Title
Domain-inspired machine learning for hypothesis extraction in biological data

Permalink
https://escholarship.org/uc/item/59q3w1bp

Author
Kumbier, Karl

Publication Date
2019

Peer reviewed|Thesis/dissertation
Domain-inspired machine learning for hypothesis extraction in biological data

by

Karl Kumbier

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy
in
Statistics
in the
Graduate Division
of the
University of California, Berkeley

Committee in charge:

Professor Bin Yu, Chair
Professor Jennifer Listgarten
Professor James B. Brown

Spring 2019
Domain-inspired machine learning for hypothesis extraction in biological data

Copyright 2019
by
Karl Kumbier
Abstract

Domain-inspired machine learning for hypothesis extraction in biological data

by

Karl Kumbier

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Bin Yu, Chair

Rapidly moving technologies are transforming the rate at which researchers accumulate information. Large, rich datasets hold promises of new insights into complex natural phenomena that will help advance the frontier of science. Here we aim to develop new statistics/data science principles and scalable algorithms for extracting reliable and reproducible information from these data.

Chapter 1 provides an overview of the work contained in this thesis. It discusses the growing availability of genomic data and the statistical machine learning tools that are being used to provide a systems-level understanding of genomic phenomena.

Chapter 2 introduces the predictability, computability, and stability (PCS) framework. The PCS framework builds on key ideas in machine learning, using predictability as a reality check and evaluating computational considerations in data collection, data storage and algorithm design. It augments predictability and computability with an overarching stability principle, which expands statistical uncertainty considerations to assesses how results vary with respect to choices (or perturbations) made across the data science life cycle. In this chapter, we develop PCS inference through perturbation intervals and PCS hypothesis testing to investigate the reliability of data results. We compare PCS inference with existing methods in high-dimensional sparse linear model simulations to demonstrate that our approach compares favorably to others, in terms of ROC curves, over a wide range of simulation settings. Finally, we propose documentation based on R Markdown, iPython, or Jupyter Notebook, with publicly available, reproducible codes and narratives to justify human choices made throughout an analysis.

As an example of the PCS framework in practice, chapter 3 develops the iterative Random Forest algorithm (iRF). iRF trains a feature-weighted ensemble of decision trees to detect stable, high-order interactions with same order of computational cost as Random Forests (RF). We demonstrate the utility of iRF for high-order interaction discovery in two prediction problems: enhancer activity in the early Drosophila embryo and alternative splicing of primary transcripts in human derived cell lines. In Drosophila, 80% of the pairwise transcription factor interactions iRF identified as stable have been previously reported
as physical interactions. Moreover, novel third-order interactions, e.g. between Zelda (Zld), Giant (Gt), and Twist (TwI), suggest high-order relationships that are candidates for follow-up experiments. In human-derived cells, iRF re-discovered a central role of H3K36me3 in chromatin-mediated splicing regulation, and identified novel 5th and 6th order interactions, indicative of multi-valent nucleosomes with specific roles in splicing regulation. By decoupling the order of interactions from the computational cost of identification, iRF opens new avenues of inquiry into the molecular mechanisms underlying genome biology.

Chapter 4 refines iRF to explicitly map responses as a function of interacting features. Our proposed method, signed iRF (siRF), describes “subsets” of rules that frequently occur on RF decision paths. We refer to these rule subsets as signed interactions. RF decision paths containing the same signed interaction share not only a set of interacting features but also exhibit similar thresholding behavior, and thus describe a consistent functional relationship between interacting features and responses. We formulate stable and predictive importance metrics (SPIMs) to rank signed interactions in terms of their stability, predictive accuracy, and strength of interaction. For each SPIM, we define null importance metrics that characterize its expected behavior under known structure. We evaluate siRF in biologically inspired simulations and two case studies: predicting enhancer activity and spatial gene expression patterns. In the case of spatial gene expression patterns, siRF recovered all 11 reported links in the gap gene network. In the case of enhancer activity, siRF discovered rules that identify enhancer elements in Drosophila embryos with high precision, suggesting candidate biological mechanisms for experimental studies. By refining the process of interaction discovery, siRF has the potential to guide mechanistic inquiry into systems whose scale and complexity is beyond human comprehension.
To William K. Kumbier
for guiding my first steps into science
# Contents

## List of Figures

List of Figures

## List of Tables

List of Tables

## 1 Overview

1 Overview

## 2 Three Principles of Data Science: Predictability, Computability, and Stability

2 Three Principles of Data Science: Predictability, Computability, and Stability

1. Introduction

2. The PCS framework

3. Connections in the PCS framework

4. PCS inference through perturbation analysis

5. PCS recommendation system for scientific hypothesis generation

6. PCS workflow documentation

7. Conclusion

## 3 Iterative Random Forests (iRF)

3 Iterative Random Forests (iRF)

1. Introduction

2. Our method: iterative random forests

3. Case study I: enhancer elements in *Drosophila*

4. Case study II: alternative splicing in a human-derived cell line

5. Discussion

## Appendices

Appendices

1. Algorithms

2. Remarks on iRF

3. Data collection and preprocessing

4. Evaluating *Drosophila* enhancer interactions

5. Simulation experiments

6. Computational cost of detecting high-order interaction

## Appendices

Appendices

1. Algorithms

2. Remarks on iRF

3. Data collection and preprocessing

4. Evaluating *Drosophila* enhancer interactions

5. Simulation experiments

6. Computational cost of detecting high-order interaction
# 4 Signed Iterative Random Forests (siRF)

4.1 Introduction .......................................................... 79
4.2 Background ............................................................ 82
4.3 Signed iterative random forests: an enhanced interpretation of iRF .... 83
4.4 Simulations ............................................................... 93
4.5 Extracting regulatory interactions in *Drosophila embryos* ............ 98
4.6 Discussion ............................................................... 108

## Appendices

4.A Interpretable prediction through signed interactions ................. 110
4.B Region boundaries and response surfaces .......................... 111
4.C Redundant feature selection on decision paths ....................... 119
4.D Data processing and predictive formulation .......................... 120

## Bibliography

130
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The data science life cycle</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison of PCS and traditional inference</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Connecting models with reality</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>iRF workflow</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Predicting enhancer activity using iRF</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Predicting alternative splicing using iRF</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>Prediction accuracy tuning parameters (enhancer)</td>
<td>61</td>
</tr>
<tr>
<td>3.5</td>
<td>Prediction accuracy tuning parameters (splicing)</td>
<td>62</td>
</tr>
<tr>
<td>3.6</td>
<td>Interaction recovery tuning parameters (enhancer)</td>
<td>63</td>
</tr>
<tr>
<td>3.7</td>
<td>Interaction recovery tuning parameters (splicing)</td>
<td>64</td>
</tr>
<tr>
<td>3.8</td>
<td>Prediction-based interaction importance (enhancer)</td>
<td>65</td>
</tr>
<tr>
<td>3.9</td>
<td>Prediction-based interaction importance (splicing)</td>
<td>66</td>
</tr>
<tr>
<td>3.10</td>
<td>iRF performance: order-4 AND, OR, and XOR rules</td>
<td>67</td>
</tr>
<tr>
<td>3.11</td>
<td>iRF performance: order-8 XOR rule</td>
<td>68</td>
</tr>
<tr>
<td>3.12</td>
<td>iRF weight distribution by iteration</td>
<td>69</td>
</tr>
<tr>
<td>3.13</td>
<td>iRF feature selection by marginal importance</td>
<td>70</td>
</tr>
<tr>
<td>3.14</td>
<td>iRF performance: mixture of order-8 XOR rule and order-4 AND rule</td>
<td>71</td>
</tr>
<tr>
<td>3.15</td>
<td>iRF performance: order-8 XOR rule, decaying covariance</td>
<td>72</td>
</tr>
<tr>
<td>3.16</td>
<td>iRF performance: order-8 XOR rule, block covariance</td>
<td>73</td>
</tr>
<tr>
<td>3.17</td>
<td>iRF performance: order-4 AND rule, large ( p ), noise level 0.2</td>
<td>74</td>
</tr>
<tr>
<td>3.18</td>
<td>iRF performance: order-4 AND rule, big ( p ), noise level 0.3</td>
<td>75</td>
</tr>
<tr>
<td>3.19</td>
<td>iRF performance: enhancer data simulation</td>
<td>76</td>
</tr>
<tr>
<td>3.20</td>
<td>iRF enhancer data simulation stability scores</td>
<td>77</td>
</tr>
<tr>
<td>3.21</td>
<td>Runtime comparison of iRF and Rulefit</td>
<td>78</td>
</tr>
<tr>
<td>4.1</td>
<td>Rule-based generative models</td>
<td>80</td>
</tr>
<tr>
<td>4.2</td>
<td>Signed interaction feature mapping</td>
<td>85</td>
</tr>
<tr>
<td>4.3</td>
<td>siRF workflow</td>
<td>86</td>
</tr>
<tr>
<td>4.4</td>
<td>Class prevalence enrichment (CPE)</td>
<td>89</td>
</tr>
<tr>
<td>4.5</td>
<td>Feature selection dependence (FSD)</td>
<td>90</td>
</tr>
<tr>
<td>4.6</td>
<td>Mean increase in precision (MIP)</td>
<td>91</td>
</tr>
</tbody>
</table>
4.7 Interaction recovery in simulation experiments ......................... 97
4.8 siRF predictions and response surfaces (Gt) .......................... 100
4.9 siRF predictions and response surfaces (Kni) .......................... 101
4.10 ROC and PR curves for predicting enhancer activity by stage ........ 104
4.11 siRF predictions by stage .............................................. 104
4.12 Interaction specific predictions of enhancer activity .................. 105
4.13 Enhancer activity response surfaces .................................... 106
4.14 Raw ChIP-seq profiles surrounding enhancer elements ............... 107
4.15 Response surfaces for Gaussian data AND model ..................... 113
4.16 Response surfaces for Drosophila data AND model .................. 114
4.17 Response surfaces for Gaussian data OR model ....................... 115
4.18 Response surfaces for Drosophila data OR model .................... 116
4.19 Response surfaces for Gaussian data ADD model ...................... 117
4.20 Response surfaces for Drosophila data ADD model .................. 118
4.21 Gap gene network ....................................................... 123
4.22 ROC and PR curves for predicting enhancer activity by threshold ..... 128
List of Tables

3.1 Previously reported pairwise TF interactions recovered by iRF ............... 51
4.1 Generative model parameters for siRF simulation experiments ................. 96
4.2 Gap gene interactions recovered by siRF .................................. 126
4.4 Enhancer related TF interactions ............................................ 127
4.3 Gap gene network validation .................................................. 129
Acknowledgments

Both before and during my time at UC Berkeley, I have had the great privilege to work with many exceptional scholars, scientists, mentors, and educators. These individuals have challenged me to think deeper and explore new areas, culminating in the work presented in this thesis. I am incredibly grateful for the generous support of everyone who has contributed to my intellectual journey.

First, I would like to thank my advisor Professor Bin Yu. She has played an integral role in shaping my approach to and taste in research. Prof. Yu provided me the opportunity and guidance to explore scientific problems with creativity and rigor. Her insights have led me to new and interesting problems, her constructive feedback has helped me clarify and communicate my thoughts, and her high standard of scientific research has improved the quality of my work. Prof. Yu is an exceptional role model, both through her academic principles and desire to contribute to the world around her. I am incredibly fortunate to have benefitted from her mentorship.

I am also grateful for the mentorship and support of the members of the Berkeley Drosophila genome project. I would like to thank Susan Celniker for the many hours she devoted to our work. Her extensive expertise provided direction to the scientific questions we investigated, and her precise explanations helped me learn to communicate with a broad range of scientists. I am thankful for the many stimulating discussions with Dr. Ben Brown, both statistical and biological, that helped guide the work presented in this thesis. Dr. Brown’s enthusiastic support and ambitious vision encouraged us to seek creative solutions to challenging scientific questions. I thank Dr. Erwin Frise for bringing me on to the project and taking the time to explain fundamental concepts in systems biology. Conversations with Dr. Frise challenged me to examine the complexity of biological processes and left me excited about open questions in the field. I am indebted to Dr. Ann Hammonds and Dr. William Fisher, for helping a young statistician learn about embryonic development. Their ability to distill meaningful biological insights from complex data enriched my understanding of the questions we sought to answer.

In addition to my mentors, I am grateful for my time with all the students in the Yu group. It has been a pleasure working with such thoughtful scholars. Their challenging questions and continued support have refined my thinking and helped me communicate my work with clarity. In particular, I would like to thank Siqi Wu, for working close with me as I began my journey into the field of genomics; Sumanta Basu, who is co-author on the work presented in chapters 3 and 4, for the many Skype calls spent refining our work; Jamie Murdoch, Chandan Singh, and Reza Abbasi-Asl for the stimulating debates on interpretable machine learning; Rebecca Barter, for her constructive feedback on presentations; Bryan Liu, Hue Wang, Xiao Li, and Tiffany Tang, for joining me on the genomics team; Shamindra Shrotriya, for his organizational skills and efforts to bring our work to broader audiences; Sujayam Saha, for the many meetings over coffee dreaming up creative connections between theory and practice; Adam Bloniarz, for being a calm and patient mentor in my early years as a graduate student; and all the other members that have made me think deeper about
the problems I work on: Yuanshi Chen, Simon Walter, Sören Kunzel, Raaz Dwivedi, Nick Altieri, Hanzhong Liu, Christine Kuang, and Sivaraman Balakrishnan.

Beyond the members of the Yu group, it has been a pleasure interacting with the community of students, administrators, and technical support staff within the UC Berkeley statistics department. I want to thank all the classmates who struggled through problem sets with me, helped plan SGSA events, and let loose at Friday wind downs and foosball tournaments. I owe a debt of gratitude to those who took the time and effort to help me navigate life as a student at Berkeley. La Shana Polaris made it simple and easy to handle administrative issues that came up along the way. Mary Melinn helped coordinate any department resource that I needed access to. Keyla Gomez quickly processed department reimbursements, so I could pay rent on time. Chris Paciorek, Ryan Lovett, and Luis Torres provided invaluable support for the department’s computing resources.

Finally, I would like to thank all the friends and family members who supported me along the way. You provided the foundation and perspective that shaped my interests, tolerated me when my mind drifted to other places, invested time in understanding my work, and encouraged me to push through my doubts. This thesis would not have been possible without your love, patience, and encouragement.
Chapter 1

Overview

Biological phenomena, from embryonic development to human health, are driven by dynamic, high-order interactions. For instance, interactions among transcription factors (TFs), epigenetic factors, and DNA regulate precise programs of spatio-temporal gene expression; interactions among genetic and environmental elements establish human susceptibility to disease and responsiveness to treatment; interactions among bacteria establish microbiomes that influence the physiology and fitness of host organisms. Despite the recent deluge of biological data, our understanding of how biological interactions coordinate development and function quite limited. However, advances in statistical machine learning hold the promise of guiding new insights. Sophisticated algorithms, paired with large datasets and powerful computing platforms, learn complex decision rules that accurately predict a wide range biological processes. Extracting the relationships these algorithms learn, and translating them into an understanding of how interactions drive the emergent behavior of biological systems, is an open and exciting challenge in genomics and precision medicine.

These avenues of inquiry are made possible by the recent explosion of high-dimensional, heterogeneous, biological data. Over the past several decades, new technologies have enabled the measurement of diverse molecular processes. Genomic assays such as ChIP-Chip, ChIP-seq, and ChIP-exo are being used to map the activity of TFs and other chromatin associated proteins across entire genomes [38, 23]. Imaging technologies provide detailed atlases of where and when genes are expressed [44, 134, 133, 55]. 3C-based methods, including 3C (chromosome conform capture), 4C (chromosome conform capture-on-chip), 5C (chromosome conformation capture carbon copy), and HiC, shed light on the 3D architecture of the genome and its role in regulation [27, 117, 34, 88]. These datasets, and many others, can be integrated to generate high-dimensional representations of functional regulation. However, information rich representations comes at a price: patterns in high-dimensional, heterogenous data are often beyond the scope of human comprehension and can only be identified computationally.

The statistical machine learning community has responded to this challenge by developing increasingly powerful and scalable predictive models. Supervised learning algorithms now routinely learn high-order relationships in data to achieve state-of-the-art predictive
CHAPTER 1. OVERVIEW

performance, even surpassing humans in many areas. A fundamental question is whether predictive models learn the mechanisms that govern biological systems. This question has been difficult to answer due the fact that modern algorithms are typically “black boxes,” providing little insight into the rules that make accurate prediction possible. Moreover, state-of-the-art machine learning models are considerably more sophisticated than probabilistic models that are traditionally studied in statistics. This makes uncertainty quantification a challenge in the age of “big data,” creating concerns of reproducibility for results derived from complex datasets [123, 65].

This thesis takes a step towards interpreting supervised learning algorithms to generate reliable hypotheses for the biological sciences. In chapter 2, we introduce the predictability, computability, and stability (PCS) framework for reproducible data science. The PCS framework draws from ideas in statistics, machine learning, and computer science to evaluate data results in realistic settings that deviate considerably from those considered in traditional statistical inference. It emphasizes the importance of human involvement throughout the data science life cycle as well as transparently communicating human judgement calls. Chapter 3 builds on the PCS framework through a new method: iterative Random Forests (iRF), which search for stable, predictive, and high-order interactions in biological data. Chapter 4 refines iRF through the notion of signed interactions. Signed interactions provide additional insights into how an RF combines information from interacting features to generate predictions. Moreover, it builds on ideas proposed in chapter 2, evaluating recovered interactions relative hypotheses that define simple structure in the data. Paired with emerging technologies that allow genome editing at an unprecedented level of precision, such as CRISPR cas-9, the work we describe here provides a powerful set of tools to probe complex systems in new ways and develop a deeper understanding of how they behave.

The following sections describe each chapter in more detail.

The predictability, computability, and stability (PCS) framework

The data science life cycle begins with a domain question and proceeds through collecting, processing, cleaning, exploring, visualizing, modeling, and interpreting the data to guide new actions. A critically important, but often ignored, component of the data science life cycle is the human involvement required to properly frame an analysis. This includes domain experts, who understand the connection between data and the natural processes they represent, database managers, who make decisions about how the data are stored and computed on, and data scientists, who make judgments on how to clean and model the data. The limited acknowledgement of human involvement in the data science life cycle makes it difficult to evaluate and reproduce analyses and is likely related to the high rate of false discoveries in many fields [65].

Communicating human judgment calls inherent to the data science life cycle requires an
enriched technical language. The PCS framework advocates for three principles: predictability, computability, and stability to serve as a foundation for this language. Predictability has long played an important role in science; models that accurately predict unobserved phenomena are widely acknowledged as foundational to the scientific community [106, 19, 45]. In the context of statistical machine learning, predictive accuracy has become an important tool to evaluate the quality of a model and the relationships it represents. Computability lays the groundwork for modern data science. Insights from state-of-the-art algorithms would not be possible without the ability to store data and tractably compute results. Stability describes the sensitivity of a data result to perturbations throughout the data science life cycle and is closely related to notions of scientific reproducibility. It generalizes considerations of statistical uncertainty to reflect human decisions made throughout an analysis.

In this chapter, we introduce the PCS framework and corresponding PCS inference procedures. We demonstrate that PCS inference outperforms probabilistic methods in the sparse linear model setting, which has been widely studied by the statistics community over the past several decades. Despite these promising results, the principal advantage of PCS inference is that it does not rely on probabilistic models alone, which can rarely be evaluated in practice, to make uncertainty statements. Thus PCS inference is applicable to the broad range of real-world data examples encountered in modern data science. For instance, we provide a case study of PCS-based inference for iRF on Zenodo.

Iterative random forests

Individual genomic assays measure elements that interact in vivo as components of larger molecular machines. Understanding the role of high-order interactions in biological processes, from gene regulation to organ development, presents a substantial statistical challenge. Namely, exhaustive searches for interaction candidates becomes intractable in genome-scale data due to combinatorial search spaces. For instance, there are 6M common human genetic variants in 20K genes, giving rise to $10^{170}$ possible interactions. Hence, the number of potential interactions far exceeds human capacity to examine or test.

Chapter 3 introduces the iterative Random Forest algorithm (iRF), a computationally efficient method for identifying high-order interactions in heterogeneous, high-dimensional data [9]. Building on random forests (RF) [18] and random intersection trees (RIT) [116], iRF trains a feature-weighted ensemble of decision trees to detect stable, high-order interactions with same order of computational cost as RF. By stabilizing and decoding the decision paths of RFs, iRF provides a new lens to interpret complex dependencies in state-of-the-art predictive models.

We demonstrate the utility of iRF in two genomics applications. First, we consider the problem of predicting enhancer activity in the early Drosophila melanogaster embryo. We use the wealth of experimental results on pairwise TF interactions to validate our findings. In this setting, 80% of the interactions iRF identifies have been previously reported. Beyond previously known pairwise interactions, iRF posits several higher-order interactions...
surrounding TFs that play an important regulatory role in the early *Drosophila* embryo. Second, we consider the problem of predicting alternative splicing in human-derived cell lines. Here iRF posits interactions of up to order-6 surrounding H3K36me3, which is known to regulate alternative splicing.

In addition to these real-data examples, we evaluate the performance of iRF across an extensive set of simulation experiments. These experiments are built from synthetic and real data and based on Boolean-type rules intended to reflect the stereospecific nature of biological interactions [97]. We demonstrate that iRF recovers up to order-8 interactions with high accuracy across a broad range of generative models. Moreover, we show that iRF offers substantial computational improvements relative to previous methods for high-order interaction recovery.

**Signed iterative random forests**

Identifying how predictive models map features to responses is an important step in extracting actionable information from data. From a biological perspective, mapping feature/response relationships helps predict the outcomes of specific experimental interventions. From a modeling perspective, the information can be used evaluate whether data results represent stable functional relationships.

In the final chapter of this thesis we describe a method signed iRF (siRF), which refines iRF interactions to explicitly map responses as a function of interacting features. siRF identifies “subsets” of rules, referred to as *signed interactions*, which frequently occur on RF decision paths. Decision paths that encode the same signed interaction not only share the same set of interacting features but also exhibit similar thresholding behavior, and thus describe a consistent functional relationship between interacting features and responses. We explicitly characterize this functional relationship to generate predictions associated with each signed interaction. In addition to this quantitative representation, we develop visualizations of RF response surfaces that allow us to observe what siRF “sees” in data.

We evaluate signed interactions using a range of stable and predictive importance metrics (SPIMs). These metrics are an example of PCS hypothesis testing proposed in chapter 2, and provide natural tests of whether interactions are consistent with simple data generating processes. We use SPIMs to conduct a suite of simulation experiments that demonstrate siRF accurately recovers feature response relationships in both synthetic and real data. In our simulations, siRF recovers data generating rules with greater accuracy than iRF recovers active features.

Finally, we examine regulatory interactions in *Drosophila* embryos through two genomic dataset. The first provides single-cell resolution measurements of gene expression in the early *Drosophila* embryo. We use this data to identify potential interactions based on spatial covariability, and validate our results against the well-known gap gene network, where we recover all 11 reported repressive links. The second dataset measures the relationship between TF binding and genomic sequences that drive patterned gene expression. Here
siRF identifies high-order interactions that predict enhancer activity and nervous system expression with high accuracy, suggesting candidate biological mechanisms that may play an important developmental role.
Chapter 2

Three Principles of Data Science: Predictability, Computability, and Stability

2.1 Introduction

Data science is a field of evidence seeking that combines data with prior information to generate new knowledge. The data science life cycle begins with a domain question or problem and proceeds through collecting, managing, processing/cleaning, exploring, modeling, and interpreting data to guide new actions (Figure 2.1). Given the trans-disciplinary nature of this process, data science requires human involvement from those who understand both the data domain and the tools used to collect, process, and model data. These individuals make implicit and explicit judgment calls throughout the data science life cycle. In order to transparently communicate and evaluate empirical evidence and human judgment calls, data science requires an enriched technical language. Three core principles: predictability, computability, and stability provide the foundation for such a data-driven language, and serve as minimum requirements for extracting reliable and reproducible knowledge from data.

These core principles have been widely acknowledged in various areas of data science. Predictability plays a central role in science through the notion of Popperian falsifiability [106]. It has been adopted by the statistical and machine learning communities as a goal of its own right and more generally to evaluate the reliability of a model or data result [19]. While statistics has always included prediction as a topic, machine learning emphasized its importance. This was in large part powered by computational advances that made it possible to compare models through cross-validation (CV), a technique pioneered by statisticians Stone and Allen [126, 3]. CV effectively generates pseudo-replicates from a single data set. This incorporates another important scientific principle: replication, and requires an understanding of the data generating process to justify the validity of CV pseudo replicates.

The role of computation extends beyond prediction, setting limitations on how data can
be collected, stored, and analyzed. Computability has played an integral role in computer science tracing back to Alan Turing’s seminal work on the computability of sequences [136]. Analyses of computational complexity have since been used to evaluate the tractability of statistical machine learning algorithms [57]. Kolmogorov built on Turing’s work through the notion of Kolmogorov complexity, which describes the minimum computational resources required to represent an object [81, 69]. Since Turing machine based computability notions are not computable in practice, in this paper, we treat computability as an issue of efficiency and scalability of optimization algorithms.

Stability is a common sense principle and a prerequisite for knowledge. In the context of the data science life cycle, stability\(^1\) has been advocated in [143] as a minimum requirement for reproducibility and interpretability at the modeling stage. To investigate the reproducibility of data results, modeling stage stability unifies numerous previous works including Jackknife, subsampling, bootstrap sampling, robust statistics, semi-parametric statistics, Bayesian sensitivity analysis (see [143] and references therein), which have been enabled in practice through computational advances. Econometric models with partial identification can also be viewed as a form of model stability consideration (see the book [92] and references therein). More broadly, stability is related to the notion of scientific reproducibility, which Fisher and Popper argued is a necessary condition for establishing scientific results [106, 41]. While reproducibility of results across laboratories has long been an important consideration in science, computational reproducibility has come to play an important role in data science as well. For example, [33] discuss reproducible research in the context of computational harmonic analysis. More broadly, [122] advocates for “preproducibility” to explicitly detail all steps along the data science life cycle and ensure sufficient information for quality control.

Here, we unify and expand on these ideas through the PCS framework. At the conceptual level, the PCS workflow uses predictability as a reality check, computability to ensure that an analysis is tractable, and stability to test the reproducibility of a data result against perturbations throughout the entire data science life cycle. It provides a general framework for evaluating and documenting analyses from problem formulation and data collection to conclusions, including all human judgment calls. The limited acknowledgement of human judgment calls in the data science life cycle, and therefore the limited transparency in reporting human decisions, has blurred the evidence for many data science analyses and resulted more false-discoveries than might otherwise occur. Our proposed PCS framework intends to mitigate these problems by clearly organizing and communicating human judgment calls so that they may be more easily deliberated and evaluated by others. It serves as a beneficial platform for reproducible data science research, providing a narrative (for example, to justify a bootstrap scheme is “appropriate”) in terms of both language and technical contents in the form of R Markdown or iPython (Jupyter) Notebook.

\(^1\)We differentiate between the notions of stability and robustness as used in statistics. The latter has been traditionally used to investigate performance of statistical methods across a range of distributions, while the former captures a much broader range of perturbations throughout the data science life cycle as discussed in this paper. But at a high level, stability is about robustness.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

The rest of the chapter is organized as follows. In Sec. 2.2 we introduce the PCS framework, which integrates the principles of predictability, computability, and stability across the entire data science life cycle. Sec. 2.3 examines connections within the PCS framework. In Sec. 2.4 we draw connections between the PCS framework and traditional statistical inference and propose a PCS inference framework for transparent instability or perturbation assessment in data science. Sec. 2.5 discusses PCS as a scientific hypothesis recommendation system, drawing connections with causal inference. In Sec. 2.6 we propose a format for documenting the PCS workflow with narratives and codes to justify the human judgment calls made during the data science life cycle. A case study of our proposed PCS workflow is based on the authors’ work studying gene regulation in Drosophila and available on Zenodo. We conclude by discussing areas for further work, including additional vetting of the workflow and theoretical analysis on the connections between the three principles.


**2.2 The PCS framework**

Given a domain problem and data, the data science life cycle generates conclusions and/or actions (Figure 2.1). The PCS workflow and documentation aim at ensuring the reliability and quality of this process through the three fundamental principles of data science. Predictability serves as a default reality check, though we note that other metrics, such as experimental validation or domain knowledge, may supplement predictability. Computability ensures that data results are tractable relative to a computing platform and available resources, including storage space, CPU/GPU time, memory, and communication bandwidth. Stability assesses whether results are robust to “appropriate” perturbations of choices made at each step throughout the data science life cycle. These considerations serve as minimum requirements for any data-driven conclusion or action. We formulate the PCS framework more precisely over the next four sections and address PCS documentation in Sec. 2.6.

