



Iterative random forests to discover predictive and stable high-order interactions

Sumanta Basu^{a,b,c,1}, Karl Kumbier^{d,1}, James B. Brown^{c,d,e,f,2}, and Bin Yu^{c,d,g,2}

^aDepartment of Biological Statistics and Computational Biology, Cornell University, Ithaca, NY 14853; ^bDepartment of Statistical Science, Cornell University, Ithaca, NY 14853; ^cData Driven Decisions Department, Preminon LLC, Antioch, CA 94531; ^dStatistics Department, University of California, Berkeley, CA 94720; ^eCentre for Computational Biology, School of Biosciences, University of Birmingham, Edgbaston B15 2TT, United Kingdom; ^fMolecular Ecosystems Biology Department, Biosciences Area, Lawrence Berkeley National Laboratory, Berkeley, CA 94720; and ^gDepartment of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720

Contributed by Bin Yu, December 21, 2017 (sent for review June 23, 2017; reviewed by Michael M. Hoffman and Daniel Jacobson)

Genomics has revolutionized biology, enabling the interrogation of whole transcriptomes, genome-wide binding sites for proteins, and many other molecular processes. However, individual genomic assays measure elements that interact in vivo as components of larger molecular machines. Understanding how these high-order interactions drive gene expression presents a substantial statistical challenge. Building on random forests (RFs) and random intersection trees (RITs) and through extensive, biologically inspired simulations, we developed the iterative random forest algorithm (iRF). iRF trains a feature-weighted ensemble of decision trees to detect stable, high-order interactions with the same order of computational cost as the 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*, among the 20 pairwise transcription factor interactions iRF identifies as stable (returned in more than half of bootstrap replicates), 80% have been previously reported as physical interactions. Moreover, third-order interactions, e.g., between *Zelda* (*Zld*), *Giant* (*Gt*), and *Twist* (*Tw*), suggest high-order relationships that are candidates for follow-up experiments. In human-derived cells, iRF rediscovered a central role of H3K36me3 in chromatin-mediated splicing regulation and identified interesting fifth- and sixth-order interactions, indicative of multivalent nucleosomes with specific roles in splicing regulation. By decoupling the order of interactions from the computational cost of identification, iRF opens additional avenues of inquiry into the molecular mechanisms underlying genome biology.

high-order interaction | random forests | stability | interpretable machine learning | genomics

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 the 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 (1–6). A central challenge with these data lies in the fact that chromatin immunoprecipitation sequencing (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 (5, 7); for a review see ref. 8. While several methods have been developed to identify interactions in large genomics datasets, for example refs. 9–11, 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 paper, we present a computationally efficient tool for directly identifying high-order interac-

tions 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 (*SI Appendix, section S4*). 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 (12), random forests (RFs) (13), Node Harvest (14), Forest Garrote (15), and Rulefit3 (16), as well as methods more specific to gene–gene interactions with categorical features, such as logic regression (17), multifactor dimensionality reduction (18), and Bayesian epistasis mapping (19). 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 toward 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) (20). This procedure

Significance

We developed a predictive, stable, and interpretable tool: the iterative random forest algorithm (iRF). iRF discovers high-order interactions among biomolecules with the same order of computational cost as random forests. We demonstrate the efficacy of iRF by finding known and promising interactions among biomolecules, of up to fifth and sixth order, in two data examples in transcriptional regulation and alternative splicing.

Author contributions: S.B., K.K., J.B.B., and B.Y. designed research, performed research, contributed analytic tools, analyzed data, and wrote the paper.

Reviewers: M.M.H., Princess Margaret Cancer Center; and D.J., Oak Ridge National Laboratory.

The authors declare no conflict of interest.

This open access article is distributed under [Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 \(CC BY-NC-ND\)](#).

¹S.B. and K.K. contributed equally to this work.

²To whom correspondence may be addressed. Email: binyu@stat.berkeley.edu or jbbrown@lbl.gov.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1711236115/-DCSupplemental.

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 (21), 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 (12) 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 (22, 23). Using empirical and numerical examples, we show that iRF is competitive with RF in terms of predictive accuracy and extracts both known and compelling candidate interactions in two motivating biological problems in epigenomics and transcriptomics. An open-source R implementation of iRF is available through CRAN (<https://cran.r-project.org/web/packages/iRF/index.html>).

Our Method: Iterative RFs

The iRF algorithm searches for high-order feature interactions in three steps. First, iterative feature reweighting 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 the RF through a generalization of the RIT, a computationally efficient algorithm that searches for high-order interactions in binary data (20). Finally, a bagging step assesses the stability of recovered interactions with respect to the bootstrap perturbation of the data. We briefly review the 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 nonnegative weights $w = (w_1, \dots, w_p)$, 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 ref. 24 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 $RF(1/p, \dots, 1/p)$.

iRF builds upon a generalization of the RIT, an algorithm that performs a randomized search for high-order interactions among binary features in a deterministic setting. More precisely, the 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 (20) for a complete description.

Consider a binary classification problem with n observations and p binary features. Suppose we are given data in the form (\mathcal{I}_i, Z_i) , $i = 1, \dots, n$. Here, each $Z_i \in \{0, 1\}$ is a binary label and $\mathcal{I}_i \subseteq \{1, 2, \dots, p\}$ is a feature-index subset indicating the indexes of “active” features associated with observation i . In the context of gene transcription, \mathcal{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 $S \subseteq \{1, \dots, p\}$ in the class $C \in \{0, 1\}$ is defined as

$$\mathbb{P}_n(S|Z = C) := \frac{\sum_{i=1}^n \mathbb{1}(S \subseteq \mathcal{I}_i)}{\sum_{i=1}^n \mathbb{1}(Z_i = C)},$$

where \mathbb{P}_n denotes the empirical probability distribution and $\mathbb{1}(\cdot)$ the indicator function. For given thresholds $0 \leq \theta_0 < \theta_1 \leq 1$, the RIT performs a randomized search for interactions S satisfying

$$\mathbb{P}_n(S|Z = 1) \geq \theta_1, \quad \mathbb{P}_n(S|Z = 0) \leq \theta_0. \quad [1]$$

For each class $C \in \{0, 1\}$ and a prespecified integer D , let j_1, \dots, j_D be randomly chosen indexes from the set of observations $\{i : Z_i = C\}$. To search for interactions S satisfying condition 1, the RIT takes D -fold intersections $\mathcal{I}_{j_1} \cap \mathcal{I}_{j_2} \cap \dots \cap \mathcal{I}_{j_D}$ from the randomly selected observations in class C . To reduce computational complexity, these interactions are performed in a tree-like fashion (SI Appendix, section S1, Algorithm 1), where each nonleaf node has n_{child} children. This process is repeated M times for a given class C , resulting in a collection of survived interactions $\mathcal{S} = \bigcup_{m=1}^M \mathcal{S}_m$, where each \mathcal{S}_m is the set of interactions that remains following the D -fold intersection process in tree $m = 1, \dots, M$. The prevalences of interactions across different classes are subsequently compared using condition 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.

iRFs. The iRF algorithm places interaction discovery in a supervised learning framework to identify class-specific, active index sets required for the 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 \mathcal{D} in the form $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with continuous or categorical features $\mathbf{x} = (x_1, \dots, x_p)$, and a binary label $y \in \{0, 1\}$. Our goal is to find subsets $S \subseteq \{1, \dots, p\}$ 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 \mathcal{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 the RIT. We remark on our particular choices in SI Appendix, section S2.

Generalized RIT (Through an RF). For each tree $t = 1, \dots, T$ in the output tree ensemble of an RF, we collect all leaf nodes and index them by $j_t = 1, \dots, J(t)$. Each feature–response pair (\mathbf{x}_i, y_i) is represented with respect to a tree t by $(\mathcal{I}_{i_t}, Z_{i_t})$, where \mathcal{I}_{i_t} is the set of unique feature indexes falling on the path of the leaf node containing (\mathbf{x}_i, y_i) in the t th tree. Hence, each (\mathbf{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

$$\mathcal{R} = \{(\mathcal{I}_{i_t}, Z_{i_t}) : \mathbf{x}_i \text{ falls in leaf node } i_t \text{ of tree } t\} \quad [2]$$

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 Fig. 1 and the complete workflow is presented in SI Appendix, section S1, Algorithm 2. We remark on the algorithm further in SI Appendix, section S2.

1) Iteratively reweighted RF. Given an iteration number K , iRF iteratively grows K feature-weighted RFs $RF(w^{(k)})$, $k = 1, \dots, K$, on the data \mathcal{D} . The first iteration of iRF ($k = 1$) starts with $w^{(1)} := (1/p, \dots, 1/p)$ and stores the importance

Case Study I: Enhancer Elements in *Drosophila*

Development and function in multicellular organisms rely on precisely regulated spatiotemporal 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 (28). In the early *Drosophila* embryo, a small cohort of ~40 TFs drive patterning (for a review see ref. 29), 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 (30) and 13 histone modifications (6), as well as labels of enhancer status for 7,809 genomic sequences in blastoderm (stage 5) *Drosophila* embryos (1, 31). See *SI Appendix, section S3* 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 (3,912 training, 3,897 test). We achieved an area under the precision-recall curve (AUC-PR) on the held-out test data of 0.5 for $K = 5$ (Fig. 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.

Fig. 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 pair-

wise TF interactions, 80% of our findings with stability score >0.5 have been previously reported as physical (*SI Appendix, section S4 and Table S1*). 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 (32), 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 (33, 34), and there is evidence to suggest that *Zld* facilitates binding to regulatory elements (35). 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 (36) and is broadly consistent with the idea that *Zld* is potentiating, rather than an activating factor (35).

More broadly, response surfaces associated with stable high-order interactions indicate AND-like rules (Fig. 2C). In other words, the proportion of active enhancers is substantially higher for sequences where all TFs are sufficiently bound, compared with sequences where only some of the TFs exhibit high levels of occupancy. Fig. 2C demonstrates a putative third-order interaction found by iRF ($sta(Kr-Gt-Zld) = 0.7$). In Fig. 2C, *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 in Fig. 2C, *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 Fig. 2D, we show the subset of sequences that correspond to this AND rule (highlighted in red), using a superheat map (37), which juxtaposes two separately clustered heat maps 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 (30), 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 ref. 38), and our results indicate that they are broadly important for the establishment of early zygotic transcription and therefore body patterning.

Case Study II: Alternative Splicing in a Human-Derived Cell Line

In eukaryotes, alternative splicing of primary messenger RNA (mRNA) transcripts is a highly regulated process in which multiple distinct mRNAs are produced by the same gene. In the case of 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) (39, 40), 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.

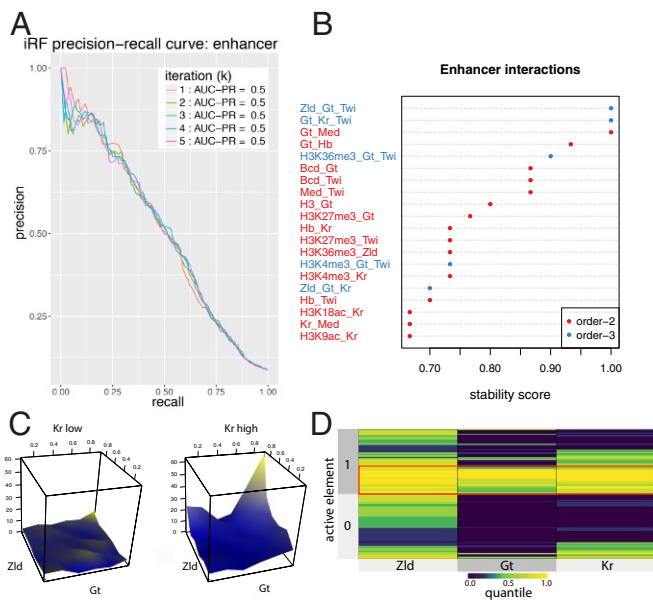


Fig. 2. (A) Accuracy of iRF (AUC-PR) in predicting active elements from TF binding and histone modification data. (B) The 20 most stable interactions recovered by iRF after five iterations. Interactions that are a strict subset of another interaction with stability score ≥ 0.5 have been removed for cleaner 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, the proportion of active enhancers does not change with *Gt* and *Zld*. On the subset of data with *Kr* binding above the median level, the 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.

Learning interactions directly from genomics data has the potential to accelerate our pace of discovery in the study of co- and posttranscriptional 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 (41–43). However, the extent to which chromatin state and DNA-binding factors interact *en masse* to modulate cotranscriptional splicing remains unknown (44). To identify interactions that form the basis of chromatin-mediated splicing, we used iRF to predict thresholded splicing rates for 23,823 exons [RNA-seq percent-spliced-in (PSI) values (<https://github.com/guigolab/ipsa-nf>); 11,911 training, 11,912 test], from ChIP-seq assays measuring enrichment of chromatin marks and TF-binding events (253 ChIP assays on 107 unique TFs and 11 histone modifications). Preprocessing methods are described in *SI Appendix, section S3*.

In this prediction problem, we achieved an AUC-PR on the held-out test data of 0.51 for $K = 2$ (Fig. 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. Fig. 3B reports stability scores of recovered interactions for $K = 2$. We find interactions involving H3K36me3, a number of interactions involving other chromatin marks, and posttranslationally 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, Fig. 3B and 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 six of these elements, indicating a potential AND-type rule related to splicing events (Fig. 3C). This observation is consistent with, and offers a quantitative model for, the previously reported predominance of cotranscriptional splicing in this cell line (45). 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.

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 promising interactions in early zygotic enhancer activation in the *Drosophila* embryo and posits more high-order interactions in splicing regulation for a human-derived system.

Validation and assessment of complex interactions in biological systems are 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 posttranslational modifications to histones (46). This may allow for tests to determine whether modifications to distinct residues at multivalent nucleosomes function in a nonadditive fashion in splicing regulation. Several of the histone marks that appear in the interactions we report, including H3K36me3 and H4K20me1, have been previously identified (47) 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 (48).

We have offered evidence that iRF constitutes a useful tool for generating 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 vs. nonadditive. 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 ref. 49 as a second consideration to work toward 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.

