*, text) for predicting height and percent best *r* (colored box, top algorithm for each species = 100% (red)). White text: the best *r* values. Violin-plots show the median and distribution of *r* values for each trait (right) and algorithm (bottom).

that performed the best (i.e. Sigmoid activation function and no L2 regularization) resulted in models with MSEs that were >500 lower than the worst performing model (Rectified Linear Unit (ReLU) activation function, no L2 regularization, and large numbers of hidden nodes; Figure 4.2C). This highlighted that, while hyperparameter selection is necessary for all non-linear algorithms, it is especially critical for building ANNs for GP problems.

Using the best set of hyperparameters for each model, we next compared the predictive performance (Pearson's correlation coefficient, *r*, between predicted and true trait values) of each algorithm on plant height. As with past efforts to benchmark GP algorithms (Heslot *et al.* 2012; Neves *et al.* 2012), no one algorithm always performed the best (white bolded; Figure 4.2D). For example, while rrBLUP performed best for maize, sorghum, and switchgrass, BA performed best for soy, and RF performed best for rice and spruce. Notably, ANNs substantially underperformed compared to other non-linear algorithms, with a median performance at 84% of the best *r* for each of the six species (i.e. 16% below the best performing algorithm for that trait/species). Notably, among the six species, ANN performed the best in soy ($r = 0.44$) relative to the species best algorithm BA ($r = 0.47$, Figure 4.2D). Soy has the largest number of training lines among the six species (5,014) and has a marker to training line ratio close to one (Figure 4.1C). Thus, we hypothesized the poor performance of the ANN models was in part due to our inability to train a network with so many features (markers) and so little training data (lines). During ANN model training, the weights assigned to each connection between nodes in neighboring layers of the network have to be estimated. Because every input marker is connected to every node in the first hidden layer, including more markers in the model will require more weights to be estimated, resulting in a more complex network that is more likely to underfit. In an ideal situation, to account for the complexity in these large networks, five to ten times more instances

(lines) than features (markers) would need to be available for training (Klimasauskas 1993). Alternatively, one can reduce model complexity by only including markers that are most likely to be associated with the trait using feature selection methods.

4.4.3 Feature selection improves performance of ANN models

ANNs and sometimes other non-linear algorithms performed poorly compared to linear methods, which could be due to an insufficient number of training lines relative to the number of markers. To address this, we used feature selection to identify and select the markers most associated with trait variation. Because the number of markers associated with a trait is dependent on the genetic architecture of the trait and is not typically known, models were built using a range of numbers of markers (p = 10~8,000) and were compared to models built using all available markers from each species. Because performing feature selection on the training and testing data can artificially inflate prediction accuracies (Bermingham *et al.* 2015), feature selection was conducted on the training set only. This was repeated 10 times, using a different subset of lines for testing for each replicate (see Methods).

Three feature selection algorithms (RF, BayesA, and Elastic Net (EN)) were compared to predict height in maize, the species with the largest number of markers (p) relative to training lines (n) (p:n = 850, Figure 4.1C). While each algorithm selected a largely different subset of markers (Figure 4.3A, Supplemental Figure 4.2A), the degree of overlap was significantly greater than random expectation. To demonstrate this, we randomly selected three sets of 8,000 maize markers and counted how many markers were present in all three sets 10,000 times and found that the 99th percentile of overlap was equal to 10, however we observed an average of 220 overlapping markers across replicates using these three feature selection approaches. When the different feature selection subsets were used to predict height in maize, there was a significant

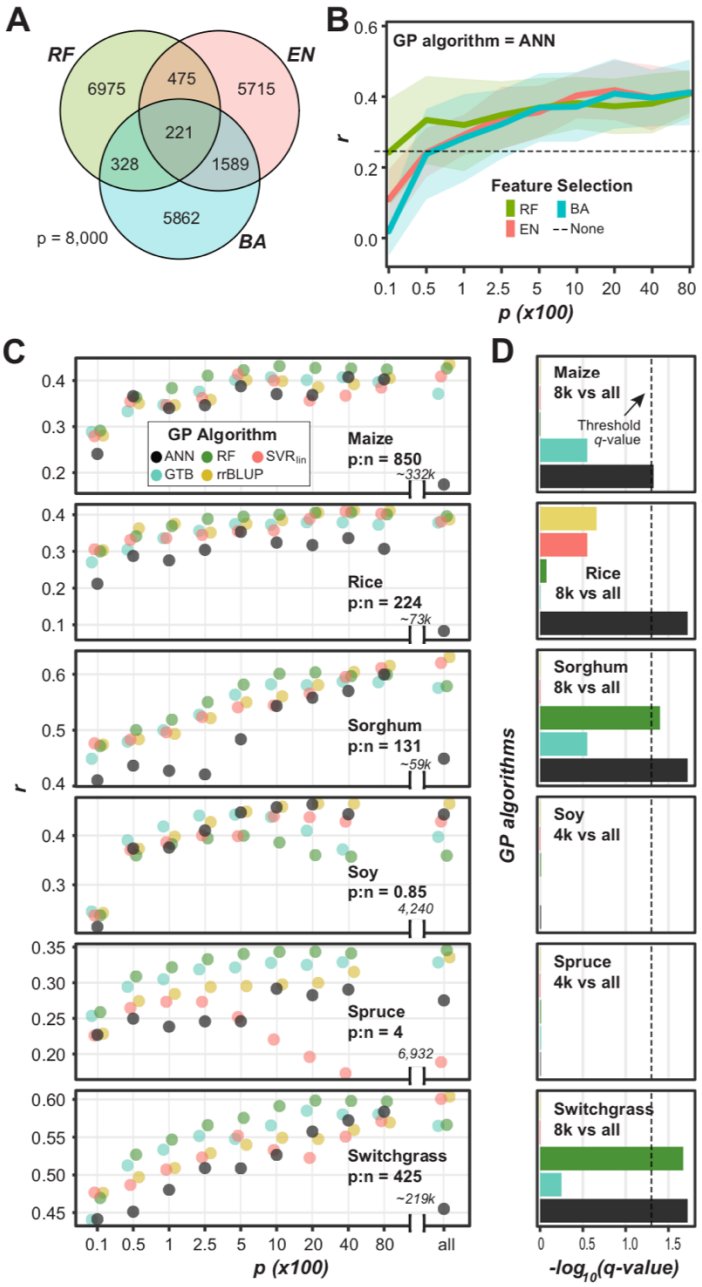


Figure 4.3. Impact of feature selection on GP algorithm performance.

(A) Average number of overlapping markers in the top 8,000 markers selected by three feature selection algorithms for predicting height in maize across ten replicates. EN: Elastic Net. (B) Change in ANN predictive performance (*r*) at predicting height in maize as the number of input markers (p) selected by three feature selection algorithms (BayesA: BA, EN, and Random

Figure 4.3 (cont'd)

Forest: RF) increases. Dashed line: mean *r* when all 332,178 maize markers were used. (C) Mean *r* of rrBLUP, SVR_{lin}, RF, GTB, and ANN models for predicting height using subsets or all (X-axis) markers as features across 10 replicate feature selection and ML runs for each of six species with their ratios of numbers of markers (p) to numbers of lines (n) shown. Data points were jittered horizontally for ease of visualization. (D) The significance ($-\log_{10}$(*q*-value), paired Wilcoxon Signed-Rank test test) of the difference in *r* between models from different GP algorithms (colored as in Figure 4.3C) generated using a subset of 4,000 or 8,000 and all markers as input. Dotted line designates significant differences (*p*-value < 0.05).

interaction between the number of available markers (p) and the feature selection method (repeat measures ANOVA: *p*-value = $1.7*10^{-12}$). Exploring this interaction further, we found that, while feature selection algorithms performed similarly with large n, RF tended to perform the best when fewer markers were selected for GP (Figure 4.3B; Supplemental Figure 4.2B) and was therefore used to test the impact of feature selection on predicting height in the other five species.

For species with a low p:n ratio (i.e. soy and spruce), for all GP algorithms tested, as p increased the model performance tended to increase continuously (e.g. all GP algorithms in sorghum) or, in some cases, the model performance reached a maximum (or a plateau) quickly (e.g. in soy after 2,500 markers were used) (Figure 4.3C). For these species, there was no significant improvement in performance after feature selection (all vs. top 4,000) using any GP algorithm (one-sided, paired Wilcoxon Signed-Rank test: *q*-value = 0.98 ~ 0.99; Figure 4.3D).

For example, ANNs built using all 6,932 spruce markers performed no better than those built using the top 4,000 markers (p-value= 0.98).

For species with a large p:n ratio (i.e. maize, rice, sorghum, and switchgrass), a similar pattern was observed for rrBLUP, SVR_{lin}, and GTB, where performance increased or reached a plateau as p increased and no significant improvement in performance was found after feature selection (p=8,000) (q-value = 0.28 ~ 0.99; Figure 4.3D). However, for these four species, feature selection improved the performance of ANN models (q-value= 0.019 ~ 0.047; Figure 4.3D). For example, after feature selection prediction of height in maize using ANNs improved from r=0.17 to 0.41, a 141% increase. Ultimately, performing feature selection prior to ANN training for these four datasets with large p:n ratios, improved ANN performance (median r at 89% of the best r for each of the six species) compared to ANNs without feature selection (84% of the best r). Therefore, for the GP benchmark analysis, feature selection was performed prior to model building for additional traits for maize, rice, sorghum, and switchgrass and the top 8,000 markers were used. Because feature selection only improved the performance of RF models in sorghum and switchgrass, we did not perform feature selection before training RF models in the full benchmark study.

While feature selection notably improved ANN performance, ANNs still often underperformed compared to other GP algorithms (Figure 4.3C), meaning the they were unable to learn even the linear relationships between markers and traits that were found using the linear-based algorithms. Because ANNs should theoretically at least match the performance of linear algorithms, this suggests that the ANN hyperparameters are not optimal. Furthermore, we found that, even after feature selection, there was greater variation in performance across replicates for ANN models compared to rrBLUP, SVR_{lin}, RF, and GTB (S2C-D Figure), indicating the ANN

models did not always converge on the best solution. One potential reason for the is that the final trained network can be heavily influenced by the initial weights used in ANN, which are selected randomly. In addition, while random weight initialization, a procedure we have used thus far, reduces bias in the network, it can also result in some networks converging on a local, rather than global, optimal solution.

4.4.4 Non-random initialization of ANN starting weights and convolutional layers improve ANN performance for some species

To reduce the likelihood of ANNs converging to locally optimal solutions, we developed an approach that allowed the ANNs to utilize the relationship between markers and traits determined by another GP algorithm. In this approach, a GP algorithm was applied to the training lines, and the coefficient or importance score assigned to each marker from this algorithm was used to seed the starting weights (Figure 4.4A). Four GP algorithms were tested to seed the weights: rrBLUP, BB, BL, and RF (referred to as ANN_{rrBLUP}, ANN_{BB}, ANN_{BL}, and ANN_{RF}, respectively). Because this approach could predispose the networks to only learn the relationship already identified by the seed algorithm, two steps were taken to re-introduce randomness into the network (see Methods). First, the seeded approach was only used to initialize starting weights for 25% of the nodes in the first hidden layer, while connection weights to the remaining 75% of nodes were initialized randomly as before. Second, noise was infused into the starting weights for the 25% of nodes that were seeded.

Applying this approach to predict plant height we found that ANN performance improved for three of six species (Figure 4.4B). For example, the average performance for rice without seeding (ANN) was $r = 0.25$ and with seeding from BL (ANN_{BL}) was $r = 0.32$, a 28% improvement, while for sorghum, ANN_{BL} had <0.1% improvement over the original ANN

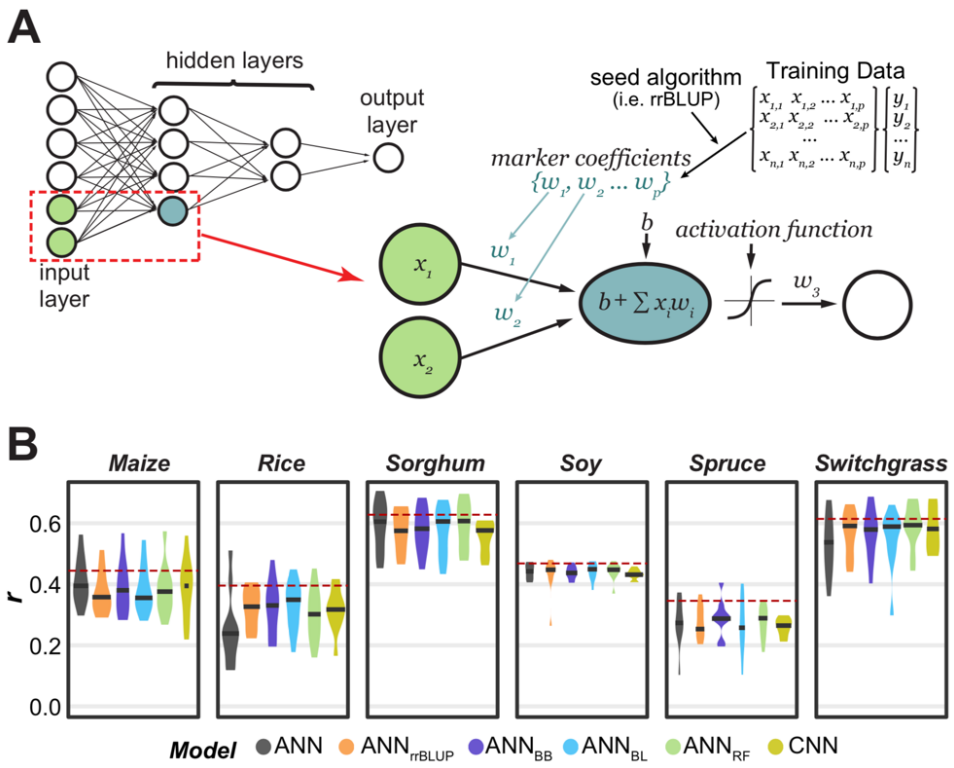


Figure 4.4. Description and performance results of the seeded ANN approach.

(A) An overview of the seeded ANN approach. The network in the top left is an example of a fully connected ANN with 6 input nodes (i.e. 6 markers), two hidden layers, and one output layer (i.e. predicted trait value). The blue node in the first hidden layer represents an example node that will have seeded weights. For this node, the weights (*w*) connecting each input node to the hidden node will be seeded from the coefficient/importance for each marker as determined by another GP algorithm using the training data. *b*: bias, which helps control the value at which the activation function will trigger. (B) The distribution of model performance (*r*) using only all random (None), 25% seeded (rrBLUP, BayesB, BL, RF) weight initialization, and convolutional neural networks (CNN). The mean performance of the overall top performing algorithm (i.e. not necessary ANN) shown as dotted red line.

methods. Seeding ANN models did not significantly reduce the amount of variation in model performance across replicates (repeated measure ANOVA: *p*-value= 0.39, Supplemental Table 4.4). Ultimately, seeded ANN models had a median performance between 89% - 90% of the best *r* for each species (compared to 89% with random initialization, Figure 4.4B). While this represented only a moderate improvement, we included the seeded ANN approach in the benchmark analysis because of how substantial the improvement was for some species (i.e. rice).

Another deep learning strategy for reducing the complexity of GP problems and consequently decreasing the likelihood of converging on local optimum is to use convolutional and pooling layers to summarize local patterns of genetic markers and learn from these summaries (Ma *et al.* 2018). We tested this approach by training Convolutional Neural Networks (CNNs) to predict plant height (S3A Figure). Notably, feature selection (n= 8,000) had either no or a negative impact on CNN performance. For example, the average performance of CNNs at predicting height in maize, the species with the most genetic markers, was $r = 0.39$, but dropped to $r = 0.37$ after feature selection. CNNs performed better than ANNs at predicting height in two of six species (yellow; Figure 4.4B), with the biggest improvement in rice where the average performance increased from $r = 0.25$ using ANNs to $r = 0.32$ using CNNs, a 32% improvement. While CNN models did not reduce the amount of variation in model performance across replicates (repeated measure ANOVA: *p*-value = 0.08, Supplemental Table 4.4), we included CNNs in the final benchmark analysis because of the promising results in rice and switchgrass.

4.4.5 No one GP algorithm performs best for all species and traits

Having established best practices for hyperparameter and feature selection for our datasets, we next compared the performance of all GP algorithms for predicting three traits in each of the six species. For maize, rice, and soy, these traits included height, flowering time, and yield (Figure

4.1C). For species where data was not available for one or more of these traits, other traits were used (see the panel labeled "Others", Figure 4.5A). As with past efforts to benchmark GP algorithms (Heslot *et al.* 2012; Neves *et al.* 2012), different algorithms performed best for different species/trait combinations (Figure 4.5A; Supplemental Table 4.5). Thus, we utilized the predictive power of multiple algorithms to establish an ensemble prediction using all (except CNN: EN_{11}) or a subset of five (EN_5) algorithms (see Methods). The ensemble models consistently performed well, with EN_5 or EN_{11} being the best (three) or tied for the best (nine) algorithm for 12 of the 18 species/trait combinations included in the benchmark and had a median performance rank of 3 (Figure 4.5B; Supplemental Table 4.6). For the remaining 6 species/trait combinations where EN_5 or EN_{11} weren't among the best performers, they tended to perform only slightly worse (median % of best r = 99.2%, Figure 4.5A). This suggests that ensemble-based predictions are more stable and more likely to result in better trait predictions than a single algorithm.

Focusing on the species/trait combinations where one of the non-ensemble algorithms was or tied for best, we found that a linear algorithm performed best for five of the species/trait combinations, a non-linear algorithm performed best for four species/trait combinations, and both a linear and a non-linear algorithm performed equally well for the remaining six species/trait combinations (Figure 4.5B). This finding suggests that linear and non-linear algorithms are equally well suited for GP. The linear algorithms BRR and BA performed best overall, being among the top performers for 9 and 8 traits, respectively, and with the top two median ranks of five and 4.5, respectively (Supplemental Table 4.6). The top performing non-linear algorithm was SVR_{poly}, which was among the top performers for 8 traits and had a median rank of 6. There was notably greater performance variation across species/traits for non-linear

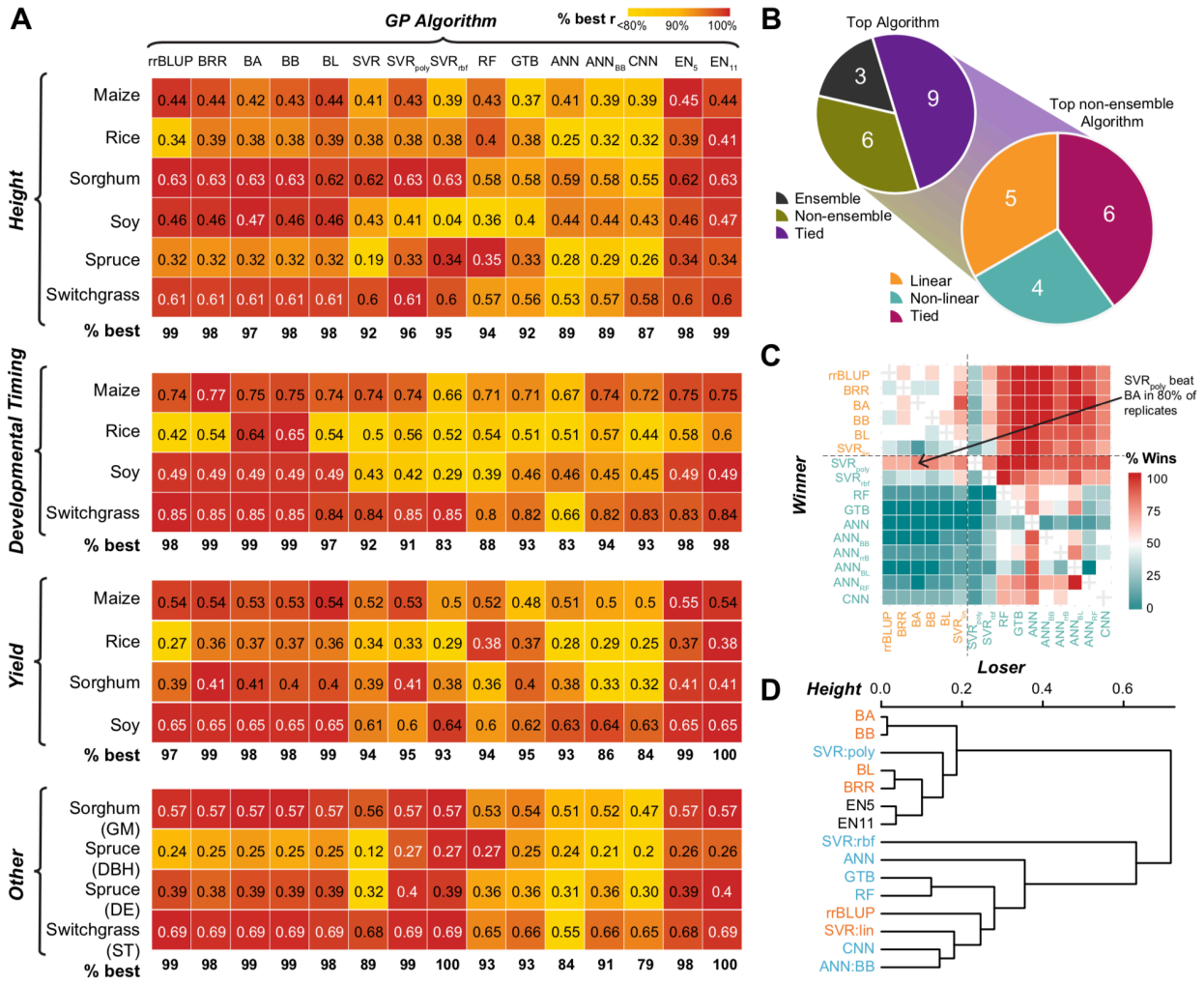


Figure 4.5. Comparison of algorithms for predicting additional traits.