**Stability assumptions initiate the data science life cycle**

The ultimate goal of the data science life cycle is to generate knowledge that is useful for future actions, be it a section in a textbook, biological experiment, business decision, or governmental policy. Stability is a useful concept to address whether an alternative, “appropriate” analysis would generate similar knowledge. At the modeling stage, stability has previously been advocated in [143]. In this context, stability refers to acceptable consistency of a data result relative to “reasonable” perturbations of the data or model. For example, jackknife [108, 107, 135], bootstrap [35], and cross validation [126, 3] may be considered reasonable or appropriate perturbations if the data are deemed approximately independent and identically distributed (i.i.d.) based on prior knowledge and an understanding of the data collection process. In addition to modeling stage perturbations, human judgment calls prior
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

to modeling also impact data results. The validity of such decisions relies on implicit stability assumptions that allow data scientists to view their data as an accurate representation of the natural phenomena they originally measured.

**Question or problem formulation:** The data science life cycle begins with a domain problem or a question. For instance, a biologist may want to discover regulatory factors that control genes associated with a particular disease. Formulating the domain question corresponds to an implicit linguistic stability, or well-definedness, assumption that both domain experts and data scientists understand the problem in the same manner. That is, the stability of meaning for a word, phrase, or sentence across different communicators is a minimum requirement for problem formulation. Since any domain problem must be formulated in terms of a natural language, linguistic stability is implicitly assumed. We note that there are often multiple translations of a domain problem into a data science problem. For example, the biologist could measure factors binding regulatory regions of the DNA that are associated with genes of interest. Alternatively, she could study how these genes covary with regulatory factors across time and/or space. From a modeling perspective, the biologist could identify important features in a random forest or through logistic regression. Stability of data results across these translations is an important consideration.

**Data collection:** To answer a domain question, data scientists and domain experts collect data based on prior knowledge and available resources. When this data is used to guide future decisions, researchers implicitly assume that the data is relevant for a future time and under future conditions. In other words, that conditions affecting data collection are stable, at least relative to some aspects of the data. For instance, a biologist could measure DNA binding of regulatory factors across the genome. To identify generalizable regulatory associations, experimental protocols must be comparable across laboratories. These stability considerations are closely related to external validity in medical research regarding the similarities between subjects in a study and subjects that researchers hope to generalize results to. We will discuss this idea more in Sec. 2.2.

**Data cleaning and preprocessing:** Statistical machine learning models or algorithms help data scientists answer domain questions. In order to use these tools, a domain question must first be translated into a question regarding the outputs of a model or an algorithm. This translation step includes cleaning and/or processing raw data into a suitable format, be it a categorical demographic feature or continuous measurements of biomarker concentrations. For instance, when multiple labs are used to produce data, the biologist must decide how to normalize individual measurements (for example see [13]). When data scientists clean or preprocess data, they are implicitly assuming that the raw and processed data contain consistent information about the underlying natural phenomena. In other words, they assume that the knowledge derived from a data result is stable with respect to their processing choices. If such an assumption can not be justified, they should use multiple reasonable processing methods and interpret only the stable data results across these methods. Others have advocated evaluating results across alternatively processed datasets under the name “multiverse analysis” [124]. Although the stability principle was developed independently of this work, it naturally leads to a multiverse-style analysis.
Exploratory data analysis (EDA): Both before and after modeling, data scientists often engage in exploratory data analysis (EDA) to identify interesting relationships in the data and interpret data results. When visualizations or summaries are used to communicate these analyses, it is implicitly assumed that the relationships or data results are stable with respect to any decisions made by the data scientist. For example, if the biologist believes that clusters in a heatmap represent biologically meaningful groups, they should expect to observe the same clusters with any appropriate choice of distance metric and with respect to appropriate data perturbations and appropriate clustering methods.

In each of these steps, stability assumptions allow data to be translated into reliable results. When these assumptions do not hold, results or conclusions apply only to the specific data from which they were drawn and with the specific data cleaning and EDA methods used. The justification is suspicious for using data results that do not generalize beyond a specific dataset to guide future actions. This makes it essential to ensure enough stability to guard against unnecessary costly future actions and false discoveries, particularly in the domains of science, business, and public policy, where data results are often used to guide large scale actions, and in medicine where people’s lives are at stake.

Predictability as reality check

After data collection, cleaning, preprocessing, and EDA, models or algorithms\textsuperscript{2} are frequently used to identify more complex relationships in data. Many essential components of the modeling stage rely on the language of mathematics, both in technical papers and in codes on computers. A seemingly obvious but often ignored question is why conclusions presented in the language of mathematics depict reality that exists independently in nature, and to what extent we should trust mathematical conclusions to impact this external reality.\textsuperscript{3} This concern has been articulated by many others. For instance, Philip Dawid drew connections between statistical inference and prediction under the name “prequential statistics,” highlighting the importance of forecasts in statistical analyses. David Freedman argued that “If the assumptions of a model are not derived from theory, and if predictions are not tested against reality, then deductions from the model must be quite shaky [45].” Seymour Geisser advocated that statistical analyses should focus on prediction rather than parametric inference, particularly in cases where the statistical model is an inappropriate description of reality [51]. More recently, Leo Breiman championed the essential role of prediction in developing realistic models that yield sound scientific conclusions [19]. It can even be argued that the goal of most analyses is prediction at the meta level. That is, the primary value of learning relationships in data is often to predict some aspect of future observations.

\textsuperscript{2}The model or algorithm choices could correspond to different translations of a domain problem.

\textsuperscript{3}The PCS documentation in Sec. 2.6 helps users assess reliability of this connection.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

Formulating prediction

We describe a general framework for prediction with data $D = (x, y)$, where $x \in \mathcal{X}$ represents input features and $y \in \mathcal{Y}$ the prediction target. Prediction targets $y \in \mathcal{Y}$ may be observed responses (e.g. supervised learning) or extracted from data (e.g. unsupervised learning). Predictive accuracy provides a simple, quantitative metric to evaluate how well a model represents relationships in $D$. It is well-defined relative to a prediction function, testing data, and an evaluation function. We detail each of these elements below.

**Prediction function:** The prediction function

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$

represents relationships between the observed features and the prediction target. For instance, in the case of supervised learning $h$ may be a linear predictor or decision tree. In this setting, $y$ is typically an observed response, such as a class label. In the case of unsupervised learning, $h$ could map from input features to cluster centroids.

To compare multiple prediction functions, we consider

$$\{h^{(\lambda)} : \lambda \in \Lambda\},$$

where $\Lambda$ denotes a collection models/algorithms. For example, $\Lambda$ may define different tuning parameters in lasso [131] or random forests [18]. For algorithms with a randomized component, such as k-means or stochastic gradient descent, $\Lambda$ can represent repeated runs. More broadly, $\Lambda$ may describe different architectures for deep neural networks or a set of competing algorithms such as linear models, random forests, and neural networks. We discuss model perturbations in more detail in Sec. 2.2.

**Testing (held-out) data:** We distinguish between *training data* that are used to fit a collection of prediction functions, and *testing data* that are held out to evaluate the accuracy of fitted prediction functions. Testing data are typically assumed to be generated by the same process as the training data. *Internal testing* data, which are collected at the same time and under the same conditions as the training data, clearly satisfy this assumption. To evaluate how a model performs in new settings, we also consider *external testing* data gathered under different conditions from the training data. Prediction accuracy on external testing data directly addresses questions of external validity, which describe how well a result will generalize to future observations. When the goal of an analysis is prediction, testing data should only be used to report the accuracy of a prediction function $h$. When the target of an analysis extends beyond predicting unobserved responses (e.g. to feature selection), testing data may be used to screen models based on predictive accuracy before evaluating the target of interest (see Sec. 2.4). We note that domain knowledge from humans involved in data generation and analysis is essential to assess the appropriateness of different prediction settings.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

**Prediction evaluation metric:** The prediction evaluation metric

$$\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$$  \hspace{1cm} (2.3)

quantifies the accuracy of a prediction function $h \in \mathcal{H}$ by measuring the similarity between $h(x)$ and $y$ in the testing data. We adopt the convention that $\ell(h, x, y) = 0$ implies perfect prediction accuracy while increasing values imply worse predictive accuracy. When the goal of an analysis is prediction, testing data should only be used in reporting the accuracy of a prediction function $h$. When the goal of an analysis extends beyond prediction (e.g. to feature selection), testing data may be used to filter models from equation (2.2) based on their predictive accuracy.

Despite its quantitative nature, prediction always requires human input to formulate, including the choice of prediction and evaluation functions, the preferred structure of a model/algorithm and what it means by achieving predictability. For example, a biologist studying gene regulation may believe that the simple rules learned by decision trees offer an appealing representation of interactions that exhibit thresholding behavior [139]. If the biologist is interested in a particular cell-type or developmental stage, she might evaluate prediction accuracy on internal test data measuring only these environments. If her responses are binary with a large proportion of class-0 responses, she may choose an evaluation function $\ell$ to handle the class imbalance.

All of these decisions and justifications should be documented by researchers and assessed by independent reviewers (see the documentation section) so that others can evaluate the strength of the conclusions based on transparent evidence. The accompanying PCS documentation provides a detailed example of our predictive formulation in practice.

**Cross validation**

As alluded to earlier, cross-validation (CV) has become a powerful work horse to select regularization parameters by estimating the prediction error through multiple pseudo held-out data within a given data set [126, 3]. As a regularization parameter selection tool, CV works more widely than as a prediction error estimate, which can incur high variability due to the often positive dependences between the estimated prediction errors in the summation of the CV error [50].

CV divides data into blocks of observations, trains a model on all but one block, and evaluates the prediction error over each held-out block. That is, CV evaluates whether a model accurately predicts the responses of pseudo-replicates. Just as peer reviewers make judgment calls on whether a lab’s experimental conditions are suitable to replicate scientific results, data scientists must determine whether a removed block represents a justifiable pseudo replicate of the data, which requires information from the data collection process and domain knowledge.
Computability

In a broad sense, computability is the gate-keeper of data science. If data cannot be generated, stored, managed, and analyzed efficiently and scalably, there is no data science. Modern science relies heavily on information technology as part of the data science life cycle. Each step, from raw data collection and cleaning, to model building and evaluation, rely on computing technology and fall under computability broadly. In a narrow sense, computability refers to the computational feasibility of algorithms or model building.

Here we use the narrow-sense computability, which is closely associated with the rise of machine learning over the last three decades. Just as scientific instruments and technologies determine what processes can be effectively measured, computing resources and technologies determine the types of analyses that can be carried out. Moreover, computational constraints can serve as a device for regularization. For example, stochastic gradient descent is widely used for optimization in machine learning problems \[111\]. Both the stochasticity and early stopping of a stochastic gradient algorithm play the role of implicit regularization.

Computational considerations and algorithmic analyses have long been an important part of machine learning. These analyses consider the number of computing operations and required storage space in terms of the number of observations \(n\), number of features \(p\), and tuning (hyper) parameters. When the computational cost of addressing a domain problem or question exceeds available computational resources, a result is not computable. For instance, the biologist interested in gene regulation may want to model interaction effects in a supervised learning setting. However, there are \(O(p^s)\) possible order-\(s\) interactions among \(p\) regulatory factors. For even a moderate number of factors, exhaustively searching for high-order interactions is not computable. In such settings, data scientists must restrict modeling decisions to draw conclusions. Thus it is important to document why certain restrictions were deemed appropriate and the impact they may have on conclusions (see Sec. 2.6).

Increases in computing power also provide an unprecedented opportunity to enhance analytical insights into complex natural phenomena. We can now store and process massive datasets and use these data to simulate large scale processes. Simulations provide concrete and quantitative representations of a natural phenomena relative to known input parameters, which can be perturbed to assess the stability of data results. As a result, simulation experiments inspired by observed data and domain knowledge become a powerful set of tools to understand how results may behave in real-world settings. They represent a best effort to emulate complex processes, where the reliability of data results is not always clear. Pairing such simulation studies with empirical evidence makes the data science life cycle more transparent for peers and users to review, aiding in the objectivity of science.

Stability after data collection

Computational advances have fueled our ability to analyze the stability of data results in practice. At the modeling stage, stability measures how a data result changes as the data and/or model are perturbed. Stability extends the concept of sampling variability in statis-
tics, which is a measure of instability relative to other data that could be generated from the same distribution. Statistical uncertainty assessments implicitly assume stability in the form of a distribution that generated the data. This assumption highlights the importance of other data sets that could be observed under similar conditions (e.g. by another person in the lab or another lab at a later time).

The concept of a true distribution in traditional statistics is a construct. When randomization is explicitly carried out, the true distribution construct is physical. Otherwise, it is a mental construct whose validity must be justified based on domain knowledge and an understanding of the data generating process. Statistical inference procedures or uncertainty assessments use distributions to draw conclusions about the real world. The relevance of such conclusions is determined squarely by the empirical support for the true distribution, especially when the construct is not physical. In data science and statistical problems, practitioners often do not make much of an attempt to justify or provide support for this mental construct. At the same time, they take the uncertainty conclusions very seriously. This flawed practice is likely related to the high rate of false discoveries [123, 65]. It is a major impediment to true progress of science and to data-driven knowledge extraction in general.

While the stability principle encapsulates uncertainty quantification (when the true distribution construct is well supported), it is intended to cover a much broader range of perturbations, such as pre-processing, EDA, randomized algorithms, data perturbation, and choice of models/algorithms. A complete consideration of stability across the entire data science life cycle is necessary to ensure the quality and reliability of data results. For example, the biologist studying gene regulation must choose both how to normalize raw data and what algorithm(s) she will use. When there is no principled approach to make these decisions, the knowledge data scientists can extract from analyses is limited to conclusions that are stable across reasonable choices [124, 1, 9]. This ensures that another scientist studying the same data will come to similar conclusions, despite slight variation in their independent choices.

**Formulating stability at the modeling stage**

Stability at the modeling stage is defined with respect to a target of interest, a “reasonable” or “appropriate” perturbation to the data and/or algorithm/model, and a stability metric to measure the change in target that results from perturbation. We describe each of these in detail below.

**Stability target:** The stability target

\[ T(D, \lambda), \tag{2.4} \]

corresponds to the data result of interest. It depends on input data \( D \) and a specific model/algorithm \( \lambda \) used to analyze the data. For simplicity, we will sometimes suppress the dependence on \( D \) and \( \lambda \) in our notation. As an example, \( T \) can represent responses predicted by \( h(\lambda) \) when the goal of an analysis is prediction. Other examples of \( T \) include features selected by lasso with penalty parameter \( \lambda \) or saliency maps derived from a convolutional neural network with architecture \( \lambda \).
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

Data and model perturbations: To evaluate the stability of a data result, we measure the change in target $\mathbf{T}$ that results from a perturbation to the input data or learning algorithm. More precisely, we define a collection of data perturbations $\mathbf{D}$ and model/algorithm perturbations $\Lambda$ and compute the stability target distribution

$$\{\mathbf{T}(D, \lambda) : D \in \mathbf{D}, \lambda \in \Lambda\}. \quad (2.5)$$

For example, reasonable data perturbations include bootstrap sampling if observations are approximately i.i.d. and generative models that are supported by subject matter knowledge (see Sec. 2.2). They even include probabilistic models that are justified from an understanding of the data generating process or explicit randomization. When different prediction functions are deemed equally appropriate based on domain knowledge, each may represent an appropriate model perturbation (see Sec. 2.2).

It can be argued that the subjectivity surrounding “appropriate” perturbations makes it difficult to evaluate results within the PCS workflow. Indeed, perturbation choices are both subjective human judgment calls and critical considerations in PCS. The degree to which a data result can be trusted depends on the degree to which a perturbation can be justified, which requires an understanding of the data domain and generating process. This is true if the perturbation comes from a probabilistic model, as in traditional statistical inference, or some broader set of perturbations, as in PCS. The goal of PCS is to use and explicitly document perturbations that are best suited to assess stability in complex, high-dimensional data rather than relying on probabilistic models alone, which have little objective meaning when the model is not justified (see Sec. 2.4). To ensure that results can be evaluated, the case for an appropriate perturbation must be made in the publication and in the PCS documentation (see Sec. 2.6). These transparent narratives allow for discussion of different perturbations to determine which should be expected for a particular field and/or type of data, aiding the objectivity of science.

Stability evaluation metric: The stability evaluation metric $s(\mathbf{T}; \mathbf{D}, \Lambda)$ summarizes the stability target distribution in expression (2.5). For instance, if $\mathbf{T}$ indicates features selected by a model $\lambda$ trained on data $D$, we may report the proportion of data perturbations $D \in \mathbf{D}$ that recover each feature for each model $\lambda \in \Lambda$. When a stability analysis reveals that the $\mathbf{T}$ is unstable (relative to a threshold meaningful in a domain context), it is advisable to search for another target of interest that achieves the desired level of stability. This creates the possibility of “data-snooping” or overfitting and hence should be viewed as part of the iterative process between data analysis and knowledge generation described by [15]. Before defining a new target, it may be necessary to collect new data to avoid overfitting.

Data perturbation

The goal of data perturbation under the stability principle is to mimic a pipeline that could have been used to produce final input data but was not. This includes human decisions, such as preprocessing and data cleaning, as well as data generating mechanisms. When we
focus on the change in target under possible realizations of the data from a well-supported probabilistic model, we arrive at well-justified sampling variability considerations in statistics. Hence data perturbation under the stability principle includes, but is much broader than, the concept of sampling variability. It formally recognizes many other important steps and considerations in the data science life cycle beyond sampling variability. Furthermore, it provides a framework to build confidence measures for estimates of $T$ when a probabilistic model is not well justified and hence sampling interpretations are not applicable.

Data perturbations can also be used to reduce variability in the estimated target, which corresponds to a data result of interest. Random Forests incorporate subsampling data perturbations (of both the data units and predictors) to produce predictions with better generalization error [18]. Generative adversarial networks (GANs) use synthetic adversarial examples to re-train deep neural networks and produce predictions that are more robust to such adversarial data points [53]. Empirically supported generative models, including PDE models, can also be used to produce “good” data points or synthetic data that encourage stability of data results with respect to mechanistic rules based on domain knowledge (for examples see [12]).

**Algorithm or model perturbation**

The goal of algorithm or model perturbation is to understand how alternative analyses of the same data affect the target estimate. A classical example of model perturbation is from robust statistics, where one searches for a robust estimator of the mean of a location family by considering alternative models with heavier tails than the Gaussian model. Another example of model perturbation is sensitivity analysis in Bayesian modeling [119, 14]. Many of the model conditions used in causal inference are in fact stability concepts that assume away confounding factors by asserting that different conditional distributions are the same [105, 58].

Modern algorithms often have a random component, such as random projections or random initial values in gradient descent and stochastic gradient descent. These random components provide natural model perturbations that can be used to assess variability or instability of $T$. In addition to the random components of a single algorithm, multiple models/algorithms can be used to evaluate stability of the target. This is useful when there are many appropriate choices of model/algorithm and no established criteria or established domain knowledge to select among them. The stability principle calls for interpreting only the targets of interest that are stable across these choices of algorithms or models [1].

As with data perturbations, model perturbations can help reduce variability or instability in the target. For instance, [95] selects lasso coefficients that are stable across different regularization parameters. Dropout in neural networks is a form of algorithm perturbation that leverages stability to reduce overfitting [121]. Our previous work [9] stabilizes Random Forests to interpret the decision rules in an ensemble of decision trees, which are perturbed using random feature selection.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

Dual roles of generative models in PCS

Generative models include both probabilistic models and partial differential equations (PDEs) with initial or boundary conditions (if the boundary conditions come from observed data, then such PDEs become stochastic as well). These models play dual roles with respect to PCS. On one hand, they can concisely summarize past data and prior knowledge. On the other hand, they can be used to generate synthetic observations that provide a form of data perturbation.

When a generative model is used to summarize data, a common target of interest is the model’s parameters. Generative models with known parameters may be used for prediction or to advance understanding through the mechanistic rules they represent. Such models correspond to infinite data, though finite under computational constraints. Generative models with unknown parameters can be used to motivate surrogate loss functions through maximum likelihood and Bayesian modeling methods. Mechanistic interpretations of such models should not be used to draw scientific conclusions. They are simply useful starting points to optimize algorithms that must be subjected to empirical validation.

Generative models that approximate the data generating process (a human judgment call) can be used as a form of data perturbation. Here synthetically generated data augment the observed data, serving the purpose of domain-inspired regularization. The amount of synthetic data to combine with the observed data reflects our degree of belief in the models. Using synthetic data for domain inspired regularization allows the same algorithmic and computing platforms to be applied to the combined data. This approach to analysis is reminiscent of AdaBoost and its variants, which use the current data and model to modify the data used in the next iteration without changing the base-learner [47].

2.3 Connections in the PCS framework

Although we have discussed the components of the PCS framework individually, they share important connections. Computational considerations can limit the predictive models/algorithms that are tractable, particularly for large, high-dimensional datasets. These computability issues are often addressed in practice through scalable optimization methods such as gradient descent (GD) or stochastic gradient descent (SGD). Evaluating predictability on held-out data is a form of stability analysis where the training/test sample split represents a data perturbation. Other perturbations used to assess stability require multiple runs of similar analyses. Parallel computation is well suited for modeling stage PCS perturbations. Stability analyses beyond the modeling stage requires a streamlined computational platform that is future work.
2.4 PCS inference through perturbation analysis

When data results are used to guide future decisions or actions, it is important to assess the quality of the target estimate. For instance, suppose a model predicts that an investment will produce a 10% return over one year. Intuitively, this prediction suggests that “similar” investments return 10% on average. Whether or not a particular investment will realize a return close to 10% depends on whether returns for “similar” investments ranged from −20% to 40% or from 8% to 12%. In other words, the variability of a prediction conveys important information about how much one should trust it.

In traditional statistics, confidence measures describe the uncertainty of an estimate due to sampling variability under a well-justified probabilistic model. In the PCS framework, we define perturbation intervals to quantify the stability of target estimates relative to different perturbations. Perturbation intervals are conceptually similar to confidence intervals. The primary difference is that they are explicitly connected to perturbations, which are justified in PCS documentation (Sec. 2.6) and evaluated by independent reviewers and domain experts. For instance, perturbation intervals for one estimation method based on bootstrap sampling specialize to traditional confidence intervals based on the bootstrap. More broadly, perturbation intervals quantify the variability of a target parameter across the entire data science lifecycle. For example, a data scientist may consider multiple preprocessing, subsampling, and modeling strategies to predict investment returns. The resulting perturbation intervals describe the range of expected returns across worlds represented by each perturbation. Their reliability lies squarely on whether the set of perturbations captures the full spectrum of “reasonable” analyses, which should be evaluated by domain experts and independent reviewers. This highlights the importance of perturbations that could plausibly generate the observed data, represent the full range of uncertainty surrounding an analysis (e.g. relative to data generation or modeling decisions), and are transparently documented for others to evaluate (see Sec. 2.6).

As a starting point, we focus on a basic form of PCS inference that generalizes traditional statistical inference. The proposed PCS inference includes a wide range of data and algorithm/model perturbations to encompasses the broad range of settings encountered in modern practice.

PCS perturbation intervals

The reliability of PCS quality measures lies squarely on the appropriateness of each perturbation. Consequently, perturbation choices should be seriously deliberated, clearly communicated, and evaluated by objective reviewers as alluded to earlier. To obtain quality measures for a target estimate\(^4\), which we call PCS perturbation intervals, we propose the

\(^4\)A domain problem can be translated into multiple data science problems with different target estimates. For the sake of simplicity, we discuss only one translation in the basic PCS inference framework.
following steps:\footnote{The PCS perturbation intervals cover different problem translations through $\Lambda$ and are clearly extendable to include perturbations in the pre-processing step through $D$.}

1. **Problem formulation:** Translate the domain question into a data science problem that specifies how the question can be addressed with available data. Define a prediction target $y$, “appropriate” data $D$ and/or model $\Lambda$ perturbations, prediction function(s) $\{h(\lambda) : \lambda \in \Lambda\}$, training/test split, prediction evaluation metric $\ell$, stability metric $s$, and stability target $T(D, \lambda)$, which corresponds to a comparable quantity of interest as the data and model/algorithm vary. Document why these choices are appropriate in the context of the domain question.

2. **Prediction screening:** For a pre-specified threshold $\tau$, screen out models that do not fit the data (as measured by prediction accuracy)

   $$\Lambda^* = \{\lambda \in \Lambda : \ell(h(\lambda), x, y) < \tau\}.$$  

   Examples of appropriate threshold include domain accepted baselines, the top $k$ performing models, or models whose accuracy is suitably similar to the most accurate model. If the goal of an analysis is prediction, the testing data should be held-out until reporting the final prediction accuracy. In this setting, equation (2.6) can be evaluated using a training or surrogate sample-splitting approach such as cross validation. If the goal of an analysis extends beyond prediction, equation (2.6) may be evaluated on held-out test data.

3. **Target value perturbation distributions:** For each of the survived models $\Lambda^*$ from step 2, compute the stability target under each data perturbation $D$. This results in a joint distribution of the target over data and model perturbations as in equation (2.5).

4. **Perturbation interval (or region) reporting:** Summarize the target value perturbation distribution using the stability metric $s$. For instance, this summary could be the 10th and 90th percentiles of the target estimates across each perturbation or a visualization of the entire target value perturbation distribution.

At a high level, PCS uses perturbation intervals to identify the stable part of accurate models. If perturbation intervals reveal instability among accurate models, PCS inference can be used to interpret aspects that are shared (i.e. stable) across these models. In this setting, PCS can be viewed as an implicit application of Occam’s razor. That is, it draws conclusions from the stable portion of predictive models to simplify data results, making them more reliable and easier to interpret. If perturbation intervals reveal that complex models are both stable and accurate, PCS inference provides evidence supporting learned relationships and justification for the added complexity.
PCS hypothesis testing

Hypothesis testing from traditional statistics is commonly used in decision making for science and business alike. The heart of Fisherian testing [40] lies in calculating the p-value, which represents the probability of an event more extreme than in the observed data under a null hypothesis or distribution. Smaller p-values correspond to stronger evidence against the null hypothesis or the scientific theory embedded in the null hypothesis. For example, we may want to determine whether a particular gene is differentially expressed between breast cancer patients and a control group. Given random samples from each population, we could address this question in the classical hypothesis testing framework using a t-test. The resulting p-value describes the probability of seeing a difference in means more extreme than observed if the genes are not differentially expressed.

While hypothesis testing is valid philosophically, many of the assumptions it relies on are unrealistic in practice. For instance, unmeasured confounding variables can bias estimates of causal effects. These issues are particularly relevant in the social sciences, where randomized trials are difficult or impossible to conduct. Resource constraints can limit how data are collected, resulting in samples that do not reflect the population of interest and introducing selection bias that distorts probabilistic interpretations. Moreover, hypothesis testing assumes empirical validity of probabilistic data generating models. When randomization is not carried out explicitly, a particular null distribution must be justified from domain knowledge of the data generating mechanism. Such issues are seldom taken seriously in practice, resulting in settings where the null distribution is far from the observed data. As a result, p-values as small as $10^{-5}$ or $10^{-8}$ are now common to report, despite the fact that there are rarely enough data to reliably calculate these values, especially when multiple hypotheses (e.g. thousands of genes) are evaluated. When results are so far off on the tail of the null distribution, there is no empirical evidence as to why the tail should follow a particular parametric distribution. Moreover, hypothesis testing as practiced today often relies on analytical approximations or Monte Carlo methods, where issues arise for such small probability estimates. In fact, there is a specialized area of importance sampling to deal with simulating small probabilities [113, 21], but the hypothesis testing does not seem to have taken advantage of these ideas.

PCS hypothesis testing builds on perturbation intervals to address these practical issues and the cognitively misleading nature of small p-values. It uses the null hypothesis to define perturbations that represent a plausible, constrained data generating process. This includes probabilistic models, when they are well founded, as well as other data and/or algorithm perturbations. For instance, generative models based on PDEs can be used to simulate data according to established physical laws. By allowing for a broad class of data perturbations, PCS hypothesis testing allows us to compare observed data with data that respects some simple structure known to represent important characteristics of the domain question. Of course, the appropriateness of a perturbation is a human judgment call that

---

6Under conditions, Freedman [46] showed that some tests can be approximated by permutation tests when data are not generated from a probabilistic model, but these results are not broadly applicable.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

should be clearly communicated in PCS documentation and debated by researchers. Much like scientists deliberate over appropriate controls in an experiment, data scientists should debate the reasonable perturbations in a PCS analysis.