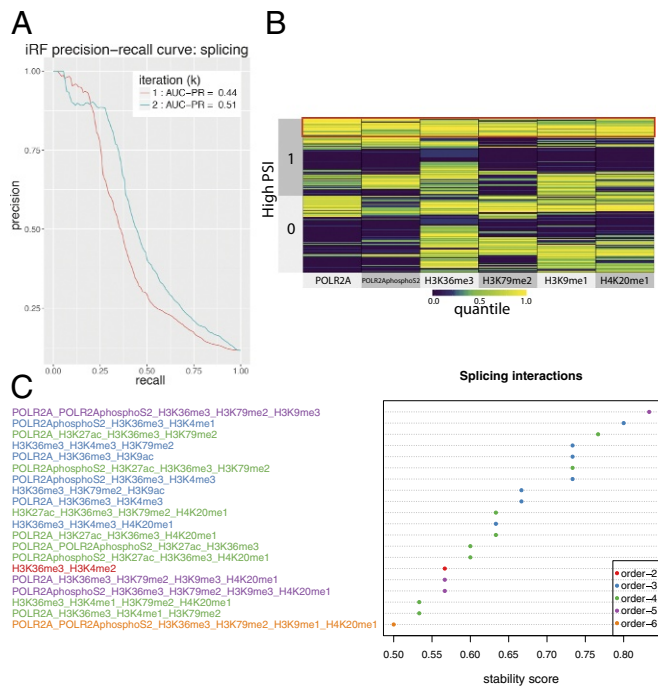


Fig. 3. (A) Accuracy of iRF (AUC-PR) in classifying included exons from excluded exons in held-out test data. iRF shows 7% increase in AUC-PR over RF. (B) An order-6 interaction recovered by iRF (stability score 0.5) displayed on a superheat map which juxtaposes two separately clustered heat maps of exons with high and low splicing rates. Coenrichment of all six 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) The 20 most stable interactions recovered in the second iteration of iRF. Interactions that are a strict subset of another interaction with stability score ≥ 0.5 have been removed for cleaner visualization.

ACKNOWLEDGMENTS. We thank P. Bickel and S. Shrotiya for helpful comments, T. Arbel for preparing the *Drosophila* dataset, and S. Celniker for help in vetting the *Drosophila* data and for consultation on TF interactions. This research was supported in part by Grants National Human Genome Research Institute (NHGRI) U01HG007031, Army Research Office W911NF1710005, Office of Naval Research N00014-16-1-2664, Department of Energy (DOE) DE-AC02-05CH11231, NHGRI R00 HG006698, DOE (SBIR/STTR) Award DE-SC0017069, and National Science Foundation (NSF) DMS-1613002. We thank the Center for Science of Information, a US

NSF Science and Technology Center, under Grant CCF-0939370. Research reported in this publication was supported by the National Library of Medicine of the NIH under Award T32LM012417. B.Y. acknowledges support from the Miller Institute for her Miller Professorship in 2016–2017. S.B. acknowledges the support of University of California, Berkeley, and Lawrence Berkeley National Laboratory, where he conducted work on this paper as a postdoc. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

1. Fisher WW, et al. (2012) DNA regions bound at low occupancy by transcription factors do not drive patterned reporter gene expression in *Drosophila*. *Proc Natl Acad Sci USA* 109:21330–21335.
2. Thomas S, et al. (2011) Dynamic reprogramming of chromatin accessibility during *Drosophila* embryo development. *Genome Biol* 12:R43.
3. Li XY, et al. (2008) Transcription factors bind thousands of active and inactive regions in the *Drosophila* blastoderm. *PLoS Biol* 6:e27.
4. Breeze CE, et al. (2016) eFORGE: A tool for identifying cell type-specific signal in epigenomic data. *Cell Rep* 17:2137–2150.
5. Hoffman MM, et al. (2012) Integrative annotation of chromatin elements from encode data. *Nucleic Acids Res* 41:827–841.
6. ENCODE Project Consortium (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature* 489:57–74.
7. Dong X, et al. (2012) Modeling gene expression using chromatin features in various cellular contexts. *Genome Biol* 13:1–17.
8. Hota SK, Bruneau BG (2016) ATP-dependent chromatin remodeling during mammalian development. *Development* 143:2882–2897.
9. Zhou J, Troyanskaya OG (2014) Global quantitative modeling of chromatin factor interactions. *PLoS Comput Biol* 10:e1003525.
10. Lundberg SM, et al. (2016) Chromnet: Learning the human chromatin network from all ENCODE ChIP-seq data. *Genome Biol* 17:82.
11. Yoshida K, Yoshimoto J, Doya K (2017) Sparse kernel canonical correlation analysis for discovery of nonlinear interactions in high-dimensional data. *BMC Bioinformatics* 18:108.
12. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) *Classification and Regression Trees* (CRC Press, Belmont, CA).
13. Breiman L (2001) Random forests. *Machine Learn* 45:5–32.
14. Meinshausen N (2010) Node harvest. *Ann Appl Stat* 4:2049–2072.
15. Meinshausen N (2009) Forest garrote. *Electron J Stat* 3:1288–1304.
16. Friedman JH, Popescu BE (2008) Predictive learning via rule ensembles. *Ann Appl Stat* 2:916–954.
17. Ruczinski CKI, LeBlanc ML, Hsu L (2001) Sequence analysis using logic regression. *Genet Epidemiol* 21:5626–5631.
18. Ritchie MD, et al. (2001) Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *Am J Hum Genet* 69:138–147.
19. Zhang Y, Liu JS (2007) Bayesian inference of epistatic interactions in case-control studies. *Nat Genet* 39:1167–1173.
20. Shah RD, Meinshausen N (2014) Random intersection trees. *J Machine Learn Res* 15:629–654.
21. Li G, et al. (2012) Extensive promoter-centered chromatin interactions provide a topological basis for transcription regulation. *Cell* 148:84–98.
22. Landt SG, et al. (2012) ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res* 22:1813–1831.
23. Li Q, Brown JB, Huang H, Bickel PJ (2011) Measuring reproducibility of high-throughput experiments. *Ann Appl Stat* 5:1752–1779.
24. Amaratunga D, Cabrera J, Lee YS (2008) Enriched random forests. *Bioinformatics* 24:2010–2014.
25. Anaissi A, Kennedy PJ, Goyal M, Catchpole DR (2013) A balanced iterative random forest for gene selection from microarray data. *BMC Bioinformatics* 14:261.
26. Breiman L (1996) Bagging predictors. *Machine Learn* 24:123–140.
27. Nelson DL, Lehninger AL, Cox MM (2008) *Lehninger Principles of Biochemistry* (Macmillan, New York).
28. Levine M (2010) Transcriptional enhancers in animal development and evolution. *Curr Biol* 20:R754–R763.
29. Rivera-Pomar R, Jäckle H (1996) From gradients to stripes in *Drosophila* embryogenesis: Filling in the gaps. *Trends Genet* 12:478–483.
30. MacArthur S, et al. (2009) Developmental roles of 21 *Drosophila* transcription factors are determined by quantitative differences in binding to an overlapping set of thousands of genomic regions. *Genome Biol* 10:R80.
31. Berman BP, et al. (2002) Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the *Drosophila* genome. *Proc Natl Acad Sci USA* 99:757–762.
32. Nüsslein-Volhard C, Wieschaus E (1980) Mutations affecting segment number and polarity in *Drosophila*. *Nature* 287:795–801.
33. Liang HL, et al. (2008) The zinc-finger protein Zelda is a key activator of the early zygotic genome in *Drosophila*. *Nature* 456:400–403.
34. Harrison MM, Li XY, Kaplan T, Botchan MR, Eisen MB (2011) Zelda binding in the early *Drosophila melanogaster* embryo marks regions subsequently activated at the maternal-to-zygotic transition. *PLoS Genet* 7:e1002266.
35. Foo SM, et al. (2014) Zelda potentiates morphogen activity by increasing chromatin accessibility. *Curr Biol* 24:1341–1346.
36. Xu Z, et al. (2014) Impacts of the ubiquitous factor Zelda on Bicoid-dependent DNA binding and transcription in *Drosophila*. *Genes Dev* 28:608–621.
37. Barter RL, Yu B (2015) Superheat: Supervised heatmaps for visualizing complex data. arXiv:1512.01524.
38. Levine M (2013) Computing away the magic? *eLife* 2:e01135.
39. So BR, et al. (2016) A U1 snRNP-specific assembly pathway reveals the SMN complex as a versatile hub for RNP exchange. *Nat Struct Mol Biol* 23:225–230.
40. Stoiber MH, et al. (2015) Extensive cross-regulation of post-transcriptional regulatory networks in *Drosophila*. *Genome Res* 25:1692–1702.
41. Kolasinska-Zwiercz P, et al. (2009) Differential chromatin marking of introns and expressed exons by H3K36me3. *Nat Genet* 41:376–381.
42. Sims III RJ, Reinberg D (2009) Processing the H3K36me3 signature. *Nat Genet* 41:270–271.
43. Kornblihtt AR (2012) CTCF: From insulators to alternative splicing regulation. *Cell Res* 22:450–452.
44. Allemand E, et al. (2016) A broad set of chromatin factors influences splicing. *PLoS Genet* 12:e1006318.
45. Tilgner H, et al. (2012) Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs. *Genome Res* 22:1616–1625.
46. Hilton IB, et al. (2015) Epigenome editing by a CRISPR-Cas9-based acetyltransferase activates genes from promoters and enhancers. *Nat Biotechnol* 33:510–517.
47. Hallmann RS, Schneeweiss LG, Correa E, Zamora J (1998) Fine needle aspiration biopsy of thymic carcinoid tumor: A case with immunocytochemical correlation. *Acta Cytol* 42:1042–1043.
48. Weiner A, et al. (2016) Co-ChIP enables genome-wide mapping of histone mark co-occurrence at single-molecule resolution. *Nat Biotechnol* 34:953–961.
49. Yu B (2013) Stability. *Bernoulli* 19:1484–1500.

Supporting Information Appendix

Contents

S1 Algorithms	3
S2 Remarks on iRF	4
S2.1 Iterative re-weighting	4
S2.2 Generalized RIT	4
S2.3 Node sampling	4
S2.4 Bagged stability scores	4
S2.5 Relation to AdaBoost	5
S2.6 Sensitivity to tuning parameters	5
S2.7 Regression and multiclass classification	5
S2.8 Grouped features and replicate assays	6
S2.9 Interaction evaluation through prediction	6
S3 Data processing	8
S3.1 <i>Drosophila</i> enhancers	8
S3.2 Alternative splicing	8
S4 Evaluating <i>Drosophila</i> enhancer interactions	8
S5 Simulation experiments	9
S5.1 Simulation 1: Boolean rules	10
S5.2 Simulation 2: marginal importance	11
S5.2.1 Noise level	11
S5.2.2 Mixture model	12
S5.2.3 Correlated features	13
S5.3 Simulation 3: big p	13
S5.4 Simulation 4: enhancer data	13
S6 Computational cost of detecting high-order interaction	14
S7 List of datasets	16

List of Figures

S1	Enhancer prediction tuning	17
S2	Splicing prediction tuning	18
S3	Enhancer interaction recovery tuning	19
S4	Splicing interaction recovery tuning	20
S5	Prediction-based interaction importance: enhancer data	21
S6	Prediction-based interaction importance: splicing data	22
S7	iRF performance for order-4 AND, OR, and XOR rules	23
S8	iRF performance for order-8 XOR rule	24
S9	Order-8 XOR rule weight distribution by iteration	25
S10	Feature entry depth by marginal importance	26
S11	iRF performance for mixture of order-8 XOR rule and order-4 AND rule	27
S12	iRF performance for order-8 XOR rule, decaying covariance	28
S13	iRF performance for order-8 XOR rule, block covariance	29
S14	iRF performance for order-4 AND rule, noise level 0.2	30
S15	iRF performance for order-4 AND rule, noise level 0.3	31
S16	iRF performance for enhancer data simulation	32
S17	iRF enhancer data simulation stability scores	33
S18	Runtime comparison of iRF and Rulefit	34

S1 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 (Shah and Meinshausen, 2014). For a full description of iRF, we refer readers to Section 2.

Algorithm 1: Random Intersection Trees [Shah and Meinshausen \(2014\)](#)

Input: $\{(\mathcal{I}_i, Z_i); \mathcal{I}_i \subseteq \{1, \dots, p\}, Z_i \in \{0, 1\}\}_{i=1}^n, C \in \{0, 1\}$
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 $0, \dots, D - 1$ 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 $j = 1, \dots, J$, let i_j be a uniform sample from the set of class C observations $\{i : Z_i = C\}$.
- 3 Set $S_1 = \mathcal{I}_{i_1}$
- 4 **for** $j = 2$ **to** J **do**
- 5 $S_j \leftarrow \mathcal{I}_{i_j} \cap S_{pa(j)}$
- 6 **end**
- 7 **return** $S_m = \{S_j : depth(j) = D\}$
- 8 **end**

Output: $S = \cup_{m=1}^M S_m$

Algorithm 2: iterative Random Forests

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

- 1 (1) **for** $k \leftarrow 1$ **to** K **do**
- 2 Fit $RF(w^{(k)})$ on \mathcal{D}
- 3 $w^{(k+1)} \leftarrow$ Gini importance of $RF(w^{(k)})$
- 4 **end**
- 5 (2) **for** $b \leftarrow 1$ **to** B **do**
- 6 Generate bootstrap samples $\mathcal{D}_{(b)}$ of the form $\{\mathbf{x}_{b(i)}, y_{b(i)}\}$ from \mathcal{D}
- 7 Fit $RF(w^{(K)})$ on $\mathcal{D}_{(b)}$
- 8 $\mathcal{R}_{(b)} \leftarrow \{(\mathcal{I}_{i_t}, Z_{i_t}) : \mathbf{x}_{b(i)} \text{ falls in leaf node } i_t \text{ of tree } t\}$
- 9 $S_{(b)} \leftarrow$ RIT($\mathcal{R}_{(b)}, C$)
- 10 **end**
- 11 (3) **for** $S \in \cup_{b=1}^B S_{(b)}$ **do**
- 12 $sta(S) = \frac{1}{B} \cdot \sum_{b=1}^B \mathbb{1}[S \in S_{(b)}]$
- 13 **end**

Output: $\{S, sta(S)\}_{S \in \cup_{b=1}^B S_{(b)}}$
Output: $\{RF(w^{(K)}) \text{ on } \mathcal{D}\}$

S2 Remarks on iRF

S2.1 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 S16).