(A) Mean model performance (*r;* text) for each species/trait combination (y-axis) for each GP algorithm (x-axis). White text: *r* of the best performing algorithm(s) for a species. Colored boxes: percent of best performance (*r*) for a species, with the top algorithm for each species = 100% (red). The median % of best performance for each GP algorithm for each type of trait (i.e. height, developmental timing, yield, other) is shown below each heatmap. GM: sorghum grain moisture. DBH and DE: diameter at breast height and wood density, respectively, for spruce. ST: standability for switchgrass. (B) Top left: summary of the number of species/trait combinations that were predicted best by an ensemble (gray) or a non-ensemble model (yellow), or predicted

Figure 4.5 (cont'd)

equally well by both (purple). Bottom right: among non-ensemble models that performed or tied for the best, the number of species/trait combinations that were predicted best by a linear (blue) or a non-linear model (green) or predicted equally well by both (orange). (C) Percent of replicates where one GP algorithm (y-axis, winner) outperformed another GP algorithm (x-axis, loser) for predicting height in switchgrass. Orange and cyan texts: linear and non-linear algorithms, respectively. (D) Hierarchical clustering of GP algorithms based on mean predictive performance across all species/trait combinations. Algorithm colored as in (C).

algorithms (mean variance = 1.03%) compared linear algorithms (mean variance = 0.65%) (Supplemental Table 4.6). For example, SVR_{rbf} performed poorly at predicting developmental timing traits (median 83% of the best *r*), however it had or was tied for the best prediction for three of the four "other" traits (median 100% of the best *r*) (Figure 4.5A). Results from ANN models using randomly initialized (ANN) and BB seeded (ANN_{BB}) weights are shown because ANN_{BB} had the best performance of the seeded ANN models (see S5, Supplemental Table 4.6 for results from other seeded ANNs). Notably, none of the randomly initialized ANN (median rank = 13.5), the ANN_{BB} (median rank = 13), or the CNN (median rank = 15.5) models performed best for any trait (Supplemental Table 4.6).

One limitation of comparing the mean score or performance rank is that small but consistent differences in model performance could be missed. To account for this, we also calculated the number of times an algorithm outperformed another algorithm for each trait across the replicates. Using this metric, we were able to identify algorithms that consistently outperformed others for a given trait/species combination (Figure 4.5C, Supplemental Figure

4.4). We frequently observed that linear algorithms had higher win percentages than nonlinear algorithms, this was the case for all three traits in maize and soybean for example (S4 Figure). However, there were plenty of exceptions. RF and SVR_{rbf} had higher win percentages than linear algorithms for predicting height and diameter at breast height (DBH) in spruce and ANN_{BB} had a higher win percentage than all algorithms except BA and BB for predicting flowering time in rice (S3 Figure). In a few cases, assessing win percentages allowed us to identify winners when mean predictive performance (*r*) was tied. For example, for predicting height in switchgrass. SVR_{poly} had the same average performance ($r = 0.61$) as multiple of the linear algorithms (i.e. rrBLUP, BA, etc.), however, it outperformed those algorithms in 70-80% of replicates (Figure 4.5C).

In order to determine which algorithms perform similarly, we performed hierarchical clustering of the algorithms based on their performance across the 18 species/trait combinations (from Figure 4.5A). Interestingly, linear and non-linear algorithms did not clearly separate from each other (Figure 4.5D). For example, rrBLUP and SVR_{lin} were more similar to the neural network based models (i.e. CNN and ANN_{BB}), than they were to the linear Bayesian algorithms (i.e. BA, BB, BL, and BRR). Notably, while the Bayesian algorithms tended to cluster together closely performance-wise, the non-linear algorithms tended to have a greater distance between them. Finally, in order to identify if algorithm performance was similar for specific types of traits (e.g. whether similar algorithms perform well at predicting traits related to developmental timing) or across species/population composition (e.g. whether similar algorithms perform well on diversity panels), we performed hierarchical clustering of each species/trait based on performance of all 14 algorithms (from Figure 4.5A). Surprisingly, species/trait combinations with similar patterns of algorithm performance were often not the same species, trait, or

population type (Supplemental Figure 4.5), suggesting that we cannot generalize easily the differences in performance based on species, trait, or population type.

4.5 Discussion

We conducted a benchmarking comparison of GP algorithms on 18 species/trait combinations that differ in the type and size of the training data set and of the marker data available. Similar to previous GP algorithm benchmark studies conducted on smaller datasets (Heslot *et al.* 2012; Blondel *et al.* 2015), a key result from this analysis is that no one model performs best for all species and all traits. We further demonstrate that, while similar algorithms perform similarly across the 18 species/trait combinations, algorithm performance was not clearly related to the trait type or population composition. With that said, linear algorithms tend to perform consistently well, while the performance of non-linear algorithms varied widely by trait. Studies of gene networks have shown that non-additive interactions (e.g. epistasis, dominance) are important for development and regulation of complex traits (Holland 2007; Monir and Zhu 2018). One may expect approaches that can consider non-linear combinations would therefore be better suited for modeling complex trait. This was not the case and we found the inconsistency of non-linear algorithms surprising.

We have three, non-mutually exclusive, explanations for why linear algorithms often outperform non-linear algorithms. First, the traits included in this study vary in their genetic architecture (i.e. the number and distribution of allele effects), therefore we may be observing that linear algorithms outperform non-linear algorithms when the trait has a predominantly additive genetic basis. Second, there is evidence that even highly complex biological systems generate allelic patterns that are consistent with a linear, additive genetic model because of the discrete nature of DNA variation and the fact that many markers have extreme allele frequencies

(Hill *et al.* 2008). The proportion of dominance and epistatic variance that can be captured by an additive (i.e. linear) model increases when allele frequencies are extreme (Hill *et al.* 2008). This phenomenon is even more important with inbred lines (e.g. soy and rice); where, at each locus there are only 2 possible variants (e.g. AA and TT); thus, the additive model fully captures the single-locus genetic variance. However, the fraction of epistatic variance that can be captured by an additive model depends on how many multi-locus genotypes are present in the data and this depends on allele frequencies. Thus, the distribution of allele frequency (which due to mutation, selection, and drift is often enriched at extreme values) is one of the reasons why additive models often capture and perform very well at predicting traits that at the biological level are affected by complex epistatic networks. Finally, a third explanation is that the amount of training data available for most GP problems was insufficient for learning non-linear interactions between large numbers of markers, therefore the linear models, which focus on modeling linear relationships, outperform the non-linear models.

Three findings from our study suggest that limited training data plays a role. First, we found that non-linear algorithms performed better at predicting traits in species with a small marker number to population size (p:n) ratio. For example, RF, SVR_{poly}, and SVR_{rbf} performed best at predicting traits in spruce and ANN models tended to perform better at predicting traits in soy, the species with the second smallest and smallest p:n, respectively. Second, the ANN models significantly improved after feature selection. This was not the case for other algorithms in our study or with previous efforts to use feature selection for GP (Vazquez *et al.* 2010; Bermingham *et al.* 2015). For example, for predicting traits in Holstein cattle, the top 2,000 markers had only 95% of the predictive ability of all the markers using BL (Vazquez *et al.* 2010). With a fixed training data size, prediction accuracy is a function of how much genetic

variation is captured by markers in linkage disequilibrium with quantitative trait loci and the accuracy of the estimated effects (Goddard 2009). Because feature selection removes markers from the model, such decreases in performance after feature selection for non-ANN models are likely due to the reduction in the amount of genetic variation captured without a subsequent increase in the accuracy of the estimated effects. However, we hypothesize that feature selection significantly improved performance for ANNs because it improved the accuracy of the estimated effects (i.e. the connection weights) more than it reduced the amount of genetic variation captured. Third, ANNs that have been trained on small datasets often have unstable performance likely because ANNs are sensitive to the initialized weight values when they do not have enough training data to learn from (LeBaron and Weigend 1998; Shaikhina and Khovanova 2017). We observed greater instability in performance across replicates for ANNs compared to other algorithms (**S2C-D Figure**), suggesting that our ANN models may have benefitted from additional training data.

However, a recent study involving large sample size (n~80,000) in humans compared linear models with two types of ANN algorithms, multilayer perceptron and convolutional neural networks, and did not find any clear superiority of the ANN methods relative to linear models, if anything the linear model offered higher predictive power than the ANNs (Bellot *et al.* 2018). While they also found that feature selection improved the performance of their ANN models, using the top 10k of the 50k markers, these models still did not outperform the linear models (Bellot *et al.* 2018). Given that these results are from a single study in humans, we believe it will be informative to benchmark ANNs on a larger crop dataset in the future.

While there is a great deal of excitement about the uses of deep learning in the field of genetics, there is still much work to be done to improve performance of deep learning-based models. In

this study we identified dimensionality as a major limitation to training ANNs for GP. Additional areas of deep learning research also need to be further explored. For example, in this study we limited the ANN hyperparameter space searched because the grid search method was too computationally intensive to be more thorough. Because changes in hyperparameters had a large impact on model performance, further hyperparameter tuning could lead to better performing models. For example, we limited our search to include nine possible network architectures with between one and three hidden layers each containing between 5-100 nodes (**Supplemental Table 4.1**), but it is possible that ANNs with different network architectures, such as more hidden layers, or different combinations of layer sizes, could have performed better. Similarly, given that the hyperparameter space for CNN models was only tested for one species and trait (height in rice), it is likely that model-specific hyperparameter selection could improve the performance of CNN models beyond what we were able to achieve here.

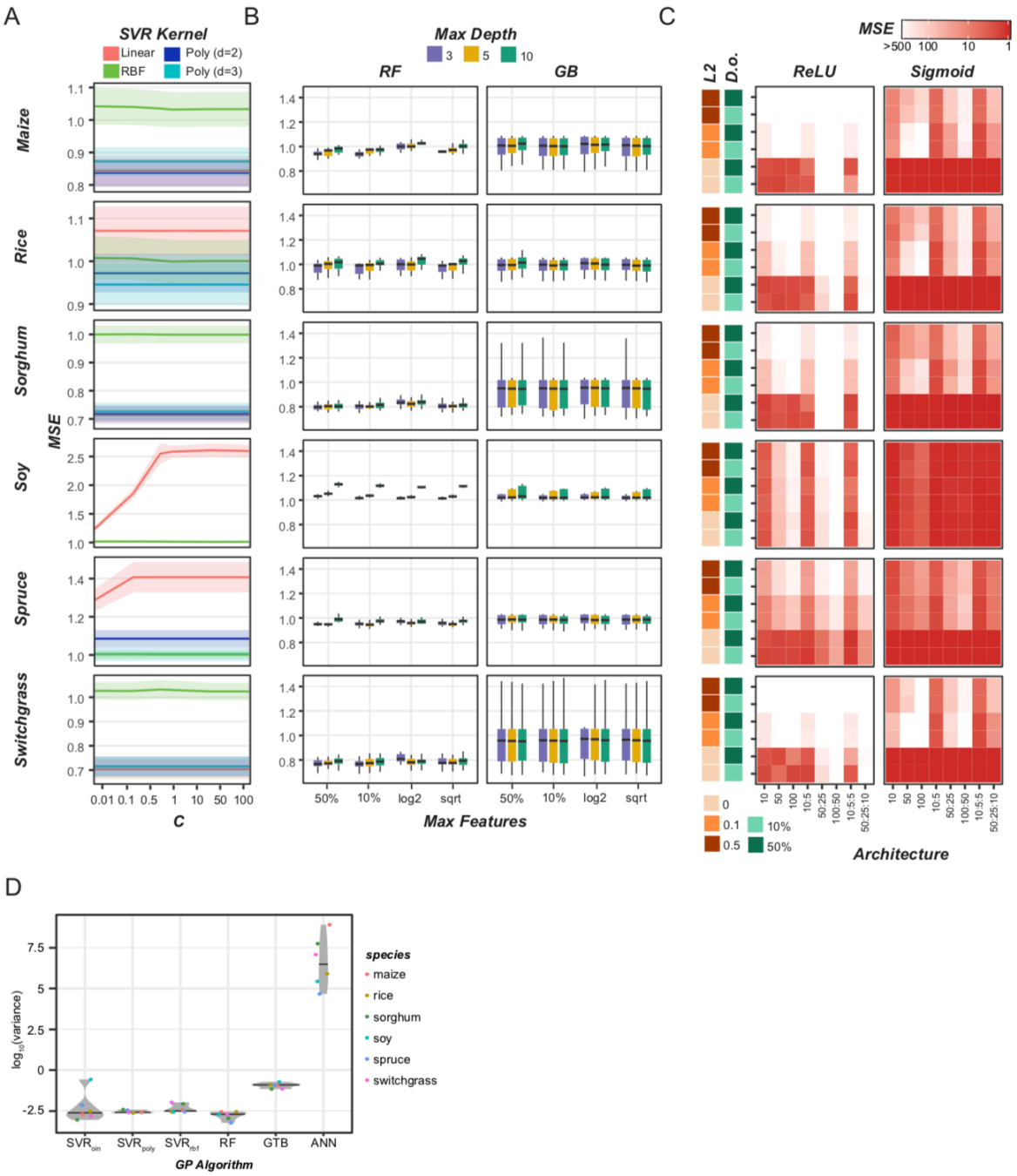
In summary, we provided a thorough comparison of 12 GP algorithms and two ensembles for predicting diverse traits in six plant species with a range of marker types and numbers and population types and sizes. We found that no GP algorithm was best for all species/trait combinations and that trait type or population type were not closely associated with which algorithms worked best. While neural network approaches did not tend to outperform linear or other non-linear models, strategies to tailor neural networks for GP problems (e.g. non-random initialization of stating weights, convolutional and pooling layers) show promise. Unlike previous GP algorithm benchmark studies (Heslot *et al.* 2012), we found that the performance of ensemble models, generated by combining predictions from multiple individual GP algorithms, consistently tied with or exceeded the performance of the best individual algorithm. Taken together, these finds lead us to recommend that breeders test the performance of multiple

algorithms on their training population to identify which algorithm or combination of algorithms performs best for traits important to their breeding program.

4.6 Acknowledgements

We thank Peipei Wang and John Lloyd from the Shiu lab, Gabriel Rovere from the MSU QuantGen group, and Fouad Bahrpeyma from the Insight Center for their valuable suggestions to our project. This work was supported by the National Science Foundation (NSF) Graduate Research Fellowship [Fellow ID: 2015196719], Graduate Research Opportunities Abroad (GROW) Fellowship to C.B.A.; NSF PlantGenomics Research Experiences for Undergraduate to E.B.; the U.S. Department of Energy Great Lakes Bioenergy Research Center [BER DE-SC0018409] and National Science Foundation [IOS-1546617, DEB-1655386] to S.-H.S.; and the National Institute of Health [R01GM099992, R01FM101219] to G.D.L.C..

APPENDIX

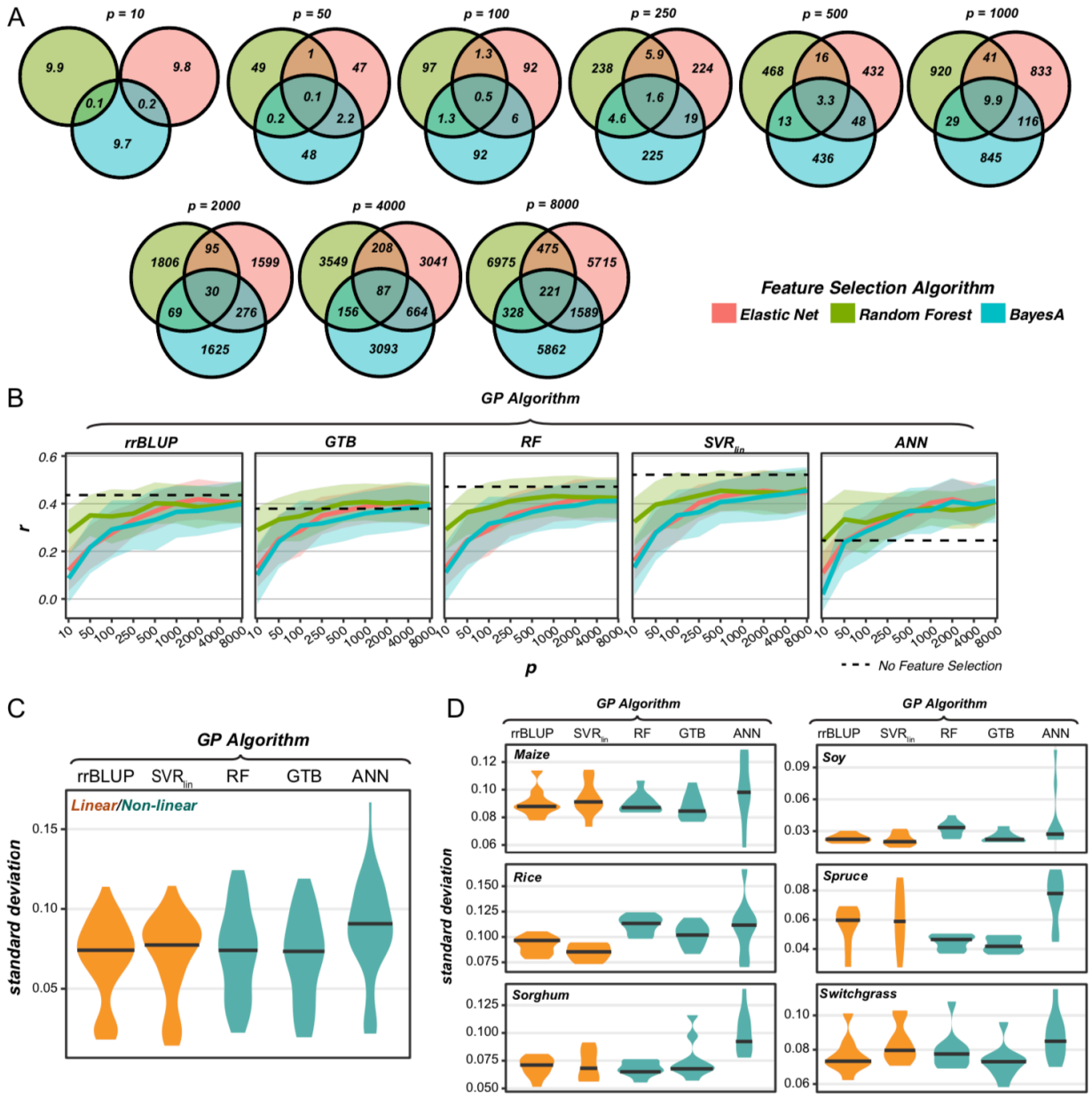


Supplemental Figure 4.1. Height prediction performance for non-linear GP algorithms during hyperparameter grid search.

(A) Average (line) and standard deviation (shadow) of mean squared error (MSE) over hyperparameter space for SVR based models predicting height as the penalty (*C*) (X-axis) change. SVR_{rbf} and SVR_{poly} results are shown using gamma=$1x10^{-5}$ and $1x10^{-4}$, respectively. (B)

Supplemental Figure 4.1 (cont'd)

Distribution of the MSE across hyperparameter space for RF (left) and GTB (right) as the maximum features available to each tree (Max Features; X-axis) and maximum tree depth (color) change. GTB results are shown using a learning rate = 0.01. (C) Average MSE across hyperparameter space for ANN models with different network architectures (X-axis), degrees of regularization using dropout (D.o.) or L2 regularization (L2), using either the Rectified Linear Unit (ReLU; left) or Sigmoid (right) activation function. (D) Distribution of the variance in MSE across the hyperparameter space for predicting height in each species using each GP algorithm. Black bar represents the median variance across the species for each GP algorithm.