Formalizing PCS hypothesis testing

Formally, we consider settings with observable input features $x \in X$, prediction target $y \in Y$, prediction functions $\{h_{\lambda}^{(\lambda)} : \lambda \in \Lambda \}$, and a null hypothesis that qualitatively describes some aspect of the domain question. PCS hypothesis testing translates the null hypothesis into a constrained perturbation and generates data

$$D_0 = \{x_0, y_0\}$$

according to this perturbation\(^7\). The particular choice of constrained perturbation should be explicitly documented and justified by domain knowledge. We use the constrained perturbation to construct and compare perturbation intervals for both $D_0$ and $D$ and evaluate whether the observed data is consistent with the embedded hypothesis.

As an example, [37] consider the null hypothesis that population level structure in single neuron data is the expected byproduct of primary features (e.g. correlations across time). The authors use a maximum entropy approach, whose constraint is represented by the number of moments, to generate data that share primary features with the observed data but are otherwise random and compare population level findings between the observed and simulated data. In the accompanying PCS documentation, we consider the null hypothesis that genomic interactions appear with equal frequency among different classes of genomic elements. We use a sample splitting strategy which treats inactive elements (class-0 observations) as a baseline to determine whether interactions appear with “unusual” frequency among active elements (class-1 observations). Once again, these comparisons rely on human judgment to determine when perturbation intervals are sufficiently different. These choices depend on the domain context and how the problem has been translated. They should be transparently communicated by the researcher in the PCS documentation.

Simulation studies of PCS inference in the linear model setting

In this section, we consider the proposed PCS perturbation intervals through data-inspired simulation studies. We focus on feature selection in sparse linear models to demonstrate that PCS inference provides favorable results, in terms of ROC analysis, in a setting that has been intensively investigated by the statistics community in recent years. Despite its favorable performance in this simple setting, we note that the principal advantage of PCS inference is its generalizability to new situations faced by data scientists today. That is, PCS can be applied to any algorithm or analysis where one can define appropriate perturbations.

\(^7\)A null hypothesis may correspond to multiple data or model/algorithm perturbations. We focus on a single data perturbation here for simplicity.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

For instance, in the accompanying PCS case study, we demonstrate the ease of applying PCS inference in the problem of selecting high-order, rule-based interactions in a high-throughput genomics problem (whose data the simulation studies below are based upon).

To evaluate feature selection in the context of linear models, we considered data (as in the case study) for 35 genomic assays measuring the binding enrichment of 23 unique TFs along 7809 segments of the genome [91, 85, 84]. We augmented this data with 2nd order polynomial terms for all pairwise interactions (excluding quadratic terms $x_i^2$), resulting in a total of $p = 630$ features. For a complete description of the data, see the accompanying PCS documentation. We standardized each feature and randomly selected $s = \lfloor \sqrt{p} \rfloor = 25$ active features to generate responses

$$y = \mathbf{x}^T \beta + \epsilon$$

(2.8)

where $\mathbf{x} \in \mathbb{R}^{7809 \times 630}$ denotes the normalized matrix of features, $\beta_j = 1$ for any active feature $j$ and 0 otherwise, and $\epsilon$ represents mean 0 noise drawn from a variety of distributions. In total, we considered 6 distinct settings with 4 noise distributions: i.i.d. Gaussian, Students $t$ with 3 degrees of freedom, multivariate Gaussian with block covariance structure, Gaussian with variance $\sigma^2 \propto \| \mathbf{x}_i \|_2^2$ and two misspecified models: i.i.d. Gaussian noise with 12 active features removed prior to fitting the model, i.i.d. Gaussian noise with responses generated as

$$y = \sum_{S_j \in \mathcal{S}} \beta_{S_j} \prod_{k \in S_j} 1(x_k > t_k) + \epsilon$$

(2.9)

where $\mathcal{S}$ denotes a set of randomly sampled pairs of active features.

Simple PCS perturbation intervals

We evaluated selected features using the PCS perturbation intervals proposed in Sec. 2.4. Below we outline each step for constructing such intervals in the context of linear model feature selection.

1. Our prediction target was the simulated responses $y$ and our stability target $\mathcal{T} \subseteq \{1, \ldots, p\}$ the features selected by lasso when regressing $y$ on $\mathbf{x}$. To evaluate prediction accuracy, we randomly sampled 50% of observations as a held-out test. The default values of lasso penalty parameter in the R package glmnet represent a collection of model perturbations. To evaluate the stability of $\mathcal{T}$ with respect to data perturbation, we repeated this procedure across $B = 100$ bootstrap replicates.

2. We formed a set of filtered models $\Lambda^*$ by taking $\lambda$ corresponding to the 10 most accurate models in terms of $\ell_2$ prediction error. Since the goal of our analysis was feature selection, we evaluated prediction accuracy on the held-out test data. We repeated the steps below on each half of the data and averaged the final results.
3. For each $\lambda \in \Lambda^*$ and $b = 1, \ldots, 100$ we let $T(x^{(b)}, \lambda)$ denote the features selected for bootstrap sample $b$ with penalty parameter $\lambda$.

4. The distribution of $T$ across data and model perturbations can be summarized into a range of stability intervals. Since our goal was to compare PCS with classical statistical inference, which produces a single p-value for each feature, we computed a single stability score for each feature $j = 1, \ldots, p$:

$$\text{sta}(j) = \frac{1}{B \cdot |\Lambda^*|} \sum_{b=1}^{100} \sum_{\lambda \in \Lambda^*} 1(j \in T(x^{(b)}, \lambda))$$

Intuitively, stability scores reflect our certainty that a given feature is active in the model, with higher scores implying a higher degree of certainty. In practice, these scores could be used to rank features and identify the most reliable collection for further consideration (e.g. experimental validation). We note that the stability selection proposed in [95] is similar, but without the prediction error screening.

**Results**

We compared the above PCS stability scores with asymptotic normality results applied to features selected by lasso and selective inference [129]. We note that asymptotic normality and selective inference both produce p-values for each feature, while PCS produces stability scores.

Figure 2.2 shows ROC curves for feature selection averaged across 100 replicates of the above experiments. The ROC curve is a useful evaluation criterion to assess both false positive and false negative rates when experimental resources dictate how many selected features can be evaluated in further studies, including physical experiments. In particular, ROC curves provide a balanced evaluation of each method’s ability to identify active features while limiting false discoveries. Across all settings, PCS compares favorably to the other methods. The difference is particularly pronounced in settings where other methods fail to recover a large portion of active features ($n < p$, heteroskedastic, and misspecified model). In such settings, stability analyses allow PCS to recover more active features while still distinguishing them from inactive features. While its performance in this setting is promising, the principal advantage of PCS is its conceptual simplicity and generalizability. That is, the PCS perturbation intervals described above can be applied in any setting where data or model/algorithm perturbations can be defined, as illustrated in the genomics case study in the accompanying PCS documentation. Traditional inference procedures cannot handle multiple models easily in general.
Figure 2.2: ROC curves comparing feature selection for PCS and traditional inference in linear model setting with $n = 250$ (top) and $n = 1000$ (bottom) observations. Each plot corresponds to a different generative model.
2.5 PCS recommendation system for scientific hypothesis generation

In general, causality implies predictability and stability over many experimental conditions; but not vice versa. The causal inference community has long acknowledged connections between stability and estimates of causal effects. For instance, many researchers have studied paradoxes surrounding associations that lead to unstable estimates of causal effects [39, 25, 11]. Estimates in the Neyman-Rubin potential outcomes framework rely on a stable treatment across observational units [98, 112]. Sensitivity analyses test the stability of a causal effect relative to unmeasured confounding [26, 31]. Stable predictions across experimental conditions has even been proposed as a criteria to estimate causal effects [105].

PCS inference builds on these ideas, using stability and predictability to rank target estimates for further studies, including follow-up experiments. In our recent works on DeepTune [1] and iterative random forests (iRF) [9], we use PCS in the modeling stage to make such recommendations. For example, PCS analysis suggested potential relationships between areas of the visual cortex and visual stimuli in neuroscience applications as well as 3rd and 4th order interactions among biomolecules that are candidates for regulating gene expression. Predictability and stability do not replace physical experiments to prove or disprove causality. However, we hope computationally tractable analyses that demonstrate high predictability and stability suggest hypotheses or intervention experiments that have higher yields than otherwise. This hope is supported by the fact that 80% of the 2nd order interactions identified by iRF had been verified in the literature by other researchers through physical experiments.

2.6 PCS workflow documentation

The PCS workflow requires an accompanying R Markdown or iPython (Jupyter) notebook, which seamlessly integrates analyses, codes, and narratives. These narratives are necessary to describe the domain problem and support assumptions and choices made by the data scientist regarding computational platform, data cleaning and preprocessing, data visualization, model/algorithm, prediction metric, prediction evaluation, stability target, data and algorithm/model perturbations, stability metric, and data conclusions in the context of the domain problem. These narratives should be based on referenced prior knowledge and an understanding of the data collection process, including design principles or rationales. In particular, the narratives in the PCS documentation help bridge or connect the two parallel universes of reality and models/algorithms that exist in the mathematical world (Figure 2.3). In addition to narratives justifying human judgment calls (possibly with data evidence), PCS documentation should include all codes used to generate data results with links to sources of data and meta data.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

Figure 2.3: A depiction of the two worlds that must be connected through transparent documentation. Assumptions made throughout the data science life cycle allow researchers to use models as an approximation of reality. Narratives provided in PCS documentation can help justify assumptions to connect these two worlds.

We propose the following steps in a notebook:

1. Domain problem formulation (narrative). Clearly state the real-world question one would like to answer and describe prior work related to this question.

2. Data collection and relevance to problem (narrative). Describe how the data were generated, including experimental design principles, and reasons why data is relevant to answer the domain question.

3. Data storage (narrative). Describe where data is stored and how it can be accessed by others.

4. Data cleaning and preprocessing (narrative, code, visualization). Describe steps taken to convert raw data into data used for analysis, and why these preprocessing steps are justified. Ask whether more than one preprocessing methods should be used and examine their impacts on the final data results.

5. PCS inference (narrative, code, visualization). Carry out PCS inference in the context of the domain question. Specify appropriate model and data perturbations. If neces-

---

8This list is reminiscent of the list in the “data wisdom for data science” blog that one of the authors wrote at http://www.odbms.org/2015/04/data-wisdom-for-data-science/
sary, specify null hypothesis and associated perturbations (if applicable). Report and post-hoc analysis of data results.

6. Draw conclusions and/or make recommendations (narrative and visualization) in the context of domain problem.

This documentation of the PCS workflow gives the reader as much as possible information to make informed judgments regarding the evidence and process for drawing a data conclusion in a data science life cycle. A case study of the PCS workflow on a genomics problem in R Markdown is available on Zenodo.
CHAPTER 2. THREE PRINCIPLES OF DATA SCIENCE: PREDICTABILITY, COMPUTABILITY, AND STABILITY

2.7 Conclusion

This paper discusses the importance and connections of three principles of data science: predictability, computability and stability. Based on these principles, we proposed the PCS framework with corresponding PCS inference procedures (perturbation intervals and PCS hypothesis testing). In the PCS framework, prediction provides a useful reality check, evaluating how well a model/algorithm reflects the physical processes or natural phenomena that generated the data. Computability concerns with respect to data collection, data storage, data cleaning, and algorithm efficiency determine the tractability of an analysis. Stability relative to data and model perturbations provides a minimum requirement for interpretability and reproducibility of data driven results [143].

We make important conceptual progress on stability by extending it to the entire data science life cycle (including problem formulation and data cleaning and EDA before and after modeling). In addition, we view data and/or model perturbations as a means to stabilize data results and evaluate their variability through PCS inference procedures, for which prediction also plays a central role. The proposed inference procedures are favorably illustrated in a feature selection problem through data-inspired sparse linear model simulation studies and in a genomics case study. To communicate the many human judgment calls in the data science life cycle, we proposed PCS documentation, which integrates narratives justifying judgment calls with reproducible codes. This documentation makes data-driven decisions as transparent as possible so that users of data results can determine their reliability.

In summary, we have offered a new conceptual and practical framework to guide the data science life cycle, but many open problems remain. Integrating stability and computability into PCS beyond the modeling stage requires new computing platforms that are future work. The PCS inference proposals, even in the modeling stage, need to be vetted in practice well beyond the case studies in this paper and in our previous works, especially by other researchers. Based on feedback from practice, theoretical studies of PCS procedures in the modeling stage are also called for to gain further insights under stylized models after sufficient empirical vetting. Finally, although there have been some theoretical studies on the simultaneous connections between the three principles (see [24] and references therein), much more work is necessary.

\textsuperscript{9}For a precise definition of interpretability, we refer to our recent paper [96]
Chapter 3

Iterative Random Forests (iRF)

3.1 Introduction

High throughput, genome-wide measurements of protein-DNA and protein-RNA interactions are driving new insights into the principles of functional regulation. For instance, databases generated by the Berkeley Drosophila Transcriptional Network Project (BDTNP) and ENCODE consortium provide maps of transcription factor (TF) binding events and chromatin marks for substantial fractions of the regulatory factors active in the model organism Drosophila melanogaster and human-derived cell lines respectively [42, 130, 85, 16, 63, 38]. A central challenge with these data lies in the fact that ChIP-seq, the principal tool used to measure DNA-protein interactions, assays a single protein target at a time. In well studied systems, regulatory factors such as TFs act in concert with other chromatin-associated and RNA-associated proteins, often through stereospecific interactions [63, 32], and for a review see [64]. While several methods have been developed to identify interactions in large genomics datasets, for example [147, 90, 142], these approaches either focus on pairwise relationships or require explicit enumeration of higher-order interactions, which becomes computationally infeasible for even moderate-sized datasets. In this chapter, we present a computationally efficient tool for directly identifying high-order interactions in a supervised learning framework. We note that the interactions we identify do not necessarily correspond to biomolecular complexes or physical interactions. However, among the pairwise Drosophila TF interactions identified as stable, 80% have been previously reported. The empirical success of our approach, combined with its computational efficiency, stability, and interpretability, make it uniquely positioned to guide inquiry into the high-order mechanisms underlying functional regulation.

Popular statistical and machine learning methods for detecting interactions among features include decision trees and their ensembles: CART [20], random forests (RFs) [18], Node Harvest [94], Forest Garotte [93], and Rulefit3 [48], as well as methods more specific to gene-gene interactions with categorical features: logic regression [114], Multifactor Dimensionality Reduction [109], and Bayesian Epistasis mapping [146]. With the exception of RFs,
the above tree-based procedures grow shallow trees to prevent overfitting, excluding the possibility of detecting high-order interactions without affecting predictive accuracy. RFs are an attractive alternative, leveraging high-order interactions to obtain state-of-the-art prediction accuracy. However, interpreting interactions in the resulting tree ensemble remains a challenge.

We take a step towards overcoming these issues by proposing a fast algorithm built on RFs that searches for stable, high-order interactions. Our method, the iterative random forest algorithm (iRF), sequentially grows feature-weighted RFs to perform soft dimension reduction of the feature space and stabilize decision paths. We decode the fitted RFs using a generalization of the Random Intersection Trees algorithm (RIT) [116]. This procedure identifies high-order feature combinations that are prevalent on the RF decision paths. In addition to the high predictive accuracy of RFs, the decision tree base learner captures the underlying biology of local, combinatorial interactions [80], an important feature for biological data, where a single molecule often performs many roles in various cellular contexts. Moreover, invariance of decision trees to monotone transformations [20] to a large extent mitigates normalization issues that are a major concern in the analysis of genomics data, where signal-to-noise ratios vary widely even between biological replicates [77, 82]. Using empirical and numerical examples, we show that iRF is competitive with RF in terms of predictive accuracy, and extracts both known and compelling, novel interactions in two motivating biological problems in epigenomics and transcriptomics. An open source R implementation of iRF is available through CRAN [8].
3.2 Our method: iterative random forests

The iRF algorithm searches for high-order feature interactions in three steps. First, iterative feature re-weighting adaptively regularizes RF fitting. Second, decision rules extracted from a feature-weighted RF map from continuous or categorical to binary features. This mapping allows us to identify prevalent interactions in RF through a generalization of RIT, a computationally efficient algorithm that searches for high-order interactions in binary data [116]. Finally, a bagging step assesses the stability of recovered interactions with respect to the bootstrap-perturbation of the data. We briefly review feature-weighted RF and RIT before presenting iRF.

Preliminaries: Feature-weighted RF and RIT

To reduce the dimensionality of the feature space without removing marginally unimportant features that may participate in high-order interactions, we use a feature-weighted version of RF. Specifically, for a set of non-negative weights \( w = (w_1, \ldots, w_p) \), where \( p \) is the number of features, let \( RF(w) \) denote a feature-weighted RF constructed with \( w \). In \( RF(w) \), instead of taking a uniform random sample of features at each split, one chooses the \( j^{th} \) feature with probability proportional to \( w_j \). Weighted tree ensembles have been proposed in [4] under the name “enriched random forests” and used for feature selection in genomic data analysis. Note that with this notation, Breiman’s original RF amounts to \( RF(1/p, \ldots, 1/p) \).

iRF build upon a generalization of RIT, an algorithm that performs a randomized search for high-order interactions among binary features in a deterministic setting. More precisely, RIT searches for co-occurring collections of \( s \) binary features, or order-\( s \) interactions, that appear with greater frequency in a given class. The algorithm recovers such interactions with high probability (relative to the randomness it introduces) at a substantially lower computational cost than \( O(p^s) \), provided the interaction pattern is sufficiently prevalent in the data and individual features are sparse. We briefly present the basic RIT algorithm and refer readers to the original paper [116] for a complete description.

Consider a binary classification problem with \( n \) observations and \( p \) binary features. Suppose we are given data in the form \((I_i, Z_i), i = 1, \ldots, n\). Here, each \( Z_i \in \{0, 1\} \) is a binary label and \( I_i \subseteq \{1, 2, \ldots, p\} \) is a feature-index subset indicating the indices of “active” features associated with observation \( i \). In the context of gene transcription, \( I_i \) can be thought of as a collection of TFs and histone modifications with abnormally high or low enrichments near the \( i^{th} \) gene’s promoter region, and \( Z_i \) can indicate whether gene \( i \) is transcribed or not. With these notations, prevalence of an interaction \( S \subseteq \{1, \ldots, p\} \) in the class \( C \in \{0, 1\} \) is defined as

\[
P_n(S|Z = C) := \frac{\sum_{i=1}^{n} \mathbb{1}(S \subseteq I_i)}{\sum_{i=1}^{n} \mathbb{1}(Z_i = C)},
\]

where \( P_n \) denotes the empirical probability distribution and \( \mathbb{1}(\cdot) \) the indicator function. For given thresholds \( 0 \leq \theta_0 < \theta_1 \leq 1 \), RIT performs a randomized search for interactions \( S \) satisfying
For each class $C \in \{0, 1\}$ and a pre-specified integer $D$, let $j_1, \ldots, j_D$ be randomly chosen indices from the set of observations $\{i : Z_i = C\}$. To search for interactions $S$ satisfying condition (3.1), RIT takes $D$-fold intersections $I_{j_1} \cap I_{j_2} \cap \ldots \cap I_{j_D}$ from the randomly selected observations in class $C$. To reduce computational complexity, these interactions are performed in a tree-like fashion (Sec. 3.A), where each non-leaf node has $n_{\text{child}}$ children. This process is repeated $M$ times for a given class $C$, resulting in a collection of survived interactions $S = \bigcup_{m=1}^{M} S_m$, where each $S_m$ is the set of interactions that remain following the $D$-fold intersection process in tree $m = 1, \ldots, M$. The prevalences of interactions across different classes are subsequently compared using condition (3.1). The main intuition is that if an interaction $S$ is highly prevalent in a particular class, it will survive the $D$-fold intersection with high probability.

Iterative random forests

The iRF algorithm places interaction discovery in a supervised learning framework to identify class-specific, active index sets required for RIT. This framing allows us to recover high-order interactions that are associated with accurate prediction in feature-weighted RFs.

We consider the binary classification setting with training data $D$ in the form $\{(x_i, y_i)\}_{i=1}^{n}$, with continuous or categorical features $x = (x_1, \ldots, x_p)$, and a binary label $y \in \{0, 1\}$. Our goal is to find subsets $S \subseteq \{1, \ldots, p\}$ of features, or interactions, that are both highly prevalent within a class $C \in \{0, 1\}$, and that provide good differentiation between the two classes. To encourage generalizability of our results, we search for interactions in ensembles of decision trees fitted on bootstrap samples of $D$. This allows us to identify interactions that are robust to small perturbations in the data. Before describing iRF, we present a generalized RIT that uses any RF, weighted or not, to generate active index sets from continuous or categorical features. Our generalized RIT is independent of the other iRF components in the sense that other approaches could be used to generate the input for RIT. We remark on our particular choices in Sec. 3.B.

Generalized RIT (through an RF): For each tree $t = 1, \ldots, T$ in the output tree ensemble of an RF, we collect all leaf nodes and index them by $j_t = 1, \ldots, J(t)$. Each feature-response pair $(x_i, y_i)$ is represented with respect to a tree $t$ by $(I_{i_t}, Z_{i_t})$, where $I_{i_t}$ is the set of unique feature indices falling on the path of the leaf node containing $(x_i, y_i)$ in the $t^{th}$ tree. Hence, each $(x_i, y_i)$ produces $T$ such index set and label pairs, corresponding to the $T$ trees. We aggregate these pairs across observations and trees as

$$\mathcal{R} = \{(I_{i_t}, Z_{i_t}) : x_i \text{ falls in leaf node } i_t \text{ of tree } t\}$$

and apply RIT on this transformed dataset $\mathcal{R}$ to obtain a set of interactions.

We now describe the three components of iRF. A depiction is shown in Figure 3.1 and the complete workflow is presented in Sec. 3.A. We remark on the algorithm further in Sec. 3.B.
1. Iteratively re-weighted RF: Given an iteration number $K$, iRF iteratively grows $K$ feature-weighted RFs $RF(w^{(k)})$, $k = 1, \ldots, K$, on the data $D$. The first iteration of iRF ($k = 1$) starts with $w^{(1)} := (1/p, \ldots, 1/p)$, and stores the importance (mean decrease in Gini impurity) of the $p$ features as $v^{(1)} = (v^{(1)}_1, \ldots, v^{(1)}_p)$. For iterations $k = 2, \ldots, K$, we set $w^{(k)} = v^{(k-1)}$ and grow a weighted RF with weights set equal to the RF feature importance from the previous iteration. Iterative approaches for fitting RFs have been previously proposed in [5] and combined with hard thresholding to select features in microarray data.

2. Generalized RIT (through $RF(w^{(K)})$): We apply generalized RIT to the last feature-weighted RF grown in iteration $K$. That is, decision rules generated in the process of fitting $RF(w^{(K)})$ provide the mapping from continuous or categorical to binary features required for RIT. This process produces a collection of interactions $S$.

3. Bagged stability scores: In addition to bootstrap sampling in the weighted RF, we use an “outer layer” of bootstrapping to assess the stability of recovered interactions. We generate bootstrap samples of the data $D_b, b = 1, \ldots, B$, fit $RF(w^{(K)})$ on each bootstrap sample $D_b$, and use generalized RIT to identify interactions $S_b$ across each bootstrap sample. We define the stability score of an interaction $S \in \bigcup_{b=1}^{B} S_b$ as

$$sta(S) = \frac{1}{B} \sum_{b=1}^{B} 1\{S \in S_b\},$$

representing the proportion of times (out of $B$ bootstrap samples) an interaction appears as an output of RIT. This averaging step is exactly the Bagging idea of Breimain [17].

**iRF tuning parameters**

The iRF algorithm inherits tuning parameters from its two base algorithms, RF and RIT. The predictive performance of RF is known to be highly resistant to choice of parameters [18], so we use the default parameters in the R randomForest package. Specifically, we set the number of trees $ntree = 500$, the number of variables sampled at each node $mtry = \sqrt{p}$, and grow trees to purity. For the RIT algorithm, we use the basic version or Algorithm 1 of [116], and grow $M = 500$ intersection trees of depth $D = 5$ with $n_{child} = 2$, which empirically leads to a good balance between computation time and quality of recovered interactions. We find that both prediction accuracy and interaction recovery of iRF are fairly robust to these parameter choices (Sec. 3.B).

In addition to the tuning parameters of RF and RIT, the iRF workflow introduces two additional tuning parameters: (i) number of bootstrap samples $B$ (ii) number of iterations $K$. Larger values of $B$ provide a more precise description of the uncertainty associated with each interaction at the expense of increased computation cost. In our simulations and case studies we set $B \in (10, 30)$ and find that results are qualitatively similar in this range. The number of iterations controls the degree of regularization on the fitted RF. We find that the quality of recovered interactions can improve dramatically for $K > 1$ (Sec. 3.E). In Sec. 3.3 and Sec. 3.4, we report interactions with $K$ selected by 5-fold cross validation.
Figure 3.1: iRF workflow. Iteratively re-weighted RF (blue boxes) are trained on full data $D$ and pass Gini importance as weights to the next iteration. In iteration $K$ (red box), feature-weighted RF are grown using $w^{(K)}$ on $B$ bootstrap samples of the full data $D_{(1)}, \ldots, D_{(B)}$. Decision paths and predicted leaf node labels are passed to RIT (green box) which computes prevalent interactions in the RF ensemble. Recovered interactions are scored for stability across (outer-layer) bootstrap samples.
CHAPTER 3. ITERATIVE RANDOM FORESTS (IRF)

3.3 Case study I: enhancer elements in Drosophila

Development and function in multicellular organisms rely on precisely regulated spatio-temporal gene expression. Enhancers play a critical role in this process by coordinating combinatorial TF binding, whose integrated activity leads to patterned gene expression during embryogenesis [79]. In the early Drosophila embryo, a small cohort of ∼40 TFs drive patterning, for a review see [110], providing a well-studied, simplified model system in which to investigate the relationship between TF binding and enhancer activities. Extensive work has resulted in genome-wide, quantitative maps of DNA occupancy for 23 TFs [91] and 13 histone modifications [38], as well as labels of enhancer status for 7809 genomic sequences in blastoderm (stage 5) Drosophila embryos [42, 10]. See Sec. 3.C for descriptions of data collection and preprocessing.

To investigate the relationship between enhancers, TF binding, and chromatin state, we used iRF to predict enhancer status for each of the genomic sequences (3912 training, 3897 test). We achieved an area under the precision-recall curve (AUC-PR) on the held-out test data of 0.5 for $K = 5$ (Figure 3.2A). This corresponds to a Matthews correlation coefficient (MCC) of 0.43 (positive predictive value (PPV) of 0.71) when predicted probabilities are thresholded to maximize MCC in the training data.

Figure 3.2B reports stability scores of recovered interactions for $K = 5$. We note that the data analyzed are whole-embryo and interactions found by iRF do not necessarily represent physical complexes. However, for the well-studied case of pairwise TF interactions, 80% of our findings with stability score > 0.5 have been previously reported as physical (Table 3.1). For instance, interactions among gap proteins Giant (Gt), Krüppel (Kr), and Hunchback (Hb), some of the most well characterized interactions in the early Drosophila embryo [102], are all highly stable ($sta(Gt - Kr) = 1.0$, $sta(Gt - Hb) = 0.93$, $sta(Hb - Kr) = 0.73$). Physical evidence supporting high-order mechanisms is a frontier of experimental research and hence limited, but our excellent pairwise results give us hope that high-order interactions we identify as stable have a good chance of being confirmed by follow-up work.

iRF also identified several high-order interactions surrounding the early regulatory factor Zelda (Zld) ($sta(Zld - Gt - Twi) = 1.0$, $sta(Zld - Gt - Kr) = 0.7$). Zld has been previously shown to play an essential role during the maternal-zygotic transition [86, 56], and there is evidence to suggest that Zld facilitates binding to regulatory elements [43]. We find that Zld binding in isolation rarely drives enhancer activity, but in the presence of other TFs, particularly the anterior-posterior (AP) patterning factors Gt and Kr, it is highly likely to induce transcription. This generalizes the dependence of Bicoid—induced transcription on Zld binding to several of the AP factors [141], and is broadly consistent with the idea that Zld is potentiating, rather than an activating factor [43].