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 S10). 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 (Deng and Runger, 2012) may also lead to improved interaction recovery, though we do not explore them in this paper.

S2.2 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 \mathbf{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.

S2.3 Node sampling

In the generalized RIT step of iRF, we represent each observation $i = 1, \dots, n$ by T rule-response pairs, determined by the leaf nodes containing observation i in each tree $t = 1, \dots, T$ of an RF. We accomplish this by replicating each rule-response pair $(\mathcal{I}_{j_t}, Z_{j_t})$ 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 \mathcal{D} , made more robust to sampling perturbations through rules derived from bootstrap samples of \mathcal{D} . Our representation is equivalent to sampling rule-response pairs $(\mathcal{I}_{j_t}, Z_{j_t})$ 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.

S2.4 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 \mathbf{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.

S2.5 Relation to AdaBoost

In his original paper on RF (Breiman, 2001), Breiman conjectured that in the later stages of iteration, AdaBoost (Freund and Schapire, 1995) 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.

S2.6 Sensitivity to tuning parameters

The predictive performance of RF is known to be highly robust to choice of tuning parameters (Breiman, 2001). 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. Figures S1 and S2 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 Sections S5.1-S5.4 extensively examine the relationship between the number of iterations and recovered interactions. Figures S3 and S4 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.

S2.7 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.

S2.8 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.

S2.9 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 (Nelson et al., 2008) 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 (Yu, 2013) 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 $S \subseteq \{1, \dots, p\}$ 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 $i = 1, \dots, n$ we evaluate its predicted value from each tree $t = 1, \dots, T$ with respect to an interaction S as

$$\hat{y}_i(t; S) = \begin{cases} Z_{i_t} & \text{if } S \subseteq \mathcal{I}_{i_t} \\ \mathbb{P}_n(y = 1) & \text{else} \end{cases}$$

where Z_{i_t} is the prediction of the leaf node containing observation i in tree t , \mathcal{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

$$\hat{y}_i(S) = \frac{1}{T} \cdot \sum_{t=1}^T \hat{y}_i(t; S). \tag{1}$$

Predictions from equation (1) 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 $S \in \mathcal{S}$ 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 (Breiman, 2001). 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 $X_{\pi_{S^c}}$ denote the feature matrix with all columns in S^c permuted, where S^c is the complement of S . We evaluate predictions on permuted data $X_{\pi_{S^c}}$, 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: Figures S5 and S6 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 S5). 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 S6).

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 $S' \subset S \subseteq \{1, \dots, p\}$. 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.

S3 Data processing

S3.1 *Drosophila* enhancers

In total, 7809 genomic sequences have been evaluated for their enhancer activity (Berman et al., 2002; Fisher et al., 2012; Frise et al., 2010; Kvon et al., 2014) 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 (Tautz and Pfeifle, 1989; Weiszmann et al., 2009). 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 (Li et al., 2008) of ChIP-seq and ChIP-chip assays (MacArthur et al., 2009; ENCODE Project Consortium, 2012) for each genomic segment. The processed data are provided in Supporting Data 1.

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 R `randomForest` package, and 500 Random Intersection Trees of depth 5 with $n_{child} = 2$ were grown to capture candidate interactions.

S3.2 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 (ENCODE Project Consortium, 2012). To identify critical interactions that form the basis of chromatin mediated splicing, we used splicing rates (Percent-spliced-in, PSI values, (Pervouchine et al., 2012, 2016)) 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, <https://www.encodeproject.org/>). A complete description of the assays, including accession numbers, is provided in Supporting Data 2.

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. The processed data are provided in Supporting Data 3.

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 R `randomForest` package, and 500 binary random intersection trees of depth 5 with $n_{child} = 2$ were grown to capture candidate interactions.

S4 Evaluating *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 Rivera-Pomar and Jäckle

(1996) and Jaeger (2011). We compared our results against these previously reported physical interactions to evaluate interactions found by iRF. Table S1 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 S1. 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 (Levine, 2013). These limited cases are not sufficient to conduct a comprehensive analysis of the high-order interactions we identify using iRF.

Table S1: Previously identified pairwise TF interactions recovered by iRF with stability score > 0.5

interaction (S)	$sta(S)$	references
Gt, Zld	1	Harrison et al. (2011); Nien et al. (2011)
Twi, Zld	1	Harrison et al. (2011); Nien et al. (2011)
Gt, Hb	1	Kraut and Levine (1991a,b); Eldon and Pirrotta (1991)
Gt, Kr	1	Kraut and Levine (1991b); Struhl et al. (1992); Capovilla et al. (1992); Schulz and Tautz (1994)
Gt, Twi	1	Li et al. (2008)
Kr, Twi	1	Li et al. (2008)
Kr, Zld	0.97	Harrison et al. (2011); Nien et al. (2011)
Gt, Med	0.97	–
Bcd, Gt	0.93	Kraut and Levine (1991b); Eldon and Pirrotta (1991)
Bcd, Twi	0.93	Li et al. (2008)
Hb, Twi	0.93	Zeitlinger et al. (2007)
Med, Twi	0.93	Nguyen and Xu (1998)
Kr, Med	0.9	–
D, Gt	0.87	–
Med, Zld	0.83	Harrison et al. (2011)
Hb, Zld	0.80	Harrison et al. (2011); Nien et al. (2011)
Hb, Kr	0.80	Nüsslein-Volhard and Wieschaus (1980); Jäckle et al. (1986); Hoch et al. (1991)
D, Twi	0.73	–
Bcd, Kr	0.67	Hoch et al. (1991, 1990)
Bcd, Zld	0.63	Harrison et al. (2011); Nien et al. (2011)

S5 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 (Nelson et al., 2008). In this section, we examine interaction recovery and predictive accuracy of iRF in a variety of simulation settings.

For all simulations in Sections S5.1-S5.3, 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 \mathcal{S} recovered by iRF. For responses generated from an interaction $S^* \subseteq \{1, \dots, p\}$, 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 $s = 2, \dots, |S|$, 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 $\mathcal{S} = \mathcal{S}_T \cup \mathcal{S}_F$ represent the set of interactions recovered by iRF, where \mathcal{S}_T and \mathcal{S}_F are the sets of recovered true and false positive interactions respectively. For a given interaction order $s = 2, \dots, |S|$, we calculate

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

This metric measures the aggregate weight of stability scores for false positive order- s interactions, $S \in \mathcal{S}_F : |S| = s$, relative to all recovered order- s interactions, $S \in \mathcal{S} : |S| = s$. 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.

S5.1 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 $\mathbf{x} = (x_1, \dots, x_{50})$ from independent, standard Cauchy distributions to reflect heavy-tailed data, and generated the binary responses from three rule settings (OR, AND, and XOR) as

$$y^{(OR)} = \mathbb{1}[x_1 > t_{OR} | x_2 > t_{OR} | x_3 > t_{OR} | x_4 > t_{OR}], \quad (2)$$

$$y^{(AND)} = \prod_{i=1}^4 \mathbb{1}[x_i > t_{AND}], \quad (3)$$

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

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 (2), (3), and (4) 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, \dots, 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, \dots , 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, \dots , 1000 observations. We report the prediction accuracy and interaction recovery for iterations $k \in \{1, 2, \dots, 5\}$ 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 (Liaw and Wiener,

2002), $M = 100$ 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 S7A 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 (2), (3), and (4). Iterative re-weighting restricts the form of functions fitted by RF and may hurt predictive performance when the generative model is not sparse.

Figure S7B 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 S7C 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 S7D). 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 S7B).

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) (Lim and Yu, 2015), which would balance trade-offs between interpretability and predictive accuracy.

S5.2 Simulation 2: marginal importance

Section S5.1 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 $\mathbf{x} = (x_1, \dots, x_{100})$ from independent, standard Cauchy distributions, and generated the binary response y as

$$y = \mathbb{1} \left[\sum_{i \in S_{XOR}} \mathbb{1}(x_i > t_{XOR}) \equiv 1 \pmod{2} \right], \quad (5)$$

$S_{XOR} = \{1, \dots, 8\}$. We set $t_{XOR} = 2$, 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 `ntree`, which was set to 200 for computational purposes. We ran iRF for $k \in \{1, \dots, 5\}$ iterations with 10 bootstrap samples and grew $M = 100$ RITs of depth 5 with $n_{child} = 2$. Each simulation was replicated 10 times to evaluate performance stability.

S5.2.1 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 (5), and swapped labels for 10%, 15%, and 20% of randomly selected responses.

Figure S8 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 S8C) reinforces the idea that regularization is critical for recovering high-order interactions. Figure S9 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 S10 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.

S5.2.2 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 (5), and the remaining proportion $1 - \pi$ of observations were generated as

$$y = \prod_{i \in S_{AND}} \mathbb{1}[x_i > t_{AND}]. \tag{6}$$

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 $S_{AND} = \{9, 10, 11, 12\}$, $t_{AND} = -0.5$ 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 S11 shows performance in terms of predictive accuracy (A) and interaction recovery of XOR (B) and AND (C) rules at each level of π . When one rule is clearly dominant (AND: $\pi = 0.5$; XOR: $\pi = 0.9$), iRF fail to recover the the other (Figure S11 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 S11). 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.

S5.2.3 Correlated features

In our next set of simulations, we examined the effect of correlated features on interaction recovery. Responses were generated using equation (5), with features $\mathbf{x} = (x_1, \dots, x_{100})$ drawn from a Cauchy distribution with mean 0 and covariance Σ , and active set S_{XOR} , $|S_{XOR}| = 8$ sampled uniformly at random from $\{1, \dots, 100\}$. 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 $G_l \subseteq \{1, \dots, p\}$ and $l = 1, \dots, L$ partition $\{1, \dots, p\}$ 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 ρ (Figures S12, S13), while interaction recovery degrades for larger values of ρ , particularly in the block covariance setting (Figure S13B,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 ρ , 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 Section S2.8.

S5.3 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), 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.

Figures S14 and S15 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 (Figures S14A, S15A). For $k = 1$, iRF rarely recovers correct interactions and never recovers interactions of order > 2 , while later iterations recover many true interactions (Figures S14C, S15C). 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 $p \gg n$ and interactions have no marginally important features.

S5.4 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 Section 4 (see also Section S3.1 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):

$$y = \mathbb{1}[x_{kr} > 1.25 \& x_{hb} > 1.25 \& x_D > 1.25 \& x_{twi} > 1.25 \& x_{zld} > 75]. \quad (7)$$

The active TFs and thresholds were selected to ensure that the proportion of positive responses was comparable to the true data ($\sim 10\%$ 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 $i = 1, \dots, n$. 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 π_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 (7) was active. This noising procedure introduced an additional $\sim 5\%$ of class 1 observations beyond the $\sim 10\%$ of observations that were class 1 as a result of equation (7). 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 π_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, \dots , 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 R `randomForest` package, $M = 100$ random intersection trees 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 S16A 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 S16). 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 S16C,D). We note that true positive interactions are consistently identified as more stable (Figure S17), suggesting that the large proportion of weighted false discoveries in Figure S16D is the result of many false positives with low stability scores.

S6 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” (Friedman and Popescu, 2008). 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 $p \in \{10, 20, \dots, 80\}$. 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 (Friedman and Popescu, 2008)). 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 S18.

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.

S7 List of datasets

Scripts and data used for the case studies and simulations described in this paper are available on [Zenodo](#).

Scripts

1. `enhancer.R`: R script used to run `iRF` on the enhancer data.
2. `splicing.R`: R script used to run `iRF` on the splicing data.
3. `booleanSimulations.R`: R script used to run `iRF` for boolean generative models (Sections [S5.1-S5.3](#)).
4. `enhancerSimulations.R`: R script used to run `iRF` for enhancer data simulations (Section [S5.4](#)).
5. `runtime/irf.R`: R script used to run the runtime analysis for `iRF` (Section [S6](#)).
6. `runtime/rulefit.R`: R script used to run the runtime analysis for `Rulefit3` (Section [S6](#)).
7. `runtime/rulefit`: R package for running `Rulefit3` ([Friedman and Popescu, 2008](#)). The package we provide is set up for use on linux systems. Other versions are available through [statweb.stanford.edu](#).

Datasets

1. `irfSuppData1.csv`: Processed data for the enhancer case study (Supporting Data 1).
2. `irfSuppData2.csv`: Description of the splicing assays including ENCODE accession number, assay name, and assay type (Supporting Data 2).
3. `irfSuppData3.csv`: Processed data used for the splicing case study (Supporting Data 3).
4. `enhancer.Rdata`: An `Rdata` file containing all variables required to run the `enhancer.R` script:
 - `X`: 7809×80 feature matrix, rows corresponding to genomic regions and columns corresponding to assays.
 - `Y`: length 7809 response vector, 1 indicating active element.
 - `train.id`: length 3912 vector giving the indices of training observations.
 - `test.id`: length 3897 vector giving the indices of testing observations.
 - `varnames.all`: 80×2 data frame, the first column giving a unique identifier for each assay and the second column giving collapsed terms used to group replicate assays.
5. `splice.Rdata`: An `Rdata` file containing all variables required to run the `splicing.R` script:
 - `x`: 23823×270 feature matrix, rows corresponding to exons and columns corresponding to assays.
 - `y`: length 23823 response vector, 1 indicating a highly spliced exon.
 - `train.id`: length 11911 vector giving the indices of training observations.
 - `test.id`: length 11912 vector giving the indices of testing observations.
 - `varnames.all`: 270×2 data frame, the first column giving a unique identifier for each assay and the second column giving collapsed terms used to group replicate assays.
6. `rfSampleSplitNoise.Rdata`: An `Rdata` file containing RF predicted probabilities used for noising the enhancer simulation:
 - `pred.prob`: 7809×20 matrix, giving the predicted probability that each genomic element is active. These probabilities were generated using the sample splitting procedure described in Section [S5.4](#) and used to noise the enhancer simulation.