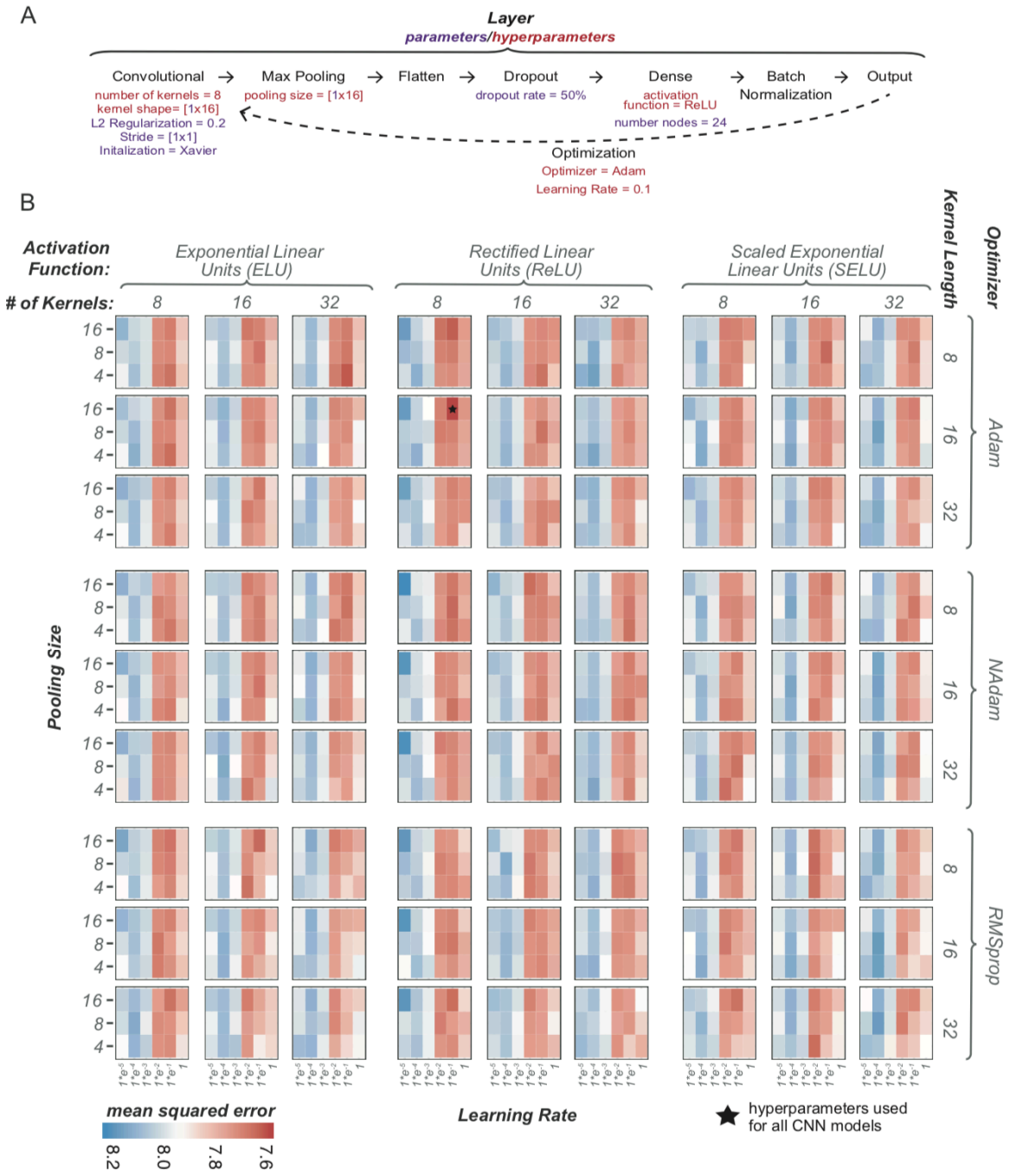


Supplemental Figure 4.2. Comparison of feature selection algorithms and change in performance variation after feature selection.

(A) Average number of overlapping markers in the top markers (p) selected by three different feature selection algorithms for predicting height in maize across ten replicates for p=10 ~ 8,000. (B) Change in model performance (*r*) using five GP algorithms at predicting height in maize as the number of input markers (p) selected by three different feature selection algorithms increases.

Supplemental Figure 4.2 (cont'd)

Dashed line: the mean *r* for each GP algorithm when all maize markers were used. Colored lines: mean *r* of models using features selection subsets using algorithms colored as in (A). Colored areas: standard deviation around the mean. (C) Distribution and median of the standard deviation of model performance (*r*) across replicates for all feature selection subsets (p=10 ~ 8,000) combined across all species for each GP algorithm (D) Distribution and median of the standard deviation of model performance across replicates for all feature selection subsets (p=10 ~ 8,000) by species for each GP algorithm.



Supplemental Figure 4.3. Hyperparameter random search results from predicting height in spruce.

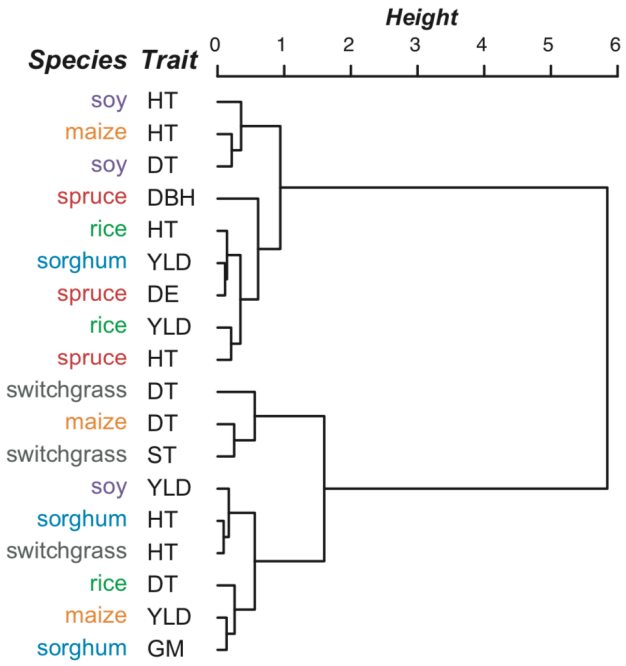
(A) Overview of the architecture and parameters used to train the CNN models. The parameters listed below for each layer (black) were either pre-set (value shown in purple) or the value for

Supplemental Figure 4.3 (cont'd)

that parameter was selected using RandomSearchCV (values tested shown in red). (B) Average mean squared error (MSE) across the hyperparameter space for predicting height in rice (replicate #1). RMSprop: Root Mean Square propagation.

Supplemental Figure 4.4. Number of wins between each pair of GP algorithm.

Percent of replicates where one GP algorithm (y-axis) outperformed another GP algorithm (x-axis) for predicting each species/trait combination.



Supplemental Figure 4.5. Similarity between traits and datasets in model performance. Hierarchical clustering of trait:species combinations based on mean predictive performance across all algorithms included in the benchmark. HT: height. DT: developmental timing. YLD: yield, GM: grain moisture. DBH: diameter at breast height. DE: wood density. ST: standability.

REFERENCES

REFERENCES

Angermueller, C., T. Pärnamaa, L. Parts, and O. Stegle, 2016 Deep learning for computational biology. Molecular Systems Biology 12: 878–16.

Beaulieu, J., T. K. Doerksen, J. MacKay, A. Rainville, and J. Bousquet, 2014 Genomic selection accuracies within and between environments and small breeding groups in white spruce. BMC Genomics 15: 1048.

Bellot, P., G. de los Campos, and M. Pérez-Enciso, 2018 Can Deep Learning Improve Genomic Prediction of Complex Human Traits? Genetics genetics.301298.2018.

Benjamini, Y., and Y. Hochberg, 1995 Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. J. R. Stat. Soc. Series B Stat. Methodol. 57: 289–300.

Bermingham, M. L., R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan *et al.*, 2015 Application of high-dimensional feature selection: evaluation for genomic prediction in man. Scientific Reports 1–12.

Blondel, M., A. Onogi, H. Iwata, and N. Ueda, 2015 A Ranking Approach to Genomic Selection. PLoS ONE 10: e0128570–23.

Breiman, L., 2001 Random Forests. Machine Learning 45: 5–32.

de los Campos, G., D. Gianola, G. J. M. Rosa, K. A. Weigel, and J. Crossa, 2010 Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. Genet. Res. 92: 295–308.

de los Campos, G., J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, and M. P. L. Calus, 2013 Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding. Genetics 193: 327–345.

de los Campos, G., H. Naya, D. Gianola, J. Crossa, A. Legarra *et al.*, 2009 Predicting quantitative traits with regression models for dense molecular markers and pedigree. Genetics 182: 375–385.

Chapelle, O., V. Vapnik, O. Bousquet, and S. Mukherjee, 2002 Choosing Multiple Parameters for Support Vector Machines. Mach. Learn. 46: 131–159.

Desta, Z. A., and R. Ortiz, 2014 Genomic selection: genome-wide prediction in plant improvement. Trends in Plant Science 19: 592–601.

Dietterich, T. G., 2000 Ensemble methods in machine learning. International workshop on multiple classifier systems.

Ehret, A., D. Hochstuhl, D. Gianola, and G. Thaller, 2015 Application of neural networks with back-propagation to genome-enabled prediction of complex traits in Holstein-Friesian and German Fleckvieh cattle. Genet. Sel. Evol. 47: 22.

Endelman, J. B., 2011 Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome.

Evans, J., E. Crisovan, K. Barry, C. Daum, J. Jenkins *et al.*, 2015 Diversity and population structure of northern switchgrass as revealed through exome capture sequencing. Plant J. 84: 800–815.

Evans, J., M. D. Sanciangco, K. H. Lau, E. Crisovan, K. Barry *et al.*, 2017 Extensive Genetic Diversity is Present within North American Switchgrass Germplasm. Plant Genome.

Fernandes, S. B., K. O. G. Dias, D. F. Ferreira, and P. J. Brown, 2017 Efficiency of multi-trait, indirect, and trait-assisted genomic selection for improvement of biomass sorghum. Theor. Appl. Genet.

Friedman, J. H., 2001 Greedy function approximation: A gradient boosting machine. Ann. Statist. 29: 1189–1232.

Gianola, D., R. L. Fernando, and A. Stella, 2006 Genomic-assisted prediction of genetic value with semiparametric procedures. Genetics 173: 1761–1776.

Girija, S. S., 2016 Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Glorot, X., and Y. Bengio Understanding the difficulty of training deep feedforward neural networks. 2010.

Goddard, M., 2009 Genomic selection: prediction of accuracy and maximisation of long term response. Genetica 136: 245–257.

González-Camacho, J. M., G. de los Campos, P. Pérez, D. Gianola, J. E. Cairns *et al.*, 2012 Genome-enabled prediction of genetic values using radial basis function neural networks. Theoretical and Applied Genetics 125: 759–771.

González-Camacho, J. M., J. Crossa, P. Pérez-Rodríguez, L. Ornella, and D. Gianola, 2016 Genome-enabled prediction using probabilistic neural network classifiers. BMC Genomics 1–16.

González-Camacho, J. M., L. Ornella, P. Pérez-Rodríguez, D. Gianola, S. Dreisigacker *et al.*, 2018 Applications of Machine Learning Methods to Genomic Selection in Breeding Wheat for Rust Resistance. Plant Genome 11:.

González-Recio, O., and S. Forni, 2011 Genome-wide prediction of discrete traits using Bayesian regressions and machine learning. Genet. Sel. Evol. 43: 7.

González-Recio, O., J. A. Jiménez-Montero, and R. Alenda, 2013 The gradient boosting algorithm and random boosting for genome-assisted evaluation in large data sets. J. Dairy Sci. 96: 614–624.

Habier, D., R. L. Fernando, K. Kizilkaya, and D. J. Garrick, 2011 Extension of the bayesian alphabet for genomic selection. BMC Bioinformatics 12: 186.

Hansey, C. N., J. M. Johnson, R. S. Sekhon, S. M. Kaeppler, and N. de Leon, 2011 Genetic diversity of a maize association population with restricted phenology. Crop Sci. 51: 704–715.

Heffner, E. L., M. E. Sorrells, and J.-L. Jannink, 2009 Genomic Selection for Crop Improvement. Crop Science 49: 1–12.

Heslot, N., H.-P. Yang, M. E. Sorrells, and J.-L. Jannink, 2012 Genomic Selection in Plant Breeding: A Comparison of Models. Crop Science 52: 146–15.

Hill, W. G., M. E. Goddard, and P. M. Visscher, 2008 Data and theory point to mainly additive genetic variance for complex traits. PLoS Genet. 4: e1000008.

Hirsch, C. N., J. M. Foerster, J. M. Johnson, R. S. Sekhon, G. Muttoni *et al.*, 2014 Insights into the maize pan-genome and pan-transcriptome. Plant Cell 26: 121–135.

Holland, J. B., 2007 Genetic architecture of complex traits in plants. Curr. Opin. Plant Biol. 10: 156–161.

Jonas, E., and D.-J. de Koning, 2013 Does genomic selection have a future in plant breeding? Trends Biotechnol. 31: 497–504.

Kasnavi, S. A., M. A. Afshar, M. M. Shariati, N. E. J. Kashan, and M. Honarvar, 2017 Performance evaluation of support vector machine (SVM)-based predictors in genomic selection. Indian J. Anim. Sci. 87: 1226–1231.

Kingma, D. P., and J. Ba, 2014 Adam: A Method for Stochastic Optimization.

Klimasauskas, C. C., 1993 Applying neural networks. Neural networks in finance and investing 47–72.

Kuhn, M., and K. Johnson, 2013 *Applied Predictive Modeling*.

LeBaron, B., and A. S. Weigend, 1998 A bootstrap evaluation of the effect of data splitting on financial time series. IEEE Transactions on Neural Networks 9: 213–220.

Lipka, A. E., F. Lu, J. H. Cherney, E. S. Buckler, M. D. Casler *et al.*, 2014 Accelerating the Switchgrass (Panicum virgatum L.) Breeding Cycle Using Genomic Selection Approaches. PLoS ONE 9: e112227–7.

Long, N., D. Gianola, G. J. M. Rosa, and K. A. Weigel, 2011 Application of support vector regression to genome-assisted prediction of quantitative traits. Theor. Appl. Genet. 123: 1065–1074.

Lorenz, A. J., S. Chao, F. G. Asoro, E. L. Heffner, T. Hayashi *et al.*, 2011 *Chapter 2: Genomic Selection in Plant Breeding: Knowledge and Prospects*. Elsevier Inc.

Ma, W., Z. Qiu, J. Song, J. Li, Q. Cheng *et al.*, 2018 A deep convolutional neural network approach for predicting phenotypes from genotypes. Planta 248: 1307–1318.

Meuwissen, T. H. E., 2009 Accuracy of breeding values of'unrelated'individuals predicted by dense SNP genotyping. Genet. Sel. Evol. 41: 35.

Meuwissen, T. H. E., B. J. Hayes, and M. E. Goddard, 2001 Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps. Genetics 1–11.

Monir, M. M., and J. Zhu, 2018 Dominance and Epistasis Interactions Revealed as Important Variants for Leaf Traits of Maize NAM Population. Front. Plant Sci. 9: 627.

Moser, G., B. Tier, R. E. Crump, M. S. Khatkar, and H. W. Raadsma, 2009 A comparison of five methods to predict genomic breeding values of dairy bulls from genome-wide SNP markers. Genet. Sel. Evol. 41: 56.

Neves, H. H., R. Carvalheiro, and S. A. Queiroz, 2012 A comparison of statistical methods for genomic selection in a mice population. 1–17.

Norman, A., J. Taylor, J. Edwards, and H. Kuchel, 2018 Optimising Genomic Selection in Wheat: Effect of Marker Density, Population Size and Population Structure on Prediction Accuracy. G3 8: 2889–2899.

Okut, H., D. Gianola, G. J. M. Rosa, and K. A. Weigel, 2011 Prediction of body mass index in mice using dense molecular markers and a regularized neural network. Genet. Res. 93: 189–201.

Parker, D. B., 1987 Optimal algorithms for adaptive networks: Second order backpropagation, second order direct backpropagation, and second order hebbing learning.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, 2011 Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12: 2825–2830.

Pérez, P., and G. de los Campos, 2014 Genome-wide regression and prediction with the BGLR statistical package. Genetics 198: 483–495.

Pouladi, F., H. Salehinejad, and A. M. Gilani, 2015 Deep Recurrent Neural Networks for Sequential Phenotype Prediction in Genomics.

Prechelt, L., 1998 Early Stopping - But When?, pp. 55–69 in *Neural Networks: Tricks of the Trade*, edited by G. B. Orr and K.-R. Müller. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ramstein, G. P., J. Evans, S. M. Kaeppler, R. B. Mitchell, K. P. Vogel *et al.*, 2016 Accuracy of Genomic Prediction in Switchgrass (Panicum virgatum L.) Improved by Accounting for Linkage Disequilibrium. G3 6: 1049–1062.

Ribaut, J.-M., and M. Ragot, 2007 Marker-assisted selection to improve drought adaptation in maize: the backcross approach, perspectives, limitations, and alternatives. Journal of Experimental Botany 58: 351–360.

Riedelsheimer, C., F. Technow, and A. E. Melchinger, 2013 Comparison of whole-genome prediction models for traits with contrasting genetic architecture in a diversity panel of maize inbred lines. 1–9.

Roorkiwal, M., A. Rathore, R. R. Das, M. K. Singh, A. Jain *et al.*, 2016 Genome-Enabled Prediction Models for Yield Related Traits in Chickpea. Front. Plant Sci. 7: 1666.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986 Learning internal representation by error propagation, Parallel Distributed Processing, DE Rumelhart and JL McClelland, eds.

Shaikhina, T., and N. A. Khovanova, 2017 Handling limited datasets with neural networks in medical applications: A small-data approach. Artificial Intelligence in Medicine 75: 51–63.

Spindel, J., H. Begum, D. Akdemir, P. Virk, B. Collard *et al.*, 2015 Genomic Selection and Association Mapping in Rice (Oryza sativa): Effect of Trait Genetic Architecture, Training Population Composition, Marker Number and Statistical Model on Accuracy of Rice Genomic Selection in Elite, Tropical Rice Breeding Lines. PLoS Genetics 11: e1004982–25.

Usai, M. G., M. E. Goddard, and B. J. Hayes, 2009 LASSO with cross-validation for genomic selection. Genet. Res. 91: 427–436.

Vazquez, A. I., G. J. M. Rosa, K. A. Weigel, G. de los Campos, D. Gianola *et al.*, 2010 Predictive ability of subsets of single nucleotide polymorphisms with and without parent average in US Holsteins. J. Dairy Sci. 93: 5942–5949.

Webb, S., 2018 Deep learning for biology. Nature 554: 555–557.

Xavier, A., W. M. Muir, and K. M. Rainey, 2016 Assessing Predictive Properties of Genome-Wide Selection in Soybeans. G3 6: 2611–2616.

Xu, Y., X. Wang, X. Ding, X. Zheng, Z. Yang *et al.*, 2018 Genomic selection of agronomic traits in hybrid rice using an NCII population. Rice 11: 32.

Zou, H., and T. Hastie, 2005 Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series.

CHAPTER FIVE: TRANSCRIPTOME-BASED PREDICTION OF COMPLEX TRAITS IN MAIZE [1]

[1] The work described in this chapter has been published in the following manuscript

Christina B. Azodi, Jeremy Pardo, Robert VanBuren, Gustavo de los Campos, and Shin-Han Shiu (2019) Transcriptome-based prediction of complex traits in maize. *The Plant Cell.* DOI: 10.1105/tpc.19.00332

5.1 Abstract

The ability to predict traits from genome-wide sequence information (i.e. genomic prediction), has improved our understanding of the genetic basis of complex traits and transformed breeding practices. Transcriptome data may also be useful for genomic prediction. However, it remains unclear how well transcript levels can predict traits, particularly when traits are scored at different development stages. Using maize genetic markers and transcript levels from seedlings to predict mature plant traits, we found transcript and genetic marker models have similar performance. When the transcripts and genetic markers with the greatest weights (i.e. the most important) in those models were used in one joint model, performance increased. Furthermore, genetic markers important for predictions were not close to or identified as regulatory variants for important transcripts. These findings demonstrate that transcript levels are useful for predicting traits and that their predictive power is not simply due to genetic variation in the transcribed genomic regions. Finally, genetic marker models identified only one of 14 benchmark flowering time genes, while transcript models identified five. Highlighting that, in addition to being useful for genomic prediction, transcriptome data can provide a link between traits and variation that cannot be readily captured at the sequence level.