More broadly, response surfaces associated with stable high-order interactions indicate AND-like rules (Figure 3.2C). In other words, the proportion of active enhancers is substantially higher for sequences where all TFs are sufficiently bound, compared to sequences where only some of the TFs exhibit high levels of occupancy. Figure 3.2C demonstrates a putative third order interaction found by iRF ($sta(Kr - Gt - Zld) = 0.7$). To the left, the
Gt-Zld response surface is plotted using only sequences for which Kr occupancy is lower than the median Kr level, and the proportion of active enhancers is uniformly low (< 10%). The response surface to the right, is plotted using only sequences where Kr occupancy is higher than median Kr level and shows that the proportion of active elements is as high as 60% when both Zld and Gt are sufficiently bound. This points to an order-3 AND rule, where all three proteins are required for enhancer activation in a subset of sequences. In Figure 3.2D, we show the subset of sequences that correspond to this AND rule (highlighted in red) using a superheat map [7], which juxtaposes two separately clustered heatmaps corresponding to active and inactive elements. Note that the response surfaces are drawn using held-out test data to illustrate the generalizability of interactions detected by iRF. While overlapping patterns of TF binding have been previously reported [91], to the best of our knowledge this is the first report of an AND-like response surface for enhancer activation. Third-order interactions have been studied in only a handful of enhancer elements, most notably eve stripe 2, for a review see [78], and our results indicate that they are broadly important for the establishment of early zygotic transcription, and therefore body patterning.
Figure 3.2: [A]: Accuracy of iRF (AUC-PR) in predicting active elements from TF binding and histone modification data. [B]: 20 most stable interactions recovered by iRF after 5 iterations. Interactions that are a strict subset of another interaction with stability score ≥ 0.5 have been removed for cleaner visualization. iRF recovers known interactions among Gt, Kr and Hb and interacting roles of master regulator Zld. [C]: Surface maps demonstrating the proportion of active enhancers by quantiles of Zld, Gt, and Kr binding (held-out test data). On the subset of data where Kr binding is lower than the median Kr level, proportion of active enhancers does not change with Gt and Zld. On the subset of data with Kr binding above the median level, structure of the response surface reflects an order-3 AND interaction: increased levels of Zld, Gt, and Kr binding are indicative of enhancer status for a subset of observations. [D]: Quantiles of Zld, Gt, and Kr binding grouped by enhancer status (balanced sample of held-out test data). The block of active elements highlighted in red represents the subset of observations for which the AND interaction is active.
3.4 Case study II: alternative splicing in a human-derived cell line

In eukaryotes, alternative splicing of primary mRNA transcripts is a highly regulated process in which multiple distinct mRNAs are produced by the same gene. In the case of messenger RNAs (mRNAs), the result of this process is the diversification of the proteome, and hence the library of functional molecules in cells. The activity of the spliceosome, the ribonucleoprotein responsible for most splicing in eukaryotic genomes, is driven by complex, cell-type specific interactions with cohorts of RNA binding proteins (RBP) [120, 125], suggesting that high-order interactions play an important role in the regulation of alternative splicing. However, our understanding of this system derives from decades of study in genetics, biochemistry, and structural biology. Learning interactions directly from genomics data has the potential to accelerate our pace of discovery in the study of co- and post-transcriptional gene regulation.

Studies, initially in model organisms, have revealed that the chromatin mark H3K36me3, the DNA binding protein CTCF, and a few other factors all play splice-enhancing roles [68, 118, 70]. However, the extent to which chromatin state and DNA binding factors interact en masse to modulate co-transcriptional splicing remains unknown [2]. To identify interactions that form the basis of chromatin mediated splicing, we used iRF to predict thresholded splicing rates for 23823 exons (RNA-seq Percent-spliced-in (PSI) values [104]; 11911 train, 11912 test), from ChIP-seq assays measuring enrichment of chromatin marks and TF binding events (253 ChIP assays on 107 unique transcription factors and 11 histone modifications). Preprocessing methods are described in Sec. 3.C.

In this prediction problem, we achieved an AUC-PR on the held-out test data of 0.51 for $K = 2$ (Figure 3.3A). This corresponds to a MCC of 0.47 (PPV 0.72) on held-out test data when predicted probabilities are thresholded to maximize MCC in the training data. Figure 3.3B reports stability scores of recovered interactions for $K = 2$. We find interactions involving H3K36me3, a number of novel interactions involving other chromatin marks, and post-translationally modified states of RNA Pol II. In particular, we find that the impact of serine 2 phosphorylation of Pol II appears highly dependent on local chromatin state. Remarkably, iRF identified an order-6 interaction surrounding H3K36me3 and S2 phospho-Pol II (stability score 0.5, Figure 3.3B,C) along with two highly stable order 5 subsets of this interaction (stability scores 1.0). A subset of highly spliced exons highlighted in red is enriched for all 6 of these elements, indicating a potential AND-type rule related to splicing events (Figure 3.3C). This observation is consistent with, and offers a quantitative model for the previously reported predominance of co-transcriptional splicing in this cell line [132]. We note that the search space of order-6 interactions is $> 10^{11}$, and that this interaction is discovered with an order-zero increase over the computational cost of finding important features using RF. Recovering such interactions without exponential speed penalties represents a substantial advantage over previous methods and positions our approach uniquely for the discovery of complex, nonlinear interactions.
Figure 3.3: [A]: Accuracy of iRF (AUC-PR) in classifying included exons from excluded exons in held-out test data. iRF shows 7% increase in AUC-PR over RF. [B]: An order-6 interaction recovered by iRF (stability score 0.5) displayed on a superheat map which juxtaposes two separately clustered heatmaps of exons with high and low splicing rates. Co-enrichment of all the 6 plotted features reflects an AND-type rule indicative of high splicing rates for the exons highlighted in red (held-out test data). The subset of Pol II, S2 phospho-Pol II, H3K36me3, H3K79me2, and H4K20me1 was recovered as an order-5 interaction in all bootstrap samples (stability score 1.0). [C]: 20 most stable interactions recovered in the second iteration of iRF. Interactions that are a strict subset of another interaction with stability score $\geq 0.5$ have been removed for cleaner visualization.
3.5 Discussion

Systems governed by nonlinear interactions are ubiquitous in biology. We developed a predictive and stable method, iRF, for learning such feature interactions. iRF identified known and novel interactions in early zygotic enhancer activation in the Drosophila embryo, and posit new high-order interactions in splicing regulation for a human-derived system.

Validation and assessment of complex interactions in biological systems is necessary and challenging, but new wet-lab tools are becoming available for targeted genome and epigenome engineering. For instance, the CRISPR system has been adjusted for targeted manipulation of post-translational modifications to histones [60]. This may allow for tests to determine if modifications to distinct residues at multivalent nucleosomes function in a non-additive fashion in splicing regulation. Several of the histone marks that appear in the interactions we report, including H3K36me3 and H4K20me1, have been previously identified [54] as essential for establishing splicing patterns in the early embryo. Our findings point to direct interactions between these two distinct marks. This observation generates interesting questions: What proteins, if any, mediate these dependencies? What is the role of Phospho-S2 Pol II in the interaction? Proteomics on ChIP samples may help reveal the complete set of factors involved in these processes, and new assays such as Co-ChIP may enable the mapping of multiple histone marks at single-nucleosome resolution [137].

We have offered evidence that iRF constitutes a useful tool for generating new hypotheses from the study of high-throughput genomics data, but many challenges await. iRF currently handles data heterogeneity only implicitly, and the order of detectable interaction depends directly on the depth of the tree, which is on the order of $\log_2(n)$. We are currently investigating local importance measures to explicitly relate discovered interactions to specific observations. This strategy has the potential to further localize feature selection and improve the interpretability of discovered rules. Additionally, iRF does not distinguish between interaction forms, for instance additive versus non-additive. We are exploring tests of rule structure to provide better insights into the precise form of rule-response relationships.

To date, machine learning has been driven largely by the need for accurate prediction. Leveraging machine learning algorithms for scientific insights into the mechanics that underlie natural and artificial systems will require an understanding of why prediction is possible. The Stability Principle, which asserts that statistical results should at a minimum be reproducible across reasonable data and model perturbations, has been advocated in [143] as a second consideration to work towards understanding and interpretability in science. Iterative and data-adaptive regularization procedures such as iRF are based on prediction and stability and have the potential to be widely adaptable to diverse algorithmic and computational architectures, improving interpretability and informativeness by increasing the stability of learners.
Appendix
3. A Algorithms

The basic versions of the Random Intersection Trees (RIT) and iterative Random Forests (iRF) algorithms are presented below. For a complete description of RIT, including analysis of computational complexity and theoretical guarantees, we refer readers to the original paper [116]. For a full description of iRF, we refer readers to Section 2.

Algorithm 1: Random Intersection Trees [116]

**Input:** \{(I_i, Z_i); I_i \subseteq \{1, \ldots, p\}, Z_i \in \{0, 1\}\}_{i=1}^n, C \in \{0, 1\}

**Tuning Parameters:** (D, M, n_{child})

1. for tree \(m \leftarrow 1\) to \(M\) do
2. Let \(m\) be a tree of depth \(D\), with each node \(j\) in levels 0, \ldots, \(D-1\) having \(n_{child}\) children, and denote the parent of node \(j\) as \(pa(j)\). Let \(J\) be the total number of nodes in the tree, and index the nodes such that for every parent-child pair, larger indices are assigned to the child than the parent. For each node \(j = 1, \ldots, J\), let \(i_j\) be a uniform sample from the set of class \(C\) observations \(\{i: Z_i = C\}\).
3. Set \(S_1 = I_{i_1}\)
4. for \(j = 2\) to \(J\) do
5. \(S_j \leftarrow I_{i_j} \cap S_{pa(j)}\)
6. end
7. return \(S_m = \{S_j: \text{depth}(j) = D\}\)
8. end

**Output:** \(S = \bigcup_{m=1}^M S_m\)

Algorithm 2: iterative Random Forests

**Input:** \(D, C \in \{0, 1\}, B, K, w^{(1)} \leftarrow (1/p, \ldots, 1/p)\)

1. (1) for \(k \leftarrow 1\) to \(K\) do
2. Fit \(RF(w^{(k)})\) on \(D\)
3. \(w^{(k+1)} \leftarrow \text{Gini importance of } RF(w^{(k)})\)
4. end
5. (2) for \(b \leftarrow 1\) to \(B\) do
6. Generate bootstrap samples \(D_b\) of the form \(\{x_{b(i)}, y_{b(i)}\}\) from \(D\)
7. Fit \(RF(w^{(K)})\) on \(D_b\)
8. \(R_{(b)} \leftarrow \{(I_{i_t}, Z_{i_t}) : x_{b(i)} \text{ falls in leaf node } i_t \text{ of tree } t\}\)
9. \(S_{(b)} \leftarrow \text{RIT}(R_{(b)}, C)\)
10. end
11. (3) for \(S \in \bigcup_{b=1}^B S_{(b)}\) do
12. \(\text{sta}(S) = \frac{1}{B} \sum_{b=1}^B 1 [S \in S_{(b)}]\)
13. end

**Output:** \(\{S, \text{sta}(S)\}_{S \in \bigcup_{b=1}^B S_{(b)}}\)

**Output:** \(\{RF(w^{(K)})\) on \(D\}\)
3.B Remarks on iRF

Iterative re-weighting

Generalized RIT can be used with any Random Forest (RF) method, weighted or not. We find that iterative re-weighting acts as a soft dimension reduction step by encouraging RF to select a stable set of features on decision paths. This leads to improved recovery of high-order interactions in our numerical simulations and in real data settings. For instance, without feature re-weighting \((k = 1)\) iRF rarely recovers interactions of order \(> 2\) in our simulations. Feature re-weighting \((k > 1)\) allows iRF to identify order-8 data generating rules as highly stable interactions for comparable parameter settings. In the enhancer case study, iRF \((k = 5)\) recovers 9 order-3 interactions with stability score \(> 0.5\). Without iterative re-weighting, iRF \((k = 1)\) does not recover any order-3 interactions with stability score \(> 0.5\). The fourth iteration of iRF also recovers many additional order-3, order-4, and order-5 interactions with lower stability scores that are not recovered in the first iteration. Although it is unclear which of these high-order interactions represent true biological mechanisms without experimental follow-up, our simulation based on the enhancer data suggests that the overall quality of recovered interactions improves with iteration (Figure 3.19).

Iterative re-weighting can be viewed as a form of regularization on the base RF learner, since it restricts the form of functions RF is allowed to fit in a probabilistic manner. In particular, we find that iterative re-weighting reduces the dimensionality of the feature space without removing marginally unimportant features that participate in high-order interactions (Figure 3.13). Moreover, we find that iteratively re-weighted and unweighted RF achieve similar predictive accuracy on held out test data. We note that other forms of regularization such as [29] may also lead to improved interaction recovery, though we do not explore them in this paper.

Generalized RIT

The RIT algorithm could be generalized through any approach that selects active features from continuous or categorical data. However, the feature selection procedure affects recovered interactions and is thus an important consideration in generalizing RIT to continuous or categorical features. There are several reasons we use an RF-based approach. First, RFs are empirically successful predictive algorithms that provide a principled, data-driven procedure to select active features specific to each observation. Second, randomness inherent to tree ensembles offers a natural way to generate multiple active index sets for each observation \(x_i\), making the representations more robust to small data perturbations. Finally, our approach allows us to interpret (in a computationally efficient manner given by RIT) complex, high-order relationships that drive the impressive predictive accuracy of RFs, granting new insights into this widely used class of algorithms.
CHAPTER 3. ITERATIVE RANDOM FORESTS (IRF)

Node sampling

In the generalized RIT step of iRF, we represent each observation $i = 1, \ldots, n$ by $T$ rule-response pairs, determined by the leaf nodes containing observation $i$ in each tree $t = 1, \ldots, T$ of an RF. We accomplish this by replicating each rule-response pair $(I_{jt}, Z_{jt})$ in tree $t$ based on the number of observations in the corresponding leaf node. We view this as a natural representation of the observations in $D$, made more robust to sampling perturbations through rules derived from bootstrap samples of $D$. Our representation is equivalent to sampling rule-response pairs $(I_{jt}, Z_{jt})$ in RIT with probability proportional to the number of observations that fall in the leaf node. However, one could sample or select a subset of leaf nodes based on other properties such as homogeneity and/or predictive accuracy. We are exploring how different sampling strategies impact recovered interactions in our ongoing work.

Bagged stability scores

iRF uses two layers of bootstrap sampling. The “inner” layer takes place when growing weighted RF. By drawing a separate bootstrap sample from the input data before growing each tree, we can learn multiple binary representations of each observation $x_i$ that are more robust to small data perturbations. The “outer” layer of bootstrap sampling is used in the final iteration of iRF. Growing $RF(w^{(K)})$ on different bootstrap samples allows us to assess the stability, or uncertainty, associated with the recovered interactions.

Relation to AdaBoost

In his original paper on RF [18], Breiman conjectured that in the later stages of iteration, AdaBoost [47] emulates RF. iRF inherits this property, and in addition shrinks the feature space towards more informative features. As pointed out by a reviewer, there is an interesting connection between AdaBoost and iRF. Namely, AdaBoost improves on the least reliable part of the data space, while iRF zooms in on the most reliable part of feature space. This is primarily motivated by the goals of the two learners — AdaBoost’s primary goal is prediction, whereas iRF’s primary goal is to select features or combinations of features while retaining predictive power. We envision that zooming in on both the data and feature space simultaneously may harness the strengths of both learners. As mentioned in the conclusion, we are exploring this direction through local feature importance.

Sensitivity to tuning parameters

The predictive performance of RF is known to be highly robust to choice of tuning parameters [18]. To test iRF’s sensitivity to tuning parameters, we investigated the stability of both prediction accuracy (AUC-PR) and interaction recovery across a range of parameter settings. Results are reported for both the enhancer and splicing datasets presented in our case studies.
The prediction accuracy of iRF is controlled through both RF parameters and number of iterations. Figure 3.4 and Figure 3.5 report 5-fold cross-validation prediction accuracy as a function of number of iterations ($k$), number of trees in the RF ensemble ($ntree$), and the number of variables considered for each split ($mtry$). We do not consider tree depth as a tuning parameter since deep decision trees (e.g. grown to purity) are precisely what allows iRF to identify high-order interactions. Aside from iteration $k = 1$ in the splicing data, prediction accuracy is highly consistent across parameter choices. For the first iteration in the splicing data, prediction accuracy increases as a function of $mtry$. We hypothesize that this is the result of many extraneous features that make it less likely for important features to be among the $mtry$ selected features at each split. Our hypothesis is consistent with the improvement in prediction accuracy that we observe for iterations $k > 1$, where re-weighting allows iRF to sample important features with higher probability. This finding also suggests a potential relationship between iterative re-weighting and RF tuning parameters. The extent to which RF tuning parameters can be used to stabilize decision paths and allow for the recovery of high-order interactions is an interesting question for further exploration.

The interactions recovered by iRF are controlled through RIT parameters and the number of iterations. Our simulations in Sec. 3.E extensively examine the relationship between the number of iterations and recovered interactions. Figure 3.6 and Figure 3.7 report the stability scores of recovered interactions in the enhancer and splicing data as a function of RIT parameters. In general, the stability scores of recovered interactions are highly correlated between different RIT parameter settings, indicating that our results are robust over the reported range of tuning parameters. The greatest differences in stability scores occur for low values of depth ($D$) and number of children ($n_{child}$). In particular, a subset of interactions that are highly stable for larger values of $n_{child}$ are less stable with $n_{child} = 1$. In contrast, a subset of interactions that are highly stable for $D = 3$ are considered less stable for larger values of $D$. We note that the findings in our case studies are qualitatively unchanged as tuning parameters are varied. Interactions we identified as most stable under the default parameter choices remain the most stable under different parameter choices.

Regression and multiclass classification

We presented iRF in the binary classification setting, but our algorithm can be naturally extended to multiclass or continuous responses. The requirement that responses are binary is only used to select a subset of leaf nodes as input to generalized RIT. In particular, for a given class $C \in \{0, 1\}$, iRF runs RIT over decision paths whose corresponding leaf node predictions are equal to $C$. In the multiclass setting, we select leaf nodes with predicted class or classes of interest as inputs to RIT. In the regression setting, we consider leaf nodes whose predictions fall within a range of interest as inputs to generalized RIT. This range could be determined in domain-specific manner or by grouping responses through clustering techniques.
CHAPTER 3. ITERATIVE RANDOM FORESTS (IRF)

Grouped features and replicate assays

In many classification and regression problems with omics data, one faces the problem of drawing conclusion at an aggregated level of the features at hand. The simplest example is the presence of multiple replicates of a single assay, when there is neither a standard protocol to choose one assay over the other, nor a known strategy to aggregate the assays after normalizing them individually. Similar situations arise when there are multiple genes from a single pathway in the feature sets, and one is only interested in learning interactions among the pathways and not the individual genes.

In linear regression based feature selection methods like Lasso, grouping information among features is usually incorporated by devising suitable grouped penalties, which requires solving new optimization problems. The invariance property of RF to monotone transformations of features and the nature of the intersection operation used by RIT provide iRF a simple and computationally efficient workaround to this issue. In particular, one uses all the unnormalized assays in the tree growing procedure, and collapses the grouped features or replicates into a “super feature” before taking random intersections. iRF then provide interaction information among these super features, which could be used to achieve further dimension reduction of the interaction search space.

Interaction evaluation through prediction

We view the task of identifying candidate, high-order interactions as a step towards hypothesis generation in complex systems. An important next step will be evaluating the interactions recovered by iRF to determine whether they represent domain-relevant hypotheses. This is an interesting and challenging problem that will require subject matter knowledge into the anticipated forms of interactions. For instance, biomolecules are believed to interact in stereospecific groups [97] that can be represented through Boolean-type rules. Thus, tests of non-additivity may provide insight into which iRF-recovered interactions warrant further examination in biological systems.

We do not consider domain-specific evaluation in this paper, but instead assess interactions through broadly applicable metrics based on both stability and predictability. We incorporated the Stability Principle [143] through both iterative re-weighting, which encourages iRF to use a consistent set of features along decision paths, and through bagged stability scores, which provide a metric to evaluate how consistently decision rules are used throughout an RF. Here we propose two additional validation metrics based on predictive accuracy.

Conditional prediction: Our first metric evaluates a recovered interaction \( S \subseteq \{1, \ldots, p\} \) based on the predictive accuracy of an RF that makes predictions using only leaf nodes for which all features in \( S \) fall on the decision path. Specifically, for each observation \( i = 1, \ldots, n \) we evaluate its predicted value from each tree \( t = 1, \ldots, T \) with respect to an interaction \( S \) as
\[
\hat{y}_i(t; S) = \begin{cases} 
Z_{i_t} & \text{if } S \subseteq I_{i_t} \\
\mathbb{P}_n(y = 1) & \text{else}
\end{cases}
\]

where \(Z_{i_t}\) is the prediction of the leaf node containing observation \(i\) in tree \(t\), \(I_{i_t}\) is the index set of features falling on the decision path for this leaf node, and \(\mathbb{P}_n(y = 1)\) is the empirical proportion of class 1 observations \(\{i : y_i = 1\}\). We average these predictions across the tree ensemble to obtain the RF-level prediction for observation \(i\) with respect to an interaction \(S\)

\[
\hat{y}_i(S) = \frac{1}{T} \cdot \sum_{t=1}^{T} \hat{y}_i(t; S).
\] (3.3)

Predictions from equation (3.3) can be used to evaluate predictive accuracy using any metric of interest. We report AUC-PR using predictions \(\hat{y}_i(S)\) for each interaction \(S \in S\) recovered by iRF. Intuitively, this metric asks whether the leaf nodes that rely on an interaction \(S\) are good predictors when all other leaf nodes make a best-case random guess.

**Permutation importance:** Our second metric is inspired by Breiman’s permutation-based measure of feature importance [18]. In the single feature case, Breiman proposed permuting each column of the data matrix individually and evaluating the change in prediction accuracy of an RF. The intuition behind this measure of importance is that if an RF’s predictions are heavily influenced by a particular feature, permuting it will lead to a drop in predictive accuracy by destroying the feature/response relationship. The direct analogue in our setting would be to permute all features in a recovered interaction \(S\) and evaluate the change in predictive accuracy of iRF. However, this does not capture the notion that we expect features in an interaction to act collectively. By permuting a single feature, we destroy the interaction/response relationship for any interaction that the feature takes part in. If \(S\) contains features that are components of distinct interactions, permuting each feature in \(S\) would destroy multiple interaction/response relationships. To avoid this issue, we assess prediction accuracy using only information from the features contained in \(S\) by permuting all other features.

Specifically, let \(X_{\pi_{sc}}\) denote the feature matrix with all columns in \(S^c\) permuted, where \(S^c\) is the compliment of \(S\). We evaluate predictions on permuted data \(X_{\pi_{sc}}\), and use these predictions to assess accuracy with respect to a metric of interest, such as the AUC-PR. Intuitively, this metric captures the idea that if an interaction is important independently of any other features, making predictions using only this interaction should lead to improved prediction over random guessing.

**Evaluating enhancer and splicing interactions:** Figure 3.8 and Figure 3.9 report interactions from both the enhancer and splicing data, evaluated in terms of our predictive metrics. In the enhancer data, interactions between collections of TFs Zld, Gt, Hb, Kr, and Twi are ranked highly, as was the case with stability scores (Figure 3.8). In the splicing data,
POL II, S2 phospho-Pol II, H3K36me3, H3K79me2, H3K9me1, and H4K20me1 consistently appear in highly ranked interactions, providing further validation of the order-6 interaction recovered using the stability score metric (Figure 3.9).

While the interaction evaluation metrics yield qualitatively similar results, there is a clear difference in how they rank interactions of different orders. Conditional prediction and stability score tend to favor lower-order interactions and permutation importance higher-order interactions. To see why this is the case, consider interactions $S' \subset S \subseteq \{1, \ldots, p\}$. As a result of the intersection operation used by RIT, the probability (with respect to the randomness introduced by RIT) that the larger interaction $S$ survives up to depth $D$ will be less than or equal to the probability that $S'$ survives up to depth $D$. Stability scores will reflect the difference by measuring how frequently an intersection survives across bootstrap samples. In the case of conditional prediction, the leaf nodes for which $S$ falls on the decision path form a subset of leaf nodes for which $S'$ falls on the decision path. As a result, the conditional prediction with respect to $S$ uses more information from the forest and thus we would generally expect to see superior predictive accuracy. In contrast, permutation importance uses more information when making predictions with $S$ since fewer variables are permuted. Therefore, we would generally expect to see higher permutation importance scores for larger interactions. We are currently investigating approaches for normalizing these metrics to compare interactions of different orders.

Together with the measure of stability, the two importance measures proposed here capture different qualitative aspects of an interaction. Conceptually, the stability measure attempts to capture the degree of uncertainty associated with an interaction by perturbing the features and responses jointly. In contrast, the importance measures based on conditional prediction and permutation are similar to effect size, i.e., they attempt to quantify the contribution of a given interaction to the overall predictive accuracy of the learner. The conditional prediction metric accomplishes this by perturbing the predicted responses, while permutation importance perturbs the features.

3.C Data collection and preprocessing

*Drosophila* enhancers

In total, 7809 genomic sequences have been evaluated for their enhancer activity [10, 42, 49, 75] in a gold-standard, stable-integration transgenic assay. In this setting, a short genomic sequence (100-3000nt) is placed in a reporter construct and integrated into a targeted site in the genome. The transgenic fly line is amplified, embryos are collected, fixed, hybridized and immunohistochemistry is performed to detect the reporter [128, 138]. The resultant stained embryos are imaged to determine: a) whether or not the genomic segment is sufficient to drive transcription of the reporter construct, and b) where and when in the embryo expression is driven. For our prediction problem, sequences that drive patterned expression in blastoderm (stage 5) embryos were labeled as active elements. To form a set of features for predicting
enhancer status, we computed the maximum value of normalized fold-enrichment \cite{85} of ChIP-seq and ChIP-chip assays \cite{91, 38} for each genomic segment.

Our processing led to a binary classification problem with approximately 10% of genomic sequences labeled as active elements. It is important to note that the tested sequences do not represent a random sample from the genome — rather they were chosen based on prior biological knowledge and may therefore exhibit a higher frequency of positive tests than one would expect from genomic sequences in general. We randomly divided the dataset into training and test sets of 3912 and 3897 observations respectively, with approximately equal portions of positive and negative elements, and applied iRF with $B = 30$, $K = 5$. The tuning parameters in RF were set to default levels of the \texttt{R randomForest} package, and 500 Random Intersection Trees of depth 5 with $n_{\text{child}} = 2$ were grown to capture candidate interactions.

### Alternative splicing

The ENCODE consortium has collected extensive genome-wide data on both chromatin state and splicing in the human-derived erythroleukemia cell line K562 \cite{38}. To identify critical interactions that form the basis of chromatin mediated splicing, we used splicing rates (Percent-spliced-in, PSI values, \cite{103, 104}) from ENCODE RNA-seq data, along with ChIP-seq assays measuring enrichment of chromatin marks and transcription factor binding events (253 ChIP assays on 107 unique transcription factors and 11 histone modifications, \url{https://www.encodeproject.org/}).

For each ChIP assay, we computed the maximum value of normalized fold-enrichment over the genomic region corresponding to each exon. This yielded a set of $p = 270$ features for our analysis. We took our response to be a thresholded function of the PSI values for each exon. Only internal exons with high read count (at least 100 RPKM) were used in downstream analysis. Exons with Percent-spliced-in index (PSI) above 70% were classified as frequently included ($y = 1$) and exons with PSI below 30% were classified as frequently excluded exons ($y = 0$). This led to a total of 23823 exons used in our analysis.

Our threshold choice resulted in $\sim 90\%$ of observations belonging to class 1. To account for this imbalance, we report AUC-PR for the class 0 observations. We randomly divided the dataset into balanced training and test sets of 11911 and 11912 observations respectively, and applied iRF with $B = 30$ and $K = 2$. The tuning parameters in RF were set to default levels of the \texttt{R randomForest} package, and 500 binary random intersection trees of depth 5 with $n_{\text{child}} = 2$ were grown to capture candidate interactions.