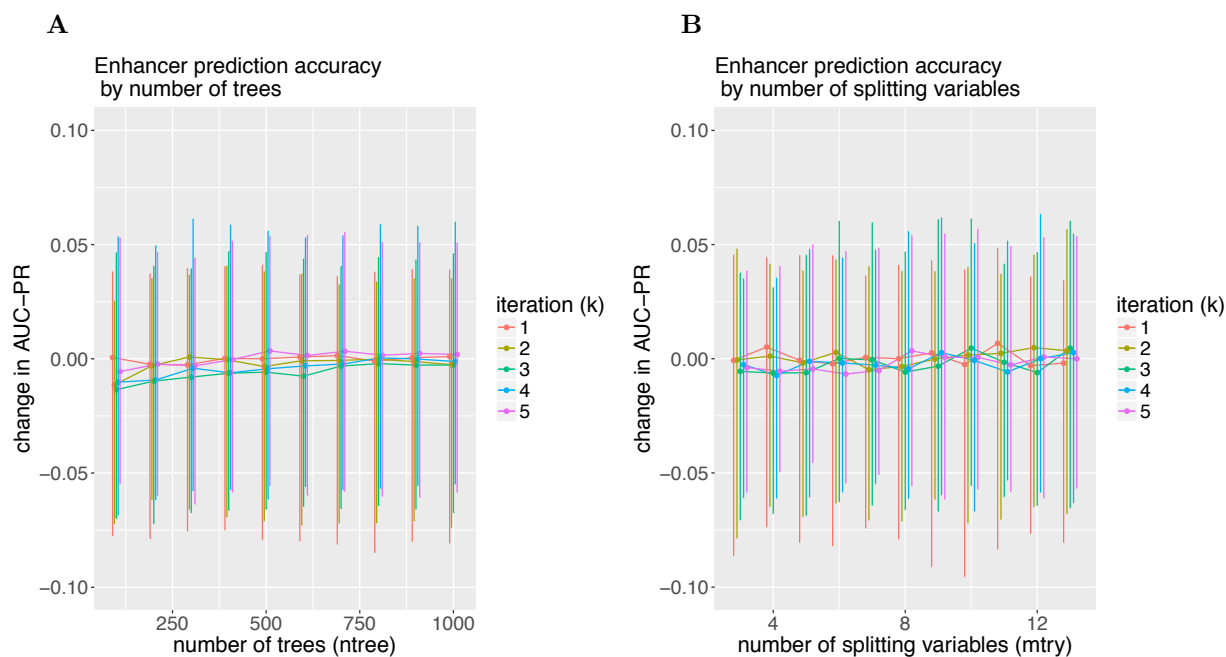


Figure S1: Enhancer data cross-validation AUC-PR change from baseline as a function of RF tuning parameters, evaluated over 5 folds. Baseline performance is given by Random Forest ($k = 1$) with default parameters ($\text{ntree} = 500$, $\text{mtry} = 8$). Error bars indicate the minimum and maximum change in AUC-PR across folds. **[A]** Prediction accuracy as a function of number of trees (ntree), with number of splitting variables (mtry) set to default ($\lfloor \sqrt{p} \rfloor = 8$). **[B]** Prediction accuracy as a function of mtry , with ntree set to default (500).

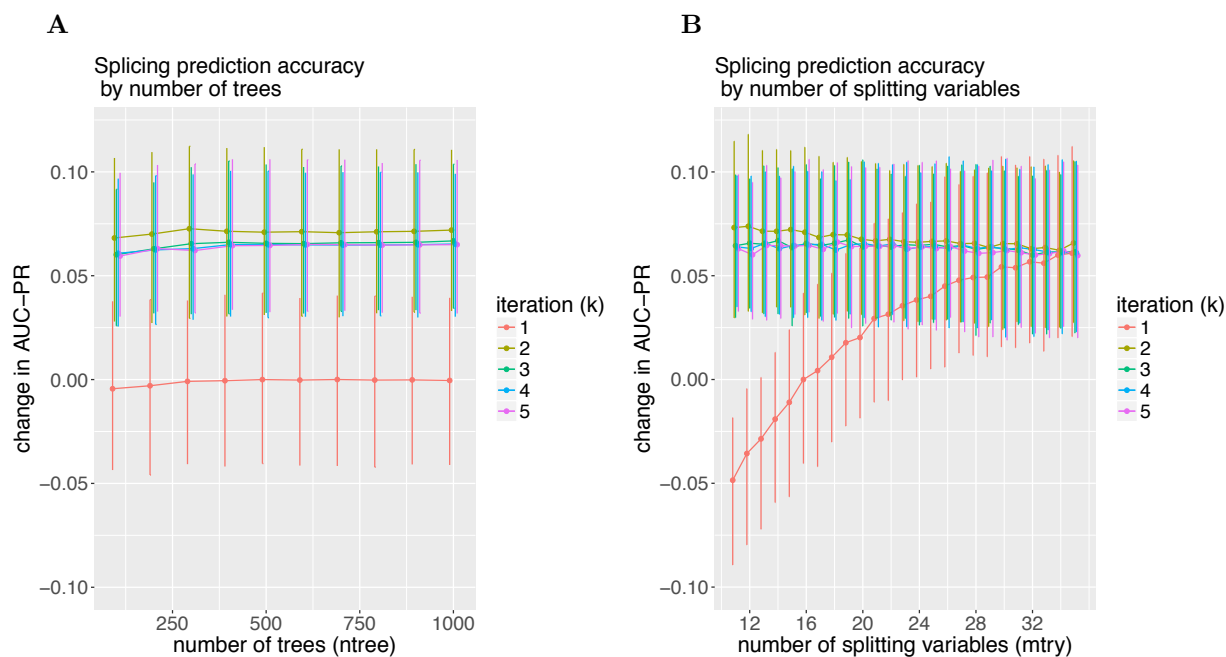


Figure S2: Splicing data cross-validation AUC-PR change from baseline as a function of RF tuning parameters, evaluated over 5-folds. Baseline performance is given by Random Forest ($k = 1$) with default parameters ($\text{ntree} = 500$, $\text{mtry} = 16$). Error bars indicate the minimum and maximum 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 (ntree), with the number of splitting variables (mtry) set to default ($\lfloor \sqrt{p} \rfloor = 16$). **[B]** Prediction accuracy as a function of mtry , with ntree set to default (500).

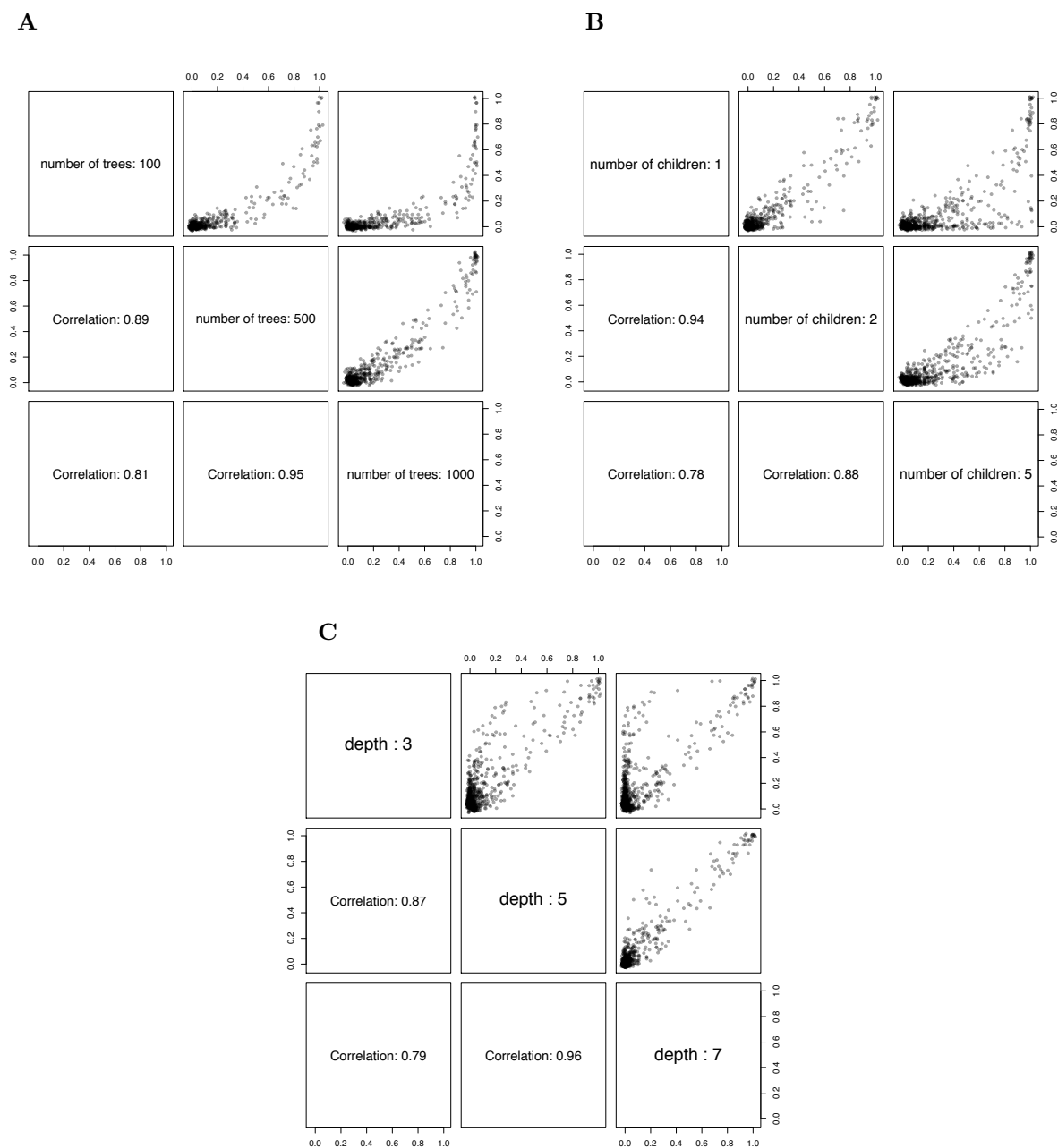


Figure S3: Enhancer data interaction stability scores as a function of RIT parameters. Each point represents a single interaction, and the point's coordinates indicate its stability score under two parameter settings. Lower panels give Pearson correlation between interaction stability scores across pairs 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.

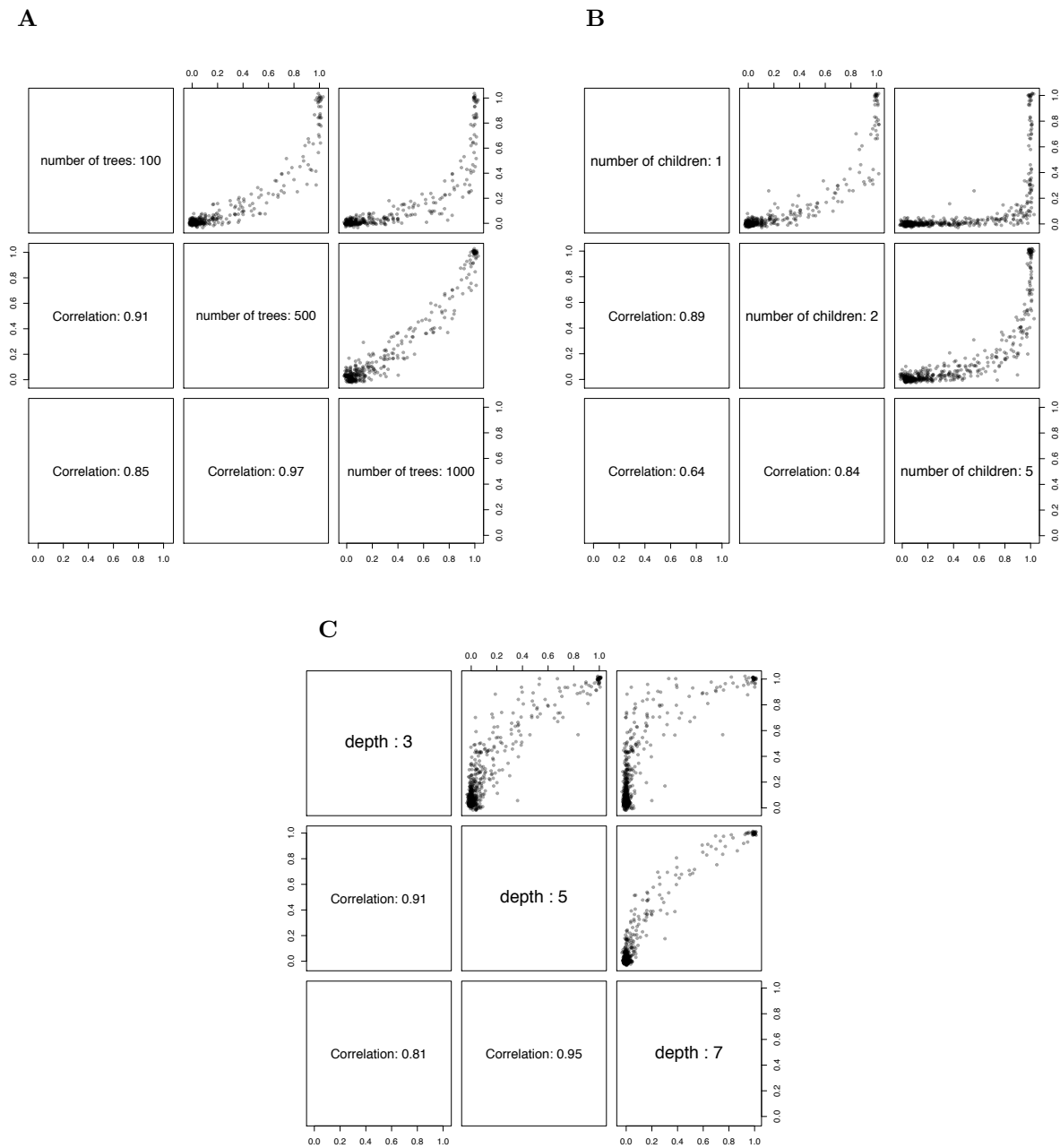
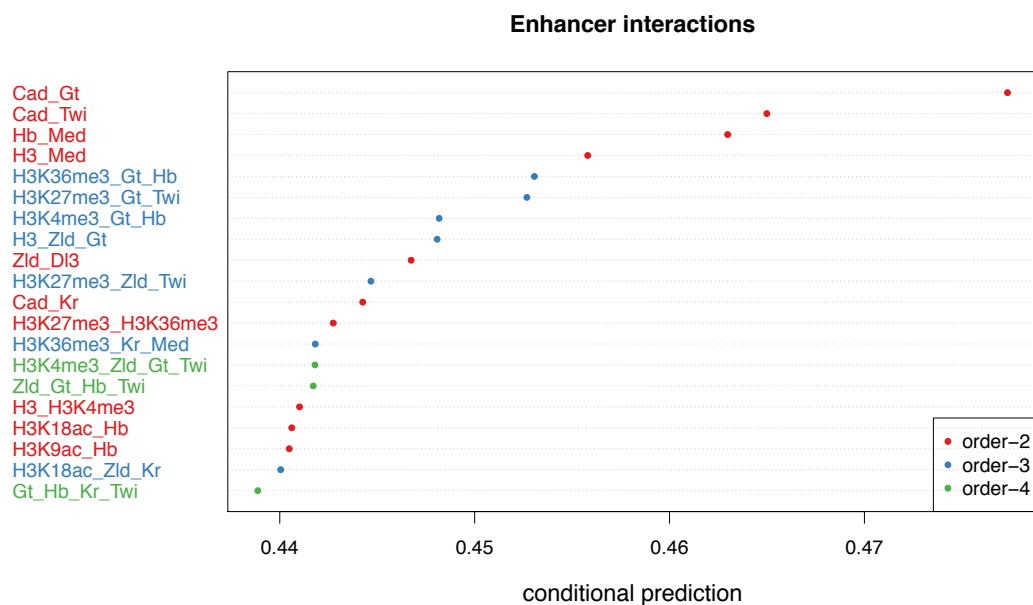


Figure S4: Splicing data interaction stability scores as a function of RIT parameters. Each point represents a single interaction, and the point's coordinates indicate its stability score under two parameter settings. Lower panels give Pearson correlation between interaction stability scores across pairs 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.