5.2 Introduction

The prediction of complex traits from genetic data is a grand challenge in biology and the outcome of such prediction has become increasingly useful for plant and animal breeding (Heffner *et al.* 2009; Jonas and de Koning 2013). Among the different approaches for connecting genotypes to phenotypes, genomic prediction (or genomic selection) using all available markers was developed to overcome the limitations of Marker-Assisted Selection, which uses only significant quantitative trait loci (QTLs), for breeding traits that are controlled by many small

effect alleles (Meuwissen *et al.* 2001; Ribaut and Ragot 2007). Using genomic prediction, breeders are able to make data driven decisions about what lines to include in their programs, speeding up and reducing the cost of developing the next generation of crops (Endelman *et al.* 2014; Spindel *et al.* 2015). Furthermore, because genomic prediction models are associating genetic signatures with phenotypes, untangling genomic prediction models has the potential to improve our understanding of the genetic basis of complex traits. However, as with related approaches such as genome wide association studies and QTL mapping, it remains difficult to go from associated genetic markers to the molecular basis for a trait (Drinkwater and Gould 2012; Solberg Woods 2014).

There are a number of factors contributing to this difficulty. The variation in markers associated with phenotypes may not be the causal variants but are linked to the genes that control the trait in question. Considering that linkage disequilibrium distance can range from 1 kilobase (kb) in diverse maize populations (Tenaillon *et al.* 2001) to ~250 kb in *Arabidopsis thaliana* (Nordborg *et al.* 2002), the linked candidate genes can range from a few to a few hundreds. Even if the associated genetic variant is controlling the underlying phenotype, most variants associated with complex traits have small effect sizes and can be regulatory (Albert and Kruglyak 2015), which may not be linked to the genes they regulate. Furthermore, multiple regulatory variants that have indiscernible effects on their own, could interact epistatically to influence gene and ultimately trait expression. However, even with sufficient statistical power to detect genetic variants with small effect sizes and interactions between them, genetic information is connected to traits through multiple intermediate processes, including, for example, transcription, translation, epigenetic modification, and metabolism. Each of these intermediate processes

represent an additional level of complexity that obscures the association between genetic information and a trait.

One solution is to account for these intermediate processes by integrating relevant omics data in addition to genetic variation. This approach has led to promising, but often mixed, results in plants. Current efforts have focused primarily on predicting hybrid performance using transcriptional information from the parental lines. For example, transcript level-based distance measures generated from transcripts associated with the trait were better than genetic markers in predicting hybrid performance in maize (Frisch *et al.* 2010; Fu *et al.* 2012). However, when all transcripts were used (instead of a subset of pre-selected transcripts), model performance decreased (Zenke-Philippi *et al.* 2016). The performance of models based on transcript levels can be better or worse compared to those based on genetic markers depending on the trait. For example, transcriptome data performed better for predicting grain yield in hybrid maize populations, but genetic marker data performed better for predicting grain dry matter content in the same population (Schrag *et al.* 2018). Similarly, in a maize diversity panel, genomic prediction models that combined transcript and marker data only outperformed models using markers alone for certain traits (Guo *et al.* 2016). Finally, efforts to integrate additional omic information to predict various traits in *Drosophila melanogaster* (Li *et al.* 2019), and human diseases, such as breast cancer (González-Reymúndez *et al.* 2017), and responses to treatment interventions, including acute kidney rejection and response to infliximab in ulcerative colitis (Kang *et al.* 2017; Zarringhalam *et al.* 2018), have demonstrated the potential usefulness of transcriptome data in the field of precision medicine.

Overall, these efforts provide reasonable evidence that transcriptome data could be useful for trait prediction. However, genomic prediction-based approaches that trained on the entire

transcriptome data have not been used to better understand the genetic mechanisms for a trait. In addition, it is not known the degree to which transcriptomes obtained at a particular developmental stage can be informative for predicting phenotypes scored at a different stage. To address these questions, we used transcriptome data derived from maize whole seedling (Hirsch *et al.* 2014) to predict phenotypes (flowering time, height, and grain yield) at much later developmental stages. In addition to comparing prediction performance between genetic marker and transcriptome-based models, we also looked at whether transcripts and genetic markers that were important for the prediction models were located in the same or adjacent regions. Finally, we determined how well our models were able to identify a benchmark set of flowering time genes to explore the potential of using genomic prediction to better understand the mechanistic basis of complex traits.

5.3 Results and Discussion

5.3.1 Relationships between transcript levels, kinship, and phenotypes among maize lines

Before using the transcriptome data for genomic prediction, we first assessed properties of the transcriptome data in three areas: (1) the quantity and distribution of transcript information across the genome, (2) the amount of variation in transcript levels, and (3) the similarity in the transcriptome profile between maize lines, with an emphasis on how these properties compared to those based on the genotype data. After filtering out 16,898 transcripts that did not map to the B73 reference genome or had zero or near zero variance across lines (see Methods), we had 31,238 transcripts. While the number of transcripts was <10% of the number of genetic markers used in this study (332,178), the distribution of transcripts along the genome was similar to the genetic marker distribution (Supplemental Figure 5.1). The $\log_2$-transformed median transcript level across lines ranged from 0 to 12.4 (median=2.2) and the variance ranged from $3x10^{-30}$ to

14.5 (median= 0.13), highlighting that a subset of transcripts had relatively high variation in transcript levels across maize lines at the seedling stage. To determine how similar transcript levels were between lines, we calculated the expression Correlation (eCor) between all pairs of lines using Pearson's Correlation Coefficient (PCC). The eCor values ranged from 0.84 to 0.99 (mean=0.93). As expected, lines with similar transcriptome profiles were also genetically similar as there was a significant correlation between eCor values with values in the kinship matrix generated from the genetic marker data (Spearman's Rank $\rho = 0.27$, $p < 2.2\text{x}10^{-16}$; Figure 5.1A). As a result, we were able to find clusters of lines that had both high transcript and genetic similarities (e.g. cluster a, b; Figure 5.1B, C). However, most of the variation in eCor was not explained by kinship, which explained why we identified other clusters that had similar transcriptome profiles, but were not genetically similar (e.g. cluster c, Figure 5.1B, C).

Because the basis of genomic prediction is to predict a phenotype from genetic data, we next asked if kinship or eCor were anti-correlated with the phenotypic distances between lines (see Methods). While both kinship ($\rho = -0.03$, $p < 2.2\text{x}10^{-16}$; Figure 5.1D) and eCor ($\rho = -0.08$, $p < 2.2\text{x}10^{-16}$; Figure 5.1E) were significantly, negatively correlated with the phenotype distance, the degree of correlation was minor. Furthermore, the groups of lines that clustered together based on their eCor (e.g. clusters a, b; Figure 5.1B, 1C) did not have lower phenotypic distance (Figure 5.1F). Taken together, these findings suggest that transcriptome data may be similarly informative as genotype data but capture difference aspect of phenotypic variation. We tested both of these interpretations further in subsequent sections.

5.3.2 Predicting complex traits from transcript or genetic marker data

To test how useful transcriptome data was for genomic prediction compared to genetic marker data, we applied four approaches to predict three agronomically important traits in maize:

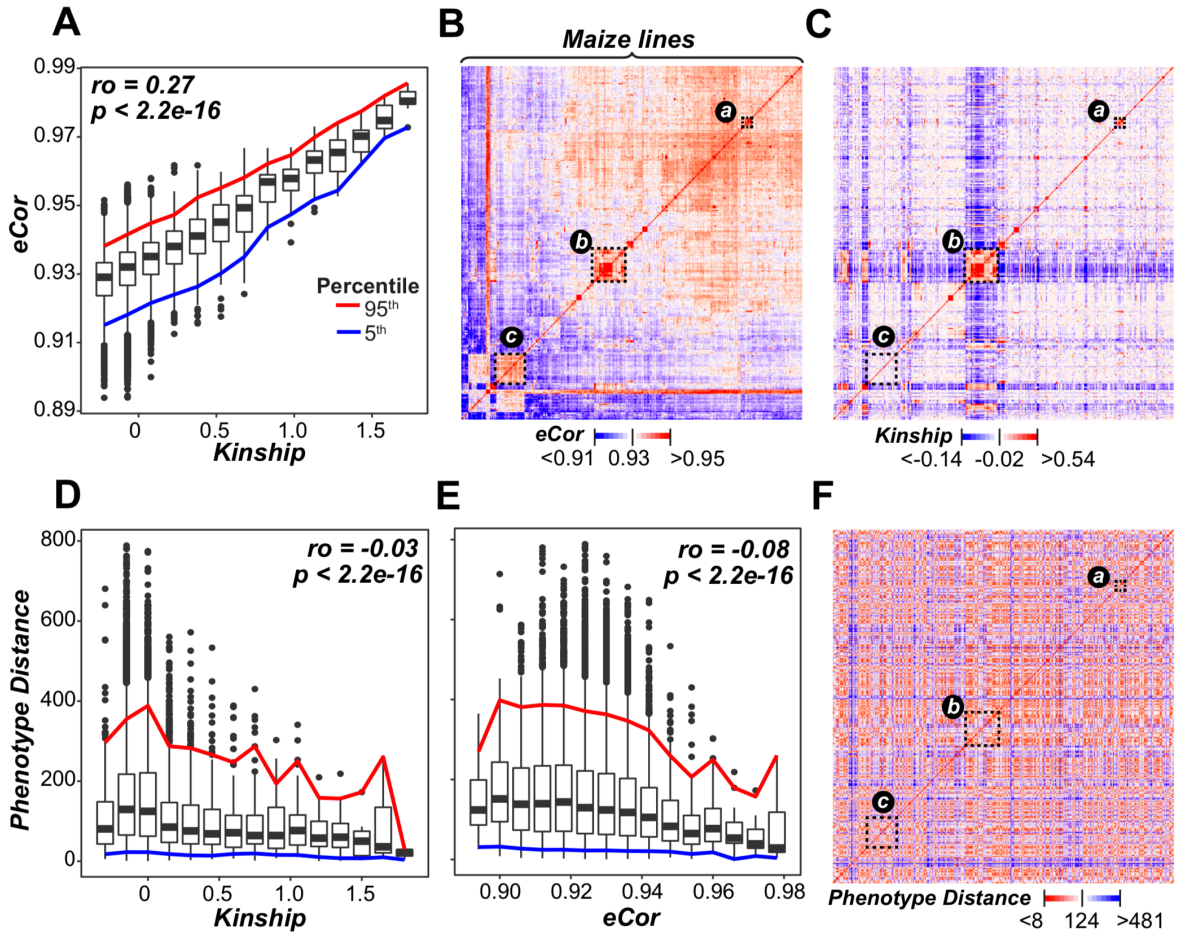


Figure 5.1. Relationship between lines from transcript and genetic marker data.

(A) Relationship between kinship based on genetic marker data (X-axis) and expression correlation (eCor, in Pearson's Correlation Coefficient (PCC)) based on transcript data (Y-axis). Boxplots show the median Y-axis value for each X-axis bin (bin size=0.15) with the 5th (blue) and 95th (red) percentile range shown. The correlation between kinship and eCor was calculated using Spear- man's Rank Coefficient (ρ). (B, C) The relationships between lines based on eCor (B) or kinship (C) for all pairs of maize lines. Lines are sorted based on hierarchical clustering results using the eCor values. The blue, white, and red color scales indicate negative, no, or positive correlations, respectively. Dotted rectangles: indicating cluster of lines discussed in the main text. (D, E) The relationships between the Euclidean distance calculated with phenotype

Figure 5.1 (cont'd)

values (Phenotype Distance: Y-axis) and kinship (D), and eCor (E). Colored line: follow those in (A). (F) The relationships between lines based on Phenotype Distance, where the lines were sorted as in (B). Red: smaller distance (more similar). Blue: greater distances (less similar).

flowering time, height, and grain yield. Because no one genomic prediction algorithm always performs best (Heslot *et al.* 2012; Spindel *et al.* 2015), we tested two linear algorithms (ridge regression Best Linear Unbiased Predictor (rrBLUP) and Bayesian-Least Absolute Shrinkage and Selection Operator (BL)), one nonlinear algorithm (random forest: RF), and one ensemble approach (En; see Methods). To establish a baseline for our genomic prediction models, we determined the amount of the phenotypic signal that could be predicted using population structure alone, defined as the first five Principal Components from the genetic marker data. Then we built models for each trait using genetic marker data (G), kinship (K) derived from G, transcript levels (T), or expression correlation (eCor) derived from T (Figure 5.2). Model performance was measured using PCC between the actual and the predicted phenotypic values.

Across algorithms and traits, the K data resulted in models with the best predictive performance, while models built using the eCor data performed the worst (Figure 5.2, Supplemental Table 5.1). Furthermore, models built using G always outperformed models using T. Regardless, eCor and T-based models were significantly better than the baseline predictions (dotted blue line, Figure 5.2), indicating transcriptome data can be informative in genomic prediction. Considering the transcriptome data is from seedling; it is particularly surprising that mature plant phenotypes can be predicted. Next, we asked if using only the most informative (i.e. the largest absolute coefficients) transcripts or genetic markers as input into our models would

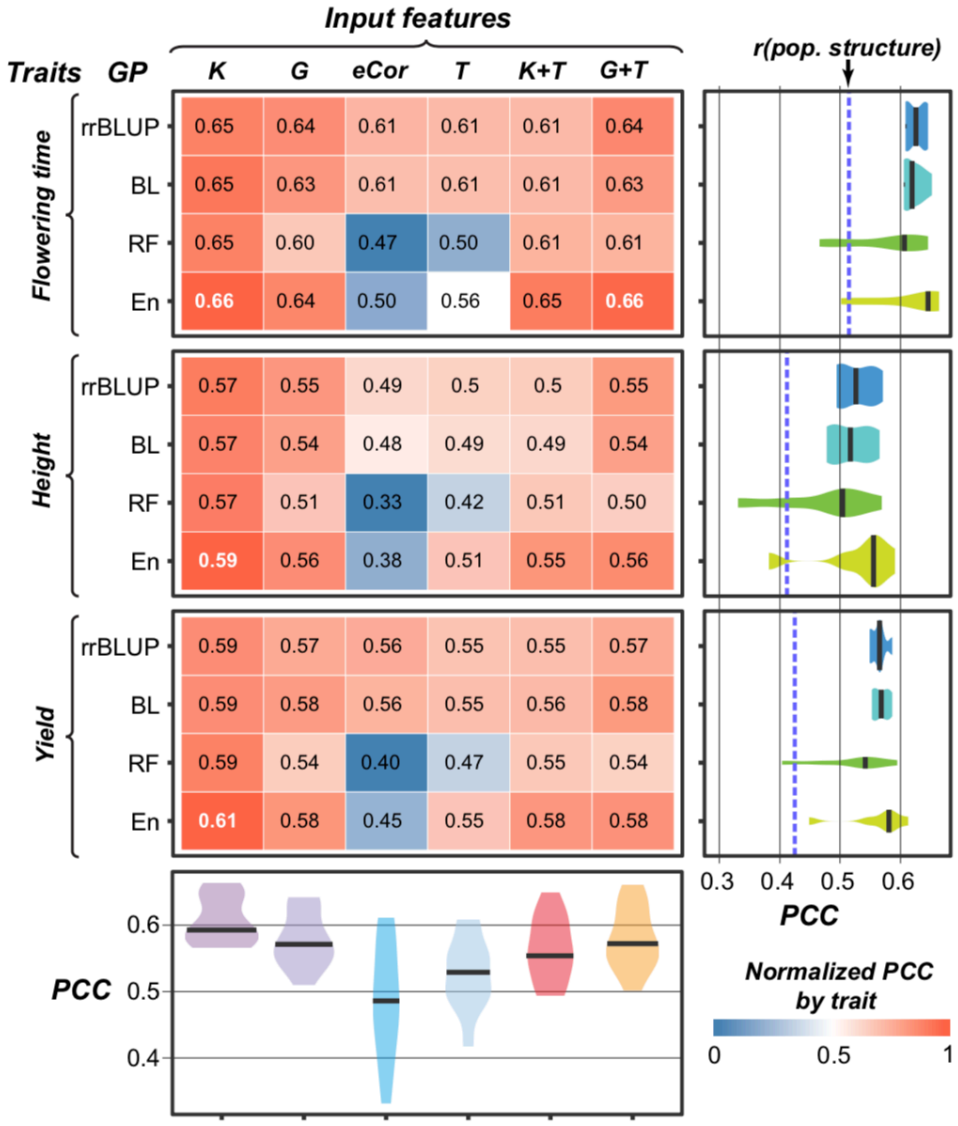


Figure 5.2. Genomic prediction model performance.

PCCs between predicted and true values for three traits and four algorithms using six different input features. The text in each box represents the absolute PCC with the best performing model for each trait in white. The box color represents the PCC normalized by trait, where the brightest red (1) corresponds to the algorithm/input feature combination that performed the best for the trait and the brightest blue (0) corresponds to the combination that performed the worst. Right violin-plots show the PCC distributions among different input features for each algorithm (right).

Figure 5.2 (cont'd)

The median PCCs are indicated with black bars. The model performance PCCs based on only population structure (first 75 principal components) are indicated with a blue dashed line. Bottom violin-plots show the PCC distributions among different algorithms for each input feature. rrB: ridge regression Best Linear Unbiased Predictor. BL: Bayesian Least Absolute Shrinkage and Selection Operator. RF: Random Forest. En: Ensemble.

improve trait predictions (see Methods). We also tested different sized subsets of transcripts with the greatest degrees of line specific expression to test if they could better predict traits. However, using rrBLUP to predict flowering time as an example, none of these subsets performed better than the full T data (Supplemental Table 5.3). We also tested setting the most variable transcripts as fixed effects in our rrBLUP models, but this also did not improve performance (Supplemental Table 5.3). Finally, consistent with earlier findings (Shen and Chou 2006; Jia *et al.* 2015), combining the predictions from multiple algorithms, known as an ensemble approach, resulted in the best predictive models (Figure 5.2), and is therefore used to illustrate most of our findings in the following sections.

5.3.3 Predicting complex traits using both transcript and genetic marker data

Because the genetic marker and transcriptome data represented different types of molecular information that could be associated with the traits of interest, we hypothesized that their combination would be more informative and next built models that used combined data, either K+T or G+T. However, adding the transcript data did not substantially improve performance over K or G alone (Figure 5.2). One possible reason for this lack of improvement could be overfitting. This is most common when there is only a small amount of training data

(i.e. few maize lines) but a very large number of predictor variables (i.e. many genetic markers/transcripts). To test this hypothesis, we trained rrBLUP models (referred to as G_{200}+T_{200}) to predict flowering time using only the 200 genetic markers and the 200 transcripts with the largest absolute coefficients from the G and T rrBLUP models, respectively (see Methods). These genetic markers and transcripts are referred to as "features". To avoid overfitting during feature selection (Bermingham *et al.* 2015), we first separated the dataset into training and testing sets. The top features were selected using the training data only. The testing data were never used to select the top features. Using the independent testing data to evaluate performance, our ability to predict flowering time improved using G_{200}+T_{200} (PCC = 0.68 +/- 0.06) compared to the full G+T model (PCC = 0.64 +/- 0.01) and to the individual G and T models (PCC = 0.64 +/- 0.01, 0.61 +/- 0.01, respectively). One explanation for this improvement could be that using only the top features of each data type reduced noise from the model. If this is the case, the G_{200} and the T_{200} models would be expected to outperform the G and the T models, respectively. But we see the opposite results (see previous section; Supplemental Table 5.3), suggesting this improvement was due to a reduction in overfitting.

To assess if G or T data features tend to be more informative in predicting traits, we further quantified the importance score of each genetic marker and transcript feature for models using G+T data. The importance score represents the impact that each feature had on model performance defined according the algorithm used (see Methods). Because the G and T data features may contain overlapping information and, thus, are not independent, the importance scores from the G+T model may be affects by issues caused by collinearity. However, given that the importance scores assigned to transcripts in the G+T models were correlated with the scores from the T-only models (Supplemental Figure 5.2A), the addition of the genetic marker features

into the model did not impact the relative importance of transcript features. The only exception was a subset of Ts that were important for the G+T, but not the T-only Bayesian LASSO (BL) models. Because RF importance measures tend to be biased toward continuous features (Strobl *et al.* 2007), we focused on rrBLUP and BL importance scores. For all three traits, the top 1,000 most important features were enriched for genetic markers relative to transcript features (Odds Ratio = 0.17 ~ 0.44; all $p < 1x10^{-16}$; Supplemental Figure 5.2B; Supplemental Table 5.2). However, the top 20 most important features tended to be enriched for transcript relative to genetic marker features (Odds Ratio = 2.66 ~ 13.0, $p = 0.087 \sim <1x10^{-16}$, Supplemental Table 5.2), with transcript features making up the top two most important feature in all cases (Supplemental Figure 5.2B). The consistency with which transcript features were the most important for the models suggests that transcript information is useful for genomic prediction.