### 3.D Evaluating \textit{Drosophila} enhancer interactions

The Drosophila embryo is one of the most well studied systems in developmental biology and provides a valuable test case for evaluating iRF. Decades of prior work have identified physical, pairwise TF interactions that play a critical role in regulating spatial and temporal patterning, for reviews see \cite{110} and \cite{67}. We compared our results against these previously
Table 3.1: Previously reported pairwise TF interactions recovered by iRF

<table>
<thead>
<tr>
<th>interaction (S)</th>
<th>sta(S)</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gt, Zld</td>
<td>1</td>
<td>[56, 100]</td>
</tr>
<tr>
<td>Twi, Zld</td>
<td>1</td>
<td>[56, 100]</td>
</tr>
<tr>
<td>Gt, Hb</td>
<td>1</td>
<td>[72, 71, 36]</td>
</tr>
<tr>
<td>Gt, Kr</td>
<td>1</td>
<td>[71, 127, 22, 115]</td>
</tr>
<tr>
<td>Gt, Twi</td>
<td>1</td>
<td>[85]</td>
</tr>
<tr>
<td>Kr, Twi</td>
<td>1</td>
<td>[85]</td>
</tr>
<tr>
<td>Kr, Zld</td>
<td>0.97</td>
<td>[56, 100]</td>
</tr>
<tr>
<td>Gt, Med</td>
<td>0.97</td>
<td></td>
</tr>
<tr>
<td>Bcd, Gt</td>
<td>0.93</td>
<td>[71, 36]</td>
</tr>
<tr>
<td>Bcd, Twi</td>
<td>0.93</td>
<td>[85]</td>
</tr>
<tr>
<td>Hb, Twi</td>
<td>0.93</td>
<td>[145]</td>
</tr>
<tr>
<td>Med, Twi</td>
<td>0.93</td>
<td>[99]</td>
</tr>
<tr>
<td>Kr, Med</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>D, Gt</td>
<td>0.87</td>
<td></td>
</tr>
<tr>
<td>Med, Zld</td>
<td>0.83</td>
<td>[56]</td>
</tr>
<tr>
<td>Hb, Zld</td>
<td>0.80</td>
<td>[56, 100]</td>
</tr>
<tr>
<td>Hb, Kr</td>
<td>0.80</td>
<td>[102, 66, 61]</td>
</tr>
<tr>
<td>D, Twi</td>
<td>0.73</td>
<td></td>
</tr>
<tr>
<td>Bcd, Kr</td>
<td>0.67</td>
<td>[61, 62]</td>
</tr>
<tr>
<td>Bcd, Zld</td>
<td>0.63</td>
<td>[56, 100]</td>
</tr>
</tbody>
</table>

reported physical interactions to evaluate interactions found by iRF. Table 3.1 indicates the 20 pairwise TF interactions we identify with stability score > 0.5, along with references that have previously reported physical interactions among each TF pair. In total, 16 (80%) of the 20 pairwise TF interactions we identify as stable have been previously reported in one of two forms: (i) one member of the pair regulates expression of the other (ii) joint binding of the TF pair has been associated with increased expression levels of other target genes. Interactions for which we could not find evidence supporting one of these forms are indicated as “−” in Table 3.1. We note that high-order interactions have only been studied in a small number of select cases, most notably eve stripe 2, for a review see [78]. These limited cases are not sufficient to conduct a comprehensive analysis of the high-order interactions we identify using iRF.
3.E Simulation experiments

We developed iRF through extensive simulation studies based on biologically inspired generative models using both synthetic and real data. In particular, we generated responses using Boolean rules intended to reflect the stereospecific nature of interactions among biomolecules [97]. In this section, we examine interaction recovery and predictive accuracy of iRF in a variety of simulation settings.

For all simulations we evaluated predictive accuracy in terms of area under the precision-recall curve (AUC-PR) for a held out test set of 500 observations. To evaluate interaction recovery, we use three metrics that are intended to give a broad sense of the overall quality of interactions $S$ recovered by iRF. For responses generated from an interaction $S^* \subseteq \{1, \ldots, p\}$, we consider interactions of any order between only active features $\{j : j \in S^*\}$ to be true positives and interactions containing any non-active variable $\{j : j \notin S^*\}$ to be false positives. This definition accounts for the fact that subsets of $S^*$ are still informative of the data generating mechanism. However, it conservatively considers interactions that includes any non-active features to be false positives, regardless of how many active features they contain.

1. **Interaction AUC:** We consider the area under the receiver operating characteristic (ROC) curve generated by thresholding interactions recovered by iRF at each unique stability score. This metric provides a rank-based measurement of the overall quality of iRF interaction stability scores, and takes a value of 1 whenever the complete data generating mechanism is recovered as the most stable interaction.

2. **Recovery rate:** We define an interaction as “recovered” if it is returned in any of the $B$ bootstrap samples (i.e. stability score $> 0$), or if it is a subset of any recovered interaction. This eliminates the need to select thresholds across a wide variety of parameter settings. For a given interaction order $s = 2, \ldots, |S|$, we calculate the proportion of the total $\binom{|S|}{s}$ true positive order-$s$ interactions recovered by iRF. This metric is used to distinguish between models that recover high-order interactions at different frequencies, particularly in settings where all models recover low-order interactions.

3. **False positive weight:** Let $S = S_T \cup S_F$ represent the set of interactions recovered by iRF, where $S_T$ and $S_F$ are the sets of recovered true and false positive interactions respectively. For a given interaction order $s = 2, \ldots, |S|$, we calculate

$$\frac{\sum_{S \in S_F : |S| = s} sta(S)}{\sum_{S \in S : |S| = s} sta(S)}.$$ 

This metric measures the aggregate weight of stability scores for false positive order-$s$ interactions, $S \in S_F : |S| = s$, relative to all recovered order-$s$ interactions, $S \in S : |S| = s$. This metric also includes all recovered interactions (stability score $> 0$), eliminating the need to select thresholds. It can be thought of as the weighted analogue to false discovery proportion.
CHAPTER 3. ITERATIVE RANDOM FORESTS (IRF)

Simulation 1: Boolean rules

Our first set of simulations demonstrates the benefit of iterative re-weighting for a variety of Boolean-type rules. We sampled features $x = (x_1, \ldots, x_{50})$ from independent, standard Cauchy distributions to reflect heavy-tailed data, and generated the binary responses from three rule settings (OR, AND, and XOR) as

$$y^{(OR)} = \mathbf{1}[x_1 > t_{OR} \mid x_2 > t_{OR} \mid x_3 > t_{OR} \mid x_4 > t_{OR}], \quad (3.4)$$

$$y^{(AND)} = \prod_{i=1}^{4} \mathbf{1}[x_i > t_{AND}], \quad (3.5)$$

$$y^{(XOR)} = \mathbf{1}\left[\sum_{i=1}^{4} \mathbf{1}(x_i > t_{XOR}) \equiv 1 \pmod{2}\right]. \quad (3.6)$$

We injected noise into these responses by swapping the labels for 20% of the observations selected at random. From a modeling perspective, the rules in equations (3.4), (3.5), and (3.6) give rise to non-additive main effects that can be represented as an order-4 interaction between the active features $x_1, x_2, x_3,$ and $x_4$. Inactive features $x_5, \ldots, x_{50}$ provide an additional form of noise that allowed us to assess the performance of iRF in the presence of extraneous features. For the AND and OR models, we set $t_{OR} = 3.2$, $t_{AND} = -1$ to ensure reasonable class balance ($\sim 1/3$ class 1 observations) and trained on samples of size 100, 200, ..., 500 observations. We set $t_{XOR} = 1$ both for class balance ($\sim 1/2$ class 1 observations) and to ensure that some active features were marginally important relative to inactive features. At this threshold, the XOR interaction is more difficult to recover than the others due to the weaker marginal associations between active features and the response.

To evaluate the full range of performance for the XOR model, we trained on larger samples of size 200, 400, ..., 1000 observations. We report the prediction accuracy and interaction recovery for iterations $k \in \{1, 2, \ldots, 5\}$ of iRF over 20 replicates drawn from the above generative models. The RF tuning parameters were set to default levels for the R randomForest package [87], $M = 100$ RITs of depth 5 were grown with $n_{child} = 2$, and $B = 20$ bootstrap replicates were taken to determine the stability scores of recovered interactions.

Figure 3.10A shows the prediction accuracy of iRF (AUC-PR), evaluated on held out test data, for each generative model and a selected subset of training sample sizes as a function of iteration number ($k$). iRF achieves comparable or better predictive performance for increasing $k$, with the most dramatic improvement in the XOR model. It is important to note that only 4 out of the 50 features are used to generate responses in equations (3.4), (3.5), and (3.6). Iterative re-weighting restricts the form of functions fitted by RF and may hurt predictive performance when the generative model is not sparse.

Figure 3.10B shows interaction AUC by generative model, iteration number, and training sample size, demonstrating that iRF ($k > 1$) tends to rank true interactions higher with respect to stability score than RF ($k = 1$). Figure 3.10C breaks down recovery by interaction...
order, showing the proportion of order-$s$ interactions recovered across any bootstrap sample (stability score $> 0$), averaged over 20 replicates. For each of the generative models, RF ($k = 1$) never recovers the true order-4 interaction while iRF ($k = 4, 5$) always identifies it as the most stable order-4 interaction given enough training observations. The improvement in interaction recovery with iteration is accompanied by an increase in the stability scores of false positive interactions (Figure 3.10D). We find that this increase is generally due to many false interactions with low stability scores as opposed to few false interactions with high stability scores. As a result, true positives can be easily distinguished through stability score ranking (Figure 3.10B).

These findings support the idea that iterative re-weighting allows iRF to recover high-order interactions without limiting predictive performance. In particular, improved interaction recovery with iteration indicates that iterative re-weighting stabilizes decision paths, leading to more interpretable models. We note that a principled approach for selecting the total number of iterations $K$ can be formulated in terms of estimation stability with cross validation (ESCV) [89], which would balance trade-offs between interpretability and predictive accuracy.

**Simulation 2: marginal importance**

Sec. 3.E demonstrates that iterative re-weighting improves the recovery of high-order interactions. The following simulations develop an intuition for how iRF constructs high-order interactions, and under what conditions the algorithm fails. In particular, the simulations demonstrate that iterative re-weighting allows iRF to select marginally important active features earlier on decision paths. This leads to more favorable partitions of the feature space, where active features that are marginally less important are more likely to be selected.

We sampled features $\mathbf{x} = (x_1, \ldots, x_{100})$ from independent, standard Cauchy distributions, and generated the binary response $y$ as

$$y = 1 \left[ \sum_{i \in S_{\text{XOR}}} 1(x_i > t_{\text{XOR}}) \equiv 1 \pmod{2} \right], \quad (3.7)$$

$S_{\text{XOR}} = \{1, \ldots, 8\}$. We set $t_{\text{XOR}} = 2$, which resulted in a mix of marginally important and unimportant active features, allowing us to study how iRF constructs interactions. For all simulations described in this section, we generated $n = 5000$ training observations and evaluated the fitted model on a test set of 500 held out observations. RF parameters were set to their default values with the exception of $\text{ntree}$, which was set to 200 for computational purposes. We ran iRF for $k \in \{1, \ldots, 5\}$ iterations with 10 bootstrap samples and grew $M = 100$ RITs of depth 5 with $n_{\text{child}} = 2$. Each simulation was replicated 10 times to evaluate performance stability.
Noise level

In the first simulation, we considered the effect of noise on interaction recovery to assess the underlying difficulty of the problem. We generated responses using equation (3.7), and swapped labels for 10%, 15%, and 20% of randomly selected responses.

Figure 3.11 shows performance in terms of predictive accuracy and interaction recovery for the 15% and 20% noise levels. At each noise level, increasing $k$ leads to superior performance, though there is a substantial drop in both absolute performance and the rate of improvement over iteration for increased noise levels.

The dramatic improvement in interaction recovery (Figure 3.11C) reinforces the idea that regularization is critical for recovering high-order interactions. Figure 3.12 shows the distribution of iRF weights, which reflect the degree of regularization, by iteration. iRF successfully recovers the full XOR interaction in settings where there is clear separation between the distribution of active and inactive variable weights. This separation develops over several iterations, and at a noticeably slower rate for higher noise levels, indicating that further iteration may be necessary in low signal-noise regimes.

Marginal importance and variable selection: iRF’s improvement with iteration suggests that the algorithm leverages informative lower-order interactions to construct the full data generating rule through adaptive regularization. That is, by re-weighting towards some active features, iRF are more likely to produce partitions of the feature space where remaining active variables are selected. To investigate this idea further, we examined the relationship between marginal importance and the average depth at which features are first selected across the forest. We define a variable’s marginal importance as the best case decrease in Gini impurity if it were selected as the first splitting feature. We note that this definition is different from the standard measure of RF importance (mean decrease in Gini impurity), which captures an aggregate measurement of marginal and conditional importance over an RF. We considered this particular definition to examine whether iterative re-weighting leads to more “favorable” partitions of the feature space, where marginally unimportant features are selected earlier on decision paths.

Figure 3.13 shows the relationship between marginal importance and feature entry depth. On average over the tree ensemble, active features enter the model earlier with further iteration, particularly in settings where iRF successfully recovers the full XOR interaction. We note that this occurs for active features with both high and low marginal importance, though more marginally important, active features enter the model earliest. This behavior supports the idea that iRF constructs high-order interactions by identifying a core set of active features, and using these, partitions the feature space in a way that marginally less important variables become conditionally important, and thus more likely to be selected.

Mixture model

Our finding that iRF uses iterative re-weighting to build up interactions around marginally important features, suggests that the algorithm may struggle to recover interactions in the
presence of other marginally important features. To test this idea, we considered a mixture model of XOR and AND rules. A proportion \( \pi \in \{0.5, 0.75, 0.9\} \) of randomly selected observations were generated using equation (3.7), and the remaining proportion \( 1 - \pi \) of observations were generated as

\[
y = \prod_{i \in S_{\text{AND}}} 1 [x_i > t_{\text{AND}}]. \tag{3.8}
\]

We introduced noise by swapping labels for 10% of the responses selected at random, a setting where iRF easily recovers the full XOR rule, and set \( S_{\text{AND}} = \{9, 10, 11, 12\} \), \( t_{\text{AND}} = -0.5 \) to ensure that the XOR and AND interactions were dominant with respect to marginal importance for \( \pi = 0.9 \) and \( \pi = 0.5 \) respectively.

Figure 3.14 shows performance in terms of predictive accuracy (A) and interaction recovery of XOR (B) and AND (C) rules at each level of \( \pi \). When one rule is clearly dominant (AND: \( \pi = 0.5 \); XOR: \( \pi = 0.9 \)), iRF fail to recover the other (Figure 3.14 B,C). This is driven by the fact that the algorithm iteratively updates feature weights using a global measure of importance, without distinguishing between features that are more important for certain observations and/or in specific regions of the feature space. One could address this with local measures of feature importance, though we do not explore the idea in this paper.

In the \( \pi = 0.75 \) setting, none of the interactions are clearly more important, and iRF recovers subsets of both the XOR and AND interactions (Figure 3.14). While iRF may recover a larger proportion of each rule with further iteration, we note that the algorithm does not explicitly distinguish between rule types, and would do so only when different decision paths in an RF learn distinct rules. Characterizing the specific form of interactions recovered by iRF is an interesting question that we are exploring in our ongoing work.

**Correlated features**

In our next set of simulations, we examined the effect of correlated features on interaction recovery. Responses were generated using equation (3.7), with features \( x = (x_1, \ldots, x_{100}) \) drawn from a Cauchy distribution with mean 0 and covariance \( \Sigma \), and active set \( S_{\text{XOR}} \), \( \mid S_{\text{XOR}} \mid = 8 \) sampled uniformly at random from \( \{1, \ldots, 100\} \). We considered both a decaying covariance structure: \( \Sigma_{ij} = \rho^{|i-j|} \), and a block covariance structure:

\[
\Sigma_{ij} = \begin{cases} 
1, & i = j \\
\rho, & i, j \in G_l \text{ and } i \neq j \\
0, & \text{else}
\end{cases}
\]

where \( G_l \subseteq \{1, \ldots, p\} \) and \( l = 1, \ldots, L \) partition \( \{1, \ldots, p\} \) into blocks of features. For the following simulations, we considered both low and high levels of feature correlation \( \rho \in \{0.25, 0.75\} \) and blocks of 10 features.

Prediction accuracy and interaction recovery are fairly consistent for moderate values of \( \rho \) (Figure 3.15, Figure 3.16), while interaction recovery degrades for larger values of \( \rho \), particu-
larly in the block covariance setting (Figure 3.16B,C). For instance when $\rho = 0.75$, iRF only recovers the full order-8 interaction at $k = 5$, and simultaneously recovers many more false positive interactions. The drop in interaction recovery rate is greater for larger interactions due to the fact that for increasing $\rho$, inactive features are more frequently selected in place of active features. These findings suggest both that iRF can recover meaningful interactions in highly correlated data, but that these interactions may also contain an increasing proportion of false positive features.

We note that the problem of distinguishing between many highly correlated features, as in the $\rho = 0.75$ block covariance setting, is difficult for any feature selection method. With a priori knowledge about the relationship between variables, such as whether variables represent replicate assays or components of the same pathway, one could group features as described in Sec. 3.B.

Simulation 3: big $p$

Our final set of synthetic data simulations tested the performance of iRF in settings where the number of features is large relative to the number of observations. Specifically, we drew 500 independent, $p$-dimensional standard Cauchy features, with $p \in \{1000, 2500\}$. Responses were generated using the order-4 AND interaction from equation (3.5), selected to reflect the form of interactions recovered in the splicing and enhancer case studies. We injected noise into the responses by swapping labels for 20% and 30% of randomly selected observations.

Figure 3.17 and Figure 3.18 show prediction accuracy and interaction recovery of iRF at each of the different noise levels. Prediction accuracy improves noticeably with iteration and stabilizes at the 20% noise level (Figure 3.17A, Figure 3.18A). For $k = 1$, iRF rarely recovers correct interactions and never recovers interactions of order > 2, while later iterations recover many true interactions (Figure 3.17C, Figure 3.18C). These findings indicate that iterative re-weighting is particularly important in this highly sparse setting and is effectively regularizing RF fitting. Based on the results from our previous simulations, we note that the effectiveness of iterative re-weighting will be related to the form of interactions. In particular, iRF should perform worse in settings where $p >> n$ and interactions have no marginally important features.

Simulation 4: enhancer data

To test iRF's ability to recover interactions in real data, we incorporated biologically inspired Boolean rules into the *Drosophila* enhancer dataset analyzed in Sec. 3.3 (see also Sec. 3.C for a description of the dataset). These simulations were motivated by our desire to assess iRF's ability to recover signals embedded in a noisy, non-smooth and realistic response surface with feature correlation and class imbalance comparable to our case studies. Specifically, we used all TF binding features from the enhancer data and embedded a 5-dimensional AND rule between Krüppel, (Kr), Hunchback (Hb), Dichaete (D), Twist (Twi), and Zelda (Zld):
\[ y = \mathbb{1}[x_{kr} > 1.25 & x_{hb} > 1.25 & x_D > 1.25 & x_{twi} > 1.25 & x_{zld} > 75]. \] (3.9)

The active TFs and thresholds were selected to ensure that the proportion of positive responses was comparable to the true data (∼10% active elements), and the interaction type was selected to match the form of interactions recovered in both the enhancer and splicing data.

In this set of simulations, we considered two types of noise. For the first, we incorporated noise by swapping labels for a randomly selected subset of 20% of active elements and an equivalent number of inactive elements. We note that this resulted in a fairly limited proportion of swapped labels among class 0 observations due to class imbalance. Our second noise setting was based on an RF/sample splitting procedure. Specifically, we divided the data into two disjoint groups of equal size. For each group, we trained an RF and used it to predict the responses of observations in the held out group. This process resulted in predicted class probabilities for each observation \( i = 1, \ldots, n \). We repeated this procedure 20 times to obtain the average predicted probability that \( y_i = 1 \). With a slight abuse of notation, we denote this predicted probability as \( \pi_i \). For each observation we sampled a Bernoulli noising variable \( \tilde{y}_i \sim \text{Bernoulli}(\pi_i) \) and used these to generate a binary response for each observation

\[ y_i = \tilde{y}_i \mathbb{1}[x_{kr} > 1.25 & x_{hb} > 1.25 & x_D > 1.25 & x_{twi} > 1.25 & x_{zld} > 75]. \]

That is, the response for observation \( i \) was to set 1 whenever the noising variable \( \tilde{y}_i \) or equation (3.9) was active. This noising procedure introduced an additional ∼5% of class 1 observations beyond the ∼10% of observations that were class 1 as a result of equation (3.9). Intuitively, this model derives its noise from rules learned by an RF. Feature interactions that are useful for classifying observations in the split data are built into the predicted class probabilities \( \pi_i \). This results in an underlying noise model that is heterogeneous, composed of many “bumps” throughout the feature space.

In each setting, we trained on samples of 200, 400, \ldots, 2000 observations and tested prediction performance on the same number of observations used to train. We repeated this process 20 times to assess variability in interaction recovery and prediction accuracy. The RF tuning parameters were set to default levels for the \texttt{R randomForest} package, \( M = 100 \) random intersection trees of depth 5 were grown with \( n_{\text{child}} = 2 \), and \( B = 20 \) bootstrap replicates were taken to determine the stability scores of recovered interactions.

Figure 3.19A shows that different iterations of iRF achieve comparable predictive accuracy in both noise settings. When the number of training observations increases beyond 400, the overall quality of recovered interactions as measured by interaction AUC improves for iterations \( k > 1 \). In some instances, there is a drop in the quality of recovered interactions for the largest values of \( k \) after the initial jump at \( k = 2 \) (Figure 3.19). All iterations frequently recover true order-2 interactions, though the weighted false positive rate for order-2 interactions drops for iterations \( k > 1 \), suggesting that iterative re-weighting helps iRF filter out false positives. Iterations \( k > 1 \) of iRF recover true high-order interactions at much greater
frequency for a fixed sample size, although these iterations also recover many false high-order interactions (Figure 3.19C,D). We note that true positive interactions are consistently identified as more stable (Figure 3.20), suggesting that the large proportion of weighted false discoveries in Figure 3.19D is the result of many false positives with low stability scores.
3.F Computational cost of detecting high-order interaction

We used the enhancer data from our case studies to demonstrate the computational advantage of iRF for detecting high-order interactions in high-dimensional data. Rulefit3 serves as a benchmark, which has competitive prediction accuracy to RF and also comes with a flexible framework for detecting nonlinear interactions hierarchically, using the so-called “H-statistic” [48]. For moderate to large dimensional datasets typically encountered in omics studies, the computational complexity of seeking high-order interactions hierarchically (select marginally important features first, then look for pairwise interaction among them, and so on) increases rapidly, while the computation time of iRF grows far more slowly with dimension.

We fit iRF and Rulefit3 on balanced training samples from the enhancer dataset (7809 samples, 80 features) using subsets of $p$ randomly selected features, where $p \in \{10, 20, \ldots, 80\}$. We ran Rulefit3 with default parameters, generating null interaction models with 10 bootstrap samples and looked for higher order interactions among features whose H-statistics are at least one null standard deviation above their null average (following [48]). The current implementation of Rulefit3 only allows H-statistic calculation for interactions of up to order 3, so we do not assess higher order interactions. We ran iRF with $B = 10$ bootstrap samples, $K = 3$ iterations, and the default RF and RIT tuning parameters. The run time (in minutes) and the AUC for different values of $p$, averaged over 10 replications of the experiment by randomly permuting the original features in enhancer data, are reported in Figure 3.21.

The plot on the left panel shows that the runtime for Rulefit3’s interaction detection increases exponentially as $p$ increases, while the increase is linear for iRF. The search space of Rulefit3 is restricted to all possible interactions of order 3, while iRF searches for arbitrarily high-order interactions, leveraging deep decision trees in RF. The linear vs. polynomial growth of computing time is not an optimization issue, it is merely a consequence of the exponentially growing search space of high-order interactions.

In addition to the comparison with Rulefit3, we profiled memory usage of the iRF R package using the splicing dataset described in Section 5 ($n = 11911$, $p = 270$) with $B = 30$ and $K = 3$. The program was run on a server using 24 cores (CPU Model: Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz, clock speed: 1200 MHz, Operating System: Ubuntu 14.04). The profiling was done using R functions `Rprof` and `summaryRprof`. iRF completed in 26 minutes 59 seconds, with a 499910 Mb memory consumption.
Figure 3.4: Enhancer data cross-validation AUC-PR change from baseline as a function of RF tuning parameters, evaluated over 5 folds. Baseline performance is given by Random Forest \( (k = 1) \) with default parameters \( (\texttt{ntree}= 500, \texttt{mtry}= 8) \). Error bars indicate the minimum and maximum change in AUC-PR across folds. [A] Prediction accuracy as a function of number of trees \( (\texttt{ntree}) \), with number of splitting variables \( (\texttt{mtry}) \) set to default \( (\lceil \sqrt{p} \rceil = 8) \). [B] Prediction accuracy as a function of \( \texttt{mtry} \), with \( \texttt{ntree} \) set to default \( (500) \).
Figure 3.5: Splicing data cross-validation AUC-PR change from baseline as a function of
RF tuning parameters, evaluated over 5-folds. Baseline performance is given by Random
Forest \((k = 1)\) with default parameters \((\text{n}\text{tre}\text{e}= 500, \text{m}\text{try}= 16)\). Error bars indicate the
minimum and maximum change in AUC-PR across folds. For iterations \(k > 1\), performance
is robust to choice of tuning parameters. [A] Prediction accuracy as a function of number
of trees \((\text{n}\text{tre}\text{e})\), with the number of splitting variables \((\text{m}\text{try})\) set to default \((\lfloor \sqrt{p} \rfloor = 16)\).
[B] Prediction accuracy as a function of \text{mtry}, with \text{n}\text{tre}\text{e} set to default \((500)\).
Figure 3.6: Enhancer data interaction stability scores as a function of RIT parameters. Each point represents a single interaction, and the point’s coordinates indicate its stability score under two parameter settings. Lower panels give Pearson correlation between interaction stability scores across pairs of parameter settings. [A] Interaction stability scores as a function of the number of trees in RIT. Number of children and depth are set to default levels of 2 and 5 respectively. [B] Interaction stability scores as a function of number of children in RIT. Number of trees and depth are set to default levels of 500 and 5 respectively. [C] Interaction stability scores as a function of depth in RIT. Number of trees and number of children are set to default levels of 500 and 2 respectively.
Figure 3.7: Splicing data interaction stability scores as a function of RIT parameters. Each point represents a single interaction, and the point’s coordinates indicate its stability score under two parameter settings. Lower panels give Pearson correlation between interaction stability scores across pairs of parameter settings. [A] Interaction stability scores as a function of the number of trees in RIT. Number of children and depth are set to default levels of 2 and 5 respectively. [B] Interaction stability scores as a function of number of children in RIT. Number of trees and depth are set to default levels of 500 and 5 respectively. [C] Interaction stability scores as a function of depth in RIT. Number of trees and number of children are set to default levels of 500 and 2 respectively.
Figure 3.8: Prediction-based validation metrics for enhancer data. Each plot shows the top 20 interactions with respect to prediction based importance metrics. Lower-order interactions that are a strict subset of some higher-order interactions have been removed for clearer visualization. The interactions reported here are qualitatively similar to those with high stability scores. [A] Conditional prediction. [B] Permutation importance.
Figure 3.9: Prediction-based validation metrics for splicing data. Each plot shows the top 20 interactions with respect to prediction based importance metrics. Lower-order interactions that are a strict subset of recovered higher-order interactions have been removed for clearer visualization. [A] Conditional prediction. [B] Permutation importance. The interactions reported here are qualitatively similar to those with high stability scores.
Figure 3.10: iRF performance for order-4 AND, OR, and XOR rules over 20 replicates. Results are shown for models trained using 100, 300, and 500 observations in the AND and OR models. Training sample size is increased to 200, 600, and 1000 in the XOR model to account for the low marginal importance of features under this rule. [A] Prediction accuracy (AUC-PR) improves with increased number of training observations and is comparable or improves for increasing $k$. [B] Interaction AUC improves with increasing $k$. For larger values of $k$, iRF always recovers the full data generating rule as the most stable interaction (AUC of 1) with enough training observations. [C] Recovery rate for interactions of all orders improves with increasing $k$. In particular, $k = 1$ fails to recover any order-4 interactions. [D] Weighted false positives increases in settings where iRF recovers high-order interactions as a result of many false positives with low stability scores.
Figure 3.11: iRF performance for order-8 XOR rule over 10 replicates as a function of noise level. All models were trained using 5,000 observations. [A] Prediction accuracy (AUC-PR) improves for increasing $k$ and at a slower rate for increased noise levels. [B] Interaction AUC improves with increasing $k$. [C] Recovery rate for interactions of all orders improves with increasing $k$. In particular, $k = 1$ does not recover any interactions of order > 2 at either noise level. Recovery of higher order interactions drops substantially at higher noise levels. [D] Weighted false positives increases in settings where iRF recovers high-order interactions as a result of many false positives with low stability scores. For order-2 interactions, later iterations of iRF filter out many of the false positives identified in earlier iterations.
CHAPTER 3. ITERATIVE RANDOM FORESTS (IRF)

Figure 3.12: iRF weights for active (blue) and inactive (red) features as a function of iteration and noise level over 10 replicates. The distribution of weights in later iterations shows a clear separation between active and inactive features, indicating that iRF has identified active features as important and incorporates them into the model with higher probability in later iterations.
Figure 3.13: Average entry depth for active (blue) and inactive (red) features across the forest as a function of marginal importance, iteration, and noise level. Results are reported for a single replicate. In later iterations, the average depth at which active variables are selected is noticeably lower than inactive variables with comparable marginal importance, indicating that the active features appear earlier on decision paths.
A

Figure 3.14: IRF performance for mixture model as a function of mixture proportion ($\pi$) over 10 replicates. All models were trained using 5,000 observations. [A] Prediction accuracy (AUC-PR) is generally poor since IRF tends to learn rules that characterize only a subset of the data. [B] Interaction AUC for the XOR rule. IRF fails to recover this marginally less important rule unless it is represented in a large proportion of the data ($\pi = 0.9$). [C] Interaction AUC for the AND rule. IRF recovers the full rule as the most stable interaction for $k \geq 3$ (AUC of 1) for $\pi = 0.5$ despite the fact that the AND interaction is only active in half of the observations. Perfect recovery of the AND rule in a setting where IRF fails to recover the XOR rule indicates that iterative re-weighting based on Gini importance encourages IRF identify rules with more marginally important features.
Figure 3.15: iRF performance for order-8 XOR rule over 10 replicates as a function of correlation level (decaying covariance structure). All models were trained using 5,000 observations.