A



B

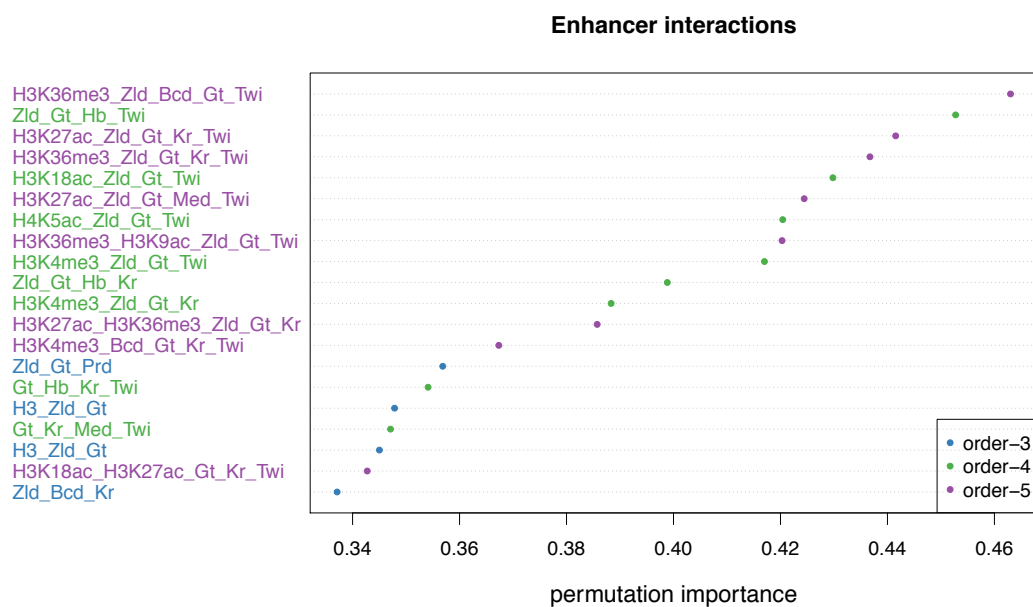
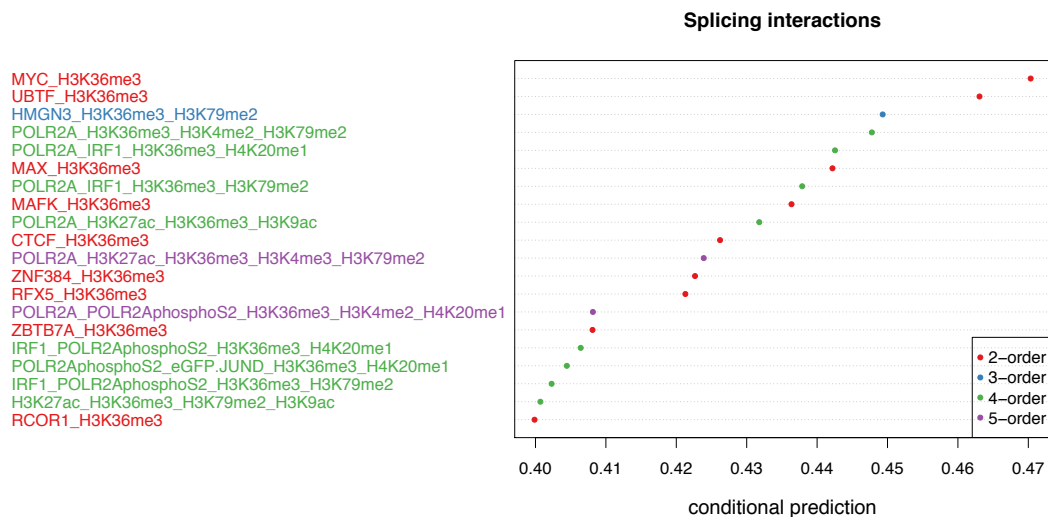


Figure S5: Prediction-based validation metrics for enhancer data. Each plot shows the top 20 interactions with respect to prediction based importance metrics. Lower-order interactions that are a strict subset of some higher-order interactions have been removed for clearer visualization. The interactions reported here are qualitatively similar to those with high stability scores. [A] Conditional prediction. [B] Permutation importance.

A



B

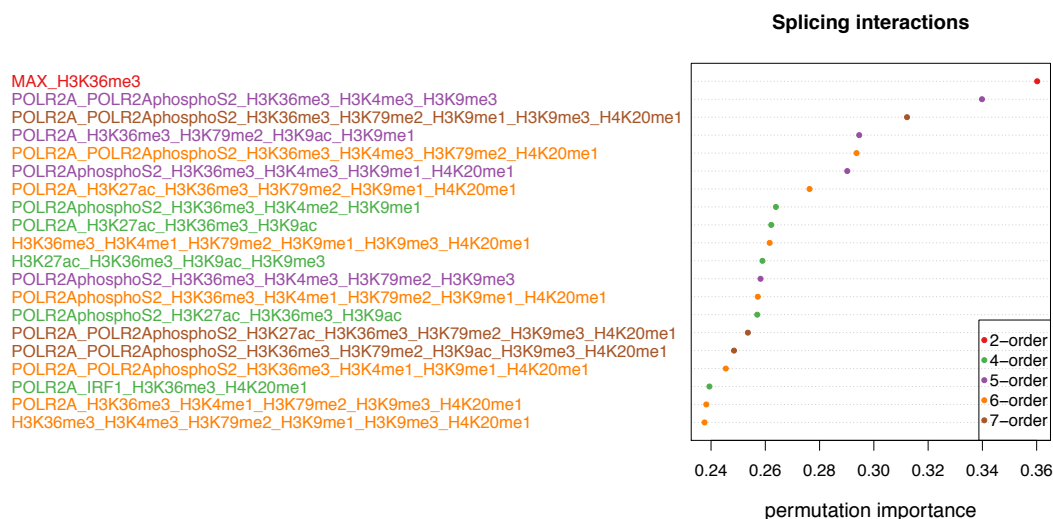


Figure S6: Prediction-based validation metrics for splicing data. Each plot shows the top 20 interactions with respect to prediction based importance metrics. Lower-order interactions that are a strict subset of recovered higher-order interactions have been removed for clearer visualization. **[A]** Conditional prediction. **[B]** Permutation importance. The interactions reported here are qualitatively similar to those with high stability scores.

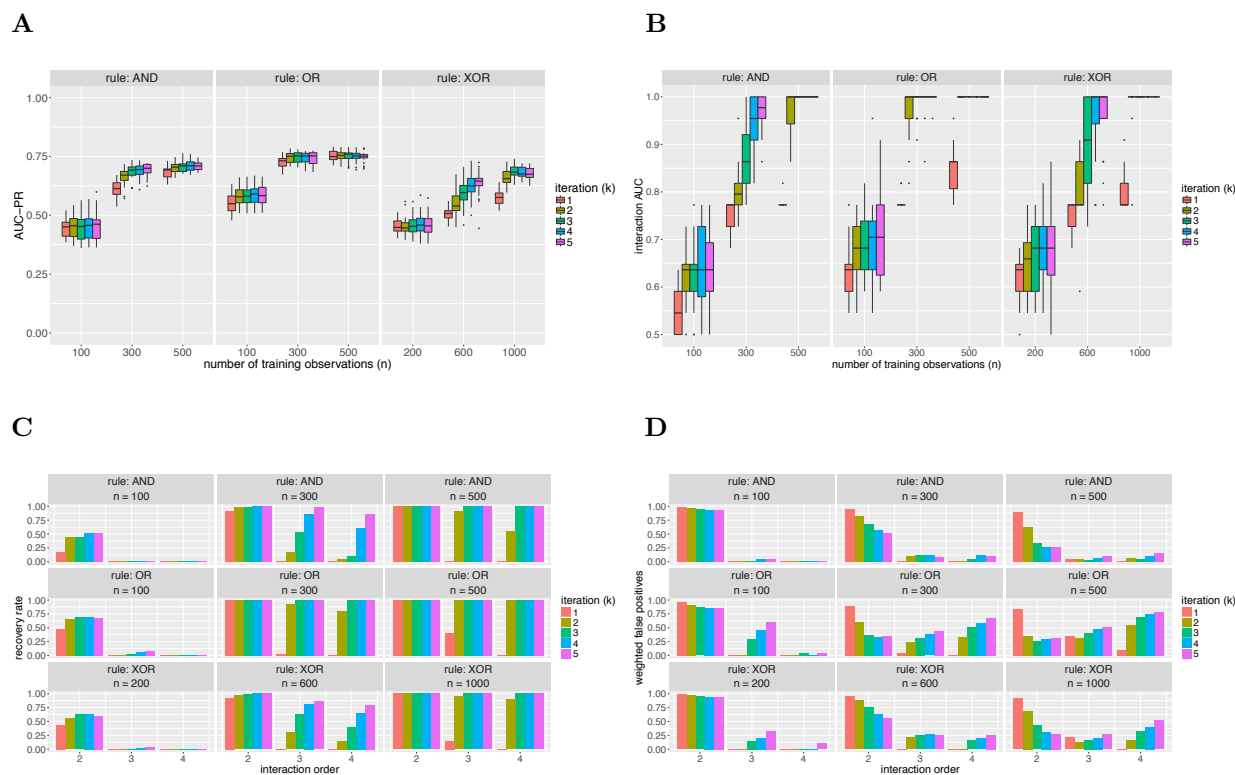


Figure S7: iRF performance for order-4 AND, OR, and XOR rules over 20 replicates. Results are shown for models trained using 100, 300, and 500 observations in the AND and OR models. Training sample size is increased to 200, 600, and 1000 in the XOR model to account for the low marginal importance of 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 increase in settings where iRF recovers high-order interactions as a result of many false positives with low stability scores.

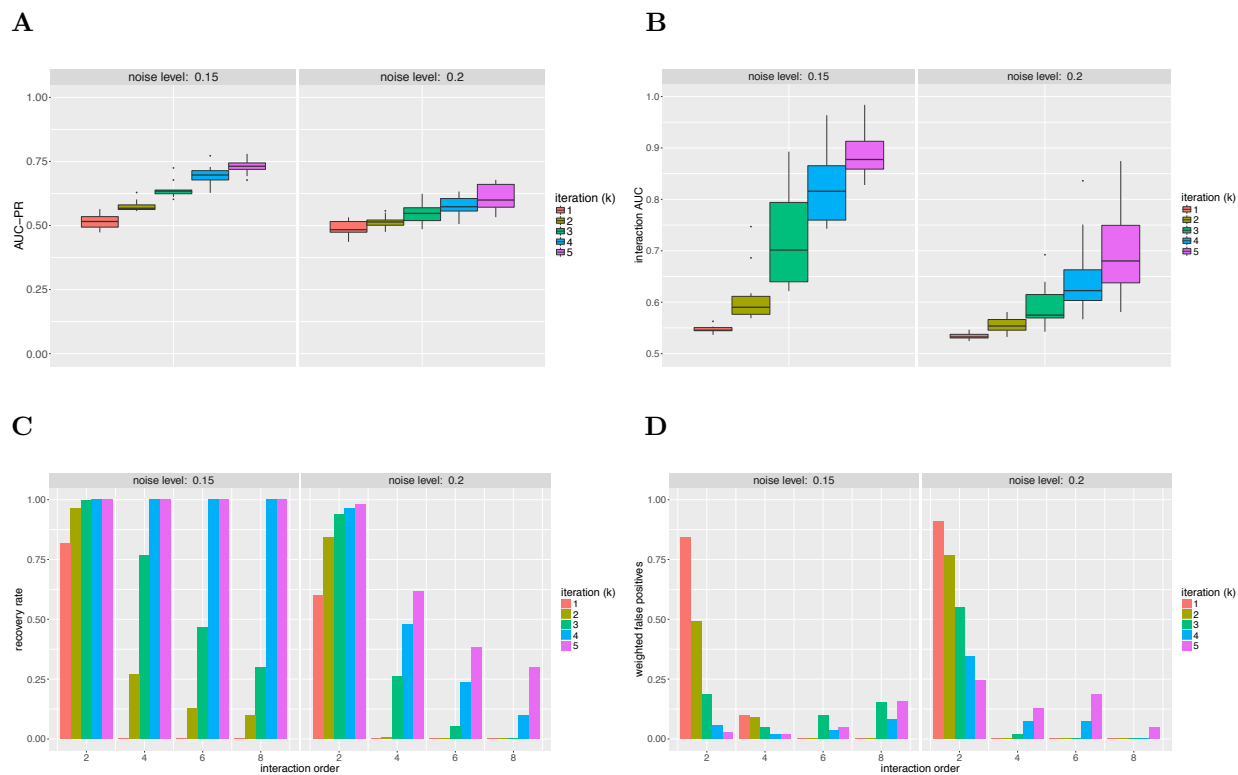


Figure S8: iRF performance for order-8 XOR rule over 10 replicates as a function of noise level. All models were trained using 5,000 observations. **[A]** Prediction accuracy (AUC-PR) improves for increasing k and at a slower rate for increased noise levels. **[B]** Interaction AUC improves with increasing 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 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.