5.3.4 Comparison of the importance of transcripts versus genetic markers for model predictions

Because models built using transcript features outperformed baseline models based solely on population structure, we know transcriptome data contained information useful for explaining phenotypic variation. Furthermore, using feature selection to combine both datasets into one predictive model ($G_{200}+T_{200}$) improved our ability to predict flowering time (Supplemental Table 5.3). Therefore, we hypothesized that these two data types capture different aspects of phenotypic variation. To address this, we assessed the extent to which the important genetic markers (from G-based models) overlapped with or neighbored the genes where the important transcripts (from T-based models) originated from (top; Figure 5.3A). We did not use the importance values from the G+T model due to concern of feature dependence. The genic region and flanking sequences within a defined window of an important transcript is referred to as the

transcript regions (see Methods). For each trait and algorithm, we compared the importance assigned to the transcript with that of the genetic marker with the highest average importance in the transcript region (T:G pair).

Multiple window sizes were explored (see Methods), and we used 2 kb (+/- 1kb from the center of a gene) where the feature importance correlation between transcripts and genetic markers was maximized (Supplemental Figure 5.3A). Using this window size, 15,049 T:G pairs were identified. At the whole genome level there appeared to be regions where both genetic markers and transcripts were identified as important (Supplemental Figure 5.4). However, when we look closer, those regions mostly do not overlap. In some cases, the important genetic markers and transcripts were in linkage disequilibrium. Using the flowering time model as an example, we found the most important genetic marker was located within a gene upstream the most important transcript (GRMZM2G171650: *MADS69*; arrow a, Figure 5.3B), but the two are in linkage disequilibrium (Hirsch *et al.* 2014). In most cases, there were no important genetic markers that were located nearby to important transcripts and if we extend the window size to 80 kb, we see *MADS69* is the exception rather than the rule (Supplemental Figure 5.3B). For example, the second most important flowering time genetic marker was not located near important transcript regions (arrow b, Figure 5.3B). Similarly, the second most important flowering time transcript (GRMZM5G865543) was over 0.6 Mb from an important genetic marker (arrow c, Figure 5.3B). Across all traits and algorithms, T:G pairs were only moderately correlated (ρ = 0.09-0.13; Figure 5.3C, Supplemental Figure 5.5A).

This lack of correlation is notable for the most important genetic markers and transcripts. For example, across the three traits, only 4-7 T:G pairs were both in the top 1% most important features from the ensemble models, and those pairs were never the top ranked genetic markers or

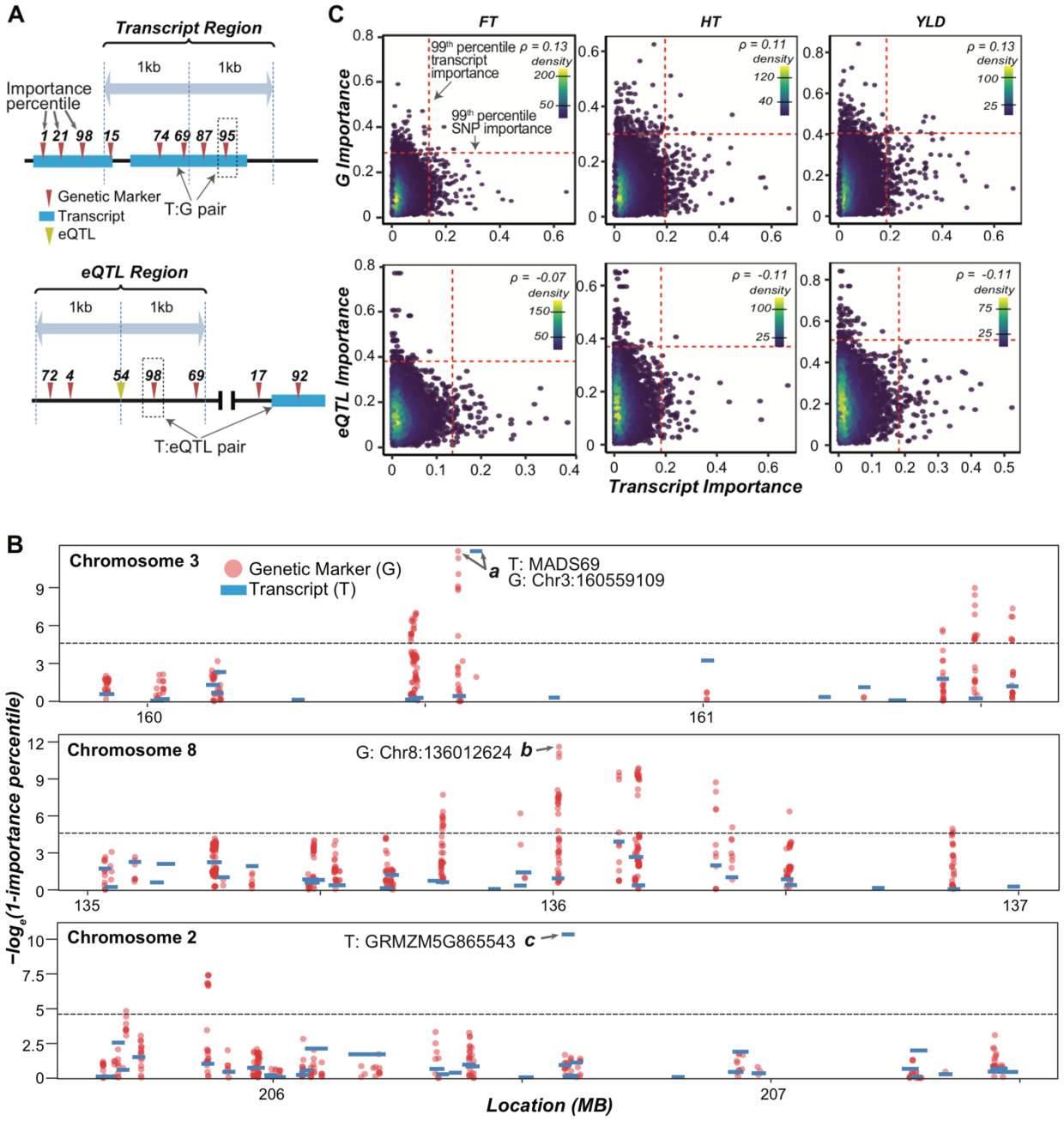


Figure 5.3. Correlation between genetic marker and transcript importance for flowering time.

(A) Illustration of how transcript (T):genetic marker (G) (top graph) and T:expression Quantitative Trait Locus (eQTL) (bottom graph) pairs were determined. Genetic marker importance percentiles are shown above the genetic markers (red triangle) and eQTL (yellow

Figure 5.3 (cont'd)

triangle). A T:G pair was defined as the transcript and the most important genetic marker within the transcript region (top graph). A T:eQTL pair was defined as the transcript and the most important genetic marker within the eQTL region (bottom graph). (B) Manhattan plots of the transcript (blue bar) and genetic marker (red dot) importance scores ($-\log_e$(1-importance percentile)) in a 2Mb window surrounding top two genetic markers (top and middle plots) and transcripts (top and bottom plots) based on the T-based and G-based Ensemble models for predicting flowering time, respectively. All genetic markers (i.e. not just the T:G pair) are shown. The threshold (gray dotted line) is set at the 99th percentile importance. (C) Density scatter plot of the importance scores (see Methods) of the genetic marker (Y-axis) and transcript (X-axis) for T:G pairs (top graphs) and of the eQTL genetic marker (Y-axis) and transcript (X-axis) for the T:eQTL pairs (bottom graphs) for three traits derived from the G-based and T-based Ensemble models, respectively. The threshold (black dotted line) was set at the 99th percentile importance score for each trait and input feature type. The correlation between importance scores between transcript and genetic marker/eQTL pairs was calculated using Spearman's rank (ρ).

transcripts from the model (Figure 5.3B). These findings argue against the notion that these two data types capture similar aspects of phenotypic variation as we hypothesized earlier. One concern was that the lack of correlation was due to the genetic marker data being derived from RNA-Seq experiments, and thus limited to the transcribed regions. However, when the experiment was repeated using ~1 million genome-wide genetic markers (G_{GW}) derived from whole-genome sequencing (Bukowski *et al.* 2018) as input features (Supplemental Figure 5.6A), the correlation between T:G_{GW} pairs did not increase (Supplemental Figure 5.6B and S6C).

In light of this, we hypothesized that the lack of correlation was because important transcripts tend to be regulated by important *trans* factors located far beyond the transcript region. To test this, we assessed the degree to which important genetic markers identified as expression QTL (eQTLs) were associated with important transcripts. We identified 58,361 *cis* (62) and *trans* (58,299) eQTL associated with 7,052 transcripts and defined T:eQTL pairs for each of these transcripts by selecting the genetic marker within +/- 1kb of an eQTL for that transcript (i.e. eQTL region) with the highest average importance. Across all traits and algorithms, the importance of transcripts and eQTL in T:eQTL pairs was actually negatively correlated (ρ = -0.15 ~ -0.06; Figure 5.3C, Supplemental Figure 5.5B).

The lack of correlation between importance scores for T:G and T:eQTL pairs was in contrast to the relatively high correlation observed in feature importance between algorithms (ρ= 0.31-0.98), with rrBLUP and BL importance scores being the most correlated (ρ= 0.87-0.98) and the average correlation between genetic markers (ρ = 0.75) being higher than for transcripts (ρ = 0.55) (correlation between algorithms; Supplemental Figure 5.7). Together with the findings that important genetic markers were not co-located and eQTL were not associated with genes that gave rise to the important transcripts for any of the three traits, these findings may suggest that transcriptome data is capturing layers of information, such as epigenetic signals, that are not captured by genome sequences. However, we cannot rule of the possibility that the eQTL approach using RNA-Seq based genetic markers is not sufficiently sensitive in identifying important *trans*-factors. Further study with more trait and high quality genome-wide genetic marker data is needed to resolve these possibilities.

5.3.5 Assessment of benchmark flowering time genes

Because the genetic basis for flowering time is well studied (Muszynski *et al.* 2006; Danilevskaya *et al.* 2010; Meng *et al.* 2011; Lazakis *et al.* 2011), we identified a set of 14 known flowering time genes (Supplemental Table 5.4). To assess the extent to which these benchmark genes can predict flowering time, we trained an rrBLUP model where we set these 14 genes as fixed, rather than random, effects and our model performance increased (PCC = 0.64 +/- 0.01; Supplemental Table 5.3) compared to when they were not fixed (PCC = 0.61). Then we compared the ability of genetic marker and transcript-based models to identify these benchmark genes as important using the T:G and T:eQTL pairs described earlier. Of the 14 benchmark genes, four had corresponding genetic markers in our T:G pair data. When we increased the flanking regions threshold to 20kb from the center of the transcript for defining T:G pairs, corresponding genetic markers were found for five additional benchmark genes. Two benchmark genes, *CCT1* and *PEBP4*, neither of which were members of a T:G pair, were associated with eQTLs. To account for differences in distribution and range of importance scores generated by different algorithms and numbers of features, the importance scores were converted to percentiles for comparison purposes.

Different benchmark genes were important (>95th percentile) for models using the two different data types, with one and five benchmark gene considered important by the genetic marker-based and the transcript-based models, respectively (Figure 4A; Supplemental Table 5.5). For example, the genetic marker located within the *RAP2* gene, which has been shown to be associated with flowering time in multiple studies (Buckler *et al.* 2009; Hirsch *et al.* 2014), was identified as important based on genetic marker (99.7th-99.9th percentile), but not transcript (59th-79th percentile) data. In contrast, *MADS69*, *MADS1*, *PEBP24*, and *PEBP8* were identified as

important using transcript data (95th-100th percentile), but not using genetic marker data (16th-93th percentile). Furthermore, with transcript data we were able to assess the importance of three genes (*ZAG6, PEPB5, and PEBP2*) that were not located near genetic markers or associated with eQTL. For example, there were no eQTL associated with or genetic markers within the 40bp window of *ZAG6*, but *ZAG6* was identified as important (98th-99.9th percentile) in the transcript-

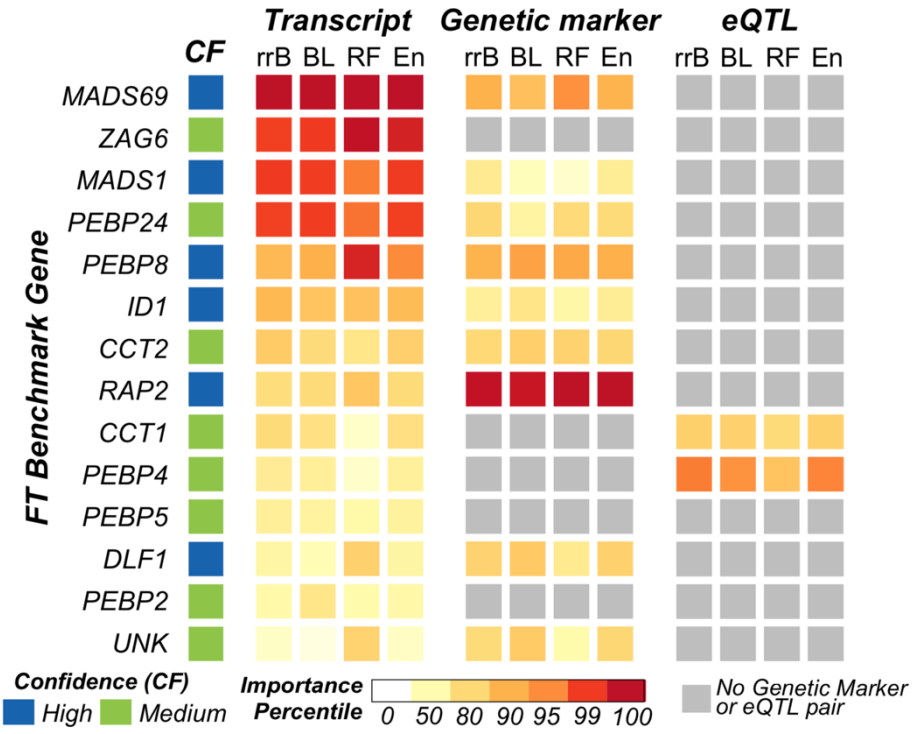


Figure 5.4. Comparison of transcript and genetic marker importance scores for benchmark flowering time genes.

Importance percentile of each transcript (left) and genetic marker (right) pair as determined by each of the 4 algorithms (X-axis). Genes are sorted based on hierarchical clustering of their importance percentiles. Gray boxes designate benchmark genes that did not have genetic markers within a 40kb window. Confidence levels (high or medium) were assigned based on the type of evidence available for the benchmark gene (see Methods). Algorithms were abbreviated as in Figure 5.2.

based models (Figure 5.4A). For some of these benchmark genes, the region most closely linked to trait variation could be outside the +/- 20kb window. For example, as described above, the important genetic marker for *MADS69* (Chr3_160559109) is ~32 kb upstream (see a arrows; Figure 5.3B). However, when we plotted the correlation between importance scores between T:G pairs using the largest window size (80 kb), we found that *MAD69* was the only gene for which this was the case (Supplemental Figure 5.3B). Taken together, these finding further highlight the usefulness of transcript data for identifying the genetic basis for variation in a trait.

5.3.6 Improving our understanding of the genetic basis of flowering time using transcriptome data

An open question was why transcript-based models were able to identify five benchmark flowering time genes as important that were not identified by genetic marker-based models and if transcriptome data could be used to better understand the genetic basis of flowering time. To understand why benchmark genes were not uniformly identified as important for flowering time when using both genetic marker and transcript data, we determined the extent to which transcript levels and the genetic marker allele (i.e. major or minor) were related to flowering time. As expected, we observed the most significant differences in flowering time for the transcripts (Figure 5.5A, Supplemental Figure 5.8A) and genetic markers (Figure 5.5B, Supplemental Figure 5.8B) that were identified as important by our models. For example, *MADS1* was important only in the transcript-based models and transcript level was significantly correlated with flowering time ($p = 0.0001$; Figure 5.5A). In contrast, lines with the major allele for the genetic marker that paired with the *MADS1* transcript (Chr9: 156980141) did not flower at a significantly different time than lines with the minor allele ($p = 0.062$; Figure 5.5B). Another

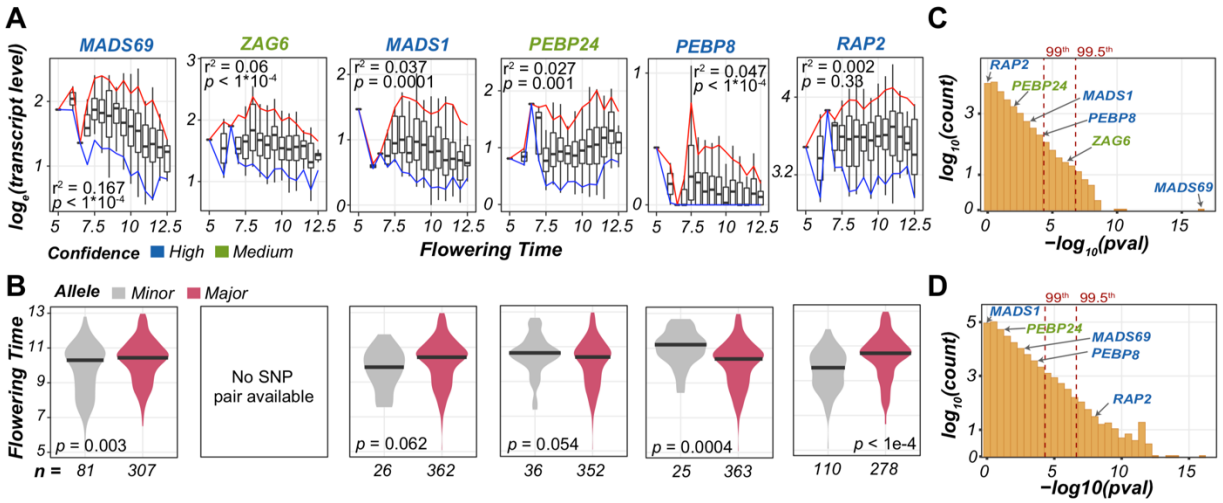


Figure 5.5. Relationship between transcript level/allele type and flowering time for benchmark genes.

(A) Boxplots show the transcript levels ($\log_e$(Fold-Change)) over flowering time bin with the 5th (blue) and 95th (red) percentile range shown. Flowering time was defined as the growing degree days/100. Linear models were fit and adjusted r^2 and p-values are shown. Confidence levels of benchmark genes were designated as in Figure 5.4. (B) Distributions of flowering time for lines with the major (red) or minor (gray) alleles for the genetic marker paired with each benchmark gene as indicated in (A). Differences in flowering time by allele were tested using t-tests. (C) Number of transcripts (Y-axis) for which transcript levels were associated with flowering time in linear models within *p*-value bins (-$\log_{10}$(*p*-value); X-axis). Benchmark genes are labeled as in (A). (D) Number of genetic markers (Y-axis) for which differences in flowering time by allele from t-tests were within p-value bins (-$\log_{10}$(*p*-value); X-axis). Benchmark genes are labeled as in (A).

example was *RAP2*, which was important only in the genetic marker-based models. Lines with the major allele in *RAP2* were more likely to flower late ($p < 1x10^{-4}$), but *RAP2* transcript levels

did not significantly correlate with changes in flowering time ($p = 0.33$). Overall, benchmark genes were more likely to have transcript levels associated with flowering time (Figure 5.5C) than genetic marker alleles associated with flowering time (Figure 5.5D).

Importantly, using the transcriptome data we were also able to understand in more detail the impact of the benchmark genes on flowering time. For example, variation in transcript levels of *MADS69* accounted for 16.7% of the variation in flowering time, more than any other transcript, where lines with lower levels of transcription flowered later. Modulation of *MADS69* expression levels has recently been patented as an approach to controlling flowering time ("US Patent Application for MODIFYING FLOWERING TIME IN MAIZE Patent Application (Application #20140366213 issued December 11, 2014) - Justia Patents Search"). Similarly, *MADS1* transcript levels explained 3.7% of the variation in flowering time, with lines with lower levels of transcription flowering later. This is consistent with what has been observed experimentally, where down-regulation of *MADS1* results in delayed flowering time (Alter *et al.* 2016). For medium confidence benchmark genes (i.e. identified through association studies), the specific roles of the genes on flowering time are not well understood, but by finding positive or negative correlations between transcript levels and the underlying phenotypes, more mechanistic details can be interred. For example, transcript levels of *ZAG6* had the second largest impact on flowering time, accounting for 6% of variation, with increased transcript levels associated with earlier flowering. Another example is *PEBP24*, with transcript levels of *PEBP24* accounting for 2.7% of the variation in flowering time. Unlike many of the other benchmark genes, increased *PEBP24* transcript levels were associated with later flowering time. Overall, the identification of these medium confidence benchmark genes as important transcript indicates the relevance of transcriptional regulation in their flowering time functions.