[A] Prediction accuracy (AUC-PR) improves for increasing $k$. [B] Interaction AUC improves with increasing $k$, but is more variable than uncorrelated settings. [C] Recovery rate for interactions of all orders improves with increasing $k$. In particular, iRF with $k = 1$ rarely recovers any interactions of order $> 2$. [D] Weighted false positives increases in settings where iRF recovers high-order interactions as a result of many false positives with low stability scores. For order-2 interactions, later iterations of iRF filter out many of the false positives identified in earlier iterations.
Figure 3.16: iRF performance for order-8 XOR rule over 10 replicates as a function of correlation level (block covariance). All models were trained on using 5,000 observations. [A] Prediction accuracy (AUC-PR) improves for increasing $k$. [B] Interaction AUC improves with increasing $k$ and drops for large values of $\rho$. Variability is comparable to the decaying covariance case and greater than in uncorrelated settings. [C] Recovery rate for interactions of all orders improves with increasing $k$. In particular, iRF with $k = 1$ rarely recovers any interactions of order > 2. [D] Weighted false positives increase in settings where iRF recovers high-order interactions as a result of many false positives with low stability scores. For order-2 interactions, later iterations of iRF filter out many of the false positives identified in earlier iterations.
Figure 3.17: iRF performance for order-4 AND rule over 10 replicates with class labels swapped for 20% of observations selected at random. All models were trained using 500 observations. [A] Prediction accuracy (AUC-PR) improves and stabilizes with increasing $k$. [B] Interaction AUC improves dramatically with increasing $k$. For $k > 3$, iRF often recovers the full order-4 AND rule as the most stable interaction (AUC of 1). [C] Recovery rate improves with increasing $k$. For $k = 1$, iRF rarely recovers any portion of the data generating rule while for $k > 3$ iRF often recovers the full data generating rule. [D] Weighted false positives are low for interactions of order $> 2$ and drop with iteration for interactions of order-2, suggesting that iRF identifies active features through iterative re-weighting.
Figure 3.18: iRF performance for order-4 AND rule over 10 replicates with class labels swapped for 30% of observations selected at random. All models were trained using 500 observations. [A] Prediction accuracy (AUC-PR) gradually improves with increasing $k$. [B] Interaction AUC gradually improves with increasing $k$ but does not achieve perfect recovery of the data generating rule. [C] Recovery rate improves with increasing $k$, but iRF recovers higher-order interactions less frequently than at lower noise levels. [D] Weighted false positives are comparable across $k$ and particularly high for order-2 interactions.
Figure 3.19: iRF performance for the enhancer data simulations by noise type. Results are shown for models trained using 400, 1200, and 2000 observations. [A] Prediction accuracy (AUC-PR) remains consistent with increasing \( k \) in both noise models. [B] Interaction AUC improves after iteration \( k = 1 \), especially for larger training samples where high-order interactions are recovered. Some settings show a drop in interaction AUC as \( k \) increases from 2 to 5, emphasizing the importance of tuning \( K \). [C] Recovery rate improves beyond \( k = 1 \) for high-order interactions and is fairly consistent for \( k = 2, \ldots, 5 \). [D] Weighted false positives drop beyond \( k = 1 \) for order−2 interactions as iterative re-weighting encourages the selection of active features. With larger training samples, iRF recovers many interactions among both active and inactive features. The stability scores of interactions among active features are consistently higher than interactions including inactive features.
Figure 3.20: Distributions of iRF stability scores for active and inactive variables by iteration ($k$) and noise type. Both models were trained using 2000 observations. Interactions among active features are consistently identified as more stable in both noise settings, and higher order interactions are only identified in later iterations.
Figure 3.21: Runtime (left) of interaction detection and Area under ROC curve (right) of prediction by Rulefit and iRF on subsets of the enhancer data with \( p \in \{10, 20, \ldots, 80\} \) features and balanced training and test sets, each of size \( n = 731 \). The results are averaged over 10 different permutations of the original features in the enhancer dataset. The two algorithms provide similar classification accuracy in test data, although computational cost of iRF grows much slower with \( p \) than that of Rulefit.
Chapter 4

Signed Iterative Random Forests (siRF)

4.1 Introduction

Biological processes are regulated by dynamic, cell-type specific interactions among individual molecular elements and their environmental contexts. Identifying these relationships in high-dimensional, heterogeneous data represents a considerable challenge for humans due to the enormous search spaces of high-order interactions. On the other hand, state-of-the-art supervised learning algorithms are well suited to discovering complex, non-linear prediction rules in massive datasets. These algorithms accurately predict a wide range of biological phenomena in both humans and model organisms, which has driven insights into the architecture of functional regulation [38, 23, 74].

In addition to accurately predicting new observations, supervised learners hold the promise of guiding inquiry into the mechanisms that govern complex processes. The relationships they learn provide models of how a system may respond to intervention, and the corresponding interactions provide candidate hypotheses of biological interactions. From this perspective, translating high-order, non-linear interactions learned by predictive algorithms into interpretable\(^1\) forms offers an invaluable opportunity to test whether learned interactions represent biological mechanisms or statistical associations.

Here we propose an approach to extract simple, rule-based interactions from an ensemble of decision trees by building on the iterative random forest algorithm (iRF) [9]. The interactions we identify correspond to collections of rules of the form

\[
\text{if: } \text{condition}_1 & \ ... & \text{condition}_k, \text{ then response.}
\]

As an example, conditions for continuous features \(x = (x_1, \ldots, x_p) \in \mathbb{R}^p\) could take the form \(x_j < t_j\), making rule-based interactions an attractive representation of the genomic processes activated by thresholded levels of regulatory factors [139].

\(^1\)For a definition and discussion of interpretable machine learning, see our paper [96]
In contrast to iRF, which searches for frequently co-occurring features along the decision paths of a random forest (RF) [18], the interactions we identify represent “subsets” of rules that exhibit similar thresholding behavior. This additional information describes both a functional relationship between interacting features/responses and the observations for which an interaction is active. To illustrate the benefit of this information, consider the following examples of a genomic process that depends on two regulatory inputs:

1. The process is activated by either low concentrations of both factors or by high concentrations of both factors (Figure 4.1 left).

2. The process is activated by high concentrations of both regulatory factors (Figure 4.1 middle).

3. The genomic process is activated by high levels of either factor (Figure 4.1 right).

All of these examples represent an order-2 interaction among the same regulatory inputs, but each defines a unique biological mechanism. Distinguishing between these settings is necessary to determine the experimental interventions that can reveal the underlying mechanism.

Our proposed approach, signed iterative random forests (siRF), scalably identifies high-order interactions, distinguishes their functional form (e.g. Figure 4.1), and implicitly localizes them to specific observations. In addition to providing more information about a fitted RF, siRF recovered interactions with higher accuracy than iRF across a wide range of simulations. We used siRF to investigate regulatory interactions in the early Drosophila embryo through multiple genomic datasets. When trained to predict spatial gene expression patterns, siRF recovered all previously reported gap gene interactions and identified novel,
high-order interactions among anterior-posterior (A-P) factors. Many of the interactions we recovered precisely define positional information, pointing to a high level of redundancy in A-P patterning. When trained to predict genomic sequences that drive patterned gene expression, siRF identified an order-3 interaction among homeothorax (Hth), knirps (Kni), and Krüppel (Kr), transcription factors (TFs) that are known to play an important role in embryonic development, that classifies a large portion of enhancer elements with near perfect accuracy.

The rest of the chapter is organized as follows. In Sec. 4.2, we review previous work on rule-based extraction from supervised learners. In Sec. 4.3, we motivate the idea of a signed interactions and present our three contributions:

1. We refine the interactions recovered by iRF through the notion of signed interactions, leading to signed iterative Random Forests (siRF). Signed interactions define a rule-based, functional relationship between interacting features and responses, and are thus more amenable to experimental follow-up. Moreover, the functional form of decision rules implicitly localizes signed interactions, describing the observations for which it is active.

2. We propose stable and predictive importance metrics (SPIMs) for signed interactions. These metrics expand upon the stability scores used by iRF, providing a more comprehensive assessment of the predictive accuracy, stability, and strength of an interaction.

3. We generalize the stability analysis framework of iRF to evaluate the uncertainty of our proposed metrics. We use null importance metrics that describe the expected behavior of our SPIMs when the data respect known structure to filter out observed results that can be reproduced by simple phenomena.

In Sec. 4.4, we compare interactions recovered by iRF and siRF in simulations based on real and synthetic data. In Sec. 4.5, we study signed interactions associated with enhancer activity and spatial gene expression patterns in the early *Drosophila* embryo. We conclude by discussing areas for further work in interpreting complex supervised learning algorithms.
4.2 Background

Current methods for extracting decision rules from data are largely based on decision trees, which partition observations into groups described by unique rules. Since rules learned by a single decision tree can be unstable with respect to small data perturbations [83], it is common to search for more reliable candidate interactions across ensembles of decision trees. One approach for identifying rules is to approximate an ensemble of decision trees through a single, stable tree [148, 52]. However, these methods rely on pseudo covariates, which can be difficult to generate in practice when the underlying data distribution is unknown. A second line of work extracts rules directly from an ensemble of decision trees: forest garrote [93], node harvest [94], rule fit [48], inTrees [28], and iRF [9]. With the exception of iRF, these methods rely on shallow trees to prevent overfitting and/or build rules through incremental search procedures that scale as $p^s$, where $p$ is the number of features and $s$ is the order of an interaction. iRF scales independently of interaction order, enabling it to identify high-order interactions that arise in genomic applications. However, the interactions iRF recovers represent co-occurring features, as opposed to a functional relationship between these features and responses.

Our proposed method, siRF, scalably extracts and evaluates rule-based interactions from the decision paths of an RF [18] through the iRF [9] framework. This allows us to identify high-order, rule-based interactions without forward-wise search procedures that scale exponentially in the size of interactions. The next section outlines the iRF algorithm. For a more detailed description, we refer readers to the original paper [9].

iterative Random Forests (iRF)

The iterative Random Forest algorithm provides a computationally efficient procedure to search for collections of interacting features that frequently co-occur along RF decision paths. This allows us to identify rules across an RF that share a common set of active features. We briefly review iRF here. For a full description of the algorithm, we refer readers to the original paper [9].

iRF trains a series of $K$ iteratively re-weighted RFs, $\{\text{RF}(w^{(k)})\}_{k=1}^K$, where features are sampled at each node with probability proportional to $w_i^{(k)} \in \mathbb{R}_+^p$, the Gini importance obtained from the previous iteration. Leaf node decision paths and predictions from the final iteration are encoded as pairs $(\mathcal{I}_i, Z_i)$, $\mathcal{I}_i \subseteq \{1, \ldots, p\}$, $Z_i \in \{0, 1\}$. $\mathcal{I}_i$ denotes features selected along the decision path of the leaf node containing observation $i = 1, \ldots, n$ in tree $t = 1, \ldots, T$ and $Z_i$ the corresponding prediction. The pairs $(\mathcal{I}_i, Z_i)$ are used as input to RIT [116], which searches for frequently co-occurring features among a particular class of observations. The process of running RIT on $(\mathcal{I}_i, Z_i)$ produces a collection of feature interactions that frequently appear on the decision paths of $\text{RF}(w^{(K)})$ and is described as generalized RIT (gRIT). Finally, iRF assess the stability of recovered interactions by repeating the gRIT step across an outer layer of bootstrap samples of the original data and evaluating the proportion of times an interaction is recovered.
4.3 Signed iterative random forests: an enhanced interpretation of iRF

We motivate siRF with a simple example. Consider a collection of binary genomic responses $y_i \in \{0, 1\}, i = 1, \ldots, n$. These responses may represent whether a segment of DNA acts as an enhancer in a particular context (i.e. 0: inactive enhancer, 1: active enhancer). For each response $i$, suppose that we measure a set of $p$ features. These features could describe the state, count, or concentration of biomolecules associated with a segment of DNA. If we can summarize the level of each biomolecule as depleted, normal, or enriched, then observations may be written as $\tilde{x}_i = (\tilde{x}_{i1}, \ldots, \tilde{x}_{ip}) \in \{-1, 0, 1\}^p$. This simplified framing is similar to [30], who binarize omics profiles based on divergence from baseline. Of course, determining whether a biomolecule is enriched or depleted is challenging in practice. We will return to this issue shortly.

Our goal is to identify states of the features, or interactions, for which there are a high proportion of active responses. More formally, we seek states $S = (S_1, \ldots, S_p) \in \{-1, 0, 1\}^p$ such that

$$P_n(y = 1|\tilde{x} = S) \geq \theta_1$$

where $P_n$ denotes an empirical distribution and $0 \leq \theta_1 \leq 1$. In our example, equation (4.1) describes the enrichment of enhancer activity given the presence/absence of specific biomolecules. The enriched biomolecules (i.e. $S_j = 1$) represent an activating association with responses while the depleted biomolecules (i.e. $S_j = -1$) represent a repressive association. Collectively, the non-zero elements of $S$ represent an order-$s$ interaction, $s = |\{j : S_j \neq 0\}|$, and a candidate biological mechanism.

One way to satisfy equation (4.1) is to select highly specific interactions that describe a very small number of observations. That is, where $|\{i : \tilde{x}_i = S\}|$ is small and $P_n(y = 1|\tilde{x} = S)$ is large. To identify interactions that generalize beyond highly specific contexts, we also seek interactions such that

$$P_n(\tilde{x} = S|y = 1) \geq \theta_2$$

where $0 \leq \theta_2 \leq 1$. In our example, equation (4.2) describes the prevalence of an interaction $S$ among all active enhancers. Of course, identifying local interactions that describe small subpopulations may be of interest for some applications. The appropriate trade off between equations (4.1) and (4.2) is thus domain dependent.

**Signed interactions**

In practice, we rarely know *a priori* which features are enriched, depleted, or within a normal range. To determine interactions that are associated with response activity, as in equations (4.1) and (4.2), we must map continuous valued observations $x \in \mathbb{R}^p \mapsto \{-1, 0, 1\}^p$. Here we propose a principled feature mapping through iteratively re-weighted RFs. A feature is deemed enriched (resp. depleted) in decision rules that rely on high (resp. low) levels for prediction.
The main idea of our approach is to extract rules from an RF by that frequently appear along decision paths. Since RFs are trained using random subsets of both observations and features, it is unlikely for identical rules to appear along multiple decision paths. Even decision paths that use the same set of active features will vary with respect to their thresholds. To address this issue, we introduce signed interactions. Intuitively, a signed interaction defines a decision path up to threshold. This allows us to compare paths that vary across an RF and identify portions of their predictive rules that are shared. From a different perspective, a signed interaction can be viewed as a coarse region of the feature space corresponding to a predictive rule with smooth decision boundaries (see Sec. 4.A).

Consider a decision tree node that splits on feature \( j \in \{1, \ldots, p\} \) with associated threshold \( t_j \in \mathbb{R} \). Observations \( x \) that arrive at this node will be sent to the left child when \( x_j < t_j \) and be sent to the right child otherwise. We use the signed feature-index \( \gamma \in \{-p, \ldots, p\} \) to describe the selected splitting feature and inequality direction associated with a child node. The left child is represented by the signed feature index \( \gamma^l = -j \) and the right child by \( \gamma^r = j \) (Figure 4.2). As a result, signed feature indices describe whether a node is associated with relatively high or low levels of a selected feature. We define the signed interaction associated with leaf node \( l = 1, \ldots, L \) as

\[
S_l = \{\gamma_1, \ldots, \gamma_k\} \subseteq \{-p, \ldots, p\},
\]

where \( \gamma_1, \ldots, \gamma_k \) are the signed feature indices corresponding to all non-root nodes along the decision path (Figure 4.3). For simplicity, we assume that each feature is selected at most once along a decision path. We discuss other cases in Sec. 4.C. We note that signed interactions are equivalent to the interactions described in Sec. 4.2. That is, we can write \( \tilde{x} \in \{-1, 0, 1\}^p \) as the signed interaction

\[
S = \{\text{sign}(j) \cdot j : \tilde{x}_j \neq 0\}
\]

Intuitively, signed interactions describe the rule associated with a decision path up to threshold.

For instance, rules of the form \( I(x_2 < \cdot) \cdot I(x_3 \geq \cdot) \) are represented by the signed interaction \( \{-2, 3\} \). A signed interactions allows us to describe \( 2^s \) rules (up to threshold) that could be defined over \( s \) interacting features. In the context of genomic data, the different rules may suggest candidates for activators or repressors of a particular cellular process (e.g. Figure 4.1). From a practical perspective, identifying how responses vary with different configurations of features can drastically reduce the number of experiments required to identify interesting biological mechanisms.

**Extracting signed interactions through iRF**

Signed interactions that frequently appear on RF decision paths correspond to decision rules that an RF consistently selects under data and model perturbations. We extract prevalent signed interactions from an RF using gRIT, iRF’s interaction search procedure. We generate
signed feature-index sets for every leaf node in an RF and represent each observation with $T$ pairs of signed feature-index sets and responses

$$(S_{it}, Z_{it}) \quad i = 1, \ldots, n, \quad t = 1, \ldots, T,$$

where $S_{it}$ represents the signed feature-index set for the leaf node containing observation $i$ in tree $t$ and $Z_{it}$ the corresponding prediction. These pairs serve as inputs to RIT [116]. We call the process of searching for signed interactions using gRIT signed iRF (siRF).

The ability to accurately describe relationships learned by a model has been advocated as an important consideration in interpretable machine learning, under the name descriptive accuracy [96]. Replacing feature-index sets with signed feature-index sets allows gRIT to target specific regions (one of $2^s$) associated with an order-$s$ interaction. This provides more accurate descriptions of how interactions influence RF predictions without without increasing the order of computational cost or prediction error. Moreover, we find that the signed representation improves the quality of recovered interactions in our simulation studies (Sec. 4.4).

**Stable and predictive importance metrics (SPIMs)**

siRF uses the decision rules learned by an RF to identify candidate interactions in data. In biological settings, our hope is that these interactions correspond to mechanistic rules. For instance, a signed interaction may describe how a collection of TFs regulates a group of enhancers. However, many factors involved in the data collection and modeling can make it difficult to differentiate between statistical associations and causal relationships.
Figure 4.3: siRF workflow: 1. Iteratively re-weighted RFs learn stable and predictive rules to differentiate active from inactive observations. 2. Decision paths are converted into signed interactions to compare rules selected across an RF. 3. gRIT searches for signed interactions that frequently appear on RF decision paths. Interactions returned by siRF are evaluated both visually using response surfaces and quantitatively using stable and predictive importance metrics (SPIMs).
To identify high-quality interaction candidates for follow up analysis, we propose importance measures based on stability and predictive accuracy, which are widely used to identify causal relationships (for a summary see [144]). In addition, we define null importance metrics that describe how our proposed importance measures behave relative to known, simple structure in the data. At a high level, the null metrics are inspired by [37], who consider the problem of generating controls for single neuron data.

**Prevalence to evaluate stability of signed interactions**

Stability of results relative to “reasonable” perturbations has been advocated as a minimum requirement to work towards reproducibility in science [143, 144]. Our proposed stability measure leverages bootstrap sampling and random feature selection to evaluate how consistently an interaction is selected across an ensemble of trees. More precisely, we define the *prevalence* of an interaction as

$$P(S | C) := \frac{1}{T} \cdot \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1(S \subseteq S_{it}) \cdot 1(Z_{it} = C)}{\sum_{i=1}^{n} 1(Z_{it} = C)}.$$  

(4.5)

Prevalence indicates the proportion of class-\(C\) observations where the interaction is active after mapping continuous features to signed interactions. In equation (4.5), perturbations enter in the form of the trees \(t = 1, \ldots, T\). Each tree defines a distinct mapping based on different bootstrap samples and randomly sampled subsets of features. Equation (4.5) is identical to the notion of prevalence described in [116], except that it is defined relative to an ensemble of decision trees. This allows us to use data and model perturbations to evaluate the stability of an interaction.

We note that equation (4.5) is closely related to the stability scores proposed by [9]. Specifically, stability scores represent the proportion of times an interaction is recovered by gRIT. The probability that an interaction is recovered by gRIT depends directly on its prevalence [116]. Thus prevalence can be viewed as a refined measure of stability. An added benefit of using prevalence to evaluate interactions is that unlike stability scores, it does not depend on RIT tuning parameters.

**Precision to evaluate the predictive accuracy of signed interactions**

Predictive accuracy quantifies how well a model represents future, unobserved data. It has been widely adopted by the machine learning community to assess the empirical evidence that supports a model. In the context of siRF, we would like to identify interactions that are strongly associated with responses (as in equation (4.1)). Toward this end, we define a measure of accuracy based on the *precision* of a signed interaction \(S\) for a class \(C\)

$$P(C | S) := \frac{1}{T} \cdot \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1(S \subseteq S_{it}) \cdot 1(y_{it} = C)}{\sum_{i=1}^{n} 1(S \subseteq S_{it})}.$$  

(4.6)
Precision measures the proportion of observations that belong to a particular class, averaged across all leaf nodes that contain a given interaction. It represents a simple measure of accuracy that can be directly evaluated from a fitted siRF. Of course, it might be desirable to evaluate an interaction based on other accuracy metrics. We describe an approach for extracting rule-based predictions associated with a signed interaction in Sec. 4.A. The predictions described in Sec. 4.A can be evaluated using any metric of interest.

As we have previously discussed, a signed interaction does not necessarily imply a mechanistic interpretation. For instance, a sample of genomic sequences may result in associations between TF binding and enhancer activity, regardless of whether the TFs play a functional role. TFs that bind many regions of the genome may co-occur at many enhancer elements, even if they do not interact. Prevalence and precision aim to evaluate interactions through broadly important notions of stability and predictive accuracy. Here we propose three null importance metrics based on the expected behavior of prevalence and precision relative to simple structure in the data. This allows us to screen out interactions that can be explained by known phenomena. We motivate each null metric in the context of enhancer prediction, where our goal is to identify TFs whose combined binding activity plays a regulatory role.

Null importance metric 1: class prevalence enrichment (CPE)

As with individual TFs [85], an interaction may be present among both active and inactive enhancers (Figure 4.4). Generally speaking, if an interaction regulates enhancer activity we expect that it should be more stable among active enhancers. To quantify this association we define the prevalence class enrichment (CPE) as

\[
CPE(S; C) := P(S|C) - P(S|1 - C). \tag{4.7}
\]

Equation (4.7) uses one class of observations as a control to determine whether a signed interaction is enriched relative to the control. In our simulations, signed interactions were recovered with high accuracy when ranked by \(CPE(S; 1)\) (Figure 4.7). In contrast, unsigned interactions included many false positives when ranked by \(CPE(S; 1)\). Intuitively, the unsigned representation “washes out” important interactions when there is a high proportion of class-1 observations defined by one signed interaction (e.g. \(\{2, 3\}\)) and a high proportion of class-0 observations defined by a different signed interaction among the same features (e.g. \(\{-2, 3\}\)).

Null importance metric 2: feature selection dependence (FSD)

Individual TFs that bind many regions of the genome will co-occur at some active enhancers, even if they do not physically interact (Figure 4.5). From a biological perspective, it is important to distinguish additive from non-additive effects, since the latter are known underlie many natural processes. From a modeling perspective, this requires us to distinguish between settings where TFs co-occur by chance and where TFs bind in a dependent manner.
CHAPTER 4. SIGNED ITERATIVE RANDOM FORESTS (SIRF)

Figure 4.4: TF interactions can appear among both active and inactive enhancers. $CPE(S; 1)$ evaluates whether an interaction is enriched among active enhancers (e.g. high levels of the blue and yellow TFs) or whether it appears frequently among both classes (e.g. high levels of the blue and orange TFs).

We quantify this dependence through feature selection dependence ($FSD$):

$$FSD(S; C) := P(S|C) - \prod_{j=1}^{|S|} P(S_j|C)$$

Equation (4.8) compares the prevalence of a signed interaction $S$ with its expected prevalence if features were selected independently of another. In our simulations, we find that $FSD$ helps differentiate between additive and non-additive components of a generative model (Figure 4.7).

Null importance metric 3: mean increase in precision ($MIP$) A TF that binds both active and inactive regions may appear in interactions as a result of sampling variability, noisy measurements, and/or deep decision trees used by siRF. If the TF does not serve a functional role, it represents a false positive that distorts our understanding of any underlying mechanism. To identify such TFs, we ask whether including it in an interaction results in more precise classification (Figure 4.6). We define the mean increase in precision ($MIP$) as

$$MIP(S; C) := \frac{1}{|S|} \sum_{S' \subset S : |S'| = |S| - 1} P(C|S) - P(C|S'),$$

Intuitively, equation (4.9) evaluates whether all features in a signed interaction are necessary.
Figure 4.5: TFs will co-occur at active enhancers whenever individual TFs bind enhancers with high probability. \( FSD(S; C) \) evaluates whether an interaction appears frequently relative to the prevalence of individual elements. For instance, the orange and blue TFs co-occur infrequently relative to the overall binding of orange and blue TFs [A]. In contrast, enhancers bound by the blue TF are almost always bound by the yellow TF [B].

to achieve the observed level of precision. This allows us to identify the minimum set of features required to produce a level of predictive accuracy.

**Generalized stability analysis to filter interactions**

The importance measures defined in Sec. 4.3 evaluate signed interactions recovered by a single siRF. To assess the uncertainty associated with these metrics, we evaluate them across siRFs trained on an “outer layer” of bootstrap samples. This extends the stability analysis of iRF, which evaluates interactions recovered by gRIT across an outer layer of bootstrap samples [9]. That is, the importance metric in iRF is simply an indicator function specifying whether an interaction is recovered for each bootstrap sample \( b = 1, \ldots, B \).

In our generalized setting, the null metrics in Sec. 4.3 describe attributes of an interaction relative to simple baselines. If any of \( CPE, FSD, MIP \leq 0 \), the simple baselines provide a reasonable description of the observed data. Placing these null metrics within the iRF stability analysis framework allows us to quantify our certainty that

1. \( S \) is enriched among class-1 leaf nodes than class-0 leaf nodes.
2. Selection of features in \( S \) is driven by dependence in their joint distribution.
3. Prediction accuracy depends on all features in \( S \).
Figure 4.6: Binding of a TF does not imply that it is useful for prediction. $MIP(C; S)$ evaluates how much an entire interaction improves precision relative to subsets. For example, binding by blue, yellow, and purple TFs does not increase the probability that an element is an enhancer relative to binding by any pair [A]. In contrast, binding by the blue, orange, and yellow TFs increases the probability an element is an enhancer to 1 from $7/12$ whenever any pair is bound [B].