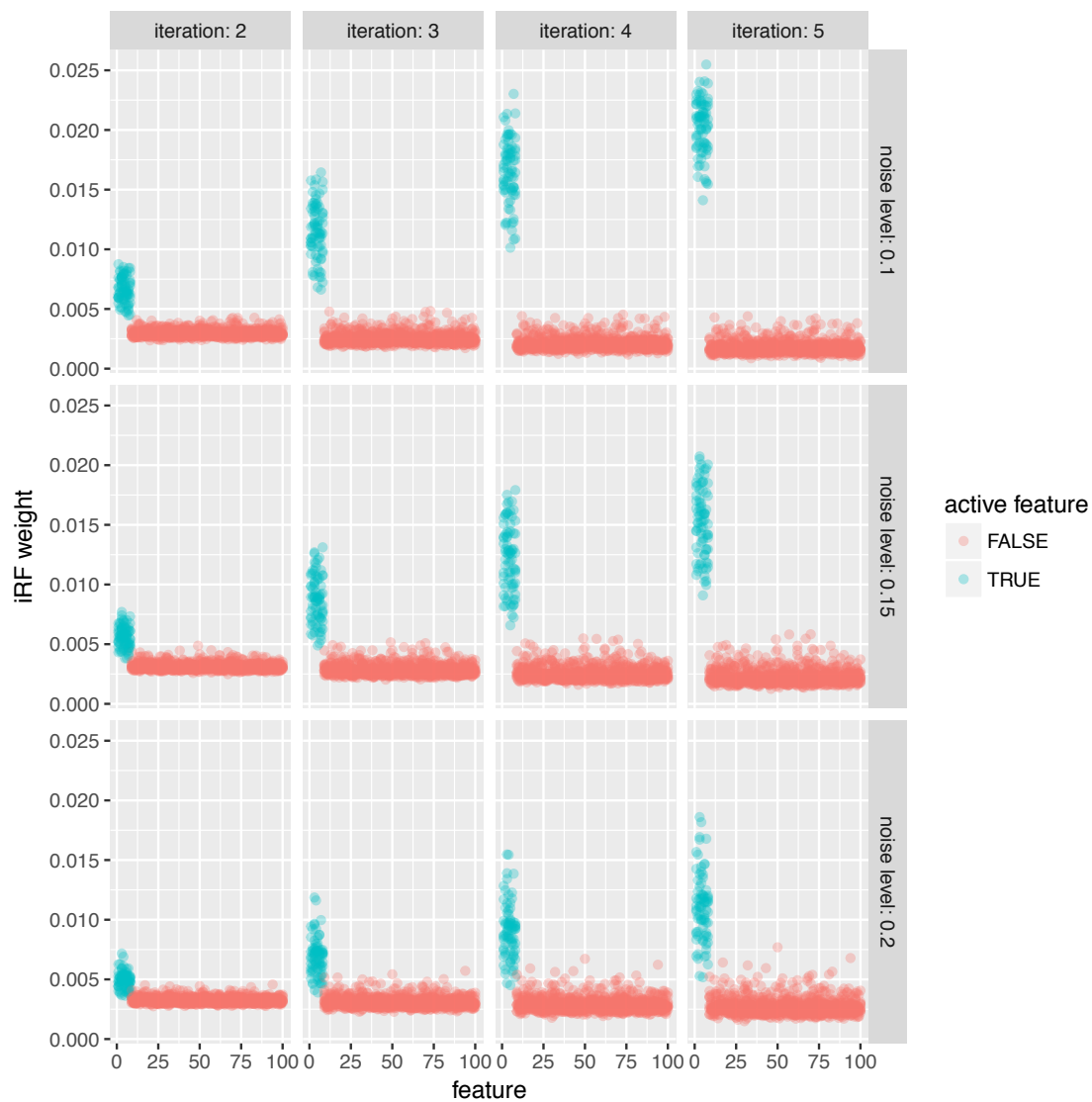


Figure S9: iRF weights for active (blue) and inactive (red) features as a function of iteration and noise level over 10 replicates. The distribution of weights in later iterations shows a clear separation between active and inactive features, indicating that iRF has identified active features as important and incorporates them into the model with higher probability in later iterations.

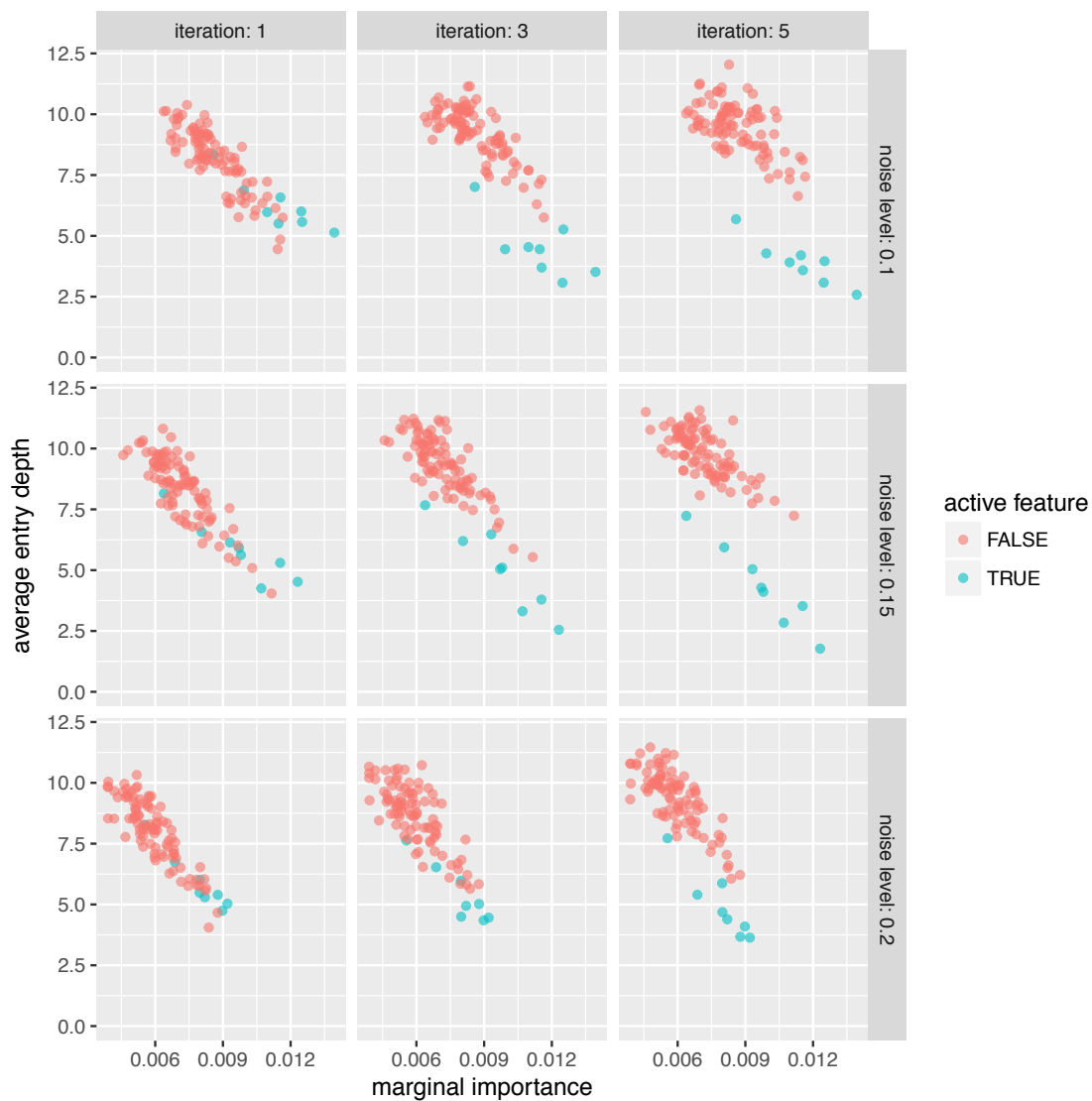


Figure S10: Average entry depth for active (blue) and inactive (red) features across the forest as a function of marginal importance, iteration, and noise level. Results are reported for a single replicate. In later iterations, the average depth at which active variables are selected is noticeably lower than inactive variables with comparable marginal importance, indicating that the active features appear earlier on decision paths.

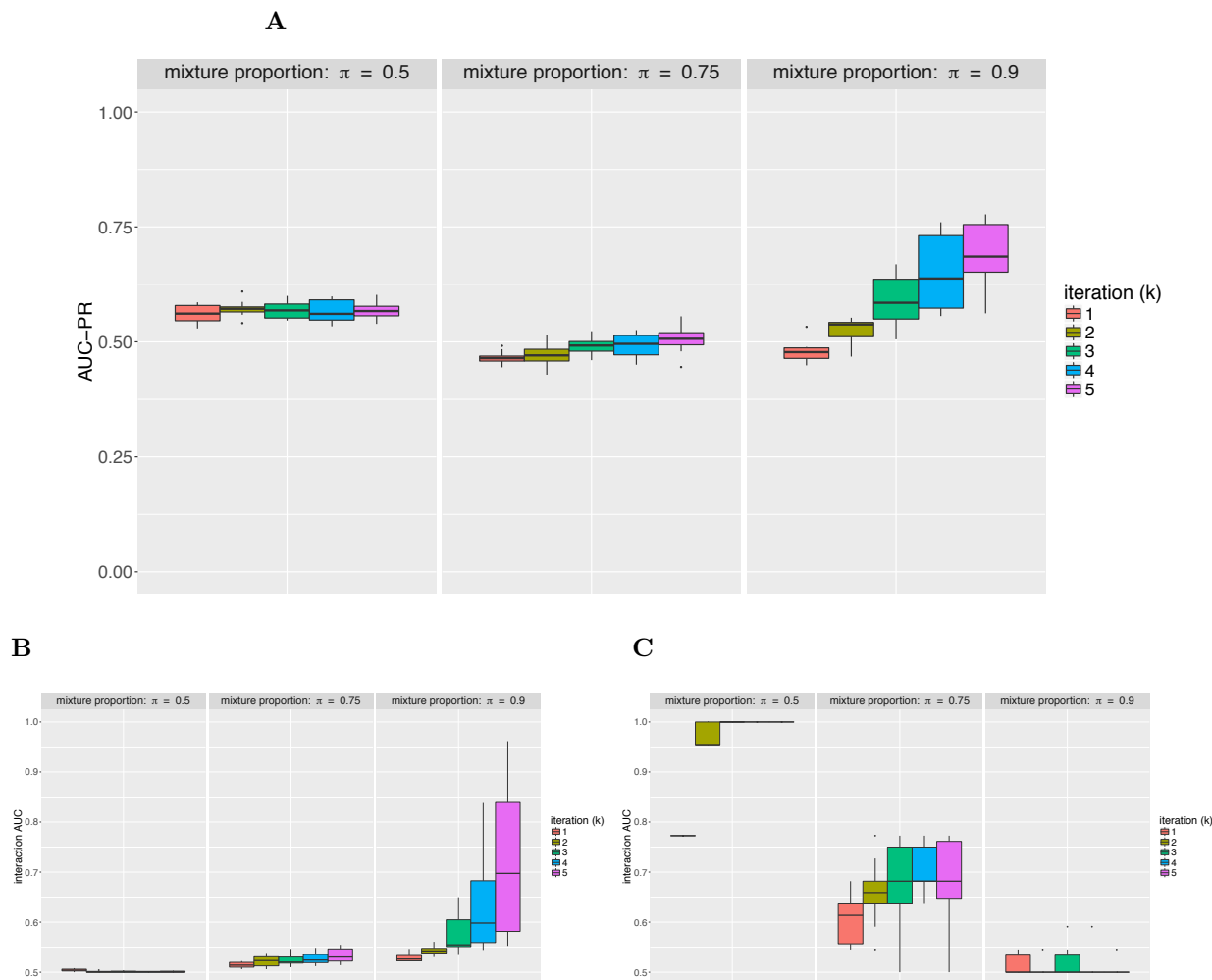


Figure S11: iRF performance for mixture model as a function of mixture proportion (π) over 10 replicates. All models were trained using 5000 observations. **[A]** Prediction accuracy (AUC-PR) is generally poor since iRF tends to learn rules that characterize only a subset of the data. **[B]** Interaction 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.