While using the benchmark genes allowed us to assess the usefulness of transcript levels compared to genetic marker information for identifying genes involved in flowering time, we should note that many non-benchmark genes were also identified by our models as important. For example, from the Ensemble model, there were 154 important, non-benchmark transcripts with importance scores falling between the two most important benchmark genes (*MADS69*, 100th percentile; *ZAG6*, 99.5th percentile; yellow, Dataset S1). While seven of those in between transcripts were annotated with the Gene Ontology (GO) term "flower development" (GO:0009908, green, Dataset S1), these 154 non-benchmark transcripts were not enriched for this GO term (q = 1.0). In fact, neither these transcripts nor any other set of important transcripts from models based on other algorithms (see Methods) were enriched for any GO terms. Therefore, from our transcript-based genomic prediction models we have identified 147 high ranking transcripts, many of which have unknown functions, that are among the most important in predicting flowering time in maize but do not play known roles in this process. For example, GRMZM5G865543 and GRMZM2G023520 (the second and third most important transcripts respectively from the Ensemble model) do not have annotated function in maize. And while they do have homologs in *Arabidopsis thaliana* and *Oryza sativa*, those homologs do not have known function in flowering time (see Supplemental Table 5.6 for similar information about the top 10 transcripts). Note that the transcriptome data is from the seedling stage. It is possible that genes of these important transcripts influence biological processes in earlier stage of development that influence flowering time later. To further our understanding of the genetic basis of flowering time control and the connections between juvenile and adult phenotypes, these important transcripts are prime candidates for future genetic studies.

5.4 Conclusions

We have generated predictive models that use genetic markers, transcripts, and their combination to predict flowering time, height, and yield in a diverse maize population. While models built using transcriptome data did not outperform models that used genotype data, transcript-based models performed well above random expectation, and in many cases, performance was similar to that of genotype-based models. We found that transcripts and genetic markers from different genomic regions were identified as important for model predictions. Furthermore, by assessing the relative importance of the features used to build the models, we found that transcript-based models identified more known flowering time associated genes than genetic marker-based models. These findings underscore the usefulness of transcript data for improving our understanding of the genetic mechanisms responsible for complex traits.

There are four possible mechanistic explanations of why transcript levels could have a similar predictive power as genetic markers. First, *cis*-regulatory variants that impact transcript levels, are all more likely to be similar between closely related individuals. Therefore, the ability of transcript data to predict phenotypes is simply a reflection of that dependency. However, we demonstrated that the most informative transcript features for predicting maize phenotypes are distinct from the most informative genetic marker features found in the transcript regions. While for some important transcripts, the associated important genetic marker could be in linkage disequilibrium but outside of the 2kb window used in our study (e.g. ~32 kb away in the case of *MADS69*), overall as we increased the transcript region window size, the correlation between the importance scores assigned to T:G pairs decreased, suggesting this is not generally the case. Thus, the second explanation is that there are *trans*-regulatory variants, e.g. due to transposon polymorphisms or transcriptional regulators, that play a major role. However, we found that the

importance of eQTLs (99.9% *trans*) and their associated transcripts were not positively correlated, suggesting that the *trans*-regulatory variation we identified cannot explain why transcript variation is predictive of phenotypic variation either. However, considering the challenges in identifying eQTLs due to mixed tissues used (Wills *et al.* 2013), in modeling epistatic interactions (Becker *et al.* 2012), and in our limited ability to find *cis*-eQTL, we cannot conclusively rule of this possibility. The third explanation is that transcription is a molecular phenotype caused by the integration of multiple genetic marker signals, both *cis* and *trans*, that may not have had strong signals individually. The fourth explanation is that there are epigenetic variants contributing to expression variation. It remains to be determined what the contribution of epigenetic variation is on our ability to use transcript data to predict phenotypes.

One surprise is that the transcript data generated using V1 seedling tissues can predict adult plant phenotypes. We reason that complex traits, such as flowering time, are influenced by more than just canonical genes that act immediately prior to the growth and developmental sequences leading to flowering. For example, early developmental events such as cotyledon damage (Hanley and May 2006), root restriction (Keever *et al.* 2015), and photoperiod and temperature changes (Song *et al.* 2013) can impact flowering time in mature plants. Therefore, early development transcript differences could eventually result in different flowering time. There were three limitations of this study that made our ability to predict adult plant phenotypes and identify known important transcripts even more surprising. First, transcript level data was derived from whole V1 seedling tissue, which should limit the predictive power of our genomic prediction models for mature plant traits. We expect that transcript information taken from tissues and timepoints more relevant to the phenotype of interest are more likely to be predictive. For example, co-expression networks derived from maize root tissues are more predictive of

accumulation of 17 different elements (e.g. Al, Fe, K, Zn) in maize seeds than co-expression networks derived from tissues not involved in element uptake and transport (Schaefer *et al.* 2018). Second, transcript levels were calculated by mapping reads to the B73 reference genome without considering structural and fragmental variations exist between diverse maize lines. Having only a B73 reference genome to map to likely results in bias or noise in our transcriptome dataset. In future studies, it will be informative to determine if correcting for such structural and fragmental variation would improve genomic prediction. Finally, a third limitation of our study is that no environmental component is considered. An area of active research in genomic prediction is the incorporation of Genotype by Environment (GxE) interactions into predictive models (Burgueño *et al.* 2012; Cuevas *et al.* 2017; Granato *et al.* 2018). Thus, a potential benefit of using transcript information for genomic prediction could be that GxE interactions would be picked up by transcript level signals. Because transcriptome data used in our study was from whole seedlings (i.e. not the same individuals that were phenotype), this could not be tested.

Our findings highlight an important benefit of using transcript data to better understand the genetic basis of a trait. While it can be difficult to associate signals from a number of small effect genetic markers or even a single large effect genetic marker back to a specific gene, transcript level information is inherently associated with genes. Because of the importance of regulatory variation on complex traits (Albert and Kruglyak 2015), the use of transcript information in genomic prediction could be crucial for deciphering the contribution of regulatory variation to the genetic basis of traits. Therefore, while we observed that in terms of predictive ability, genetic marker data outperformed transcript data, expression differences are more straightforward to interpret than sequence polymorphisms. In practice, this meant that transcript-

based models identified five benchmark flowering time genes, while genetic marker-based models only identified one and it highlighted our finding that more insight into the genetic basis of complex traits can be gained when transcriptome data are considered.

5.5 Methods

5.5.1 Genotypic, transcriptomic, and phenotypic data processing

The phenotypic (Hansey *et al.* 2011), and genotypic and transcriptomic (Hirsch *et al.* 2014) data used in this study were generated from the pan-genome population consisting of diverse inbred maize lines. Genotype, transcriptome, flowering time, height, and yield data was all available for 388 lines out of the 503 maize pan-genome panel and were used for the study (Dataset S2). Genetic marker scores derived from RNA-seq reads were converted to a [-1,0,1] format corresponding to [aa, Aa, AA] with the more common allele (AA) designated as 1. The genetic marker positions were converted from maize B73 reference genome A Golden Path v2 (AGPv2) to AGPv4.37. The AGPv2 genetic markers that did not map to AGPv4.37 and genetic markers with a minor allele frequency less than 5% were removed, resulting in 332,178 genetic markers. To determine if the use of RNA-Seq derived genetic markers biased our results, we also tested a set of genome-wide markers (G_{GW}). These markers were downloaded already processed and uplifted to AGPv4 from (Bukowski *et al.* 2018). Data was available for 149 maize lines included in the study. After removing G_{GW} with minor allele frequency less than 5% and duplicate patterns of allele calls across the 149 lines (i.e. the same criteria used for the G dataset), ~ 1.08 million markers were available for this analysis.

RNA-Seq derived transcriptomic data from whole-seedling tissue (i.e. root and shoot) at the V1 stage from (Hirsch *et al.* 2014) was processed to remove loci that did not map to AGPv4.37. The remaining maize B73 genes were filtered with default settings of the

nearZeroVar function from the R caret package to remove genes with zero or near zero variance (> 95% of the lines sharing the same transcript level) across lines. After the filtering steps, transcript counts for 31,238 genes were retained in the final dataset. The raw transcripts per million count data were transformed with a $\log_e + 1$ transformation before the data were used in subsequent analyses. Mapping rates to the B73 genome assembly were also downloaded from (Hirsch *et al.* 2014). To assess if transcriptome data had predictive power beyond random expectation, transcriptome data were permuted by gene, so each gene had the same distribution of transcript values, but the values were randomly assigned to different maize lines for building the transcriptome shuffled models. To compared important transcripts and genetic markers from genomic prediction models, transcripts were converted from AGPv2 to v4, only genes with one to one correspondence between AGPv3 and v4 were included in this analysis. To assess the impact uplifting had on expression levels we re-mapped transcript data from B73 to AGPv4 using Bowtie2 (version 2.3.2) and performed read counting using Cufflinks (version 2.2.1). The correlation between uplifted and re-mapped gene expression levels for B73 was 0.94 (PCC, p.value < $2x10^{-16}$).

5.5.2 Comparison of transcript and genetic marker data

Three different approaches were used to determine the similarity between lines based on the three different data types. For the genotype data, a kinship matrix was generated using the centered Identity By State method (Endelman and Jannink 2012) implemented in TASSEL v5.20180517 (Bradbury *et al.* 2007). The Pearson Correlation Coefficient (PCC) between RNA-Seq mapping rates and kinship with B73 was calculated using the cor.test function in R. For the transcript data, we generated an expression Correlation (eCor) matrix by calculating the PCCs of transcript values between lines using the cor.test function in R. The eCor matrix was normalized

between 0 and 1 and the diagonal was set as 1. Finally, for phenotype data, we calculated the Euclidean distance between lines using the distances package in the R environment. The correlation between kinship, eCor, and Phenotype Distance between pairs of lines was calculated using PCC.

5.5.3 Genomic prediction models and model performance

Because part of the phenotypic signal observed in genomic prediction models may be due to population structure/family relatedness within the breeding population, we established a baseline for our genomic prediction models by using the principal components (PCs) generated using the marker data alone, to predict phenotypic values for each trait. Because the relationship between the population structure and traits can vary by trait and by population, we tested the top 5, 10, 15, 20, 50, 75, and 100 PCs and selected the top 75 PCs to use as our baseline because accuracy plateaued after this point. Four methods were used for each trait, two linear-parametric methods: ridge regression-Best Linear Unbiased Predictor (rrBLUP) (Endelman 2011) and Bayesian Least absolute shrinkage and selection operator (BL) (Pérez and de los Campos 2014), and one non-linear and non-parametric method: Random Forest (RF) (Leo Breiman Statistics 2001), and one ensemble based approach (En) (Dietterich 2000). The rrBLUP models used the mixed.solve function in the "rrBLUP" package implemented in R. The BL models were also implemented in R using the "BGLR" package. RF was implemented in python using Scikit-Learn (Pedregosa *et al.* 2011). Ensemble predictions were generated by taking the mean of the predicted trait values from rrBLUP, BL, and RF. A grid-search was performed on the first 10 of the 100 cross-validation replicates to find the best combination of parameters for the RF model. Parameters tested included max tree depth (3, 5, 10, and 50) and the max number of features included in each tree (10%, 50%, 100%, square root, and $\log_2$).

The predictive performance of the models was compared using the PCC. The PCC between the predicted ($\hat{Y}$) and the true trait value (Y) and was computed using the cor() function in R for rrBLUP and BL or the NumPy corrcoef function in Python for RF. One hundred replicates of a five-fold cross validation approach were applied to maximize the data available for model training without resulting in overfitting. For each replicate, the lines were randomly divided into 5 subsets, where each subset is used as the testing set once and the rest 4 subsets combined to train the model, resulting in a total of 500 cross-validated runs. PCC was calculated using only the predicted values from the testing set for each run.

For the top 10 most important transcripts from the ensemble model, leave-one-feature-out analysis was performed using rrBLUP with 100 replicates to get a score for how much the model performance (PCC) changes when that one transcript is removed from training (Supplemental Table 5.6). Information about top BLAST matches was collected from maizeGDB (https://www.maizegdb.org/).

5.5.4 Selecting subsets of T or G for input to genomic prediction models

To determine if using smaller subsets of T or G as input to the genomic prediction models would improve our ability to predict traits, we used rrBLUP and flowering time as an example to select features. For transcript data, features were selected in three ways. First, 10, 20, 100, and 1000 transcripts with the greatest variance across the maize lines were selected and used as input to the rrBLUP models. Second, the 14 benchmark flowering time genes (see Methods: Benchmark flowering time genes) were used. Finally, 14 and 200 transcripts with the greatest absolute coefficient (i.e. weight) assigned by rrBLUP during training were selected. For this analysis, the models were re-run without cross-validation so that feature selection and model

training were performed on the training data and the testing data was only used to measure model performance, thus ensuring against overfitting. This was done for each of the 100 replicates.

5.5.5 Genetic marker/transcript importance analysis

In order to identify features important for building the genomic prediction models, feature importance information was extracted from each model established with one of four methods: rrBLUP, BL, RF, and Ensemble. For rrBLUP, the importance metric was the marker effect ($u) calculated by mixed.solve in the R rrBLUP package. For BL, the importance metric was the estimated posterior mean ($ETA) calculated using the R BGLR package. The absolute value of marker effect and estimated posterior mean were used since the features are categorical with no particular meaning for the sign of importance metrics. For RF, the importance metric was the Gini importance, collected using the _importance_score function built into the Scikit-Learn implementation of RF. The Gini importance is the total decrease in node impurity (i.e. the homogeneity of classes in a node) after a particular feature is used to split a node. Node impurity decreases as instances from one of the classes are removed from the node, leaving a greater proportion of instances from the other class. Importance metrics from rrBLUP, BL, and RF were averaged over the 100 cross-validation replicates. Ensemble importance scores were calculated by normalizing the average importance scores from each model and each method between 0 and 1, then taking the mean of normalized importance scores across the three algorithms. Enrichment for transcript compared to genetic marker features within the top 1000 or top 20 features was done using Fisher's Exact Test, where the number of transcript features in and not in the top X features was compared to the number of genetic marker features in and not in the top X features.

To determine the degree to which the importance of a transcript correlates with the importance of nearby genetic markers, the genetic marker *G* with the greatest mean importance

score within a fixed window from the center of a genomic region *R* where a transcript *T* mapped to was selected among genetic markers in region *R,* referred to as a T:G pair (Figure 5.3A). To identify the effect of window size, a series of window sizes ranging from 1-80kb were tested. For each window size, the Spearman's Correlation (ρ) was calculated between the importance scores of T:G pairs. The window size with the highest correlation (2kb) was chosen (Supplemental Figure 5.3A). For this analysis, transcripts without location information or without one-to-one mapping between AGP V3 to V4 were removed, leaving 24,412 transcripts. With a window size of 2kb, additional transcripts were dropped because there was not a genetic marker within that window, resulting in 15,049 transcripts to be included in the downstream analysis. This analysis was repeated for the genome-wide genetic markers (G_{GW}) from (Bukowski *et al.* 2018).

To determine the degree to which the importance of a transcript correlated with the importance of *trans*-regulatory variants, significant eQTLs (multiple testing corrected $p<0.05$) were identified for each transcript using the linear regression (modelLINEAR) approach from MatrixeQTL implemented in R. Benjamini-Hochberg false discovery rate correction was used to adjust p for multiple testing and eQTLs were considered significant if adjusted $p < 0.05$ (Benjamini and Hochberg 1995). The distance for considering eQTL as *cis* was 1 mega base (Zan *et al.* 2016), however, because <0.1% of eQTL identified were *cis*, all eQTL were analyzed together. The importance of an eQTL or the neighboring genetic marker located within a 2kb window of the eQTL with the greatest average importance score was compared to the importance of the transcript with the eQTL in question (T:eQTL pair).

Enrichment of Gene Ontology (GO) terms associated with important transcripts compared to the reference genome was tested using agriGO v2 (Tian *et al.* 2017). The enrichment *p*-values are corrected for multiple testing by agriGOv2 using FDR. The top 10, 25,

and 100 transcripts from each algorithm, excluding the benchmark flowering time genes, were tested against the reference genome. The top 153 transcripts excluding benchmark genes (i.e. the top transcripts between the best two benchmark genes), from the ensemble algorithm and the union of the top 10, 25, and 100 transcripts from all four algorithms were tested.

5.5.6 Benchmark flowering time genes

We compiled a list of genes known to be involved in flowering time based on evidence from knockdown experiments (Muszynski *et al.* 2006; Danilevskaya *et al.* 2010; Meng *et al.* 2011; Lazakis *et al.* 2011; Alter *et al.* 2016) and/or association studies (Salvi *et al.* 2007; Hirsch *et al.* 2014). Genes were assigned confidence levels based on the type of evidence available, with experimental evidence considered high confidence, association study evidence and significant similarity with known flowering time genes from other species considered medium confidence (Supplemental Table 5.4). Because some of these genes did not have genetic markers located within the 2kb window of the center of the transcript, progressively larger windows were used to identify the most important nearby genetic marker up to 40kb. To compared importance scores across algorithms and between models using G or T data as input, percentiles were used. To determine if transcripts or genetic markers assigned to flowering time benchmark genes were associated with flowering time in this study, linear models and t-tests, respectively, implemented in R were used.

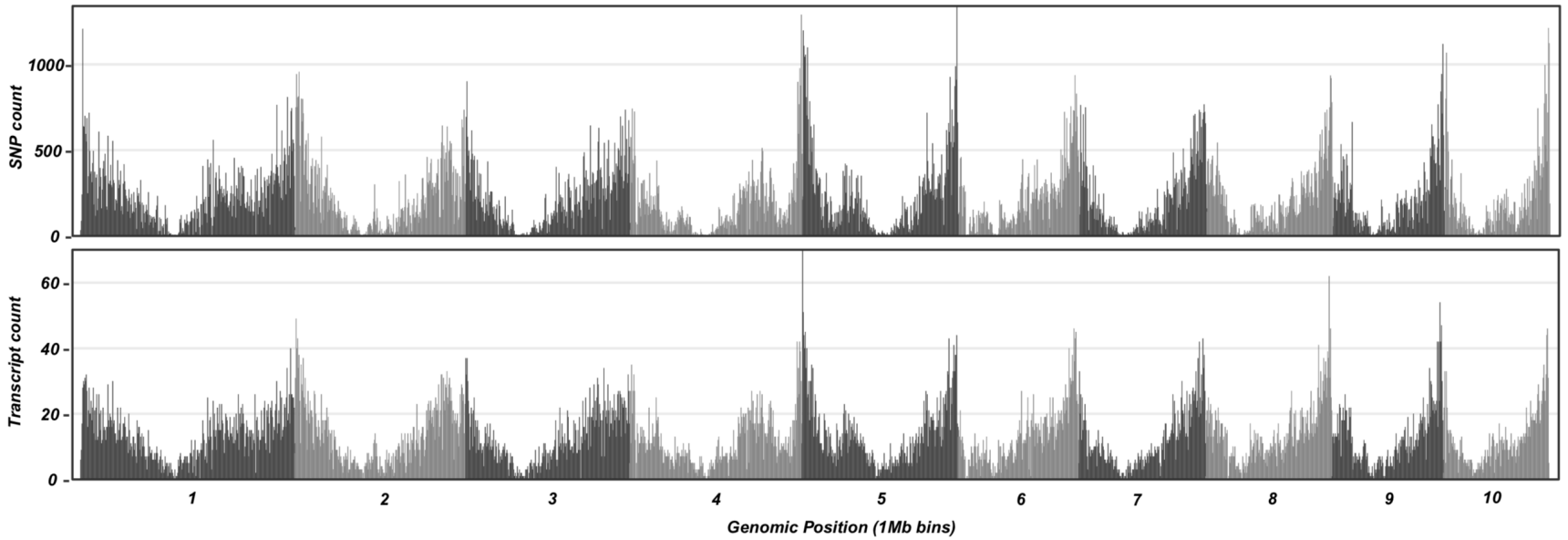
5.5.7 Data Availability

All data and code needed to reproduce the results from this study is available on GitHub (https://github.com/ShiuLab/Manuscript_Code/tree/master/2019_expression_GP/data).