Algorithm 3 outlines our procedure to search for and filter interactions at a pre-defined level $0 \leq \tau \leq 1$. In our case studies, we apply algorithm 3 with $\tau = 0.5$ to filter interactions recovered by siRF. This threshold improves the quality of recovered interactions in our simulation studies without being overly conservative. We are exploring the quality of screening under different generative models in our ongoing work.
Algorithm 3: siRF interaction search and null importance metric filtering

Input: \( \mathcal{D}, B, K, w^{(1)} \leftarrow (1/p, \ldots, 1/p), \tau \)

1. Fit iteratively re-weighted RFs on full data \( \mathcal{D} \) and extract interactions
2. for \( k \leftarrow 1 \) to \( K \) do
   3. Fit \( RF(w^{(k)}) \) on \( \mathcal{D} \)
   4. \( w^{(k+1)} \leftarrow \) Gini importance of \( RF(w^{(k)}) \)
3. end
4. Extract signed interactions from RF fitted in iteration \( K \)
5. \( S \leftarrow \text{gRIT}(RF(w^{(K)})) \)
6. Fit RFs on outer layer bootstrap samples and evaluate importance metrics for recovered interactions \( S \)
7. for \( b \leftarrow 1 \) to \( B \) do
   8. Generate bootstrap dataset: \( \mathcal{D}^{(b)} \leftarrow (x^{(b)}, y^{(b)}) \)
   9. Fit \( RF^{(b)}(w^{(K)}) \) on \( \mathcal{D}^{(b)} \)
   10. for \( S \in S \) do
      11. Evaluate \( CPE^{(b)}(S; 1) \) on \( RF^{(b)}(w^{(K)}) \)
      12. Evaluate \( FSD^{(b)}(S; 1) \) on \( RF^{(b)}(w^{(K)}) \)
      13. Evaluate \( MIP^{(b)}(S; 1) \) on \( RF^{(b)}(w^{(K)}) \)
      14. end
7. end
8. Filter interactions at specified level \( \tau \)
9. \( S \leftarrow S \setminus \{ S : \frac{1}{B} \sum_{b=1}^{B} \mathbbm{1}(CPE^{(b)}(S; 1) \leq 0) > \tau \} \)
10. \( S \leftarrow S \setminus \{ S : \frac{1}{B} \sum_{b=1}^{B} \mathbbm{1}(FSD^{(b)}(S; 1) \leq 0) > \tau \} \)
11. \( S \leftarrow S \setminus \{ S : \frac{1}{B} \sum_{b=1}^{B} \mathbbm{1}(MIP^{(b)}(S; 1) \leq 0) > \tau \} \)
12. Output: \( \{ S, CPE(S; 1), FSD(S; 1), MIP(S; 1) : S \in S \} \)
4.4 Simulations

To evaluate the signed interactions recovered by siRF, we developed a suite of simulation experiments based on Boolean-type rules intended to reflect stereospecific biological interactions. Using two datasets (1 synthetic, 1 real):

1. $n = 2500$ independent, standard Gaussian features $\mathbf{x} = (x_1, \ldots, x_{50})$
2. $n = 7808$ *Drosophila* ChIP-chip measurements for 23 TFs

We drew responses $y \sim Bernoulli(\pi)$ from three generative models:

\[
\pi^{(\text{AND})} = 0.8 \left( \prod_{j \in I_1} 1(x_j > t_{j,1}^+) \right) \tag{4.10}
\]

\[
\pi^{(\text{OR})} = 0.8 \left( \prod_{j \in I_1} 1(x_j > t_{j,1}^+) \prod_{j \in I_2} 1(x_j \leq t_{j,2}^-) + \prod_{j \in I_1} 1(x_j \leq t_{j,1}^-) \prod_{j \in I_2} 1(x_j > t_{j,2}^-) \right) \tag{4.11}
\]

\[
\pi^{(\text{ADD})} = 0.4 \left( \prod_{j \in I_1} 1(x_j > t_{j,1}^+) + \prod_{j \in I_2} 1(x_j > t_{j,2}^-) \right) \tag{4.12}
\]

Active features $I_k$ and thresholds $t_{j,k}^+, t_{j,k}^-$, $k = 1, 2$ were selected to ensure a class balance that was comparable to the *Drosophila* data described in section 3.3 (~ 10% active responses, see Table 4.1).

For each simulation experiment, we considered the collection of signed interactions represented by the data generating rule and evaluated interaction recovery relative to these “active interactions.” In particular, we labeled each interaction recovered by iRF, siRF, and siRF after filtering (algorithm 3) as “active” or “inactive” and calculated precision recall (PR) curves based on $PCE(S; 1)$ scores. Rankings based on the other metrics led to worse performance for all methods in every simulation setting.

We selected the number of iterations for iRF by maximizing prediction accuracy on out of bag samples (up to 10 iterations) and report interactions recovered across 25 bootstrap samples. We grew 500 RITs with depth and number of children set to the default values of 5 and 2 respectively. For every setting, the PR curves for signed interactions show noticeable improvement over those for unsigned interactions in terms of area under the PR curve (AUC-PR). When the generative model includes additive and non-additive components, the filtering procedure described in algorithm 3 further improves PR curvesi (Figure 4.7). We discuss the results of each simulation setting in greater detail below.

**Single component AND rule (AND)**

The AND generative model (equation (4.10)) corresponds to a single AND rule. This rule is active with high probability when all active features are present at high levels. Active signed
interactions correspond to a portion of the generative rule and active unsigned interactions to a subset of active features (Table 4.1).

Figure 4.7 (top) reports the PR curves for interaction recovery in both the *Drosophila* (left) and Gaussian (right) data, averaged over 50 simulation replicates. In the Gaussian setting all three approaches achieve near perfect interaction recovery (Figure 4.7 top right). That is, scores of active interactions almost always exceed those of inactive interactions. For the *Drosophila* data signed interactions are similarly accurate (AUC-PR 0.97 signed, 0.98 signed and filtered), while the quality of unsigned interactions drops slightly (AUC-PR 0.83). We note that an unsigned interaction $S$ represents a collection of $2^{|S|}$ signed interactions. In equation (4.10), only one of these signed interactions is active. However, unsigned interactions evaluate $PCE(S; 1)$ across the entire collection. The drop in AUC-PR reflects the “price” of evaluating a collection of interactions simultaneously. That is, inactive signed interactions in the collection wash out the scores of active interactions, resulting in worse performance.

**Multi-component AND rule (OR)**

The OR generative model (equation (4.11)) is active with high probability when either of two AND rules is active. These AND rules are defined over the same features, but describe different functional relationships between features and responses. Active signed interactions distinguish between these functional forms. That is, we define a signed interaction as active if it represents a portion of one AND rule (Table 4.1). Active unsigned interactions represent a subset of active features. This distinction makes the signed recovery problem more difficult, since signed interactions must not only identify the 4 active features but also the $2/2^4$ active states.

Figure 4.7 (middle) reports the PR curves for interaction recovery in both the *Drosophila* (left) and Gaussian (right) settings, averaged over 50 simulation replicates. In both settings, the signed interaction representation improves the quality of recovered interactions (AUC-PR Gaussian: 0.73 signed, signed and filtered 0.86, unsigned 0.48; AUC-PR *Drosophila*: 0.92 signed, signed and filtered 0.92, unsigned 0.85).

**Additive AND rules (ADD)**

The ADD generative model (equation (4.12)) depends on two AND rules defined over three distinct features. Responses are active with moderate probability when one AND rule is active and high probability when both rules are active. Active signed interactions represent a portion of one AND rule, but not if they combine portions of the two rules (Table 4.1). Similarly, active unsigned interactions represent a subset of active features for one rule, but not if they combine features across the two rules.

Figure 4.7 (bottom) reports the PR curves for interaction recovery in both the *Drosophila* (left) and Gaussian (right) settings, averaged over 50 simulation replicates. Here, iRF recovers many combinations of active features as unsigned interactions but does not distinguish
between features that participate in different AND rules. As a result, the quality of unsigned interaction is low relative to signed interactions (AUC-PR Gaussian: 0.73 unsigned, 0.92 signed; AUC-PR Drosophila: 0.26 unsigned 0.42). The improvement among signed interactions comes without explicit filtering. In other words, tracking how features are used along decision paths implicitly identifies features participating in the same rules. Explicitly filtering improves the quality of interactions further (AUC-PR Gaussian: 0.96 signed and filtered; AUC-PR Drosophila: 0.68 signed and filtered) by removing those whose features are selected independently across an RF.

**Response surfaces**

Figure 4.15—Figure 4.20 depict the response surfaces extracted from RFs for the six simulation experiments. Each plot indicates $P(y = 1)$ as a function of the interacting features, providing insights into what an RF “sees” in the data. For instance, the AND model is characterized by a single bump in the feature space (Figure 4.15 and Figure 4.16), reflecting the fact that responses are only active when all interacting features are present at high levels. The OR model is characterized by two bumps over the same set of features, showing how different configurations of these features generate response activity (Figure 4.17 and Figure 4.18). The ADD model is characterized by two bumps over different sets of features, which independently influence response activity. We provide details on generating response surfaces in Sec. 4.B.
Table 4.1: Active features, thresholds, and active signed interactions for each simulation experiment. Active unsigned interactions are given by the set of features that define an active signed interaction. We consider subsets of active interactions as active in evaluating interaction recovery. Thresholds are calculated for each active feature using the indicated quantile.
Figure 4.7: Precision recall curves for interaction recovery using *Drosophila* data (left) and standard Gaussian data (right). Responses were generated from the AND model (eq. (4.10), top), OR model (eq. (4.11), middle) and ADD model (eq. (4.12), bottom). Curves are given for unsigned interactions (blue), signed interactions (red), and filtered signed interactions (green). Across all simulation experiments, the quality of recovered interactions improves using the signed representation.
4.5 Extracting regulatory interactions in *Drosophila* embryos

In metazoan development, a single fertilized egg cell gives rise to a wide range of cell types and organ systems. This process depends on a precise program of gene expression that is coordinated by interactions among genetic and epigenetic factors. Recent efforts have made progress measuring the individual elements that play an important role in this process [44, 91, 84, 23]. Here we consider the problem of extracting candidate interactions among these elements directly from such data. The interactions we identify describe associations between combinations of TFs and genomic processes in the *Drosophila* embryo. While such relationships do not represent comprehensive, quantitative models of transcription, they do provide simplified summaries that correspond to potential hypotheses for experimental validation.

**Spatiotemporal gene expression**

Studying how genes are expressed across time and space can provide valuable insights into the transcriptional networks that drive embryogenesis. To investigate spatiotemporal covariability of gene expression, previous work developed a quantitative, virtual atlas of the *Drosophila* blastoderm [44]. Briefly, 1822 embryos representing 6 temporal cohorts were collected and imaged using fluorescence microscopy to detect the mRNA expression levels of 95 genes. Embryos were registered to a common template that represented the expression of each gene in 6078 nuclei. Below, we used siRF to extract candidate interactions from this data in the context of a well-studied regulatory network.

The gap gene network has been extensively studied and provides a set of known physical interactions that establish anterior posterior (A-P) gene expression patterns in the early *Drosophila* embryo [102, 101, 67]. We used these known interactions to evaluate siRF through spatially local signed interaction networks (SLSIN) similar to the spatially local correlation networks (SLCN) studied in [140]. For each gap gene, we fit a siRF to predict thresholded gene expression levels and extracted predictive and stable signed interactions (see Sec. 4.D for details). To identify spatially localized signed interactions, we used a region-weighted siRF. Specifically, we weighted leaf nodes based on the number of observations they contained from a given region (e.g. contiguous block of nuclei) and sampled them with probability proportional to this weight for RIT. Thus, leaf nodes containing many observations from the specified region were sampled with higher probability than those containing few. Following [140], we split gap genes expressed in multiple regions into their anterior and posterior domains and ran region-weighted siRF in each half of the embryo respectively.

Table 4.2 reports *gap interactions* recovered from predicting each gap gene expression pattern after filtering ($\tau = 0.5$). Each of these interactions is associated with a set of predictive rules in the fitted siRF based on the signed interaction mapping. We used these rules to generate predictions for each interaction (Sec. 4.A) and ranked interactions based
on the area under the receiver operating characteristic curve (AUROC) on held out test data. We further filtered the recovered interactions by removing any whose AUROC was not within 90% of the most accurate. The remaining interactions are both highly accurate and stable. For example, prediction rules defined by low levels of Knirps (Kni) and Krüppel (Kr) are highly predictive for the anterior stripes of giant (Gt) (AUROC 0.91).

We compared the putative gap interactions in Table 4.2 to previously validated interactions summarized in [67]. For each of the 11 known gap gene interactions (Figure 4.21), siRF recovered at least one signed interaction corresponding to the true physical interaction (Table 4.3). For four of the six gap genes, Gt, Kni, Kr, and Hunchback (Hb), these known interactions appear at the top of our ranked lists in Table 4.2. Moreover, the known regulators for many gap gene domains appear as a single recovered interaction, supporting previous results indicating that gap genes act collectively to establish the spatial expression domains of their targets (for a summary see [67]).

In addition to identifying well-known pairwise interactions, we also characterized relationships between high-order interactions and spatial patterning. Figure 4.8A and Figure 4.9A depict the true and predicted expression patterns for a single, held-out time cohort of Gt and Kni. In both cases, predictions associated with individual signed interactions capture much of the positional information represented in the full RF prediction. The relationships described by these signed interactions can be viewed as response surfaces, which describe the probability that the predicted gene is expressed as a function of interacting features (Figure 4.8B,C and Figure 4.8B,C). Each set of surface maps exhibits behavior that is characteristic of AND-type rules. That is, response activity increases considerably whenever all features are present at levels (high/low) indicated by the signed interaction. To the best of our knowledge, such maps represent the first quantitative characterization of high-order interaction dynamics learned by an RF.
Figure 4.8: siRF predictions and response surfaces for Gt expression. [A]: Observed Gt expression (top), RF predicted expression (middle), and predictions generated from siRF using two signed interactions (bottom), projected onto A-P axis. [B], [C]: Response surfaces showing the probability Gt is expressed as a function of Kr and Kni and of Kr, Kni, Hb, and Eve, evaluated on held-out test data. These interactions alone are sufficient to reproduce the anterior (Kr and Kni) and posterior (Kr, Kni, Hb, and Eve) stripes of Gt with high precision.
Figure 4.9: siRF predictions and response surfaces for Knï expression. [A]: Observed Knï expression (top), RF predicted expression (middle), and predictions generated from siRF using two signed interactions (bottom), projected onto A-P axis. [B], [C]: Response surfaces showing the probability Knï is expressed as a function of Gt and Knrl and of Kr, Gt, and Hb, evaluated on held-out test data. These interactions are sufficient to reproduce the anterior (Gt and Knrl) and posterior (Kr, Gt, and Hb) stripes of Gt with high precision.
Transcription factor binding and enhancer activity

Embryonic development is driven by spatially restricted gene expression patterns that initiate a range of cellular programs across distinct organ systems. Short segments of DNA known as enhancers regulate this process by recruiting TFs, which interact with one another and with other DNA-associated proteins to initiate transcription. Extensive work in *Drosophila* has shed light on the relationship between individual TFs and enhancers, particularly in early stage, blastoderm embryos [6, 85, 76]. However, identifying how TFs interact en masse throughout embryonic development remains an open challenge.

Below, we combined genome-wide maps of DNA binding for 307 TFs [73] with experimental annotations of 7705 genomic sequences [76] to investigate the relationship between TF binding and enhancer activity. Each segment \( i = 1, \ldots, 7705 \) was represented using feature vector \( \mathbf{x}_i \in \mathbb{R}^{307} \), which described TF binding activity, and response \( y_i \in \{0, 1\} \), which indicated whether the segment acts as an enhancer in at least one stage of embryonic development. We trained a siRF to predict enhancer activity based on TF binding patterns and extracted signed interactions to identify candidate regulatory interactions (see Sec. 4.D for details on data processing and prediction formulation). It is important to note that these interactions represent predictive rules rather than established regulatory mechanisms. To determine high-quality candidates for follow-up validation, we considered multiple criteria for each interaction including predictive accuracy, stability, previous evidence surrounding involved TFs, and qualitative analyses, including visualizations of raw signal.

Figure 4.10 reports the accuracy of siRF on held-out test data in terms of both area under the receiver operating characteristic curve (AUROC) and area under the precision recall curve (AUC-PR). Broadly speaking, siRF identified active enhancers (Any) with high accuracy (AUROC: 0.88, AUC-PR: 0.76). This corresponds to a precision of greater than 0.9 at a recall of 0.25 (> 0.8 at 0.5 recall), an important consideration for genome-wide predictions, since enhancers represent a small fraction (\( \sim 1\% \)) of the entire genome. To determine whether these results represented accurate prediction across embryogenesis, we also evaluated ROC and PR curves for active enhancers at each stage of embryonic development (Figure 4.10). Although enhancers are known to exhibit considerable heterogeneity throughout embryogenesis [6, 76], siRF predictions were consistently accurate across each stage. However, predicted probabilities drop considerably for enhancers that are not active until later stages of development (Figure 4.11). We attribute this to the fact that the important TFs we identify (see below) were only measured in earlier stages of embryonic development.

To identify candidate TF interactions surrounding enhancer elements, we filtered the signed interactions extracted from siRF at a level of \( \tau = 0.5 \). For each of the survived interactions, we generated and visually inspected response surfaces to determine interactions that exhibited strong effects in held-out test data (Figure 4.13). Qualitatively, the effects of an interaction appear as “spikes” in the response surfaces, which indicate that the probability of enhancer activity rises sharply when TFs are bound in a specific configuration. Figure 4.13 reports several examples of response surfaces that show strong interaction effects. Each represents a subset of active enhancers that can be predicted with high precision (\( \sim 0.7 \))
based on the binding activity of 3 to 4 TFs. Table 4.4 reports the set of 28 interactions that remained after screening.

To quantify the effect of interactions more precisely, we constructed predictive rules for each interaction using the decision paths of siRF (Sec. 4.A) and evaluated the accuracy of these rules on held-out test data. Averaged across the set of 28 screened interactions (Table 4.4), these predictions achieve comparable accuracy to the full siRF (AUROC: 0.87, AUC-PR: 0.72), offering a substantially simplified model of enhancer activity. Interactions associated with accurate predictive rules surrounded TFs known to play an important role in development: Kr, Kni, and homothorax (Hth). These TFs were measured at Kr: 0-10 hours post fertilization (hpf), Kni: 0-12 hpf, and Hth: 0-8 hpf, which potentially explains the drop in siRF predictions for stages 13-14 (10-12 hpf) and 15-16 (12+ hpf).

Figure 4.12 reports the predictions associated with any interaction containing at least one of these TFs, grouped by enhancer status at any stage of embryonic development. A large portion of both enhancers and non-enhancers are bound by at least one of these TFs. However, predictive rules involving all three of these TFs identify a large portion of active enhancers with an exceptionally high degree of accuracy (AUC-PR: 0.63). This suggests that high levels of binding by all three of these factors may play a broadly important role in establishing enhancer activity. We reviewed the images used to label these segments, and found sequences with high predicted probability based on Hth, Kni, and Kr binding that were labeled as non-enhancers drove patterned expression in a subset of embryos, providing further evidence for the combined role of these three TFs.

To identify potential targets of interactions involving Hth, Kni, and Kr, we used the predictive rules associated with each interaction as a score to rank enhancer elements. For highly ranked elements, we visualized the raw ChIP-seq profiles of interacting TFs at the enhancer element to determine whether they were consistent with the corresponding signed interactions (Figure 4.14). For instance, the signed interaction $Hth^+ \cdot Kni^+ \cdot Kr^+$ is defined by enriched binding of Hth, Kni, and Kr. In addition, we scanned the genome for Hth, Kni, and Kr motifs to determine whether putative targets contained known binding sites of these TFs. Figure 4.14 highlights the top ranked enhancer elements based on the binding activity of Hth, Kni, and Kr. These sequences exhibit pronounced peaks for each of the TFs, concentrated in a similar region of the enhancer and surrounding known motifs, making them plausible targets of a physical interaction.

Despite this evidence, we note once again that these relationships represent associations as opposed to physical interactions. However, several of these TFs are known to interact and play an important role in establishing enhancer activity. For instance, Kni, Kr, and Cad (Figure 4.13) interact as part of the widely studied gap gene network, which is responsible for establishing A-P patterning [102, 101, 67]. Although additional experimental evidence is necessary to validate these interactions, our approach represents a significant contribution to identifying how and where TFs may interact.
Figure 4.10: ROC (left) and PR (right) curves for predicting active enhancers based on TF binding. Colors indicate curves for evaluated at different stages of embryonic development. The pink curve (any) is evaluated for sequences that are active in at least one stage of embryonic development.

Figure 4.11: Average siRF predicted probabilities by stage first active, restricted to elements that once active, remain active throughout development. Error bars show the 5th and 95th percentiles of average predicted probabilities across 1000 bootstrap replicates. Enhancers that are only active in later stages of development show a considerable drop in predicted probability of enhancer activity.
Figure 4.12: Superheat map clustering genomic sequences by siRF interaction specific predictions of enhancer activity. Columns correspond to siRF predictions associated with each interaction, grouped by the number of enhancer-related TFs (Kni, Kr, Hth) contained in the interaction. Rows correspond to genomic sequences, grouped by whether the sequence is an active enhancer for at least one stage of embryonic development. Binding of all three enhancer related TFs identifies a large subset of active enhancers that can be predicted with nearly perfect precision.
Figure 4.13: Response surfaces depicting the probability a genomic sequence acts as an enhancer during any stage of embryonic development as a function of TF binding quantiles. Plots correspond to interactions among: [A] Cad+, E(bx)-, Kni+,-Kr+, [B] Hth+, Kni+,-Kr+, and [C] D+,-Hth+,-Kni+. Each response surface represents an interaction among two factors. Interactions among additional factors are depicted using multiple response surfaces, generated for observations where the additional TFs are bound at low/high levels relative to siRF thresholds. Spikes in these response surfaces show the probability of enhancer activity rises sharply when TFs are bound in particular configurations.
Figure 4.14: Raw ChIP-seq profiles for putative targets of an interaction among Hth, Kni, and Kr. Enhancers are ranked using predictive rules associated with the signed interaction, extracted from siRF. ChiP peaks show a strong correspondence to signed interactions.
4.6 Discussion

Here we proposed signed iRF, or siRF, for refining the interactions recovered by iRF to gain insights into complex data. siRF builds on the notion of signed interactions, which describe both a set of interacting features and a functional relationship between features and responses. We validated siRF across multiple simulations and real data settings. In our simulations, siRF correctly identified active features and functional relationships between features/responses in different regions of the feature space. When trained to predict spatial gene expression patterns, siRF recovered all 11 links in the gap gene network from spatial data and posits high-order interactions that are highly predictive of A-P patterning. When trained to predict genomic sequences that drive patterned gene expression, siRF identifies predictive rules that suggest candidate biological mechanisms responsible for regulating enhancer activity and nervous system gene expression.

We have offered evidence that siRF extracts interpretable and relevant information from a fitted RF. However, there are several areas for further work. siRF searches for interactions globally despite the fact that RF-based methods learn localized interactions in heterogeneous data. Although the leaf node weighting approach used in the spatial gene expression study can be used to identify local interactions, it requires a priori information on subpopulation structure. We are currently investigating strategies to identify meaningful subpopulations from siRF to gain additional insights into how iRF achieves accurate prediction. In addition, the signed interactions we recover represent individual components of the full iRF. We are exploring how these components relate to one another and how their combined behavior relates to responses to provide a more holistic representations of relationships in complex data.

Interpreting machine learning algorithms that achieve state-of-the-art predictive accuracy has the potential to offer new insights into high-dimensional, heterogeneous data. As part of this broad framework, the work we described here will help guide inquiry into the mechanisms underlying biological and other complex systems.
Appendix
4.A Interpretable prediction through signed interactions

RFs leverage high-order feature interactions for accurate prediction, but it is often difficult to interpret how these interactions influence predictions across an ensemble of trees. We use signed interactions to generate rule-based predictions that depend only on active features and that define a consistent functional relationship between features and responses.

The decision paths of an RF can be represented as rules
\[ r(x; \mathcal{R}) = \prod_{j=1}^{p} \mathbb{1}(x_j \in R_j) \] (4.13)
where \( \mathbb{1}(\cdot) \) denotes an indicator function, \( \mathcal{R} = R_1 \times \cdots \times R_p \) a hyperrectangle associated with the decision path, and \( R_j \) intervals of the form \( x_j \geq t_j \) or \( x_j < t_j \). For a signed interaction \( S \) we collect all leaf nodes throughout an RF for which \( S \) falls on the decision path. We represent this collection as a group of regions and predictions
\[ G(S) := \{(\mathcal{R}_l, Z_l) : S \subseteq S(\mathcal{R}_l)\}_{l=1}^L \] (4.14)
where \( \mathcal{R}_l \subseteq \mathbb{R}^p \) and \( Z_l \in \{0,1\} \) denote the hyperrectangle and prediction associated with each leaf node \( l = 1 \ldots , L \) in an RF. Equation (4.14) defines an ensemble of grouped rules, for which we write predictions as
\[ g(x; S) = \sum_{(\mathcal{R}, Z) \in G(S)} r(x; \mathcal{R}, \mathcal{I}) \cdot Z, \] (4.15)
where \( \mathcal{I} = \{|j| : j \in S\} \) denotes the set of active features \( S \) is defined over and
\[ r(x; \mathcal{R}, \mathcal{I}) := \prod_{j \in \mathcal{I}} \mathbb{1}(x_j \in R_j) \] (4.16)
represents a rule over only the interacting features \( \mathcal{I} \). Predictions may assign equal weight to each term in equation (4.15) or weight based on measures of importance associated with each leaf node.

Intuitively, equation (4.15) can be viewed as a decision rule with smooth boundaries, made up of many rules defined over the same features. This smooth rule takes the form
\[ r(x; S) = \prod_{\gamma \in S} h(x; \gamma) \] (4.17)
where
\[ h(x; \gamma) = \begin{cases} \mathbb{1}(x_\gamma < \cdot) & \text{if } \text{sign}(\gamma) = -1 \\ \mathbb{1}(x_\gamma \geq \cdot) & \text{if } \text{sign}(\gamma) = 1 \end{cases} \] (4.18)
and exhibits several desirable properties:
1. Smooth decision boundaries make predictions less sensitive to small changes in \( \mathbf{x} \) compared with a single decision rule defined over the same features.

2. Predictions incorporate non-linear feature interactions in the form of rules as in equation (4.16).

3. Predictions depend solely on the set of interacting features \( \mathcal{I} \).

4. Predictions are built from a collection of rules that define an identical functional relationship with responses described by (4.17).

5. Predictions are monotonic as a function of each active feature, and the sign of each \( j \in S \) defines whether predictions are increasing or decreasing as a function of the corresponding feature.

These properties ensure that \( g(\mathbf{x}; S) \) provide robust and explainable predictions based on high-order, non-linear interactions.

4.B Region boundaries and response surfaces

Since signed interactions provide a coarse representation of hyperrectangles associated with a decision rule, running RIT on the collection of signed feature-index set/response pairs can be thought of as an approximate intersection of hyperrectangles. Understanding how response behavior varies across these coarse regions may be satisfactory in many applications. For instance, if the precision of experimental interventions is limited, a practitioner may not need to know the thresholds associated with each rule in a group. However, rule groups \( G(S) \) describe the distribution of thresholds selected across an RF, which we leverage to characterize response behavior with greater precision. This may be of interest in settings where a practitioner has the ability to precisely control some system and would like to optimize response behavior.

Intuitively, our representation uses each region \( \mathcal{R} \in G(S) \) as a data-adaptive binning to generate a histogram of response values. Averaging over every region \( \mathcal{R} \in G(S) \) allows us to identify the smoothed boundaries learned by an RF. We note that it may be desirable to weight regions \( \mathcal{R} \), for instance by using response purity or node size. In our simulations, we find that the boundary estimates are improved for larger leaf nodes and we weight each region in our response surface figures by the number of observations contained in the corresponding leaf. It may also be of interest to regularize estimates of these response surfaces, though we do not explore the idea in this paper.