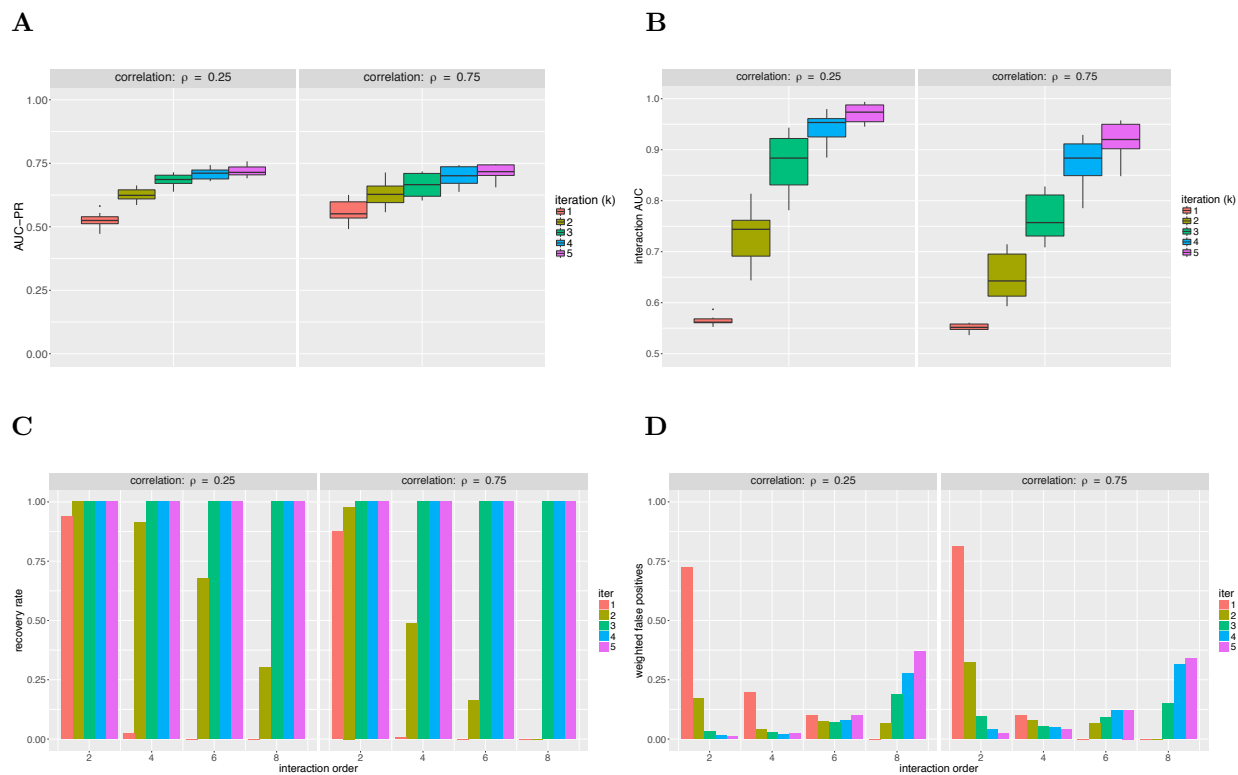


Figure S12: iRF performance for order-8 XOR rule over 10 replicates as a function of correlation level (decaying covariance structure). All models were trained using 5000 observations. **[A]** Prediction accuracy (AUC-PR) improves for increasing k . **[B]** Interaction AUC improves with increasing k , 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 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.

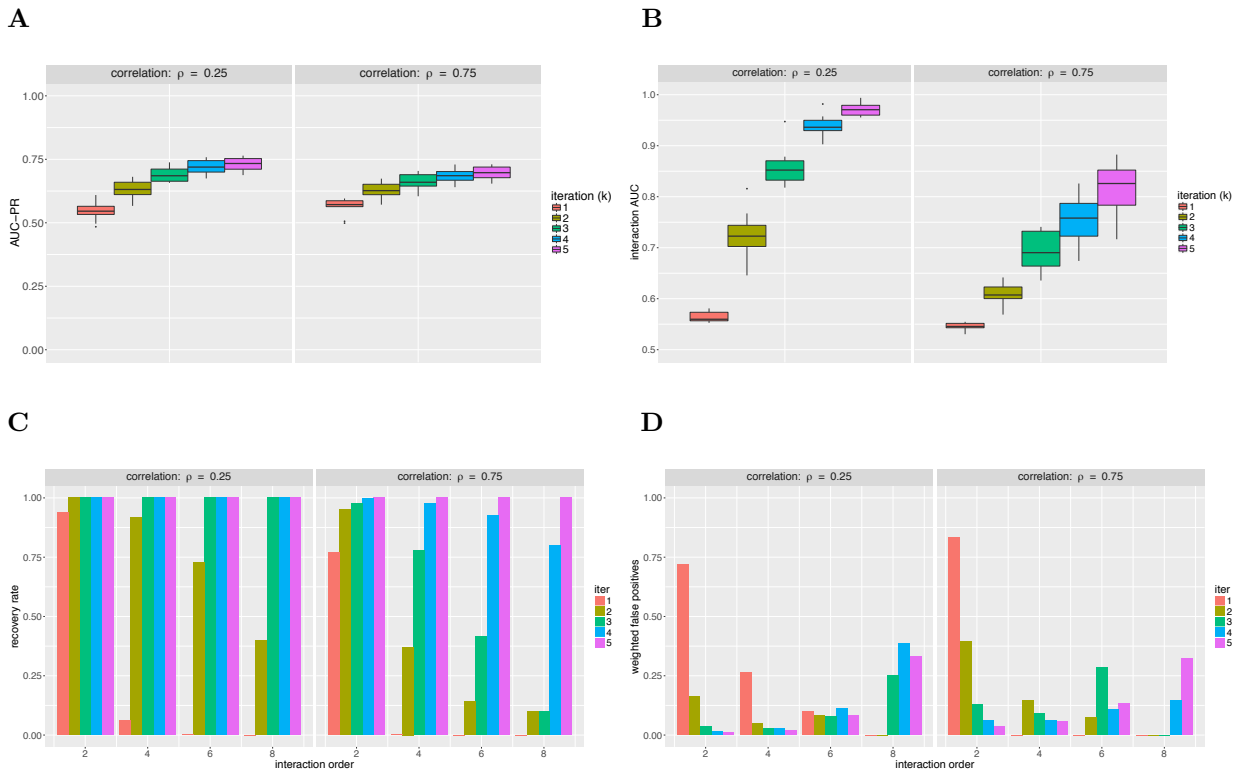


Figure S13: iRF performance for order-8 XOR rule over 10 replicates as a function of correlation level (block covariance). All models were trained on 5000 observations. **[A]** Prediction accuracy (AUC-PR) improves with increasing k . **[B]** Interaction AUC improves with increasing k and drops for large values of ρ . 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.

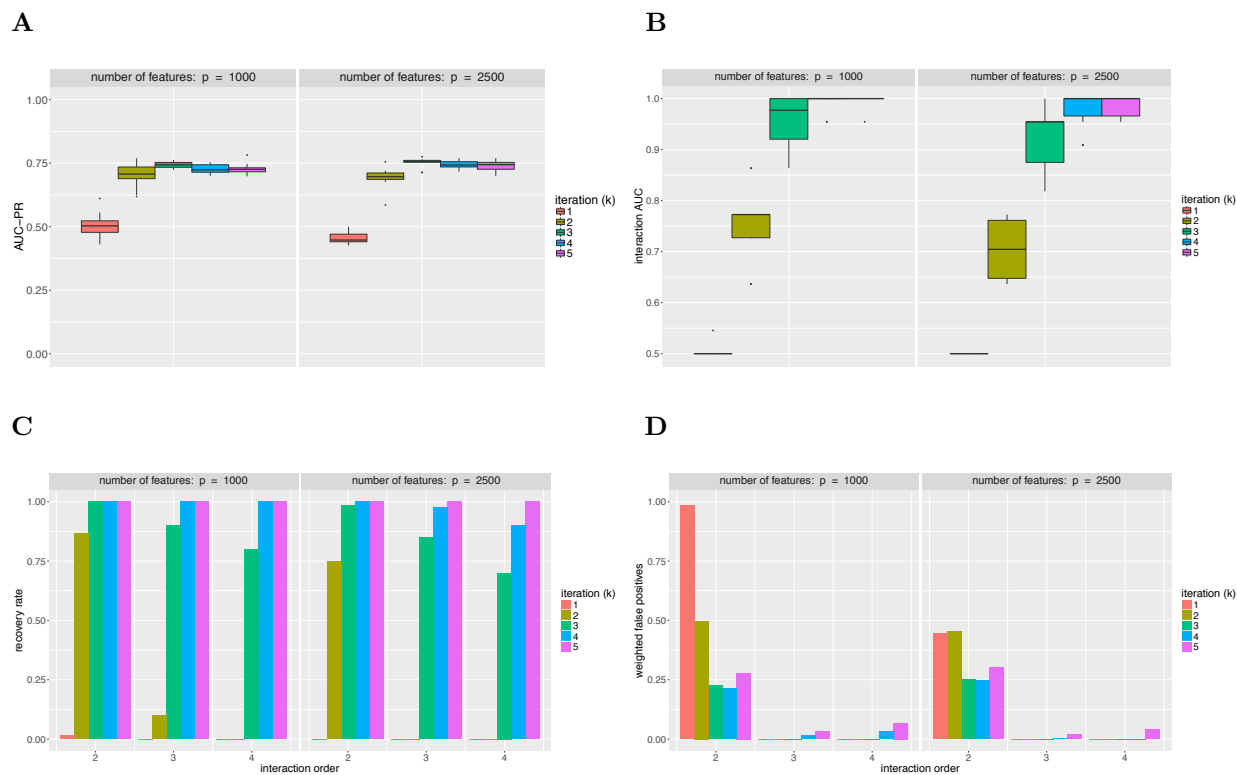


Figure S14: iRF performance for order-4 AND rule over 10 replicates with class labels swapped for 20% of observations selected at random. All models were trained using 500 observations. **[A]** Prediction accuracy (AUC-PR) improves and stabilizes with increasing k . **[B]** Interaction AUC improves dramatically 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.

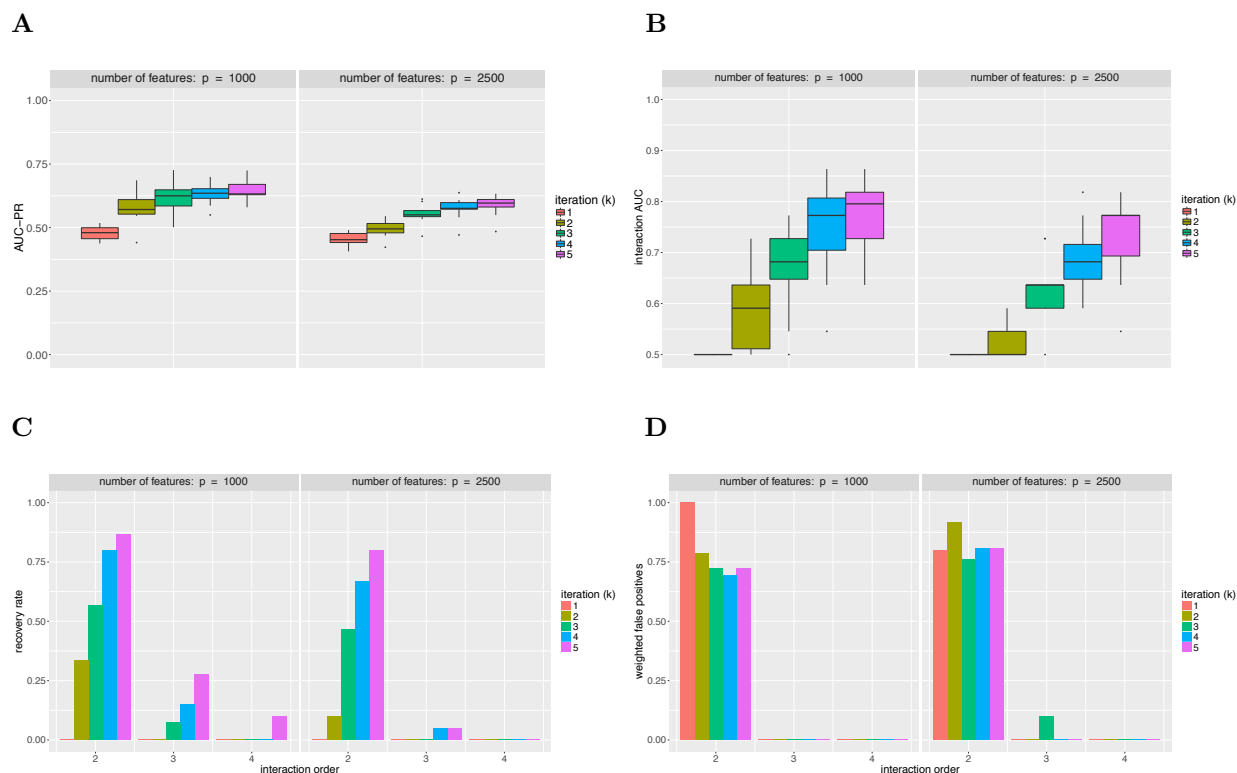


Figure S15: iRF performance for order-4 AND rule over 10 replicates with class labels swapped for 30% of observations selected at random. All models were trained using 500 observations. [A] Prediction accuracy (AUC-PR) gradually improves with increasing k . [B] Interaction AUC gradually improves with 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.