5.6 Acknowledgements

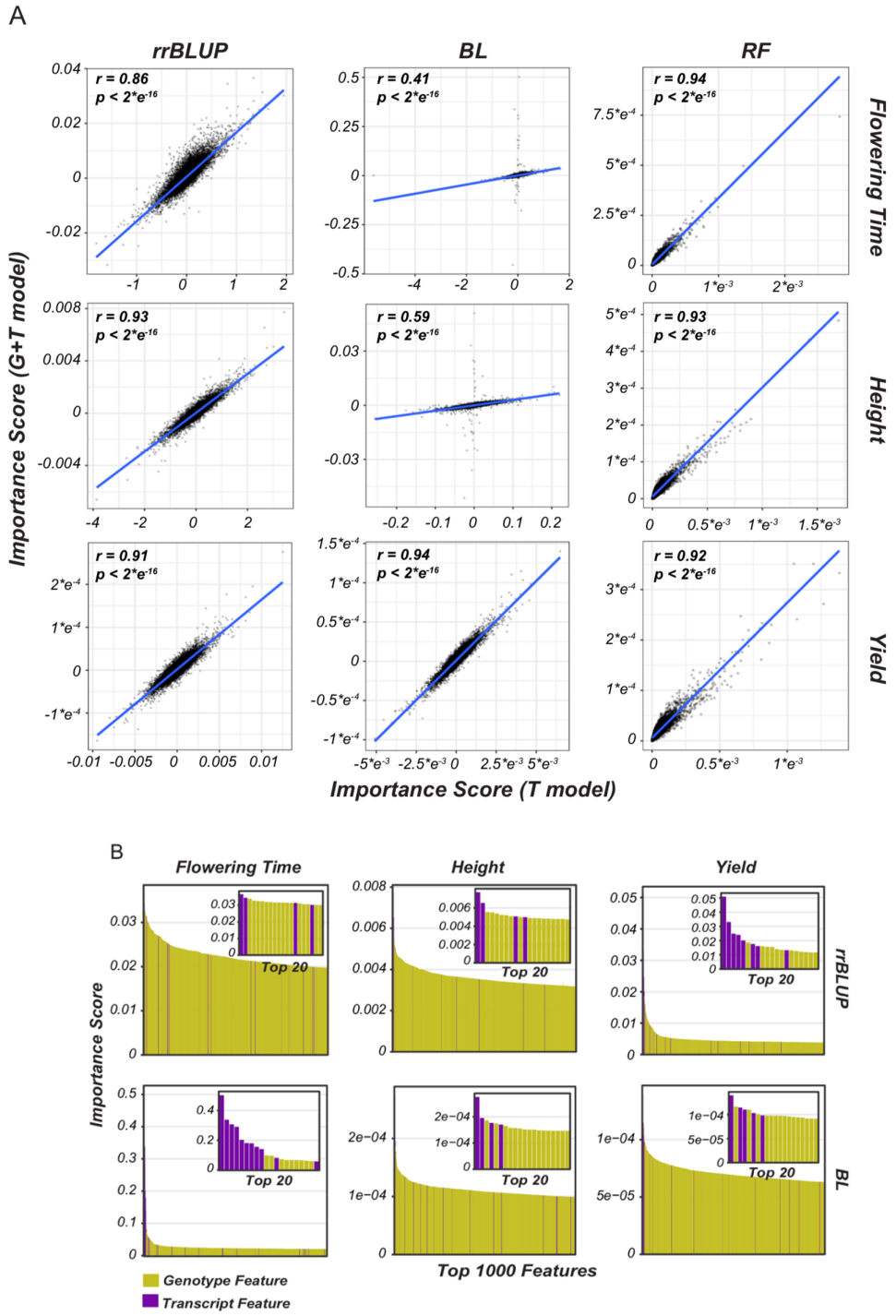
We thank Richard Amasino, Wolfgang Busch, and David Lowry for their help in interpreting our findings. This work was partly supported by NSF Graduate Research Fellowship (Fellow ID: 2015196719), Graduate Research Opportunities Abroad (GROW) Fellowship to C.B.A.; the U.S. Department of Energy Great Lakes Bioenergy Research Center (BER DE-SC0018409) and the National Science Foundation (IOS-1546617, DEB-1655386) to S.-H.S.

APPENDIX



Supplemental Figure 5.1. Distribution of genetic marker and transcript data across maize chromosomes.

Number of genetic markers (top) and transcripts (bottom) included in this study in 1 Mb bins across the maize chromosomes. Supports Figure 5.1.

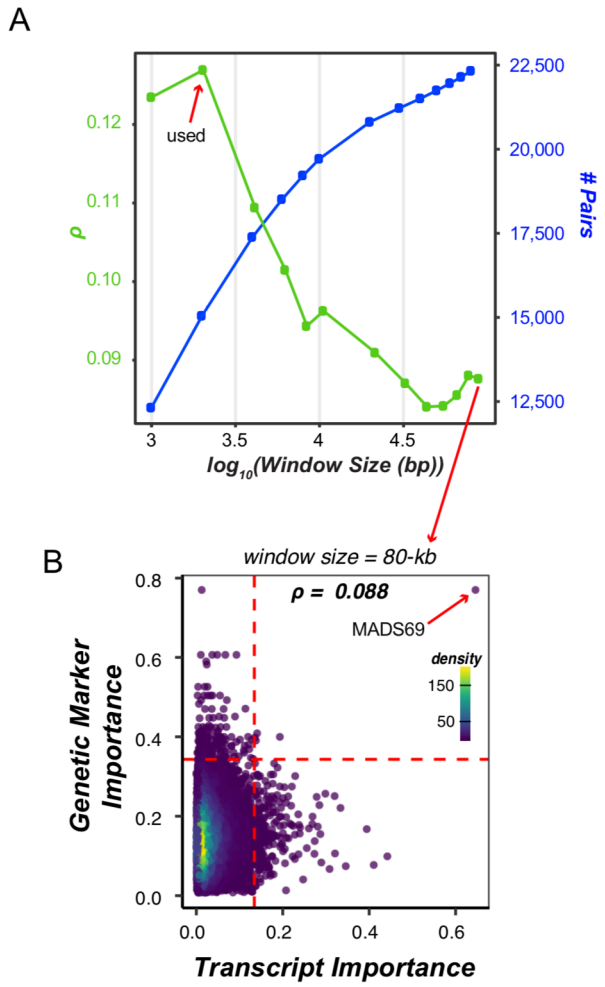


Supplemental Figure 5.2. Feature importance analysis for G+T models.

(A) Relationships between importance scores for transcripts from the T (X-axis) and G+T (Y-axis) flowering time prediction models established with rrBLUP (left column), BL (middle column), and RF (right column). The Pearson's Correlation Coefficient (r) is shown in the top

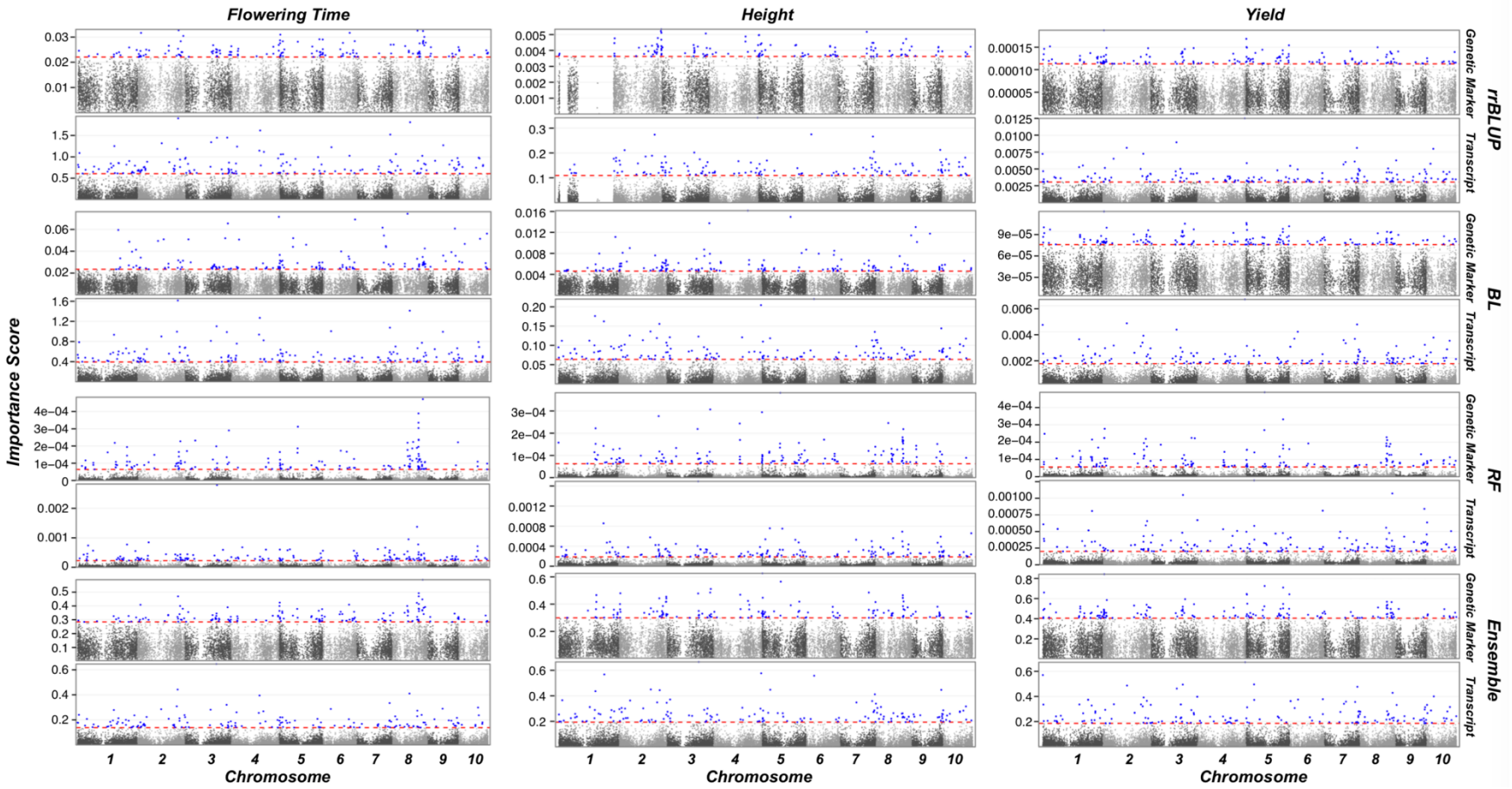
Supplemental Figure 5.2 (cont'd)

left corner. (B) Distribution of importance scores for the top 1,000 (inset = top 20) features from the G+T models for three traits using rrBLUP (top row) and BL (bottom row). Transcripts are in purple and genetic markers are in yellow. Supports Figure 5.2.



Supplemental Figure 5.3. Impact of transcript region sizes on importance correlation between transcript:genetic marker pairs.

(A) The correlation (green) between importance scores for transcript:genetic marker pairs and the number of pairs found (blue) as the transcript region size increases from 1-80 kb. (B) Density plot of the importance scores of genetic markers (Y-axis) and transcripts (X-axis) from T:G pairs using an 80 kb window size. The threshold was set (red dotted line) as the 99th percentile of the normalized importance score for each trait, algorithm, and input feature type. The correlation between transcript and genetic marker importance was calculated using Spearman's Rank (ρ). Supports Figure 5.3.



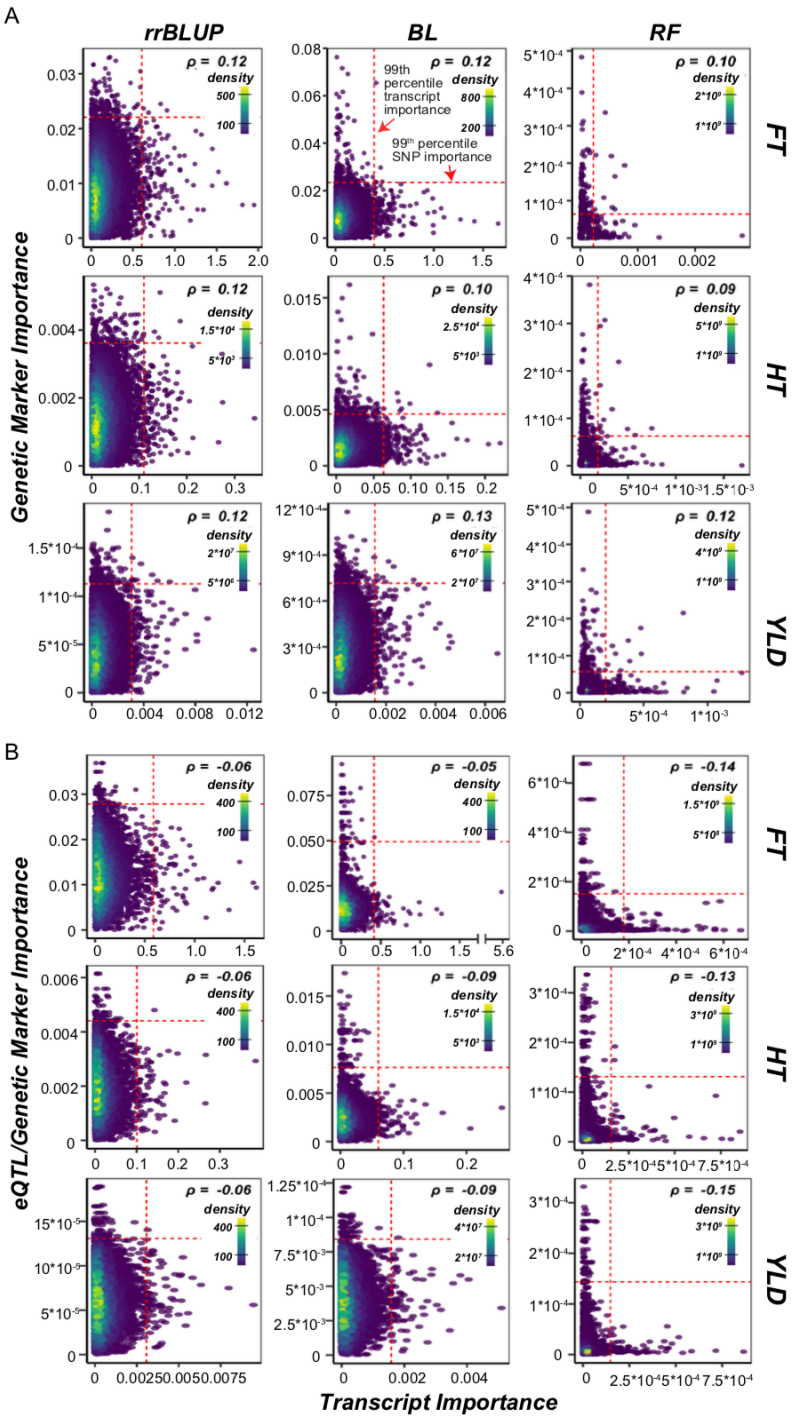
Supplemental Figure 5.4. Manhattan plot of importance scores from Genomic Prediction models.

Manhattan plots of genetic marker (top) and transcript (bottom) importance scores for predicting (A) flowering time, (B) height, and (C) yield. Threshold importance scores (dotted blue) were set at the 99th percentile importance score for each trait, algorithm, and

Supplemental Figure 5.4 (cont'd)

input feature type (i.e. genetic markers or transcripts). Genetic markers and transcripts falling above that threshold colored in blue.

Supports Figure 5.3.



Supplemental Figure 5.5. Correlation between genetic marker/eQTL and transcript importance.

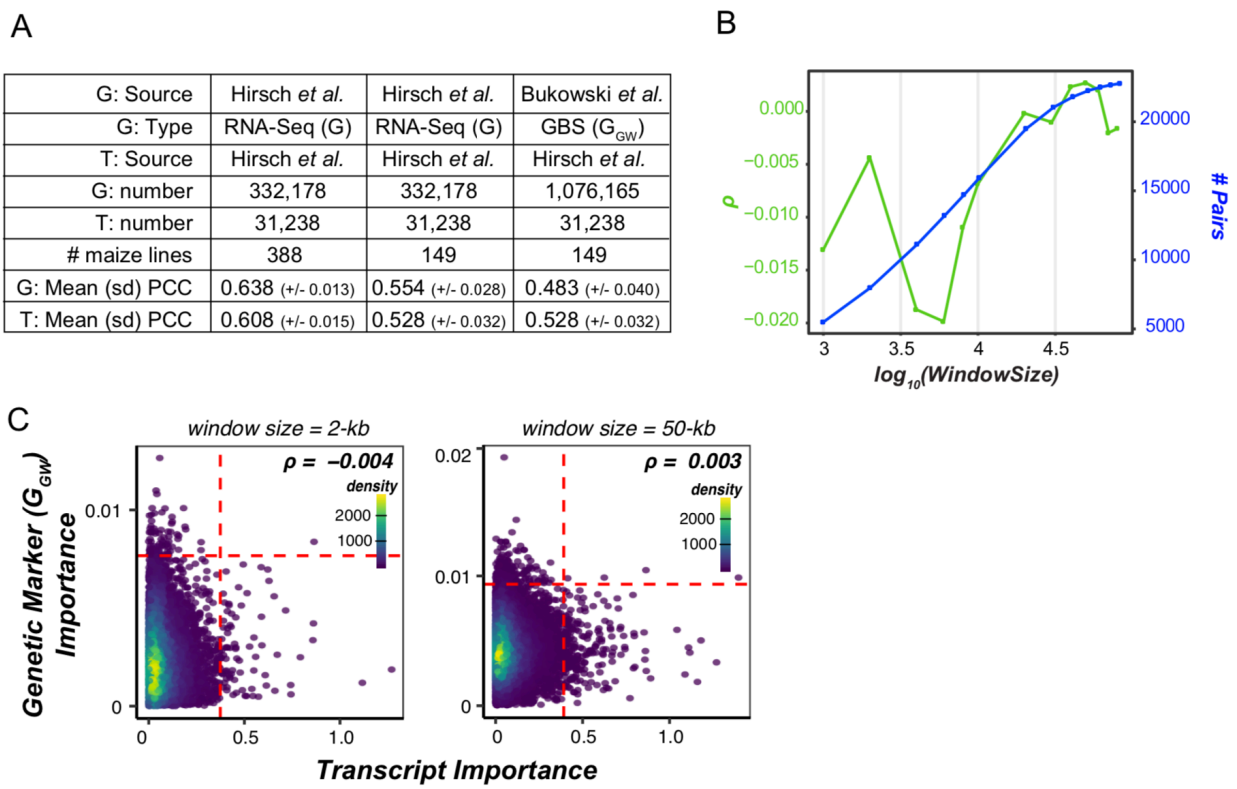
Density plot of the importance scores of (A) genetic markers (G, Y-axis) and transcripts (T, X-axis) from T:G pairs and (B) eQTL (eQTL, Y-axis) and transcripts (T, X-axis) from T:eQTL

Supplemental Figure 5.5 (cont'd)

pairs. The threshold was set (red dotted line) as the 99th percentile of the normalized importance score for each trait, algorithm, and input feature type. The correlation between transcript and genetic marker importance was calculated using Spearman's Rank (ρ). Supports Figure 5.3.