When \( |S| \) is small we can visualize the response surfaces described in equation (4.15). For example, Figure 4.15-Figure 4.20 depict response surfaces that show the probability of response activity as a function of interacting features in each of our simulation settings. For larger interactions, we visualize region boundaries marginally for each feature \( j \in \mathcal{I} \) by
considering the distribution of thresholds that describe each $R_j$, the 1-dimensional hyper-rectangle corresponding to the $j^{th}$ feature in region $\mathcal{R}$. From this perspective, variability in thresholds provides sense of the stability of boundaries under different data and model perturbations.
Figure 4.15: Each plot shows $P(y = 1)$ as a function of four interacting features: $\{x_1, x_2, x_3, x_4\}$. The different plots correspond to high/low levels of $x_3$ and $x_4$, determined by median RF threshold for each feature: $\{x_3-, x_4-\}$ (top left), $\{x_3-, x_4+\}$ (top right), $\{x_3+, x_4-\}$ (bottom left), $\{x_3+, x_4+\}$ (bottom right). Activity in the bottom right plot is indicative of an order-4 AND rule: responses are active only when all 4 features are present at high levels.
Figure 4.16: Each plot shows $P(y = 1)$ as a function of four interacting features: \{Sna, Kr, D, Zld\}. The different plots corresponds to high/low levels of D and Zld, determined by 25th percentile of RF thresholds for each feature: \{D−, Zld−\} (top left), \{D−, Zld+\} (top right), \{D+, Zld−\} (bottom left), \{D+, Zld+\} (bottom right). Activity in the bottom right plot is indicative of an order-4 AND rule: responses are active only when all 4 features are present at high levels.
Figure 4.17: Each plot shows $P(y = 1)$ as a function of four interacting features: $\{x_1, x_2, x_3, x_4\}$. The different plots correspond to high/low levels of $x_3$ and $x_4$, determined by median RF threshold for each feature: $\{x_3-, x_4-\}$ (top left), $\{x_3-, x_4+\}$ (top right), $\{x_3+, x_4-\}$ (bottom left), $\{x_3+, x_4+\}$ (bottom right). Activity in the top left plot is indicative of an order-4 AND rule: responses are active when $x_1$ and $x_2$ are present at high levels and $x_3$ and $x_4$ at low levels. Activity in the bottom right plot corresponds to another order-4 AND rule: responses are active when $x_1$ and $x_2$ are present at low levels and $x_3$ and $x_4$ at high levels.
Figure 4.18: Each plot shows $P(y = 1)$ as a function of four interacting features: \{Sna, Kr, D, Zld\}. The different plots corresponds to high/low levels of D and Zld, determined by 25th percentile of RF thresholds for each feature: \{Sna−, Zld−\} (top left), \{Sna−, Zld+\} (top right), \{Sna+, Zld−\} (bottom left), \{Sna+, Zld+\} (bottom right). Activity in the top left plot is indicative of an order-4 AND rule: responses are active when Kr and D are present at high levels and Sna and Zld at low levels. Activity in the bottom right plot corresponds to another order-4 AND rule: responses are active when Kr and D are present at low levels and Sna and Zld at high levels.
Figure 4.19: Each plot shows $P(y = 1)$ as a function of three interacting features $\{x_1, x_2, x_3\}$ (top) and $\{x_4, x_5, x_6\}$ (bottom). The different plots correspond to high/low levels of $x_3$ (top) and $x_6$ (bottom), determined by median RF threshold for each feature: $x_3^-$ (top left), $x_3^+$ (top right), $x_6^-$ (bottom left), $x_6^+$ (bottom right). Activity in the top right plot is indicative of an order-3 AND rule: responses are active when $x_1$, $x_2$, and $x_3$ are present at high levels. Activity in the bottom right plot is also indicative of an order-3 AND rule: responses are active when $x_4$, $x_5$, and $x_6$ are present at high levels.
Figure 4.20: Each plot shows $P(y = 1)$ as a function of three interacting features \{Kr, D, Gt\} (top) and \{Sna, Zld, Twi\} (bottom). The different plots correspond to high/low levels of Gt (top) and Twi (bottom), determined by 25th percentile of RF thresholds for each feature: Gt− (top left), Gt+ (top right), Twi− (bottom left), Twi+ (bottom right). Activity in the top right plot is indicative of an order-3 AND rule: responses are active when Kr, D, and Gt are present at high levels. Activity in the bottom right plot is also indicative of an order-3 AND rule: responses are active when Sna, Zld, and Twi are present at high levels.
4.C **Redundant feature selection on decision paths**

In some cases, particularly for deep decision trees used by RF, a feature $j$ may be selected multiple times on the same decision path. In these situations both signed feature indices $j$ and $-j$ could appear on the decision path. The recursive nature of decision tree partitions suggests a natural way to handle this issue. For any features that are selected multiple times on a decision path, we use only the signed feature index associated with the first split, since subsequent splits on the same feature are restricted by the initial split. This representation destroys the one-to-one correspondence between a signed interaction and feature splits along a decision path. However, we find empirically that our simplified representation performs comparably to the full decision rule in terms of predictive accuracy and allows for a high-level, interpretable grouping of decision rules.
CHAPTER 4. SIGNED ITERATIVE RANDOM FORESTS (SIRF)

4.D Data processing and predictive formulation

Spatiotemporal gene expression

**Features and responses:** Data used to predict spatiotemporal gene expression were taken from the Berkeley Drosophila Transcription Network Project (BDTNP) 3D gene expression atlas [44, 59]. The 3D gene expression atlas measures mRNA expression for 95 genes in 6078 nuclei over 6 temporal cohorts of blastoderm embryos based on measurements taken by fluorescence microscopy. Following the original study, we fit predictive models based on the expression patterns of 23 genes known to play an important regulatory role in the early embryo. For each gene of interest, we generated binary responses by clustering nuclei for the target gene into two groups based on mRNA expression levels using the k-means algorithm. Active responses were determined based on the cluster with the highest mean expression level. We predicted these responses using the expression levels of the remaining 22 genes.

**Training/test split:** Gene expression patterns are highly variable in blastoderm embryos. To ensure that the interactions recovered by siRF were representative of the entire time course measured in the 3D expression atlas, observations were split into training and test sets based on time cohort. Specifically, we used an evenly spaced set of cohorts (1, 3, 5) as a training set and evaluated predictions on the held-out time cohorts (2, 4, 6). To evaluate the stability of recovered interactions across time, we repeated this process with the training and test sets swapped, and reported interactions that were recovered in both settings. All importance measures were averaged across the sample splits. Measures of prediction accuracy and response surfaces were evaluated held-out test data.

**siRF parameters:** To predict spatiotemporal gene expression patterns for a target gene, we ran iRF for 10 iterations with 25 bootstrap samples and report interactions with $K$ selected by minimizing prediction accuracy on out of bag samples. We set RIT parameters to their default values of $n_{tree} = 500$, $n_{child} = 2$ and $d = 5$ and filtered the recovered interactions at a level of $\tau = 0.5$ to identify stable and predictive interactions. In addition, we use a more stringent version of FSD to determine whether any gene in an interaction was selected independently of the remaining genes. Specifically, we consider $S = \{(S', S'') : S', S''$ partition $S, |S'| = 1\}$ and evaluate

$$\frac{1}{|S|} \sum_{(S', S'') \in S} P(S|C) - P(S'|C)P(S''|C)$$  \hspace{1cm} (4.19)

DNA binding and enhancer activity

**Responses:** Responses for the enhancer prediction problem were taken from the Stark Lab database of fly enhancers [76]. This database contains labels of enhancer activity for 7705 sequences, representing 13.5% of the non-coding, non-repetitive Drosophila genome. Labels were generated using a high-throughput, stable-integration, transgenic assay. Briefly, a genomic sequence (100-3000nt) was placed into a reporter construct and integrated into
a targeted site in the genome. The transgenic fly line was then amplified, embryos were collected, fixed, and stained using immunohistochemistry to detect the reporter [128, 138]. The stained fly line was then imaged and annotated using a controlled vocabulary [134, 133] to determine a) where and when expression is driven b) the strength of expression in specific regions based on staining intensity (from 1 - very weak to 5 - very strong). We used these annotations and scores to define a binary classification problem for enhancer prediction: predicting which sequences represented active enhancers at any stage of embryonic development. Here any sequence with a score greater than or equal to a given threshold (2, 3, 4) for any annotation term was considered an active enhancer while any sequence scored as 0 was considered an inactive enhancer. Based on these definitions, the number of observations varied across prediction problem and threshold from 4398 - 6862. In the main text, we report results for strong enhancer elements, where our model is most accurate. Prediction for each threshold is shown in Figure 4.22.

**Features:** ChIP-seq experiments available through the modENCODE/modERN consortia provide genome-wide, quantitative maps of DNA binding for a substantial portion of TFs in the *Drosophila* embryo [73]. To predict enhancer activity and nervous system expression, we considered a total of 332 experiments, each with at least two replicates, measuring DNA binding activity of 307 unique TFs. We evaluated TF binding activity for each replicate of an experiment based on peak value before irreproducibility discovery rate (IDR) thresholding. Specifically, for a target sequence we took the maximum normalized fold enrichment of any overlapping peaks to obtain a score for each sequence and replicate pair. Scores for each pair were combined by taking the maximum across replicates. Although this approach is likely to generate noisy scores in our dataset, it also ensures that relevant signal is maintained, allowing siRF to determine which peaks are important for a given prediction problem. Most TFs were measured for a single, wide time range, selected based on RNA-seq measurements, resulting in a single score for each TF/sequence pair. However, a small handful of TFs were measured at multiple time points during embryonic development. Since the temporal resolution of most TFs was not sufficient to differentiate enhancer activity or nervous system expression at distinct time points, we focused on predicting these activities at any developmental stage and aggregated TFs measured at multiple time points by taking their maximum scores.

**Training/test split:** Studies of the 3D architecture of the genome are beginning to shed light on spatial dependencies that play an important role in gene regulation. Ideally, we would like our training/testing split to respect these dependencies. However, current knowledge of the 3D structure of the genome does not suggest a well-justified approach for sample splitting. Since we were interested in identifying candidate interactions across the entire genome, we randomly split sequences into equally sized training and test sets. All reported measures of prediction accuracy and response surfaces were evaluated on held-out test data.

**siRF parameters:** To predict enhancer and nervous system specific activity, we ran iRF for 10 iterations with 25 bootstrap samples and report interactions with \( K \) selected by minimizing prediction accuracy on out of bag samples. We set RIT parameters to their
default values of $n_{\text{tree}} = 1000$, $n_{\text{child}} = 2$ and $d = 3$ and filtered the recovered interactions at a level of $\tau = 0.5$ to identify stable and predictive interactions. The larger number of trees and lower depth were necessary to identify stable interactions in this lower-signal dataset. In addition, we use the more stringent definition of $FSD$ described in equation (4.19).
Figure 4.21: Spatial diagram of interactions in the gap gene network, originally described in [67]. Links are numbered 1 to 11 and all represent repressive interactions. Genes that are expressed in multiple regions are represented with subscripts (e.g. $gt_1$, $gt_2$). The network described in [67] does not consider the anterior domains of Tll, Kni, or Hkb. The interactions involving these genes in the above network correspond to the posterior domain.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Interaction</th>
<th>AUROC</th>
<th>AUC-PR</th>
<th>$CPE(S; 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gt1</td>
<td>Kni-$\cdot$Kr-$\cdot$</td>
<td>0.91</td>
<td>0.74</td>
<td>0.435</td>
</tr>
<tr>
<td>Gt1</td>
<td>Hb-$\cdot$Kni-$\cdot$Kr-$\cdot$</td>
<td>0.899</td>
<td>0.697</td>
<td>0.298</td>
</tr>
<tr>
<td>Gt1</td>
<td>Fkh-$\cdot$Kni-$\cdot$Kr-$\cdot$</td>
<td>0.897</td>
<td>0.695</td>
<td>0.177</td>
</tr>
<tr>
<td>Gt1</td>
<td>Knrl-$\cdot$Kr-$\cdot$</td>
<td>0.894</td>
<td>0.707</td>
<td>0.188</td>
</tr>
<tr>
<td>Gt1</td>
<td>D-$\cdot$Kni-$\cdot$Kr-$\cdot$</td>
<td>0.888</td>
<td>0.706</td>
<td>0.177</td>
</tr>
<tr>
<td>Gt1</td>
<td>Kni-$\cdot$Kr-$\cdot$Tll-$\cdot$</td>
<td>0.886</td>
<td>0.701</td>
<td>0.214</td>
</tr>
<tr>
<td>Gt1</td>
<td>Kni-$\cdot$Kr-$\cdot$Prd+$\cdot$</td>
<td>0.882</td>
<td>0.692</td>
<td>0.223</td>
</tr>
<tr>
<td>Gt1</td>
<td>Eve+$\cdot$Kr-$\cdot$</td>
<td>0.861</td>
<td>0.704</td>
<td>0.173</td>
</tr>
<tr>
<td>Gt1</td>
<td>Hkb-$\cdot$Kr-$\cdot$</td>
<td>0.843</td>
<td>0.558</td>
<td>0.184</td>
</tr>
<tr>
<td>Gt1</td>
<td>Kr-$\cdot$Prd+$\cdot$</td>
<td>0.841</td>
<td>0.59</td>
<td>0.251</td>
</tr>
<tr>
<td>Gt1</td>
<td>Fkh-$\cdot$Kr-$\cdot$</td>
<td>0.824</td>
<td>0.536</td>
<td>0.177</td>
</tr>
<tr>
<td>Gt1</td>
<td>Kr-$\cdot$Tll-$\cdot$</td>
<td>0.823</td>
<td>0.571</td>
<td>0.218</td>
</tr>
<tr>
<td>Gt1</td>
<td>D+$\cdot$Kr-$\cdot$</td>
<td>0.819</td>
<td>0.511</td>
<td>0.11</td>
</tr>
<tr>
<td>Gt2</td>
<td>Eve+$\cdot$Hb-$\cdot$Kni-$\cdot$Kr-$\cdot$</td>
<td>0.952</td>
<td>0.817</td>
<td>0.224</td>
</tr>
<tr>
<td>Gt2</td>
<td>D+$\cdot$Hb-$\cdot$Kni-$\cdot$Kr-$\cdot$</td>
<td>0.939</td>
<td>0.718</td>
<td>0.32</td>
</tr>
<tr>
<td>Gt2</td>
<td>D+$\cdot$Hb-$\cdot$Kni-$\cdot$</td>
<td>0.937</td>
<td>0.711</td>
<td>0.334</td>
</tr>
<tr>
<td>Gt2</td>
<td>Fkh-$\cdot$Hb-$\cdot$Kni-$\cdot$</td>
<td>0.936</td>
<td>0.713</td>
<td>0.247</td>
</tr>
<tr>
<td>Gene 1</td>
<td>Gene 2</td>
<td>Gene 3</td>
<td>Similarity</td>
<td>Distinctness</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
<td>--------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Gt2</td>
<td>D+_Hb-_Kni-_Kr-_Slp1-</td>
<td>0.936</td>
<td>0.708</td>
<td>0.178</td>
</tr>
<tr>
<td>Gt2</td>
<td>Fkh-_Hb-_Kni-_Kr-</td>
<td>0.936</td>
<td>0.715</td>
<td>0.239</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Tll-</td>
<td>0.931</td>
<td>0.702</td>
<td>0.305</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Kr-_Tll-</td>
<td>0.93</td>
<td>0.7</td>
<td>0.294</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Kr-_Prd+</td>
<td>0.923</td>
<td>0.694</td>
<td>0.236</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Prd+</td>
<td>0.92</td>
<td>0.682</td>
<td>0.245</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Hkb-_Kni-_Kr-</td>
<td>0.916</td>
<td>0.67</td>
<td>0.199</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Kr-</td>
<td>0.896</td>
<td>0.582</td>
<td>0.549</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Kr-_Slp1-</td>
<td>0.89</td>
<td>0.581</td>
<td>0.311</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Kr-_Zell-</td>
<td>0.889</td>
<td>0.588</td>
<td>0.21</td>
</tr>
<tr>
<td>Gt2</td>
<td>D+_Hb-_Kt-</td>
<td>0.883</td>
<td>0.54</td>
<td>0.346</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-</td>
<td>0.877</td>
<td>0.545</td>
<td>0.56</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Knr-_Kt-</td>
<td>0.87</td>
<td>0.519</td>
<td>0.171</td>
</tr>
<tr>
<td>Gt2</td>
<td>Hb-_Kni-_Slp1-</td>
<td>0.869</td>
<td>0.547</td>
<td>0.317</td>
</tr>
<tr>
<td>Gt2</td>
<td>Eve+_Hb-_Kt-</td>
<td>0.864</td>
<td>0.508</td>
<td>0.237</td>
</tr>
<tr>
<td>Hb1</td>
<td>Kni-_Kt-</td>
<td>0.709</td>
<td>0.411</td>
<td>0.248</td>
</tr>
<tr>
<td>Hb1</td>
<td>Eve+_Kt-</td>
<td>0.701</td>
<td>0.409</td>
<td>0.161</td>
</tr>
<tr>
<td>Hb1</td>
<td>Gt-_Kt-</td>
<td>0.656</td>
<td>0.373</td>
<td>0.145</td>
</tr>
<tr>
<td>Hb2</td>
<td>Gt-_Hkb-_Kni-</td>
<td>0.961</td>
<td>0.789</td>
<td>0.218</td>
</tr>
<tr>
<td>Hb2</td>
<td>Hkb-_Kni-_Kt-</td>
<td>0.9</td>
<td>0.637</td>
<td>0.192</td>
</tr>
<tr>
<td>Hb2</td>
<td>Hkb-_Kni-</td>
<td>0.895</td>
<td>0.623</td>
<td>0.27</td>
</tr>
<tr>
<td>Hb2</td>
<td>Gt-_Kni-_Kt-</td>
<td>0.893</td>
<td>0.541</td>
<td>0.345</td>
</tr>
<tr>
<td>Hb2</td>
<td>Gt-_Kni-_Slp1-</td>
<td>0.886</td>
<td>0.512</td>
<td>0.217</td>
</tr>
<tr>
<td>Hb2</td>
<td>Gt-_Kni-</td>
<td>0.871</td>
<td>0.49</td>
<td>0.438</td>
</tr>
<tr>
<td>Hb2</td>
<td>Gt-_Kt-</td>
<td>0.867</td>
<td>0.533</td>
<td>0.177</td>
</tr>
<tr>
<td>Hkb1</td>
<td>Croc+_D-</td>
<td>0.966</td>
<td>0.872</td>
<td>0.194</td>
</tr>
<tr>
<td>Hkb1</td>
<td>Eve-_Twi+</td>
<td>0.963</td>
<td>0.769</td>
<td>0.097</td>
</tr>
<tr>
<td>Hkb1</td>
<td>Croc+_Gt-</td>
<td>0.958</td>
<td>0.87</td>
<td>0.188</td>
</tr>
<tr>
<td>Hkb1</td>
<td>Eve-_Odd+</td>
<td>0.95</td>
<td>0.77</td>
<td>0.086</td>
</tr>
<tr>
<td>Hkb1</td>
<td>D-_Kni+</td>
<td>0.946</td>
<td>0.788</td>
<td>0.086</td>
</tr>
<tr>
<td>Hkb1</td>
<td>D-_Prd-</td>
<td>0.939</td>
<td>0.604</td>
<td>0.1</td>
</tr>
<tr>
<td>Hkb1</td>
<td>D-_Twi+</td>
<td>0.924</td>
<td>0.695</td>
<td>0.105</td>
</tr>
<tr>
<td>Hkb1</td>
<td>D-_Gt-</td>
<td>0.908</td>
<td>0.63</td>
<td>0.116</td>
</tr>
<tr>
<td>Hkb1</td>
<td>Knrl+_Odd+</td>
<td>0.904</td>
<td>0.746</td>
<td>0.091</td>
</tr>
<tr>
<td>Hkb1</td>
<td>D-_Odd+</td>
<td>0.891</td>
<td>0.625</td>
<td>0.113</td>
</tr>
<tr>
<td>Hkb2</td>
<td>Fkh+_Sna+</td>
<td>0.992</td>
<td>0.959</td>
<td>0.239</td>
</tr>
<tr>
<td>Hkb2</td>
<td>D-_Ftz-</td>
<td>0.983</td>
<td>0.817</td>
<td>0.104</td>
</tr>
<tr>
<td>Hkb2</td>
<td>Ftz-_Twi+</td>
<td>0.982</td>
<td>0.886</td>
<td>0.182</td>
</tr>
<tr>
<td>Hkb2</td>
<td>Fkh+_Odd+</td>
<td>0.981</td>
<td>0.921</td>
<td>0.178</td>
</tr>
<tr>
<td>Hkb2</td>
<td>Eve-_Ftz-</td>
<td>0.98</td>
<td>0.783</td>
<td>0.112</td>
</tr>
<tr>
<td>Hkb2</td>
<td>Ftz-_Sna+</td>
<td>0.978</td>
<td>0.866</td>
<td>0.126</td>
</tr>
<tr>
<td>Hkb2</td>
<td>Eve-_Hb-</td>
<td>0.971</td>
<td>0.814</td>
<td>0.132</td>
</tr>
<tr>
<td><strong>Hkb2</strong></td>
<td><strong>D-</strong></td>
<td><strong>Twi+</strong></td>
<td>0.967</td>
<td>0.744</td>
</tr>
<tr>
<td><strong>Hkb2</strong></td>
<td><strong>Eve-</strong></td>
<td><strong>Twi+</strong></td>
<td>0.967</td>
<td>0.735</td>
</tr>
<tr>
<td><strong>Hkb2</strong></td>
<td><strong>Fkh+</strong></td>
<td><strong>Kni+</strong></td>
<td>0.965</td>
<td>0.896</td>
</tr>
<tr>
<td><strong>Hkb2</strong></td>
<td><strong>D-</strong></td>
<td><strong>Hb-</strong></td>
<td>0.964</td>
<td>0.837</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Knr1+</strong></td>
<td>0.961</td>
<td>0.921</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Knr1+</strong></td>
<td><strong>Prd-</strong></td>
<td>0.959</td>
<td>0.902</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Eve-</strong></td>
<td><strong>Knr1+</strong></td>
<td>0.958</td>
<td>0.917</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Knr1+</strong></td>
<td><strong>Slp1-</strong></td>
<td>0.955</td>
<td>0.904</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Knr1+</strong></td>
<td><strong>Rho-</strong></td>
<td>0.951</td>
<td>0.895</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Kr-</strong></td>
<td>0.888</td>
<td>0.613</td>
</tr>
<tr>
<td><strong>Kni1</strong></td>
<td><strong>Croc+</strong></td>
<td><strong>Zen-</strong></td>
<td>0.883</td>
<td>0.77</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Hb-</strong></td>
<td>0.909</td>
<td>0.721</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Hb-</strong></td>
<td>0.905</td>
<td>0.698</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Knr1_</strong></td>
<td>0.876</td>
<td>0.645</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Tll-</strong></td>
<td>0.874</td>
<td>0.68</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Knr1_</strong></td>
<td>0.864</td>
<td>0.657</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Knr1+</strong></td>
<td>0.853</td>
<td>0.72</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Fkh-</strong></td>
<td><strong>Hb-</strong></td>
<td>0.837</td>
<td>0.583</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Ftz+</strong></td>
<td><strong>Hb-</strong></td>
<td>0.831</td>
<td>0.558</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>D+</strong></td>
<td><strong>Gt-</strong></td>
<td>0.826</td>
<td>0.57</td>
</tr>
<tr>
<td><strong>Kni2</strong></td>
<td><strong>Fkh-</strong></td>
<td><strong>Gt-</strong></td>
<td>0.818</td>
<td>0.548</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Hb-</strong></td>
<td>0.94</td>
<td>0.838</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Kni-</strong></td>
<td>0.94</td>
<td>0.797</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>D+</strong></td>
<td><strong>Gt-</strong></td>
<td>0.936</td>
<td>0.799</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Kni-</strong></td>
<td>0.905</td>
<td>0.668</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Kni-</strong></td>
<td>0.898</td>
<td>0.604</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>D+</strong></td>
<td><strong>Gt-</strong></td>
<td>0.898</td>
<td>0.612</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Knr1_</strong></td>
<td>0.892</td>
<td>0.555</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Ftz+</strong></td>
<td><strong>Gt-</strong></td>
<td>0.878</td>
<td>0.539</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>Gt-</strong></td>
<td><strong>Hb-</strong></td>
<td>0.867</td>
<td>0.542</td>
</tr>
<tr>
<td><strong>Kr</strong></td>
<td><strong>D+</strong></td>
<td><strong>Gt-</strong></td>
<td>0.857</td>
<td>0.497</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Kni-</strong></td>
<td><strong>Prd-</strong></td>
<td>0.897</td>
<td>0.238</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Hb+</strong></td>
<td><strong>Prd-</strong></td>
<td>0.893</td>
<td>0.215</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Eve-</strong></td>
<td><strong>Kni-</strong></td>
<td>0.891</td>
<td>0.21</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Eve-</strong></td>
<td><strong>Kr+</strong></td>
<td>0.88</td>
<td>0.323</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Eve+</strong></td>
<td><strong>Prd-</strong></td>
<td>0.867</td>
<td>0.223</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Eve-</strong></td>
<td><strong>Gt+</strong></td>
<td>0.85</td>
<td>0.184</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Ftz-</strong></td>
<td><strong>Kr+</strong></td>
<td>0.835</td>
<td>0.216</td>
</tr>
<tr>
<td><strong>Tll1</strong></td>
<td><strong>Odd+</strong></td>
<td><strong>Prd-</strong></td>
<td>0.833</td>
<td>0.134</td>
</tr>
<tr>
<td><strong>Tll2</strong></td>
<td><strong>Fkh+</strong></td>
<td><strong>Knr1-</strong></td>
<td>0.995</td>
<td>0.986</td>
</tr>
<tr>
<td><strong>Tll2</strong></td>
<td><strong>Cad-</strong></td>
<td><strong>Fkh+</strong></td>
<td>0.986</td>
<td>0.978</td>
</tr>
<tr>
<td><strong>Tll2</strong></td>
<td><strong>Eve-</strong></td>
<td><strong>Kni-</strong></td>
<td>0.986</td>
<td>0.957</td>
</tr>
</tbody>
</table>
Table 4.2: Gap gene interactions recovered by siRF. The domains for genes expressed in multiple regions are numered as 1 or 2 (e.g. Gt1 and Gt2), indicating the anterior and posterior domains respectively. Interactions are filtered at level $\tau = 0.5$, and interactions whose AUROC is less than 90% of the most accurate interaction have been removed.
Table 4.4: Enhancer TF interactions recovered by siRF after screening.

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Correlation</th>
<th>p-value</th>
<th>False Discovery Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cad+_E(Bx)-_Hth+</td>
<td>0.747</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>Cad+_Zen+</td>
<td>0.746</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>Cad+_D+</td>
<td>0.745</td>
<td>0.007</td>
<td>0.007</td>
</tr>
<tr>
<td>Cad+_E(Bx)-_Kni+_Kr+</td>
<td>0.738</td>
<td>0.012</td>
<td>0.012</td>
</tr>
<tr>
<td>Cad+_Kni+_Kr+</td>
<td>0.738</td>
<td>0.033</td>
<td>0.033</td>
</tr>
<tr>
<td>Cad+_Insv+</td>
<td>0.725</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>E(Bx)-_Insv+</td>
<td>0.712</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>Grn+_Usp+</td>
<td>0.699</td>
<td>0.006</td>
<td>0.006</td>
</tr>
<tr>
<td>Oc+_Usp+</td>
<td>0.696</td>
<td>0.013</td>
<td>0.013</td>
</tr>
</tbody>
</table>
Figure 4.22: ROC (left) and PR (right) curves for predicting active enhancers based on TF binding. Colors indicate curves for evaluated at different thresholds of enhancer activity.
### Table 4.3

Each row corresponds to one link of the gap gene network in Fig. 4.21. For each pair of gap genes, we report the signed interaction containing Gene 2 (resp Gene 1) with the highest AUROC recovered when predicting Gene 1 (resp Gene 2). For genes expressed in the anterior (resp. posterior) region of the embryo, we weight all observations in the posterior (resp. anterior) half of the embryo to 0 when searching for interactions. This procedure recovers all links in the gap gene network with the correct sign.

<table>
<thead>
<tr>
<th>Link</th>
<th>Gene 1</th>
<th>Gene 2</th>
<th>Signed interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gt₁</td>
<td>Kr</td>
<td>Gt₁ ← Kni-, Kr ← Gt-Hb-Kni-Odd⁺</td>
</tr>
<tr>
<td>2</td>
<td>Hb₁</td>
<td>Kr</td>
<td>Hb₁ ← Kni-, Kr ← Gt-Hb-Kni-Odd⁺</td>
</tr>
<tr>
<td>3</td>
<td>Hb₁</td>
<td>Kni</td>
<td>Hb₁ ← Kni-, Kni ← Gt-Hb-Kr⁻</td>
</tr>
<tr>
<td>4</td>
<td>Kni</td>
<td>Kr</td>
<td>Kni ← Gt-Hb-Kr-, Kr ← Gt-Hb-Kni-Odd⁺</td>
</tr>
<tr>
<td>5</td>
<td>Gt₂</td>
<td>Kr</td>
<td>Gt₂ ← Eve₄-Hb-Kni-Kr-, Kr ← Gt-Hb-Kni-Odd⁺</td>
</tr>
<tr>
<td>6</td>
<td>Gt₂</td>
<td>Kni</td>
<td>Kni ← Gt-Hb-Kr-, Gt₂ ← Eve₄-Hb-Kni-Kr⁻</td>
</tr>
<tr>
<td>7</td>
<td>Hb₂</td>
<td>Kni</td>
<td>Hb₂ ← Gt-Hkb-Kni-, Kni ← Gt-Hb-Kr⁻</td>
</tr>
<tr>
<td>8</td>
<td>Gt₂</td>
<td>Hb₂</td>
<td>Gt₂ ← Eve₄-Hb-Kni-Kr-, Hb₂ ← Gt-Hkb-Kni⁻</td>
</tr>
<tr>
<td>9</td>
<td>Kni</td>
<td>Tll</td>
<td>Kni ← Gt-Tll-, Tll ← Eve-Kni⁻</td>
</tr>
<tr>
<td>10</td>
<td>Gt₂</td>
<td>Tll</td>
<td>Gt₂ ← Hb-Kni-Tll-, Tll ← D-Gt⁻</td>
</tr>
<tr>
<td>11</td>
<td>Hb₂</td>
<td>Hkb</td>
<td>Hb₂ ← Gt-Hkb-Kni-, Hkb ← Eve-Hb⁻</td>
</tr>
</tbody>
</table>
Bibliography


[65] John PA Ioannidis. “Why most published research findings are false”. In: PLoS medicine 2.8 (2005), e124.


[91] Stewart MacArthur et al. “Developmental roles of 21 Drosophila transcription factors are determined by quantitative differences in binding to an overlapping set of thousands of genomic regions”. In: Genome biology 10.7 (2009), R80.