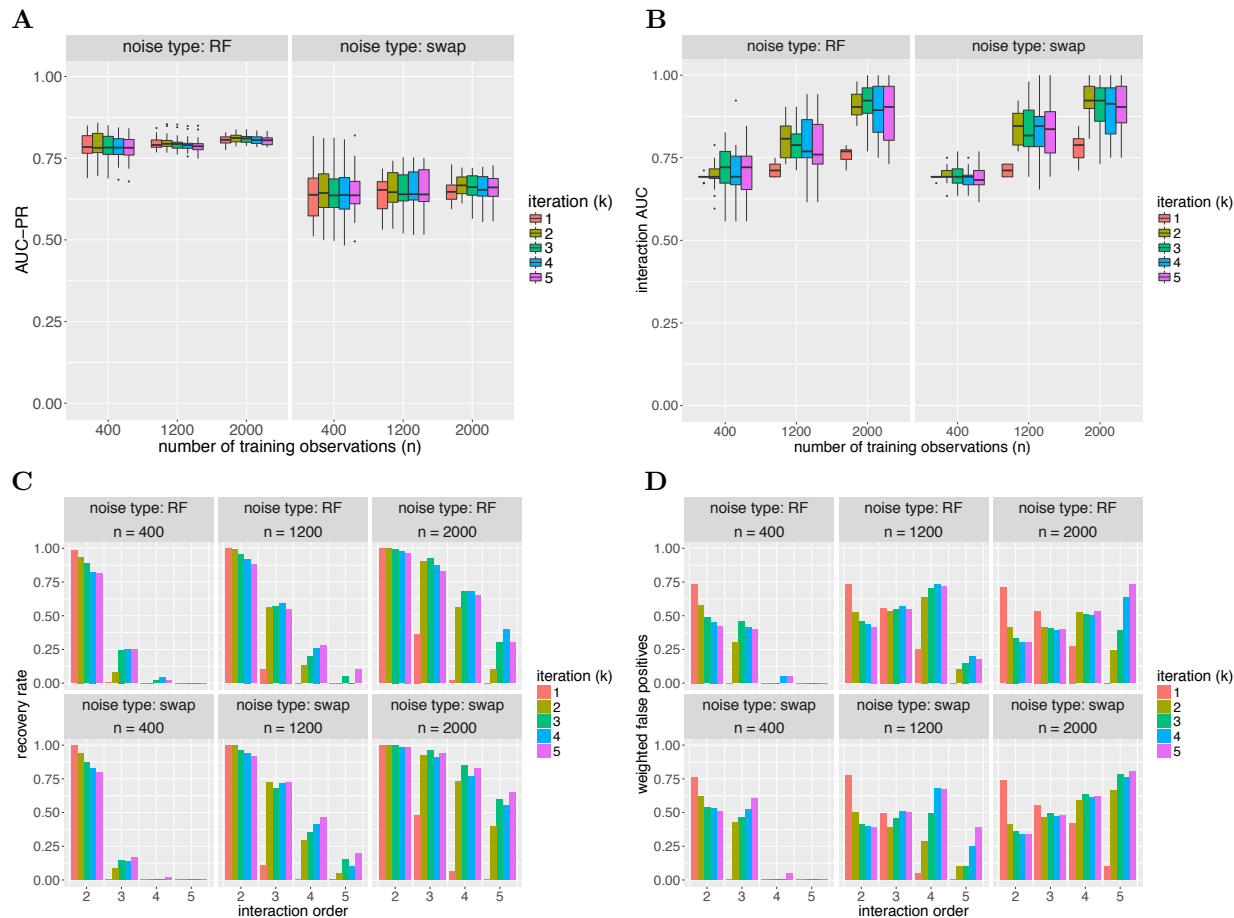


Figure S16: iRF performance for the enhancer data simulations by noise type. Results are shown for models trained using 400, 1200, and 2000 observations. [A] Prediction accuracy (AUC-PR) remains consistent with increasing k in both noise models. [B] Interaction AUC improves after iteration $k = 1$, especially 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 $k = 2, \dots, 5$. [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.

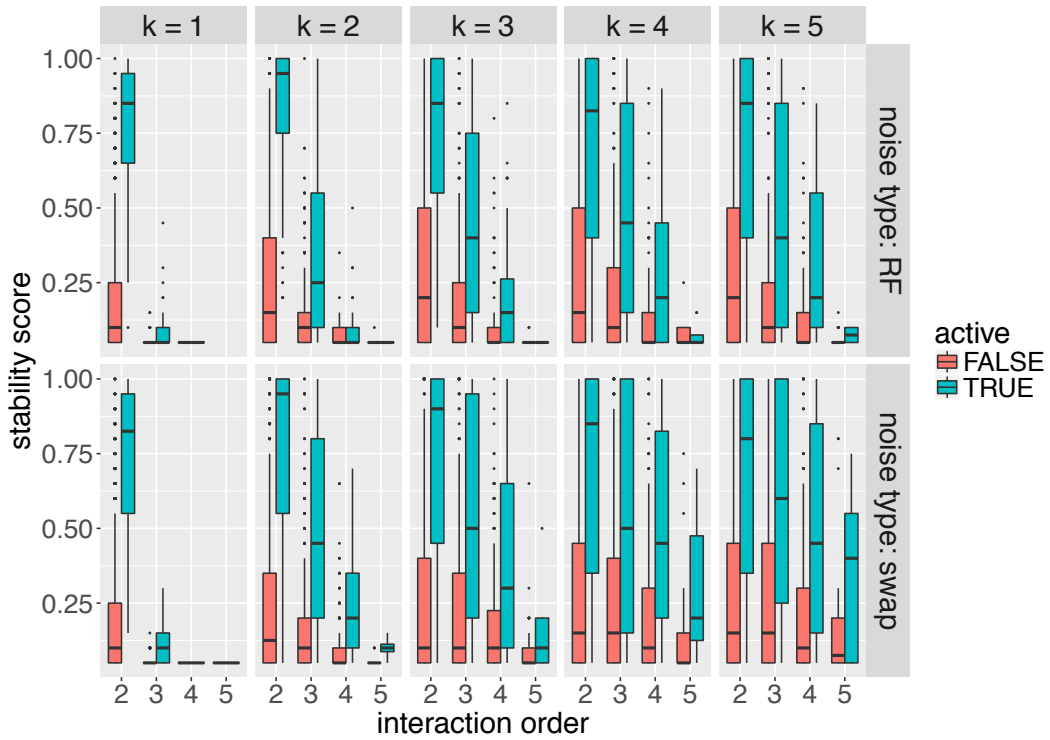


Figure S17: Distributions of iRF stability scores for active and inactive variables by iteration (k) and noise type. Both models were trained using 2000 observations. Interactions among active features are consistently identified as more stable in both noise settings, and higher order interactions are only identified in later iterations.

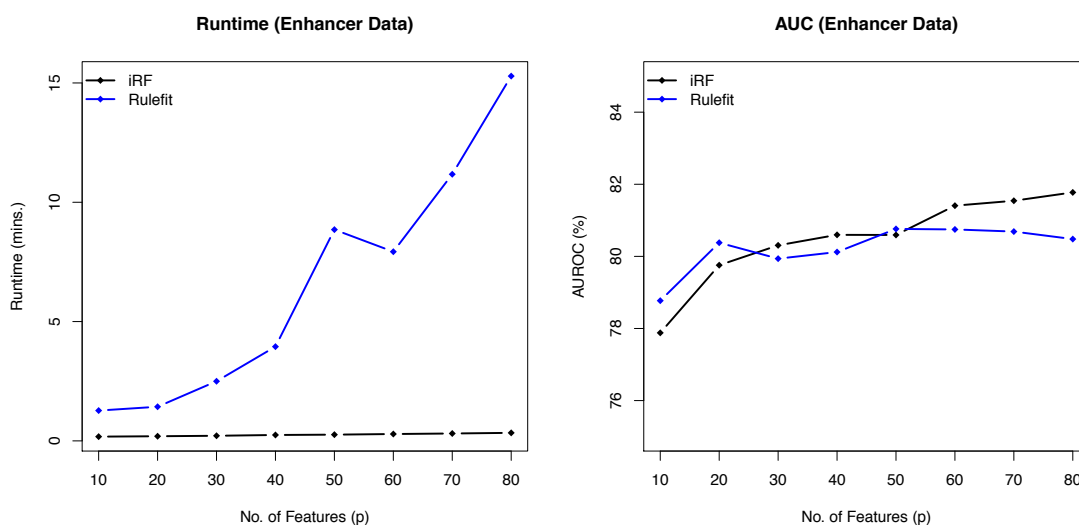


Figure S18: Runtime (left) of interaction detection and Area under ROC curve (right) of prediction by Rulefit and iRF on subsets of the enhancer data with $p \in \{10, 20, \dots, 80\}$ features and balanced training and test sets, each of size $n = 731$. The results are averaged over 10 different permutations 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 , compared to the computational cost of Rulefit.

References

- B. P. Berman, Y. Nibu, B. D. Pfeiffer, P. Tomancak, S. E. Celniker, M. Levine, G. M. Rubin, and M. B. Eisen. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the drosophila genome. *Proceedings of the National Academy of Sciences*, 99(2):757–762, 2002.
- L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- M. Capovilla, E. D. Eldon, and V. Pirrotta. The giant gene of drosophila encodes a b-zip DNA-binding protein that regulates the expression of other segmentation gap genes. *Development*, 114(1):99–112, 1992.
- H. Deng and G. Runger. Feature selection via regularized trees. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.
- E. D. Eldon and V. Pirrotta. Interactions of the drosophila gap gene giant with maternal and zygotic pattern-forming genes. *Development*, 111(2):367–378, 1991.
- ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.
- W. W. Fisher, J. J. Li, A. S. Hammonds, J. B. Brown, B. D. Pfeiffer, R. Weiszmann, S. MacArthur, S. Thomas, J. A. Stamatoyannopoulos, M. B. Eisen, P. J. Bickel, M. D. Biggin, and S. E. Celniker. DNA regions bound at low occupancy by transcription factors do not drive patterned reporter gene expression in drosophila. *Proceedings of the National Academy of Sciences*, 109(52):21330–21335, 2012.
- Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, pages 916–954, 2008.
- E. Frise, A. S. Hammonds, and S. E. Celniker. Systematic image-driven analysis of the spatial Drosophila embryonic expression landscape. *Mol. Syst. Biol.*, 6:345, 2010.
- M. M. Harrison, X.-Y. Li, T. Kaplan, M. R. Botchan, and M. B. Eisen. Zelda binding in the early drosophila melanogaster embryo marks regions subsequently activated at the maternal-to-zygotic transition. *PLoS genetics*, 7(10):e1002266, 2011.
- M. Hoch, C. Schröder, E. Seifert, and H. Jäckle. Cis-acting control elements for krüppel expression in the drosophila embryo. *The EMBO journal*, 9(8):2587, 1990.
- M. Hoch, E. Seifert, and H. Jäckle. Gene expression mediated by cis-acting sequences of the krüppel gene in response to the drosophila morphogens bicoid and hunchback. *The EMBO journal*, 10(8):2267, 1991.
- H. Jäckle, D. Tautz, R. Schuh, E. Seifert, and R. Lehmann. Cross-regulatory interactions among the gap genes of drosophila. *Nature*, 324(6098):668–670, 1986.
- J. Jaeger. The gap gene network. *Cellular and Molecular Life Sciences*, 68(2):243–274, Jan 2011.
- R. Kraut and M. Levine. Mutually repressive interactions between the gap genes giant and krüppel define middle body regions of the drosophila embryo. *Development*, 111(2):611–621, 1991a.
- R. Kraut and M. Levine. Spatial regulation of the gap gene giant during drosophila development. *Development*, 111(2):601–609, 1991b.
- E. Z. Kvon, T. Kazmar, G. Stampfel, J. O. Yanez-Cuna, M. Pagani, K. Schernhuber, B. J. Dickson, and A. Stark. Genome-scale functional characterization of Drosophila developmental enhancers in vivo. *Nature*, 512(7512):91–95, Aug 2014.

- M. Levine. Computing away the magic? *eLife*, 2:e01135, 2013.
- X.-y. Li, S. MacArthur, R. Bourgon, D. Nix, D. A. Pollard, V. N. Iyer, A. Hechmer, L. Simirenko, M. Stapleton, C. L. L. Hendriks, H. C. Chu, N. Ogawa, W. Inwood, V. Sementchenko, A. Beaton, R. Weiszmann, S. E. Celniker, D. W. Knowles, T. Gingeras, T. P. Speed, M. B. Eisen, and M. D. Biggin. Transcription factors bind thousands of active and inactive regions in the drosophila blastoderm. *PLOS Biology*, 6(2):1–24, 02 2008.
- A. Liaw and M. Wiener. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.
- C. Lim and B. Yu. Estimation stability with cross validation (escv). *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.
- S. MacArthur, X.-Y. Li, J. Li, J. B. Brown, H. C. Chu, L. Zeng, B. P. Grondona, A. Hechmer, L. Simirenko, S. V. Keränen, D. W. Knowles, M. Stapleton, P. Bickel, M. D. Biggin, and M. B. Eisen. Developmental roles of 21 drosophila transcription factors are determined by quantitative differences in binding to an overlapping set of thousands of genomic regions. *Genome Biology*, 10(7):R80, Jul 2009.
- D. L. Nelson, A. L. Lehninger, and M. M. Cox. *Lehninger principles of biochemistry*. Macmillan, 2008.
- H. T. Nguyen and X. Xu. Drosophila mef2 expression during mesoderm development is controlled by a complex array of cis-acting regulatory modules. *Developmental biology*, 204(2):550–566, 1998.
- C.-Y. Nien, H.-L. Liang, S. Butcher, Y. Sun, S. Fu, T. Gocha, N. Kirov, J. R. Manak, and C. Rushlow. Temporal coordination of gene networks by zelda in the early drosophila embryo. *PLoS genetics*, 7(10):e1002339, 2011.
- C. Nüsslein-Volhard and E. Wieschaus. Mutations affecting segment number and polarity in drosophila. *Nature*, 287(5785):795–801, 1980.
- D. D. Pervouchine, D. G. Knowles, and R. Guigó. Intron-centric estimation of alternative splicing from RNA-seq data. *Bioinformatics*, 29(2):273–274, 2012.
- D. D. Pervouchine, A. Breschi, E. Palumbo, and R. Guigo. IPSA: An integrative pipeline for splicing analysis, 2016. URL <https://github.com/pervouchine/ipsa>.
- R. Rivera-Pomar and H. Jäckle. From gradients to stripes in drosophila embryogenesis: filling in the gaps. *Trends in Genetics*, 12(11):478–483, 1996.
- C. Schulz and D. Tautz. Autonomous concentration-dependent activation and repression of krüppel by hunchback in the drosophila embryo. *Development*, 120(10):3043–3049, 1994.
- R. D. Shah and N. Meinshausen. Random intersection trees. *The Journal of Machine Learning Research*, 15(1):629–654, 2014.
- G. Struhl, P. Johnston, and P. A. Lawrence. Control of drosophila body pattern by the hunchback morphogen gradient. *Cell*, 69(2):237–249, 1992.
- D. Tautz and C. Pfeifle. A non-radioactive in situ hybridization method for the localization of specific RNAs in drosophila embryos reveals translational control of the segmentation gene hunchback. *Chromosoma*, 98(2):81–85, 1989.
- R. Weiszmann, A. S. Hammonds, and S. E. Celniker. Determination of gene expression patterns using high-throughput RNA in situ hybridization to whole-mount drosophila embryos. *Nature protocols*, 4(5):605, 2009.
- B. Yu. Stability. *Bernoulli*, 19(4):1484–1500, 2013.
- J. Zeitlinger, R. P. Zinzen, A. Stark, M. Kellis, H. Zhang, R. A. Young, and M. Levine. Whole-genome ChIP–chip analysis of dorsal, twist, and snail suggests integration of diverse patterning processes in the drosophila embryo. *Genes & development*, 21(4):385–390, 2007.