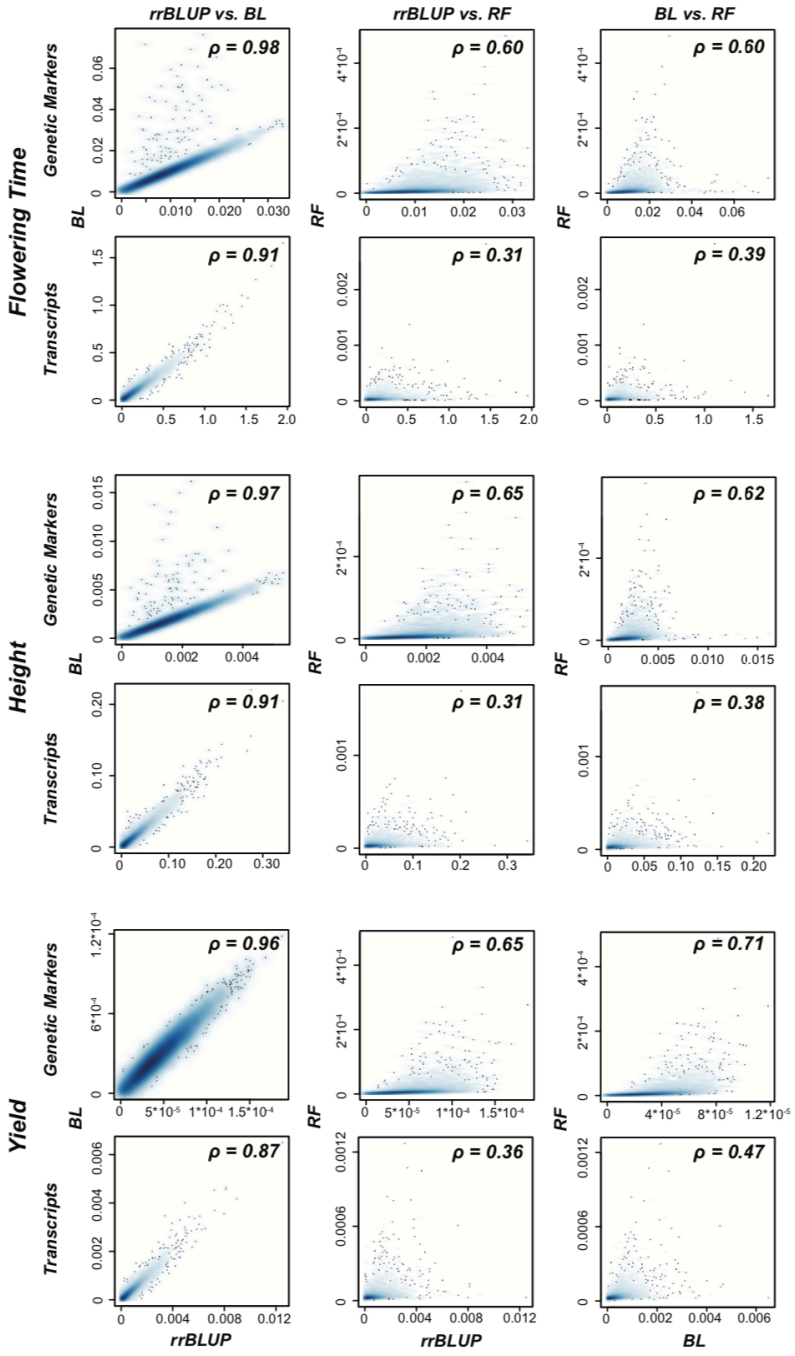
A

G: Source	Hirsch *et al.*	Hirsch *et al.*	Bukowski *et al.*
G: Type	RNA-Seq (G)	RNA-Seq (G)	GBS (G_{GW})
T: Source	Hirsch *et al.*	Hirsch *et al.*	Hirsch *et al.*
G: number	332,178	332,178	1,076,165
T: number	31,238	31,238	31,238
# maize lines	388	149	149
G: Mean (sd) PCC	0.638 (+/- 0.013)	0.554 (+/- 0.028)	0.483 (+/- 0.040)
T: Mean (sd) PCC	0.608 (+/- 0.015)	0.528 (+/- 0.032)	0.528 (+/- 0.032)

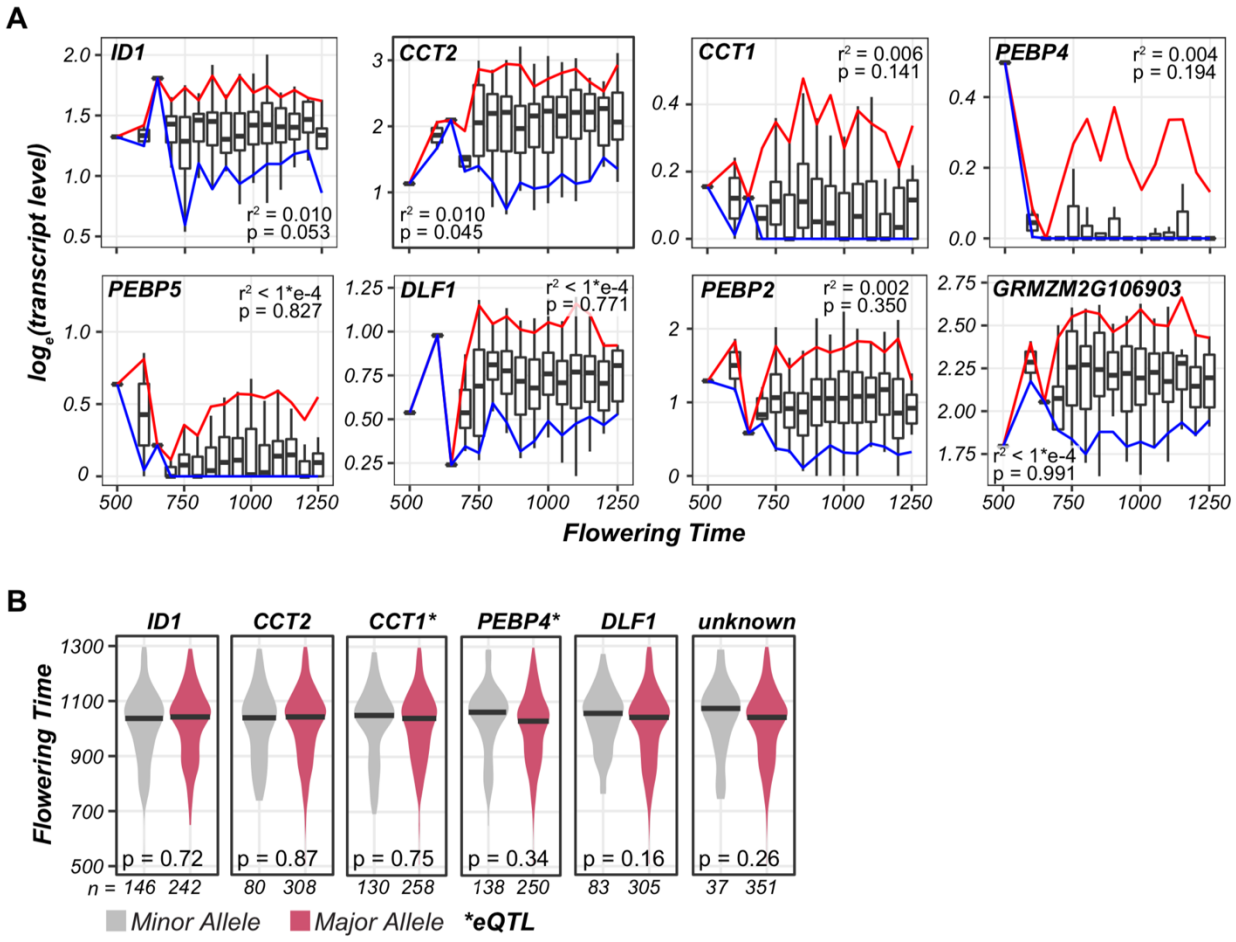


Supplemental Figure 5.6. Genomic prediction and genetic marker:transcript pairs using genome-wide genetic markers (G_{GW}).

(A) Summary of performance of rrBLUP models using the genome-wide genetic markers (G_{GW}) from Bukowski *et al*. compared to those using the RNA-Seq derived genetic markers (G) from Hirsch *et al.* (B) The correlation (green) between importance scores for Transcript (T)/G_{GW} pairs and the number of pairs found (blue) as the transcript region size increases from 1-80 kb (compare to analysis using Hirsch *et al.* genetic markers (G) in Supplemental Figure 5.3). (C) Density plot of the importance scores of G_{GW} (Y-axis) and T (X-axis) using a 2-kb (left) or a 50-kb (right) window. The threshold was set (red dotted line) as the 99th percentile of the normalized importance score for each trait, algorithm, and input feature type. The correlation between transcript and genetic marker importance was calculated using Spearman's Rank (ρ). Supports Figure 5.3.



Supplemental Figure 5.7. Correlation between feature importance between algorithms. Density scatter plot of the importance scores of genetic markers (top) and transcripts (bottom) generated with rrBLUP and BL (left), rrBLUP and RF (middle), as well as BL and RF (right). The correlation between importance scores between algorithms was calculated using Spearman's Rank (ρ). Supports Figure 5.3.



Supplemental Figure 5.8. Relationship between transcript levels and alleles and flowering time for benchmark genes.

(A) Boxplots show the median transcript level (log(Fold-Change)) for each flowering time (Growing Degree Days (GDD)/100) bin with the 95th (red) and 5th (blue) percentiles shown. Linear models were fit and adjusted r^2 and p-values are shown. (B) Violin-plots of the distribution of flowering time (GDD/100) for lines with the major (blue) or minor (gray) allele for the genetic marker paired with each benchmark gene. Significant differences in the GDD by allele were tested for using t-tests. Supports Figure 5.5.

Supplemental File 5.1. Top 1000 most important transcripts for each trait from the transcript-based Ensemble models.

Supplemental File 5.1 can be found at the following link:
http://www.plantcell.org/content/early/2019/10/22/tpc.19.00332/tab-figures-data

Supplemental File 5.2. Account of data (Genetic Marker, Transcript, Phenotype) availability for maize lines and decision to include line in the study

Supplemental File 5.2 can be found at the following link:
http://www.plantcell.org/content/early/2019/10/22/tpc.19.00332/tab-figures-data

REFERENCES

REFERENCES

Albert, F. W., and L. Kruglyak, 2015 The role of regulatory variation in complex traits and disease. Nat. Rev. Genet. 16: 197–212.

Alter, P., S. Bircheneder, L.-Z. Zhou, U. Schlüter, M. Gahrtz *et al.*, 2016 Flowering Time-Regulated Genes in Maize Include the Transcription Factor ZmMADS1. Plant Physiol. 172: 389–404.

Becker, J., J. R. Wendland, B. Haenisch, M. M. Nöthen, and J. Schumacher, 2012 A systematic eQTL study of cis-trans epistasis in 210 HapMap individuals. Eur. J. Hum. Genet. 20: 97–101.

Benjamini, Y., and Y. Hochberg, 1995 Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. J. R. Stat. Soc. Series B Stat. Methodol. 57: 289–300.

Bermingham, M. L., R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan *et al.*, 2015 Application of high-dimensional feature selection: evaluation for genomic prediction in man. Scientific Reports 1–12.

Bradbury, P. J., Z. Zhang, D. E. Kroon, T. M. Casstevens, Y. Ramdoss *et al.*, 2007 TASSEL: software for association mapping of complex traits in diverse samples. Bioinformatics 23: 2633–2635.

Buckler, E. S., J. B. Holland, P. J. Bradbury, C. B. Acharya, P. J. Brown *et al.*, 2009 The genetic architecture of maize flowering time. Science 325: 714–718.

Bukowski, R., X. Guo, Y. Lu, C. Zou, B. He *et al.*, 2018 Construction of the third-generation Zea mays haplotype map. Gigascience 7: 1–12.

Burgueño, J., G. de los Campos, K. Weigel, and J. Crossa, 2012 Genomic prediction of breeding values when modeling genotype$\times$ environment interaction using pedigree and dense molecular markers. Crop Sci. 52: 707–719.

Cuevas, J., J. Crossa, O. A. Montesinos-López, J. Burgueño, P. Pérez-Rodríguez *et al.*, 2017 Bayesian Genomic Prediction with Genotype × Environment Interaction Kernel Models. G3 7: 41–53.

Danilevskaya, O. N., X. Meng, and E. V. Ananiev, 2010 Concerted modification of flowering time and inflorescence architecture by ectopic expression of TFL1-like genes in maize. Plant Physiol. 153: 238–251.

Dietterich, T. G., 2000 Ensemble methods in machine learning. International workshop on multiple classifier systems.

Drinkwater, N. R., and M. N. Gould, 2012 The long path from QTL to gene. PLoS Genet. 8: e1002975.

Endelman, J. B., 2011 Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome.

Endelman, J. B., G. N. Atlin, Y. Beyene, K. Semagn, X. Zhang *et al.*, 2014 Optimal Design of Preliminary Yield Trials with Genome-Wide Markers. Crop Sci. 54: 48–59.

Endelman, J. B., and J.-L. Jannink, 2012 Shrinkage estimation of the realized relationship matrix. G3 2: 1405–1413.

Frisch, M., A. Thiemann, J. Fu, T. A. Schrag, S. Scholten *et al.*, 2010 Transcriptome-based distance measures for grouping of germplasm and prediction of hybrid performance in maize. Theor. Appl. Genet. 120: 441–450.

Fu, J., K. C. Falke, A. Thiemann, T. A. Schrag, A. E. Melchinger *et al.*, 2012 Partial least squares regression, support vector machine regression, and transcriptome-based distances for prediction of maize hybrid performance with gene expression data. Theor. Appl. Genet. 124: 825–833.

González-Reymúndez, A., G. de los Campos, L. Gutiérrez, S. Y. Lunt, and A. I. Vazquez, 2017 Prediction of years of life after diagnosis of breast cancer using omics and omic-by-treatment interactions. Eur. J. Hum. Genet. 25: 538–544.

Granato, I., J. Cuevas, F. Luna-Vázquez, J. Crossa, O. Montesinos-López *et al.*, 2018 BGGE: A New Package for Genomic-Enabled Prediction Incorporating Genotype × Environment Interaction Models. G3 8: 3039–3047.

Guo, Z., M. M. Magwire, C. J. Basten, Z. Xu, and D. Wang, 2016 Evaluation of the utility of gene expression and metabolic information for genomic prediction in maize. Theoretical and Applied Genetics 129: 2413–2427.

Hanley, M. E., and O. C. May, 2006 Cotyledon damage at the seedling stage affects growth and flowering potential in mature plants. New Phytol. 169: 243–250.

Hansey, C. N., J. M. Johnson, R. S. Sekhon, S. M. Kaeppler, and N. de Leon, 2011 Genetic diversity of a maize association population with restricted phenology. Crop Sci. 51: 704–715.

Heffner, E. L., M. E. Sorrells, and J.-L. Jannink, 2009 Genomic Selection for Crop Improvement. Crop Science 49: 1–12.

Heslot, N., H.-P. Yang, M. E. Sorrells, and J.-L. Jannink, 2012 Genomic Selection in Plant Breeding: A Comparison of Models. Crop Science 52: 146–15.

Hirsch, C. N., J. M. Foerster, J. M. Johnson, R. S. Sekhon, G. Muttoni *et al.*, 2014 Insights into the maize pan-genome and pan-transcriptome. Plant Cell 26: 121–135.

Jia, J., Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, 2015 iPPI-Esml: An ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC. J. Theor. Biol. 377: 47–56.

Jonas, E., and D.-J. de Koning, 2013 Does genomic selection have a future in plant breeding? Trends Biotechnol. 31: 497–504.

Kang, T., W. Ding, L. Zhang, D. Ziemek, and K. Zarringhalam, 2017 A biological network-based regularized artificial neural network model for robust phenotype prediction from gene expression data. BMC Bioinformatics 18: 565.

Keever, G. J., J. R. Kessler Jr, G. B. Fain, and D. C. Mitchell, 2015 Seedling Developmental Stage at Transplanting Affects Growth and Flowering of Medallion Flower and Globe Amaranth. J. Environ. Hortic. 33: 53–57.

Lazakis, C. M., V. Coneva, and J. Colasanti, 2011 ZCN8 encodes a potential orthologue of Arabidopsis FT florigen that integrates both endogenous and photoperiod flowering signals in maize. J. Exp. Bot. 62: 4833–4842.

Leo Breiman Statistics, L. B., 2001 Random Forests.

Li, Z., N. Gao, J. W. R. Martini, and H. Simianer, 2019 Integrating Gene Expression Data Into Genomic Prediction. Front. Genet. 10: 126.

Meng, X., M. G. Muszynski, and O. N. Danilevskaya, 2011 The FT-like ZCN8 Gene Functions as a Floral Activator and Is Involved in Photoperiod Sensitivity in Maize. Plant Cell 23: 942–960.

Meuwissen, T. H. E., B. J. Hayes, and M. E. Goddard, 2001 Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps. Genetics 1–11.

Muszynski, M. G., T. Dam, B. Li, D. M. Shirbroun, Z. Hou *et al.*, 2006 delayed flowering1 Encodes a basic leucine zipper protein that mediates floral inductive signals at the shoot apex in maize. Plant Physiol. 142: 1523–1536.

Nordborg, M., J. O. Borevitz, J. Bergelson, C. C. Berry, J. Chory *et al.*, 2002 The extent of linkage disequilibrium in Arabidopsis thaliana. Nature Genetics 30: 190–193.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, 2011 Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12: 2825–2830.

Pérez, P., and G. de los Campos, 2014 Genome-wide regression and prediction with the BGLR statistical package. Genetics 198: 483–495.

Ribaut, J.-M., and M. Ragot, 2007 Marker-assisted selection to improve drought adaptation in maize: the backcross approach, perspectives, limitations, and alternatives. Journal of Experimental Botany 58: 351–360.

Salvi, S., G. Sponza, M. Morgante, D. Tomes, X. Niu *et al.*, 2007 Conserved noncoding genomic sequences associated with a flowering-time quantitative trait locus in maize. Proc. Natl. Acad. Sci. U. S. A. 104: 11376–11381.

Schaefer, R. J., J.-M. Michno, J. Jeffers, O. Hoekenga, B. Dilkes *et al.*, 2018 Integrating Coexpression Networks with GWAS to Prioritize Causal Genes in Maize. Plant Cell 30: 2922–2942.

Schrag, T. A., M. Westhues, W. Schipprack, F. Seifert, A. Thiemann *et al.*, 2018 Beyond Genomic Prediction: Combining Different Types of omics Data Can Improve Prediction of Hybrid Performance in Maize. Genetics 208: 1373–1385.

Shen, H.-B., and K.-C. Chou, 2006 Ensemble classifier for protein fold pattern recognition. Bioinformatics 22: 1717–1722.

Solberg Woods, L. C., 2014 QTL mapping in outbred populations: successes and challenges. Physiol. Genomics 46: 81–90.

Song, Y. H., S. Ito, and T. Imaizumi, 2013 Flowering time regulation: photoperiod- and temperature-sensing in leaves. Trends in Plant Science 18: 575–583.

Spindel, J., H. Begum, D. Akdemir, P. Virk, B. Collard *et al.*, 2015 Genomic Selection and Association Mapping in Rice (Oryza sativa): Effect of Trait Genetic Architecture, Training Population Composition, Marker Number and Statistical Model on Accuracy of Rice Genomic Selection in Elite, Tropical Rice Breeding Lines. PLoS Genetics 11: e1004982–25.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn, 2007 Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics 8:.

Tenaillon, M. I., M. C. Sawkins, A. D. Long, R. L. Gaut, J. F. Doebley *et al.*, 2001 Patterns of DNA sequence polymorphism along chromosome 1 of maize (Zea mays ssp. mays L.). Proc. Natl. Acad. Sci. U. S. A. 98: 9161–9166.

Tian, T., Y. Liu, H. Yan, Q. You, X. Yi *et al.*, 2017 agriGO v2.0: a GO analysis toolkit for the agricultural community, 2017 update. Nucleic Acids Res. 45: W122–W129.

US Patent Application for MODIFYING FLOWERING TIME IN MAIZE Patent Application (Application #20140366213 issued December 11, 2014) - Justia Patents Search.

Wills, Q. F., K. J. Livak, A. J. Tipping, T. Enver, A. J. Goldson *et al.*, 2013 Single-cell gene expression analysis reveals genetic associations masked in whole-tissue experiments. Nat. Biotechnol. 31: 748–752.

Zan, Y., X. Shen, S. K. G. Forsberg, and Ö. Carlborg, 2016 Genetic Regulation of Transcriptional Variation in Natural Arabidopsis thaliana Accessions. G3 6: 2319–2328.

Zarringhalam, K., D. Degras, C. Brockel, and D. Ziemek, 2018 Robust phenotype prediction from gene expression data using differential shrinkage of co-regulated genes. Sci. Rep. 8: 1237.

Zenke-Philippi, C., A. Thiemann, F. Seifert, T. Schrag, A. E. Melchinger *et al.*, 2016 Prediction of hybrid performance in maize with a ridge regression model employed to DNA markers and mRNA transcription profiles. BMC Genomics 17: 262.

CHAPTER SIX: PERCEPTIONS OF EMERGING BIOTECHNOLOGIES [1]

[1] This chapter has been published in the following manuscript

Christina B. Azodi and Thomas Dietz (2019) Perceptions of Emerging Biotechnologies. *Environmental Research Letters*. DOI: 10.1088/1748-9326/ab44332

© IOP Publishing. Reproduced with permission. All rights reserved.

6.1 Abstract

Research on public views of biotechnology has centered on genetically modified (GM) foods. However, as the breadth of biotechnology applications grows, a better understanding of public concerns about non-agricultural biotechnology products is needed in order to develop proactive strategies to address these concerns. Here, we explore the perceived benefits and risks associated with five biotechnology products and how those perceptions translate into public opinion about the use and regulation of biotechnology in the United States. While we found greater support for non-agricultural biotechnology product, 70% of individuals surveyed showed no or little variation in their support across the products, indicating opinions about early GM products may be influencing the acceptance of emerging biotechnologies. We identified five common patterns of opinions about biotechnology and used machine learning models to integrate a wide range of factors and predict a respondent's opinion group. While the model was particularly good at identifying individuals supportive of biotechnology, differentiating between individuals from the non- and conditionally-supportive opinion groups was more challenging, emphasizing the complexity of public opinions of emerging biotechnology products.

The full article is available at the following link:

https://iopscience.iop.org/article/10.1088/1748-9326/ab4433